# Low Cost Vessel Monitoring System for Inland Water Transport in Bangladesh (LCSMS - IWTB)

By

Md. Habibur Rahman
ID: CSE 048 06349
&
Hosanuzzaman
ID: CSE 048 06325

**A Project Submitted in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**STAMFORD UNIVERSITY BANGLADESH**

**March 2017**

# APPROVAL

The Project Report "Low Cost Vessel Monitoring System for Inland Water Transport in Bangladesh (LCSMS - IWTB)" submitted by Habibur Rahman, Student ID: CSE04806349 and Hosanuzzaman, Student ID: CSE04806325, to the Department of Computer Science and Engineering, Stamford University Bangladesh, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering and approved as to its style and contents.

| | |
|---|---|
| …….. | …… |
| **(Supervisor)** | **(Chairman)** |
| Mohammad Manzurul Islam | Dr. KamruddinMdNur |
| Assistant Professor | Associate Professor |
| Department of CSE | Department of CSE |

# DECLARATION

We, hereby, declare that the work presented in this project is the outcome of the investigation performed by us under the supervision of Mohammad Manzurul Islam, Assistant Professor, Department of Computer Science and Engineering, Stamford University Bangladesh. We also declare that no part of this project and thereof has been or is being submitted elsewhere for the award of any degree or Diploma.

Countersigned

……

**(Supervisor)**

Mohammad Manzurul Islam

Assistant Professor

Department of CSE

Signature

……

**(Candidate) 1.**

Md. Habibur Rahman

Student ID: CSE 04806349

Signature

……

**(Candidate) 2.**

Hossanuzamman

Student ID: CSE 04806325

# ABSTRACT

Bangladesh is a riverine country and communication by waterway is of great importance especially in the southern region of the country. Waterway is the only means of transport and as a result a large number of people has to travel by boats and ships in the coastal areas and inland rivers. Since the early 1950's, motor launch services have become popular and in the period of 1997-98 there were 1,853 registered launches operating in 227 routes. But there is no proper vessel monitoring system in our country and this result into tragic disasters every year, incurring a heavy toll onhuman lives. At least 4,420 people died, 520 people injured and 400 people remained missing in more than 550 passenger boat and ship accidents that took place in last 38 years in Bangladesh.

In this paper a simple and cheap but effective vessel monitoring system is developed for inland vessels. Our device can detect overloading and capsize events by tracking the floating position of a ship and instantly notify the authority and/or the Potential travelers with current coordinates of the vessel, which is obtained using the Global Positioning System (GPS). This can certainly boost up the rescue process and minimize losses. It will encourage the traveler not to board on any overloaded vessel at jetty as this is one of the key reason for many deaths each year during public holidays in Bangladesh. We have used an android phone connected with Arduino via HC-05 Bluetooth device for the communication between the device installed in the vessel and server stored in web. The web server communicates with the control room and travelers. Our developed system can save many lives each year if the government comes forward to implement this project in Bangladesh.

# ACKNOWLEDGEMENTS

First and foremost, we are grateful to Allah, the Almighty, the Merciful without whose blessing, this project would not have been successful. He gave us confidence, courage and determination to overcome the obstacles we faced during this journey.

We would like to express our eternal gratitude to Mohammad Manzurul Islam, Assistant Professor, our respected supervisor, for taking the time from his hectic schedule to guide us in this project. He is the one who inspired us to take this project and supported usin every steps. Without him, this project would not have come to fruition. He provided us with valuable technical advice and pointed us in the right directions which were much required for a project like this. We are truly grateful for having a teacher like him as our supervisor.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# CHAPTER 1

## INTRODUCTION

# 1.1 Introduction

Bangladesh is a riverine country and communication by waterways is of great importance especially in the southern region of the country. Inland water transport plays a vital role in the national life. But this important mode is ridden with tragic disasters every year. At least 4,420 people died, 520 people injured and 400 people remained missing in more than 550 passenger boat and ship accidents that took place in last 38 years in Bangladesh [1]. A statistic by Bangladesh Inland Water Transport Authority (BIWTA) showed that about 20% accidents happened because of vessel overloading [2].

Our attempt is to prevent this overloading problem and save lives. We have developed a low cost fully functional safety monitoring system using Arduino microprocessor and mobile app. Our system sends an overloading alert from an overloaded vessel to the control room so that the authority can take necessary steps before the vessel departed from terminal. Our system also sends an alert notification to a user if a user set his journey on that overloaded vessel on that time through our system. Not only this, our system continuously monitor the vessel and send notification to the control room and authorize app users about the status of the vessel.

So a single overloaded vessel cannot leave vessel terminal and we can reduce more than 20% vessel accidents and it can save a lot of lives. Also, a drowning vessel can send alert message on time so that the authority can take proper action on time.

# 1.2 Objective

The objective of this project is to build a system that reduces vessel disasters and save lives.

We have built a system controlled by Arduino which determines current floating position of a vessel, if it crosses its floating limit, Arduino sends an alert notification to the control room, the vessel captain and the passengers who registered their journey on that overloaded vessel on that time.

Firstly, after getting this alert notification, vessel captain try to make that vessel neutral. If

vessel committee failed to do that then control room forced to stop that vessel in that terminal.

As passenger also get the alert notification, so they can also take their own decision.
If our system implemented properly not a single overloaded vessel can departed from vessel terminal and we can save lots of life.

## 1.3 Scope

This needs to be a government project. After implementing this system government can monitor all the vessels. Control room shows the live list of the current status of every vessel. They also show the last overloading position of a vessel in google map. So if any vessel sinks or any accident happens, the control room would have the pinpoint location of the respective vessel. So, they can send rescue team immediately.

In our system, passenger can also get alert notifications. For getting this notification, the passengers need to be registered in our system and they must add their journey to our system. To add journey, passengers just need to choose the vessel and journey time. During the time of journey if that vessel gets overloaded, our system will send a push notification to that user.

# CHAPTER 2

---

Literature Review

## 2.1 Existing technologies used for monitoring vessels

Today gadgets, instruments and software tools are always in the process of evolving as newer advancements in the navigation aids. Gone are the days when one had to anchor during mist, fog or blizzards when visibility comes down to almost zero. Also, gone are the days when one had to take a position fix taking into consideration the external reference points across the navigating channels and plotting lines using a bearing indicator.

Now a days all international vessels of more than 300 tons and all passenger vessels, irrespective of size, are mandated by the International Convention for the Safety of Life at Sea to carry a transponder that broadcasts their position, course and speed as a collision avoidance system. This tracking system, called the Vessel Monitoring System (VMS), sends information to other nearby vessels so it can be displayed and utilized in conjunction with radar[3].

### 2.1.1 VMS

The Vessel Monitoring System (VMS) allows enforcement to use 21st century technologies to monitor compliance, track violators, and provide substantial evidence for prosecution.

VMS is a satellite surveillance system primarily used to monitor the location and movement of commercial vessels. The system uses satellite-based communications from on-board transceiver units, which certain vessels are required to carry. The transceiver units send position reports that include vessel identification, time, date, and location, and are mapped and displayed on the end user's computer screen[4].

Each vessel typically sends position reports once an hour, but at increased intervals when the vessel is approaching an environmentally sensitive area. Alerts can be sent to the VMS technicians and other personnel when a particular vessel location might require additional inquiry or contact with the vessel operator.

Since the introduction of VMS, the global positioning system and other technology has progressed, and several solutions have provided greater visibility of vessel locations on a worldwide basis. Most popular VMS solutions are given below.

## 2.1.1.1 ORBCOMM Satellite VMS

ORBCOMM devices are optimized to provide near real-time, remote vessel tracking, monitoring and control for commercial fishing boats, merchant marine fleets as well as ocean buoys travelling global waters. Through its versatile suite of programmable satellite and dual-mode devices, ORBCOMM enables commercial fleets of any size to maintain vital data communications links with home ports, no matter where they are on the world's oceans [5].Some ORBCOMM devices are given below:

- **IDP-690: Vessel Tracking Device for Maritime Communications**

  The IDP-690 is ideal for tracking vessels and buoys and communicating with sea personnel. It is designed for maritime applications with its low elevation angle antenna, environmental enclosure and IEC60945 certification. Send location data, reports, text messages, e-logs and more to and from marine vessels. The IDP 600 series of devices uses the reliable, low-latency, two-way Inmarsat IsatData Pro satellite service.

- **ORBCOMM Vessel Monitoring System (VMS)**

  ORBCOMM's Vessel Monitoring System (VMS) is designed to simplify the development and deployment of VMS solutions. The kit delivers the components required to facilitate integration of a vessel tracking solution—hardware, terminal application, and services—all from a single provider at competitive prices.

- **IDP-800: Programmable Satellite Device**

  For smaller vessels, barges and containers, where power and space is at a premium, the low-profile, fully programmable, battery-powered IDP-800 provides for vital tracking and communications. Plus, use it as a vessel safety device with panic button.

## 2.1.1.2SkyHawk Vessel Monitoring Systems (VMS)

SkyHawk – Vessel Monitoring Systems (VMS) are reliable and cost competitive satellite solutions for vessel monitoring and fleet tracking. The SkyHawk VMS provides data services with integrated GPS positioning and two-way text communications that link commercial fishing vessel operators to our a web-based monitoring center and government monitoring centres and also to your loved ones.

The SkyHawk VMS transmits real-time GPS location monitoring data over the satellite network to a government-based tracking office, providing maritime authorities with the ability to monitor the location of commercial fishing vessels operating in regulated waters. If a vessel is found to be in restricted waters, then it is contacted with an appropriate message or instructions. The SkyHawk VMS also has a display terminal to allow vessel

crews to send and receive messages and weather reports to the monitoring centre and other contacts [6].

## 2.2 Vessel Monitoring System in Bangladesh

In August 31, 2013Bangladesh Inland Water Transport Corporation (BIWTC) launched modern Vessel Tracking System (VTS) for itsferries, vessels, streamers and tug vessels aiming to monitor the movement of the water vessels, reports BSS.Country's leading cellular phone operator Grameenphone has installed the information technology based tracking system that will ensure the actual position of the vessel during movement as well as efficient operation [7].

But most of the passenger vessel don’t use this system.

## 2.3 Similar Work

- **GPS-GSM based Inland Vessel Tracking System for Automatic Emergency Detection and Position Notification**

  It is an upgraded version of vehicle tracking system is developed for inland vessels. In addition to the features available in traditional VTS (Vehicle Tracking System) for automobiles, it has the capability of remote monitoring of the vessel's motion and orientation. Furthermore, this device can detect capsize events and other accidents by motion tracking and instantly notify the authority and/or the owner with current coordinates of the vessel, which is obtained using the Global Positioning System (GPS). This can certainly boost up the rescue process and minimize losses. It used GSM network for the communication between the device installed in the vessel and the ground control. So, this can be implemented only in the inland vessels [8].

## 2.4 Uniqueness of LCSMS – IWTBfrom other VMS

Not a single vessel monitoring system provides a monitoring access to a passenger. Our LCSMS-IWTB provides some access to the passengers. A passenger can set journey to a specific vessel with the information of journey time. If that vessel become overloaded during the departure time then our system sends an alert notification to that passenger. Again it also sends another alert notification to the control room. So it is a two way safety system.

# 2.5 Vessel accidents in Bangladesh

Passenger vessel accidents in our inland waterways are becoming a recurrent phenomenon. At least 4,420 people died, 520 people injured and 400 people remained missing in more than 550 passenger vessel accidents that took place in last 38 years [1].

## 2.5.1 Types and causes of vessel accidents

A study finds that all vessel accidents may be classified into five types or categories, namely collision, foundering, overloading, cyclone, and others. As depicted in the following table, the most frequent type or cause of vessel accidents is collision, followed by foundering and overloading. The least frequent accident types are others, followed by cyclones [2].

Table 2.1: Vessel disaster statistics for Bangladesh, 1977-2000

| No | Type or cause of accident | Number of accidents | percentage |
|---|---|---|---|
| 1 | Collision | 59 | 42 |
| 2 | Foundering | 42 | 30 |
| 3 | Overloading | 28 | 20 |
| 4 | Cyclone | 10 | 7 |
| 5 | Others | 1 | 1 |

## 2.5.2 Statistics of vessel accidents

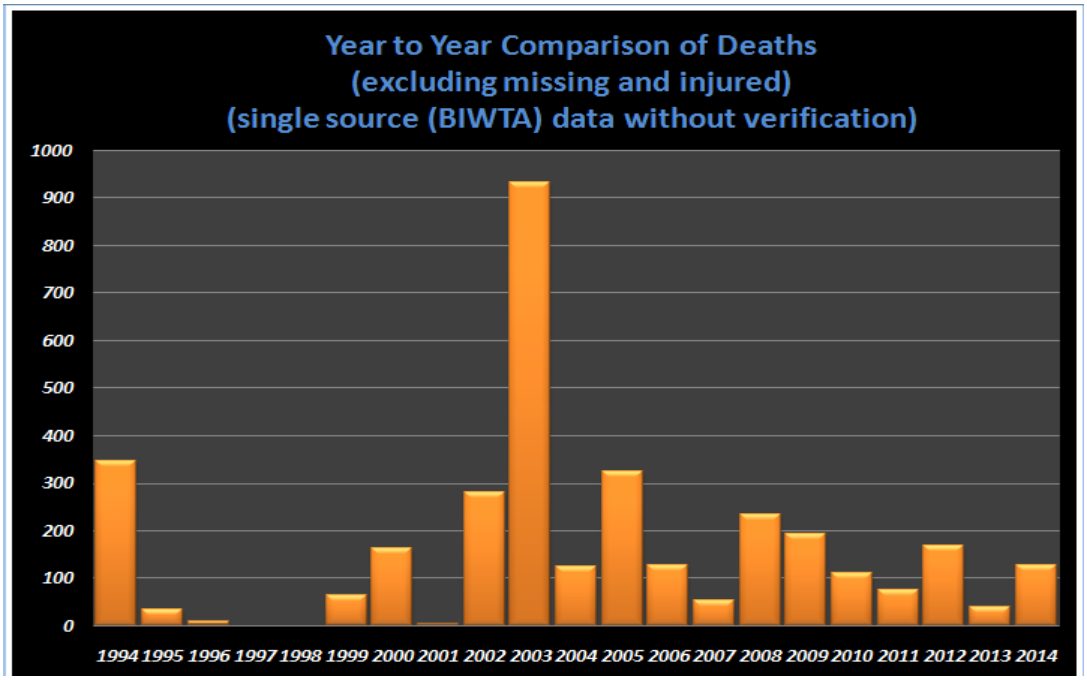


Figure 2.1:Year to year comparison of deaths in accidents in inland waterways [9]

In above figure, the column height is comparatively greater in the year of 1994, 2002, 2003 and 2005. The reasons behind the relatively higher death number in those four years are the major accidents shown in the below table.

Table 2.2: Accident data of the year 1994, 2002, 2003 and 2005 [9].

| Year | Total death | Death in single accident | Vessel (Type) | Reported Cause |
|---|---|---|---|---|
| 1994 | 346 | 273 | M.V. Dinar – 2 (passenger) | Turbulence water &over loading |
| 2002 | 281 | 245 | M.V. Salauddin -2 (passenger) | Nor`wester |
| 2003 | 931 | 131 | M.V. Mitali-3 (passenger) | Nor`wester |
| | | 625 | M.V. Nasrin-1 (passenger) | Turbulence water &over loading |

| | | | | |
|---|---|---|---|---|
| 2005 | 324 | 148 | M.V. Moharaj (passenger) | Nor`wester |

The accidental data of some other major accidents occurred in the last ten years also showsin the following table.

Table 2.3: Some other major accidents of last 10 years [9].

| Vessel (Type) | Total death | Year | Reported cause by BIWTA |
|---|---|---|---|
| M.L. Pinak-6 (passenger) | 47 | 2014 | Strong Current & Overloading |
| M.V. Miraz-4 (passenger) | 56 | 2014 | Overloading and Nor'wester |
| M.V. Shariatpur-1 (passenger) | 146 | 2012 | Collision |
| M.B. Chandpur (passenger) | 50 | 2008 | Storm & over loading |
| M.L. Shourav-1 (passenger) | 50 | 2008 | Collision |
| M.V. Lighting Sun (passenger) | 61 | 2004 | Nor`wester |

# 2.6 Arduino Mega 2650

## 2.6.1 Introduction

Arduino is an open-source project that created microcontroller-based kits for building digital devices and interactive objects that can sense and control physical devices.

The project is based on microcontroller board designs, produced by several vendors, using various microcontrollers. These systems provide sets of digital and analog input/output

(I/O) pins that can interface to various expansion boards (termed shields) and other circuits. The boards feature serial communication interfaces, including Universal Serial Bus (USB) on some models, for loading programs from personal computers. For programming the microcontrollers, the Arduino project provides an integrated development environment (IDE) based on a programming language named Processing, which also supports the languages C and C++ [10].

The LCSMS – IWTBhas utilized the Arduino 2560 which is a microcontroller board based on the ATmega1280. It has:

Table 2.4: Microcontroller board based on the ATmega1280 [11].

| | |
|---|---|
| Microcontroller | ATmega1280 |
| Operating Voltage | 5V |
| Input Voltage (recommended) | 7-12V |
| Input Voltage (limits) | 6-20V |
| Digital I/O Pins | 54 (of which 15 provide PWM output) |
| Analog Input Pins | 16 |
| DC Current per I/O Pin | 40 mA |
| DC Current for 3.3V Pin | 50 mA |
| Flash Memory | 128 KB of which 4 KB used by boot loader |
| SRAM | 8 KB |
| EEPROM | 4 KB |
| Clock Speed | 16 MHz |

## 2.6.2 Arduino 2560 Components

### 2.6.2.1 ATmega1280

The Arduino Mega is a microcontroller board based on the ATmega1280. This highperformance, low-power Atmel 8-bit AVR RISC-based microcontroller combines 128KB ISP flash memory, 8KB SRAM, 4KB EEPROM, 86 general purpose I/O lines, 32 general purpose working registers, real time counter, six flexible timer/counters with compare modes, PWM, 4 USARTs, byte oriented 2-wire serial interface, 16-channel 10-bit A/D converter, and a JTAG interface for on-chip debugging. The device achieves a throughput of 16 MIPS at 16 MHz and operates between 2.7-5.5 volts.

### 2.6.2.2 Power

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector [12].

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN:** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

- **5V:** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.

- **3.3V:** A 3.3 volt supply generated by the on-board FTDI chip. Maximum current draw is 50 mA.

- **GND:** Ground pins.

### 2.6.2.3 Memory

The ATmega1280 has 128 KB of flash memory for storing code (of which 4 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library) [13].

### 2.6.2.4 Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts [14]. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial:** 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX). Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the FTDI USB-to-TTL Serial chip.

- **External Interrupts:** 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2). These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.

- **PWM:**2 to 13 and 44 to 46. Provide 8-bit PWM output with theanalogWrite()function.

- **SPI:** 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.

- **LED:** 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

- **$I^2C$:** 20 (SDA) and 21 (SCL). Support I2C (TWI) communication using the Wire library (documentation on the Wiring website). Note that these pins are not in the same location as the I2C pins on the Duemilanove or Diecimila.

The Mega has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and analogReference() function.

There are a couple of other pins on the board:

- **AREF:** Reference voltage for the analog inputs. Used with analogReference().
- **Reset:** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

### 2.6.2.5 Communication

The Arduino Mega has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers [15]. The ATmega1280 provides four hardware UARTs for TTL (5V) serial communication. An FTDI FT232RLon the board channels one of these over USB and the FTDI drivers (included with the Arduino software) provide a virtual com port to software on the computer. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the FTDI chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows for serial communication on any of the Mega's digital pins.

The ATmega1280 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation on the Wiring website for details. To use the SPI communication, please see the ATmega1280 datasheet.

### 2.6.2.6 Programming

The Arduino Mega can be programmed with the Arduino software. The ATmega1280 on the Arduino Mega comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files) [16].

### 2.6.2.7 Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Mega is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of theFT232RL is connected to the reset line of the ATmega1280 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the

Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload [17].

This setup has other implications. When the Mega is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line.

# 2.6.2.8 USB Overcurrent Protection

The Arduino Mega has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

# 2.6.2.9 Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins [18].

The Mega is designed to be compatible with most shields designed for the Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega and Duemilanove / Diecimila.

# 2.7 HC-05 Bluetooth device

HC-05 module is easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup.

Serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses CSR Blue core 04-External single chip Bluetooth system with CMOS technology and with AFH(Adaptive Frequency Hopping Feature). It has the footprint as small as 12.7mmx27mm [19].

Figure 2.2:HC-05

## 2.7.1 Hardware Features

- Typical -80dBm sensitivity
- Up to +4dBm RF transmit power
- Low Power 1.8V Operation ,1.8 to 3.6V I/O
- PIO control
- UART interface with programmable baud rate
- With integrated antenna
- With edge connector

## 2.7.2 Software Features

- Default Baud rate: 38400, Data bits:8, Stop bit:1,Parity:No parity, Data control: has.

Supported baud rate: 9600,19200,38400,57600,115200,230400,460800.

- Given a rising pulse in PIO0, device will be disconnected.
- Status instruction port PIO1: low-disconnected, high-connected;
- Auto-connect to the last device on power as default. Permit pairing device to connect as default.
- Auto-pairing PINCODE:"0000" as default
- PIO10 and PIO11 can be connected to red and blue led separately. When master and slave are paired, red and blue led blinks 1time/2s in interval, while disconnected only blue led blinks 2times/s
- Auto-reconnect in 30 min when disconnected as a result of beyond the range of connection.

# 2.8Breadboard

A breadboard is a construction base for prototyping of electronics. Originally it was literally a bread board, a polished piece of wood used for slicing bread. In the 1970s the solderless breadboard (AKAplugboard, a terminal array board) became available and nowadays the term "breadboard" is commonly used to refer to these. "Breadboard" is also a synonym for "prototype".

Because the solder less breadboard does not require soldering, it is reusable. This makes it easy to use for creating temporary prototypes and experimenting with circuit design. For this reason, solder less breadboards are also extremely popular with students and in technological education. Older breadboard types did not have this property. Astrip board (Vero board) and similar prototyping printed circuit boards, which are used to build semi-permanent soldered prototypes or one-offs, cannot easily be reused. A variety of electronic systems may be prototyped by using breadboards, from small analog and digital circuits to complete central processing units (CPUs) [20].

# 2.9Horizontal Floating Water Level Switch

It is a very simple switch that can measure the water level. The principle is very simple. It has a floating part which can freely float with water. When the floating part touches the fixed part or upper part of our switch then the circuit become closed.

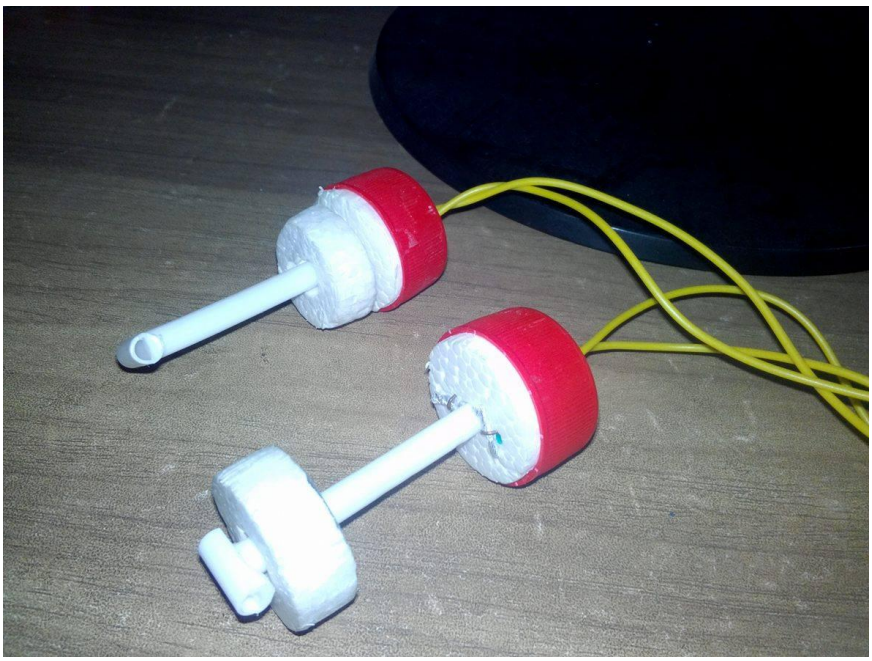

Figure 2.3: Horizontal Floating Water Level Switch

# 2.10 Android Application Technologies

## 2.10.1 Android Studio IDE

Android Studio was on May 16, 2013. Google's product manager Ellie Powers on the developer conferenceGoogle I / O announced. Shortly after this time, Google periodically new test versions [21].
After a development time of two years, Google published on 8 December 2014 Version 1.0 for Windows, OS X and Linux.
Since the alpha version 1.2 Preview 1, which was published on 10 March 2015 is based on Android Studio IntelliJ IDEA 14.With the preview version 1.3 of 28 May 2015 the SDK Manager has been fully integrated into Android Studio, further is now support for the Android NDK ( Native Development Kit ) available. In the final version 1.3 also the Android Memory Viewer and Allocation Tracker was integrated. From this version, it is also possible to inline annotations for the new app permissions of Android M Data Binding to use as well.

Since the Android Studio preview version 2.0, the feature is Instant Run available, which allows developers to modify the amended code and resources directly on the device within the current app.

## 2.10.2 Android SDK

The Android software development kit (SDK) includes a comprehensive set of development tools. These include a debugger, libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. Currently supported development platforms include computers running Linux (any modern desktop Linux distribution), Mac OS X 10.5.8 or later, and Windows 7 (previously XP) or later. As of March 2015, the SDK is not available on Android itself, but the software development is possible by using specialized Android applications. Until around the end of 2014, the officially supportedintegrated development environment (IDE) was Eclipse using the Android Development Tools (ADT) Plugin, though IntelliJ IDEA IDE (all editions) fully supports Android out of the box,[7]and NetBeans IDE also supports Android development via a plugin.· As of 2015, Android Studio, made by Google and powered by IntelliJ, is the official IDE; however, developers are free to use others. Additionally, developers may use any text editor to edit Java and XML files, then usecommand line tools (Java Development Kit andApache Ant are required) to create, build and debug Android applications as well as control attached Android devices (e.g., triggering a reboot, installing software package(s) remotely). Enhancements to Android's SDK go hand in hand with the overall Android platform development. The SDK also supports older versions of the Android platform in case developers wish to target their applications at older devices. Development tools are downloadable components, so after one has downloaded the latest version and platform, older platforms and tools can also be downloaded for compatibility testing. Android applications are packaged in .apk format and stored under  folder on the Android OS (the folder is accessible only to the root user for security reasons). APK package contains dex files (compiled byte code files called Dalvik executables), resource files, etc [22].

## 2.10.3 Gradle

Gradle is an open source build automation system that builds upon the concepts of Apache Ant and Apache Maven and introduces a Groovy-based domain-specific language (DSL) instead of the XML form used byApache Maven of declaring the project configuration. Gradle uses a directed acyclic graph ("DAG") to determine the order in which tasks can be run.
Gradle was designed for multi-project builds which can grow to be quite large and supports incremental builds by intelligently determining which parts of the build tree are up-to-date, so that any task dependent upon those parts will not need to be re-executed.

The initial plugins are primarily focused around Java, Groovy and Scale development and deployment, but more languages and project workflows are on the roadmap [23].

## 2.10.4 XML

In addition to Java code, Android projects (and their developers) have the ability to utilize XML to perform many standard tasks. Some XML usage is required, such as the definition of the projects and its components. Much of XML's usage is optional, making many common tasks easier [24]. The Android XML schema is highly flexible and may be used in combination with code, exclusively, or not at all. Below is a list of common usage of XML:

- Manifest - definition of the project
- Layout - creation of partial or complete layouts for Activities, Dialogs, and Widgets
- Colors - Constant color values used throughout the project
- Style & Themes - application of custom standardized looks of Views
- Animations - Standardized animations that may be applied to Views.
- Drawables - Some specialized icons and graphics that may not be created wholly with an image editor. (StateDrawables, TranstionDrawables, Shapes, and VectorGraphics)
- Menu - A resource used to aid in standardized menus for an Activity
- Integers, Strings, and Arrays - Constants used by the Application as resources.

## 2.10.5 Material Design

Material design is a comprehensive guide for visual, motion, and interaction design across platforms and devices.

Material Design makes more liberal use of grid-based layouts, responsive animations and transitions, padding, and depth effects such as lighting and shadows.

Google announced Material Design on June 25, 2014, at the 2014Google I/O conference. Android 5.0 Lollipop includes support for material design apps. Polymer and Angular Material projects also provide official implementations [25].

## 2.10.6 Coding language (Java)

Java is a general-purpose computer programming language that is concurrent, class-based,objectorientedand specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation.[16]Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of computer architecture. As of 2016, Java is one of the most popular programming languages in useparticularly for client-server web applications, with a reported 9 million developers. Java was originally developed by James Gosling atSun Microsystems (which has since beenacquired by Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them [26]. The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses. As of May 2007, in compliance with the specifications of the Java Community Process, Sun relicensed most of its Java technologies under the GNU General Public License. Others have also developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java (bytecode compiler), GNU Class path(standard libraries), and Iced Tea-Web (browser plugin for applets).

The latest version is Java 8, which is the only version currently supported for free by Oracle, although earlier versions are supported both by Oracle and other companies on a commercial basis.

## 2.10.7 Google Play service

With Google Play services, your app can take advantage of the latest, Google-powered features such as Maps, Google+, and more, with automatic platform updates distributed as an APK through the Google Play store. This makes it faster for your users to receive updates and easier for you to integrate the newest that Google has to offer [27].

Google Play service client library contains the interfaces to the individual Google services and allows you to obtain authorization from users to gain access to these services with their credentials. It also contains APIs that allow you to resolve any issues at runtime, such as a missing, disabled, or out-of-date Google Play services APK. The client library has a light footprint if you use ProGuard as part of your build process, so it won't have an adverse

impact on your app's file size.

## 2.10.8 Google Location services:

The Google Play services location APIs are preferred over the Android framework location APIs (android.location) as a way of adding location awareness to your app.

The Google Location Services API, part of Google Play Services, provides a more powerful, high-level framework that automatically handles location providers, user movement, and location accuracy. It also handles location update scheduling based on power consumption parameters you provide. In most cases, you'll get better battery performance, as well as more appropriate accuracy, by using the Location Services API.

## 2.10.9 Google Map Service

Google Maps is a web mapping service developed by Google. It offers satellite imagery, street maps, 360° panoramic views of streets (Street View), real-time traffic conditions (Google Traffic), and route planning for traveling by foot, car, bicycle (in beta), or public transportation [28].

Google Maps began as a C++ desktop program designed by Lars and Jens Eilstrup Rasmussen Where 2 Technologies. In October 2004, the company was acquired by Google, which converted it into a web application. After additional acquisitions of a geospatial data visualization company and a real-time traffic analyzer, Google Maps was vesseled in February 2005.[1]The service's front end utilizes JavaScript, XML, and Ajax. Google Maps offers an API that allows maps to be embedded on third-party websites,[2]and offers a locator for urban businesses and other organizations in numerous countries around the world. Google Map Maker allows users to collaboratively expand and update the service's mapping worldwide.

Google Maps' satellite view is a "top-down" or "birds eye" view; most of the high-resolution imagery of cities is aerial photography taken from aircraft flying at 800 to 1,500 feet (240 to 460 m), while most other imagery is from satellites.[3]Much of the available satellite imagery is no more than three years old and is updated on a regular basis.[4]Google Maps uses a close variant of the Mercator projection, and therefore cannot accurately show areas around the poles.

## 2.10.10 Volley Library

Volley is an HTTP library that makes networking for Android apps easier and most importantly, faster [29].

Volley offers the following benefits:

- Automatic scheduling of network requests.
- Multiple concurrent network connections.
- Transparent disk and memory response caching with standard HTTP cache coherence.
- Support for request prioritization.
- Cancellation request API. You can cancel a single request, or you can set blocks or scopes of requests to cancel.
- Ease of customization, for example, for retry and backoff.
- Strong ordering that makes it easy to correctly populate your UI with data fetched asynchronously from the network.
- Debugging and tracing tools.
- The core Volley library is developed in the open AOSP repository at frameworks/volley and contains the main request dispatch pipeline as well as a set of commonly applicable utilities, available in the Volley "toolbox." The easiest way to add Volley to your project is to add the following dependency to your app's build.Gradle file:

```
dependencies {
...
compile 'com.android.volley:volley:1.0.0'
}
```

### 2.10.11 Material DateTime Picker

Material DateTime Picker tries to offer the date and time pickers as shown in the Material Design spec, with an easy themable API. The library usesthe code from the Android frameworks as a base and tweaked it to be as close as possible to Material Design example. Support for Android 4.0 and up [30].
The easiest way to add the Material DateTime Picker library to your project is by adding it as a dependency to yourbuild.gradle.

```
dependencies {    compile
'com.wdullaer:materialdatetimepicker:2.5.0' }
```

### 2.10.12 REST API

REST is the underlying architectural principle of the web. The amazing thing about the web is the fact that clients (browsers) and servers can interact in complex ways without the client knowing anything beforehand about the server and the resources it hosts. The key constraint is that the server and client must both agree on the media used, which in the case of the web is HTML.

An API that adheres to the principles of REST does not require the client to know anything about the structure of the API. Rather, the server needs to provide whatever information the client needs to interact with the service. An HTML form is an example of this: The server specifies the location of the resource, and the required fields. The browser doesn't know in advance where to submit the information, and it doesn't know in advance what information to submit. Both forms of information are entirely supplied by the server [31]. (This principle is called HATEOAS.)

## 2.11 Web Service Technologies

### 2.11.1 Sublime Text (text editor)

Sublime Text is a proprietary cross-platformsource code editor with a Python application programming interface (API). It natively supports manyprogramming languages and markup languages, and its functionality can be extended by users with plugins, typically community-built and maintained underfree-software licenses [32].

**Features:**

- “Go to Anything," quick navigation to files, symbols, or lines
- "Command palette" uses adaptive matching for quick keyboard invocation of arbitrary commands
- Simultaneous editing: simultaneously make the same interactive changes to multiple selected areas
- Python-based plugin API
- Project-specific preferences
- Extensive customizability via JSON settings files, including project-specific and platform-specific settings
- Cross platform (Windows, OS X, Linux)
- Compatible with many language grammars from Text Mate.

## 2.11.2 PHP (Scripting Language)

PHP is a server-side scripting language designed primarily for web development but is also used as a general-purpose programming language. Originally created by Rasmus Lerdorf in 1994, the PHP reference implementation is now produced by The PHP Development Team. PHP originally stood for Personal Home Page, but it now stands for therecursive acronym PHP: Hypertext Preprocessor [33].
PHP code may be embedded into HTML code, or it can be used in combination with various web template systems, web content management systems, and web frameworks.PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications.

## 2.11.3 Slim Framework

Slim is a PHP micro framework that helps you quickly write simple yet powerful web applications and APIs [34].

```
<?phpuse\Psr\Http\Message\ServerRequestInterface as
Request;use\Psr\Http\Message\ResponseInterface as
Response;


require'vendor/autoload.php';


$app=new\Slim\App;

$app->get('/hello/{name}',function(Request $request, Response $response){

$name=$request->getAttribute('name');

$response->getBody()->write("Hello, $name");


return$response;

});

$app->run();
```

## 2.11.4 JSON

JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language [35].

JSON is built on two structures:

- A collection of name/value pairs. In various languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array.
- An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence.

These are universal data structures. Virtually all modern programming languages support them in one form or another. It makes sense that a data format that is interchangeable with programming languages also be based on these structures.

The following example shows a possible JSON representation describing a person.

```
{
"firstName": "John",
"lastName": "Smith",
"isAlive": true,
"age": 25,
"address": {
"streetAddress": "21 2nd Street",
"city": "New York",
"state": "NY",
"postalCode": "10021-3100"
},
"phoneNumbers": [
{
"type": "home",
"number": "212 555-1234"
},
{
"type": "office",
"number": "646 555-4567"
},
{
"type": "mobile",
"number": "123 456-7890"
```

}
],
"children": [],
"spouse": null
}

## 2.12Plimsoll line

The Plimsoll line is a reference mark located on a vessel's hull that indicates the maximum depth to which the vessel may be safely immersed when loaded with cargo. This depth varies with a vessel's dimensions, type of cargo, time of year, and the water densities encountered in port and at sea. Once these factors have been accounted for, a vessel's captain can determine the appropriate [36].


Figure2.4: Plimsoll Line

# CHAPTER 3

---

## SYSTEM DEVELOPMENT

# 3.1 Schematics

The Low cost vessel monitoring system for Inland water transport (LCSMS-IWT) system consists of an Arduino Mega 2560, a HC-05 Bluetooth module, four pics water level switch, two android application, one Bluetooth and internet supported android phone. The Vessel Control Application (One of our Android Application) is installed in that android mobile phone. HC-05 Bluetooth device is connected with the Arduino and this Bluetooth device will pass the information of Arduino to our mobile

Application via Bluetooth. If the water touches vessel's plimsoll line [36], Arduino sends alert notification to the Vessel Control Application. The App then sends this alert notification to our server via internet and the server sends the alert notification to the control room and also passengers of that vessel who add their journey to our system.

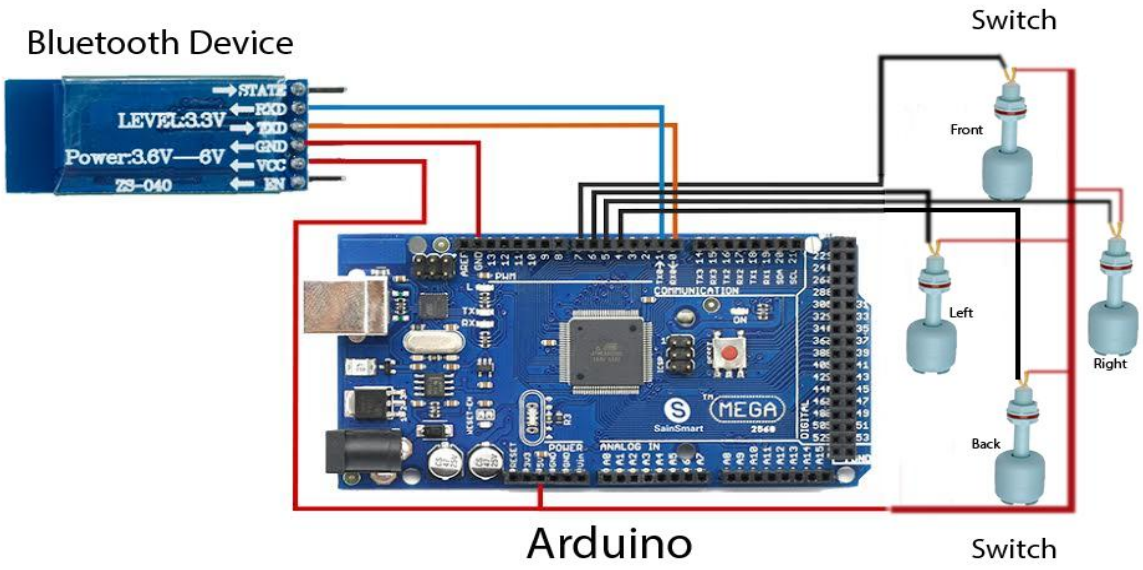


Figure 3.1: Schematics (Graphical)

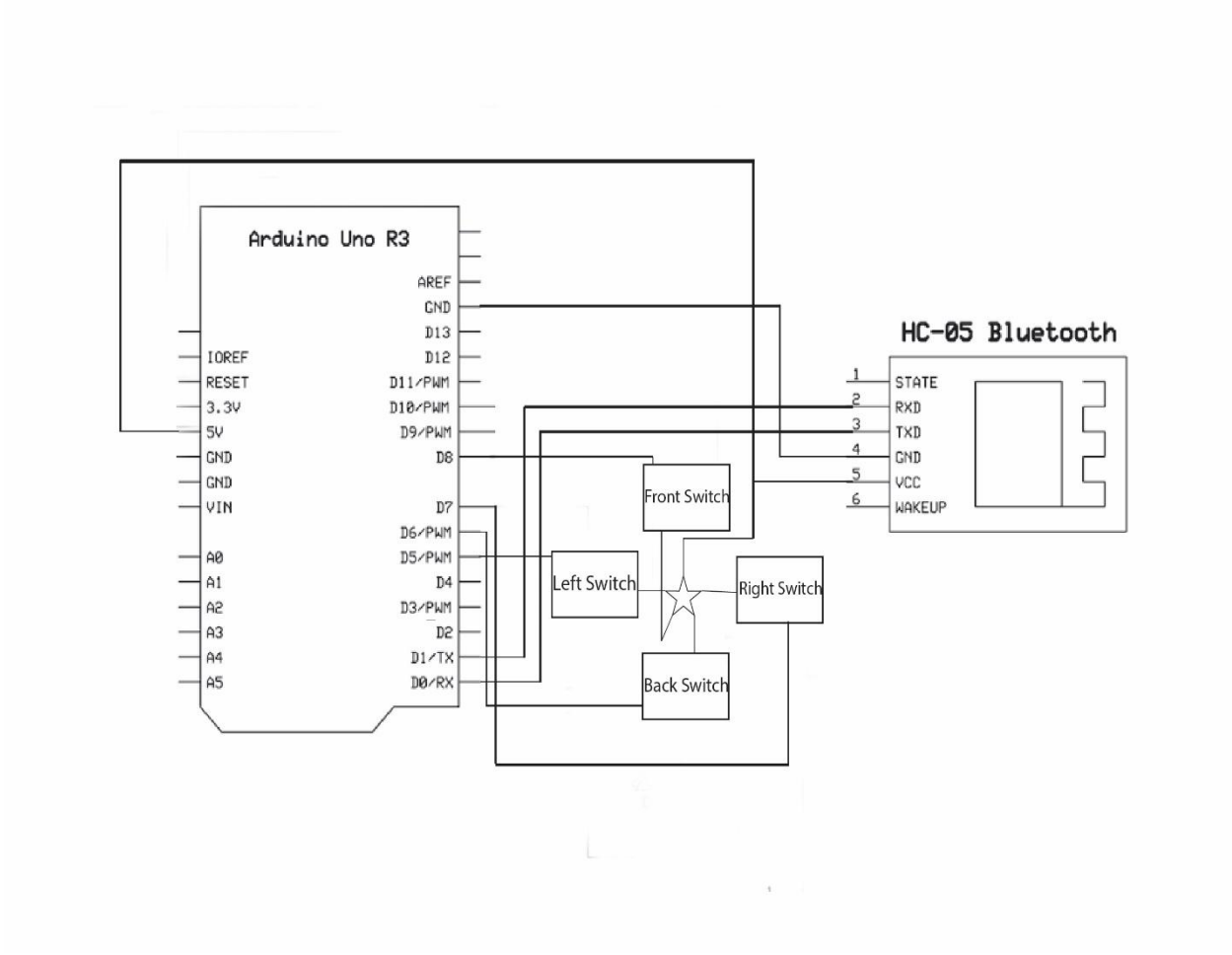


Figure 3.2: Schematics

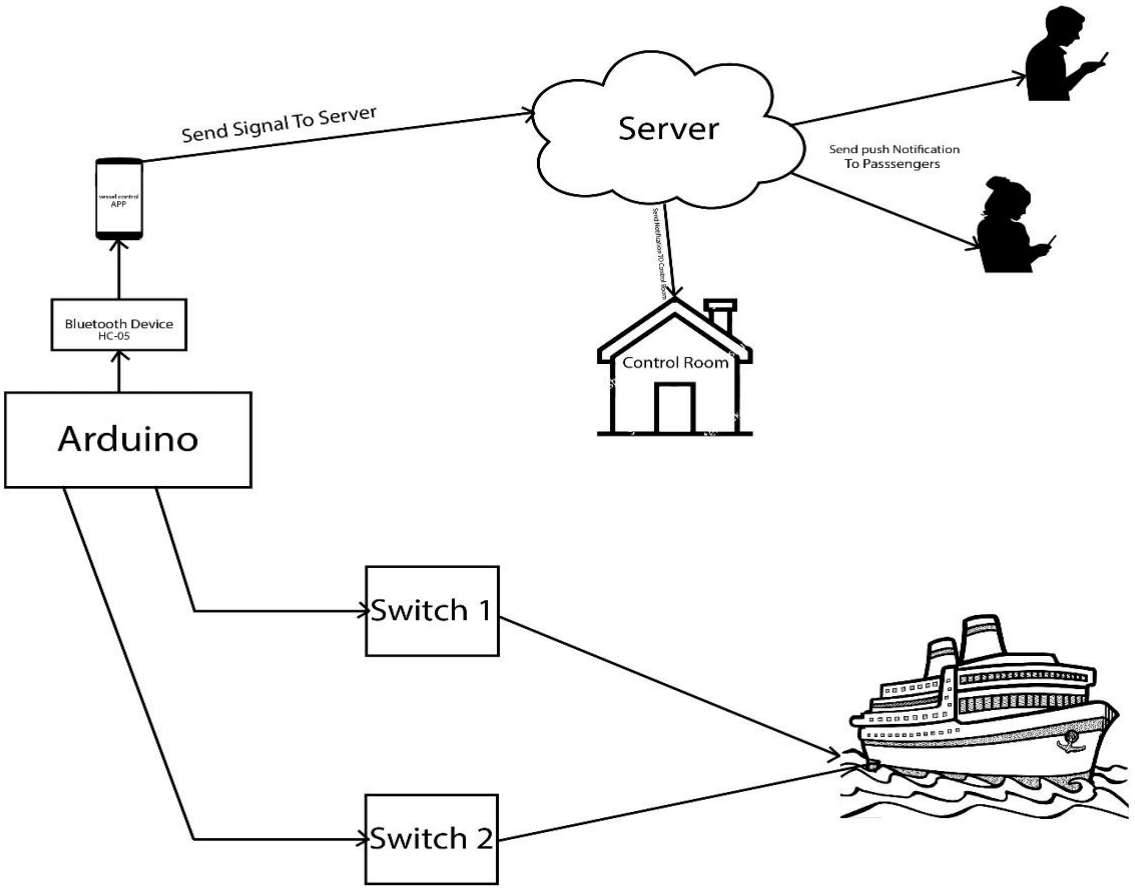


Figure 3.3: LCSMS – IWTB

## 3.2 Android Application

The system consists of two Android application – one is “User Application”, another is “Vessel Control Application”.

### 3.2.1 User Application

User needs to login first for using this application. If user does not have an account in our system, then user can register in our system through this application. There are two types of user for this application.

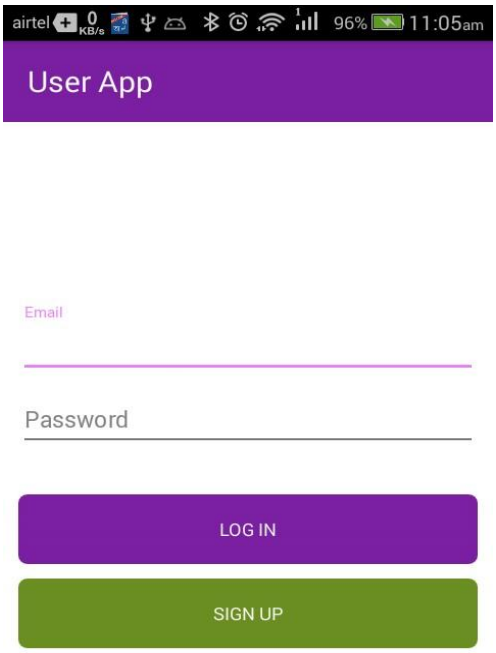


Figure 3.4: Log in page

**General User:** They are the passengers. They can create or update account to our system through our application.

**Control Room User:** They are authorized users, who worked in control room. They monitor all vessels live. They also have the access to add a new vessel in our database and also can view the pinpoint of the overloaded position of a vessel in google map. They can't register through this application. Only super admin who has the access to our database can add or modify this kind of users account.

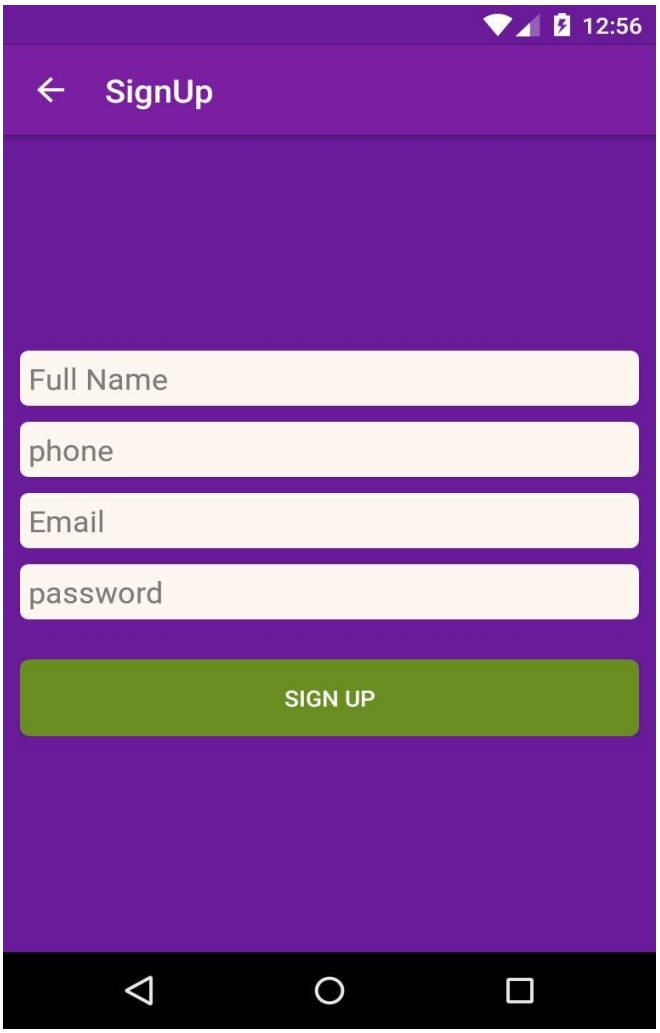


Figure 3.5: User Sign up Page (Normal User)

### 3.2.1.1Functionality for Normal User

- **Show All Journey:** After log-in user directly go the welcome page. Here all his previous journeys are showed. The user can also see the details or update of his journey by touching that journey list. In the bottom of this page there is a floating add button, which takes the user to Set Journey page.

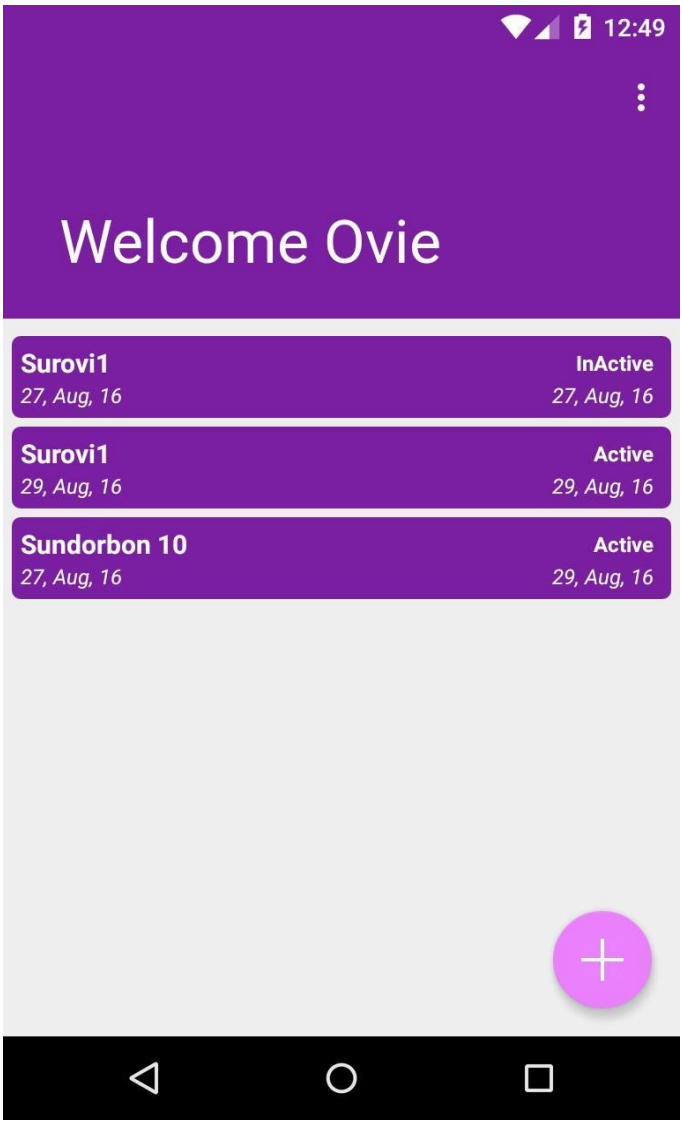

Figure 3.6: Show All Journey (Normal User)

- **Set journey:** User can add his journey by selecting the vessel name, travel start and end time.

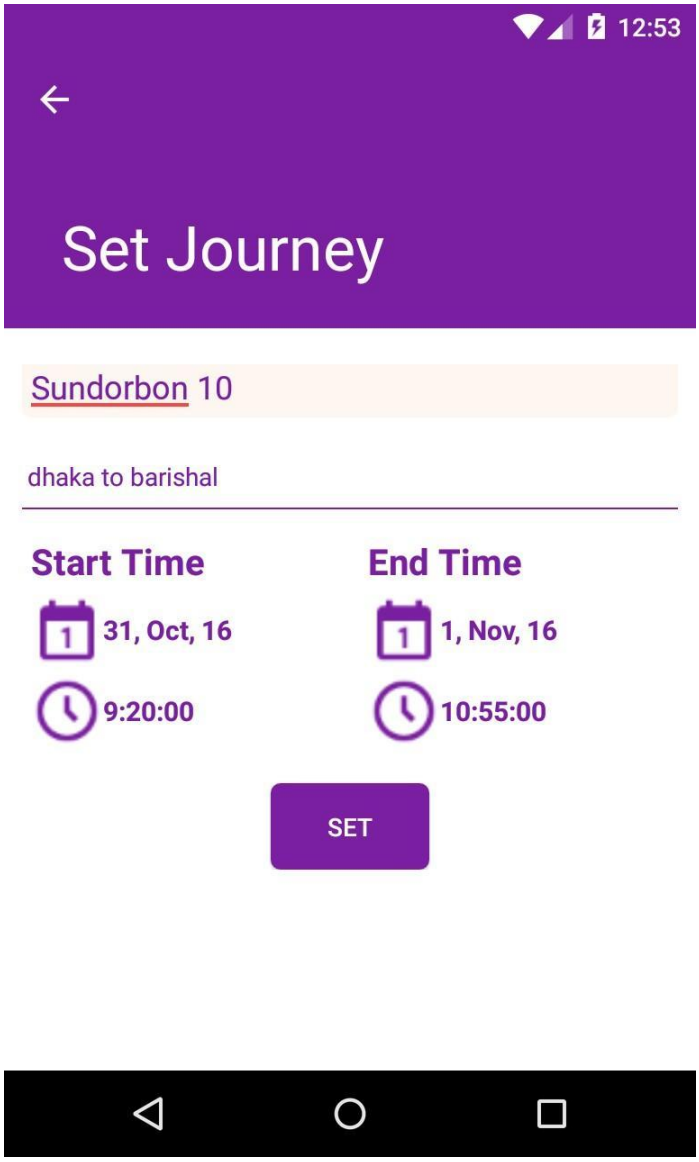


Figure 3.7: Set Journey Page

- **Update Journey:** This user can update his journey. User can also show all the alert and save notification on that vessel during their journey time.

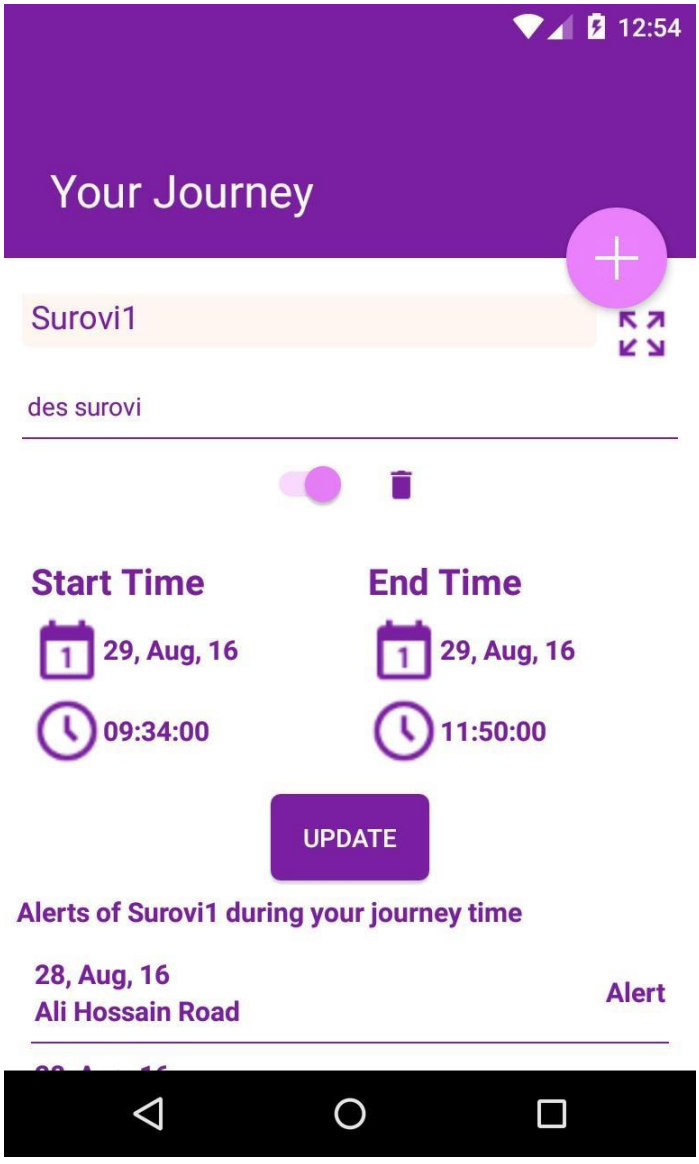


Figure 3.8: Update Journey

- **Get Push Notification:** User gets push notification if the vessel that he add to his journey gets overloaded or resolved his overloaded situation on his journey time. If the vessel gets overloaded then user gets the alert notification. On the other hand if the vessel resolved his overloaded situation then user gets a safe notification.

## 3.2.1.2Functionality for Control Room User

- **Monitoring vessels:** Here user sees all the overloaded vessels list on that time. List will update in live. In upper side of the page there is a sorting option where user can sort the time as his wish. This sorting option helps user to find the overloaded vessels list on a specific time period.

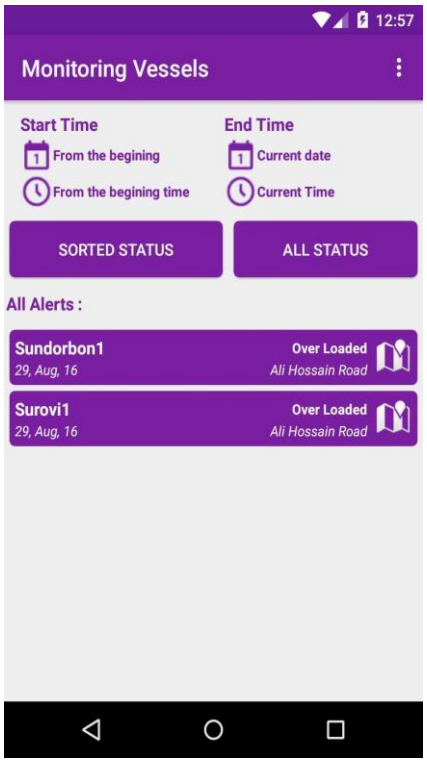


Figure 3.9: Monitoring vessels (Control Room User)

- **Position of an overloaded vessel:** Control room user has the special access to see the overloaded vessel's last overloading position in google map. After knowing the position, control room can send immediate rescue team to that location and it may saves a lot of lives.


Figure 3.10: Position of a vessel in google map (Control Room User)

- **Add Vessel:** Authorized user can add vessel in our system.

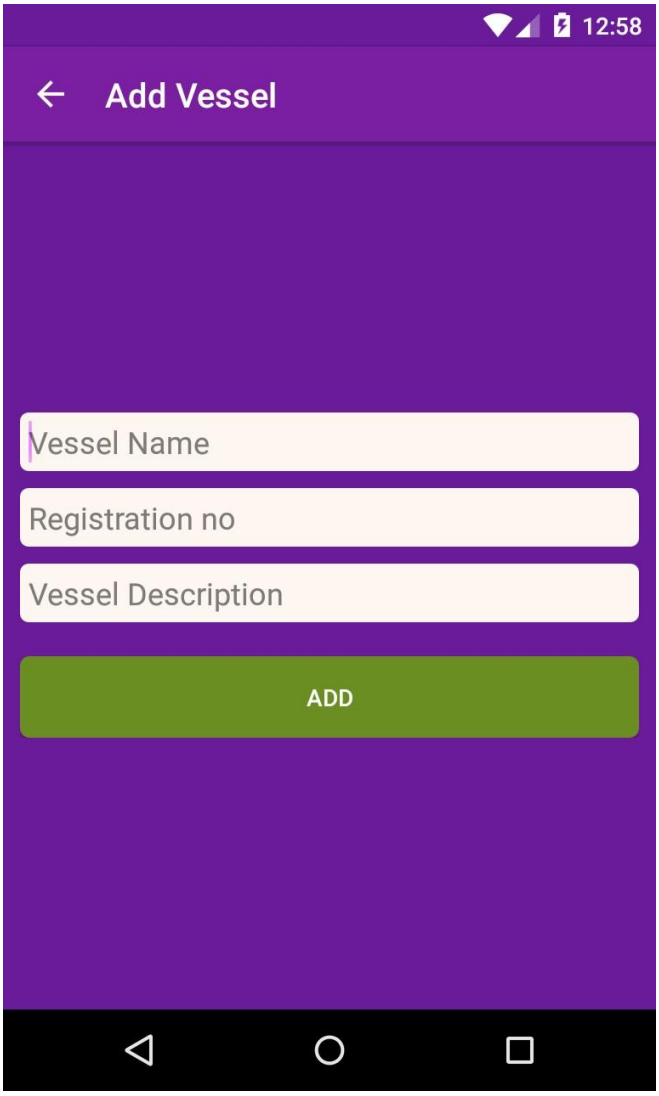


Figure 3.11: Add Vessel (Control Room User)

- **Show and update vessel:** They can show the vessel list and authorized to update a vessel. User also can search vessel.

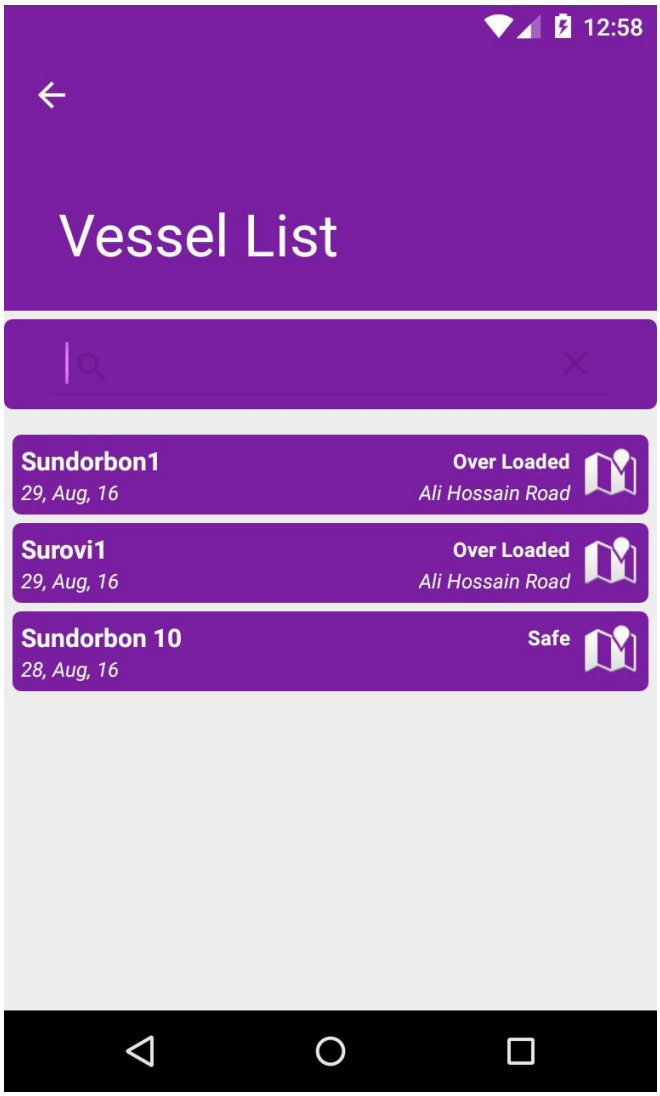


Figure 3.12: Vessel List (Control Room User)

## 3.2.2 Vessel Control Application

This application communicate with the Arduino that is placed in a vessel via Bluetooth. Arduino monitors the floating position of the vessel. If vessel gets overloaded, Arduino sends an alert signal to our application .After getting the alert signal from Arduino our app sends alert signal to the server and server sends a push notification to all the passengers who registered their journey on that vessel during that time. When vessel resolved its overloaded situation and gets its neutral position this app then sends save notification to all those passengers.

User needs to manually pair the HC-05 Bluetooth device with that phone (phone where our application was installed).

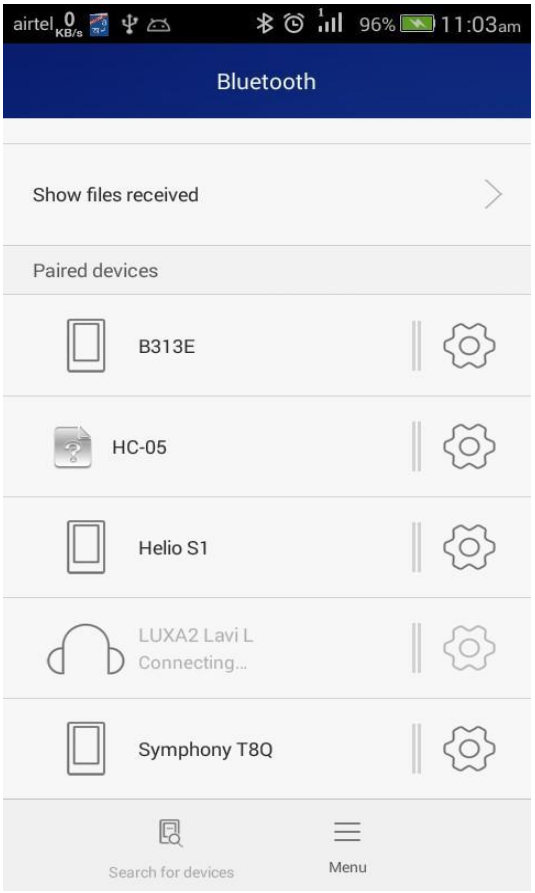


Figure 3.13: Pairing HC-05 Bluetooth device

### 3.2.2.1 Functionality of Vessel Control Application

- **Vessel Choose:** There is a vessel list in the first page of this app. Vessel captain choose his vessel from that list. Here is a search functionality for searching a specific vessel.

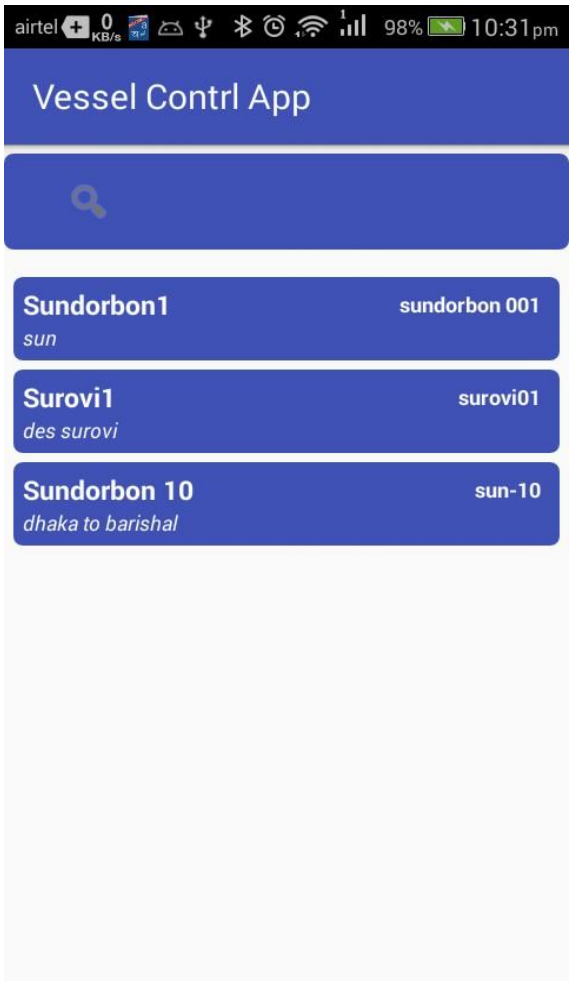


Figure 3.14: Vessel Choosing (Vessel Control App)

• **Choose Bluetooth Device:** After select vessel user sees the Bluetooth paired devices list of that phone. User needs to choose the HC-05 Bluetooth device from that list.

Figure 3.15: Bluetooth paired devices (Vessel Control App)

• **Connect HC-05 and talk with Arduino:** After select HC-05 Bluetooth device user can connect with this by clicking a button. When connection is established with our application and HC-05, our application can directly communicate with Arduino.

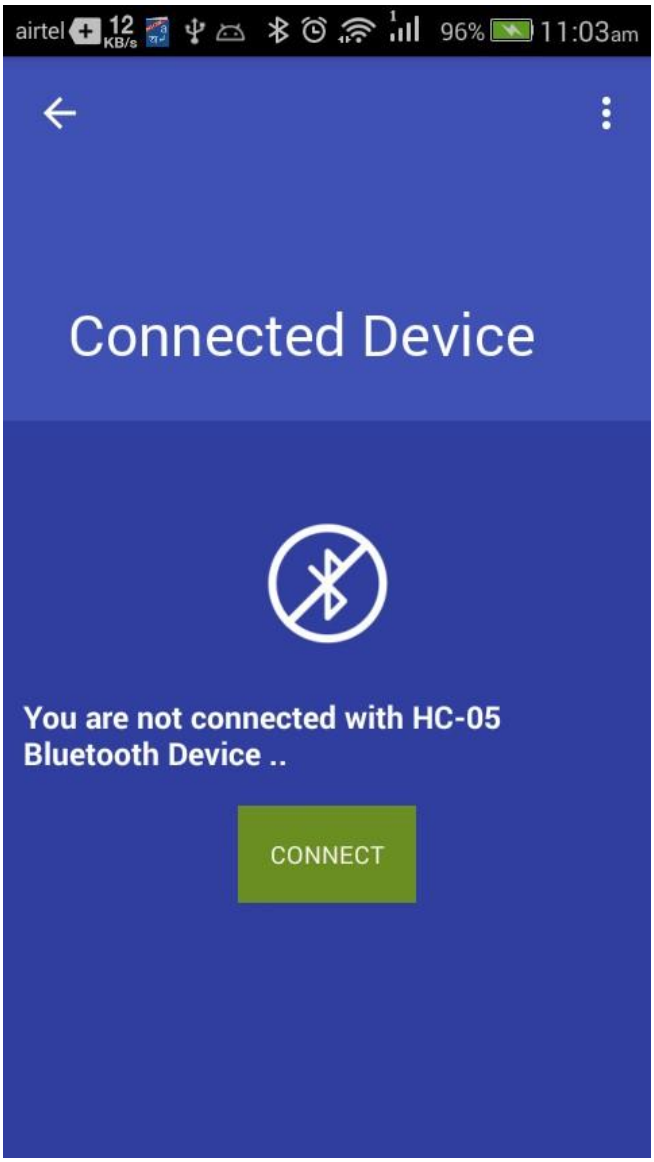

Figure 3.16: Before Connecting with HC-05 (Vessel Control App)

After stablishing the connection there is an option to disconnect option, which disconnects the Bluetooth connection of our android application and Arduino.

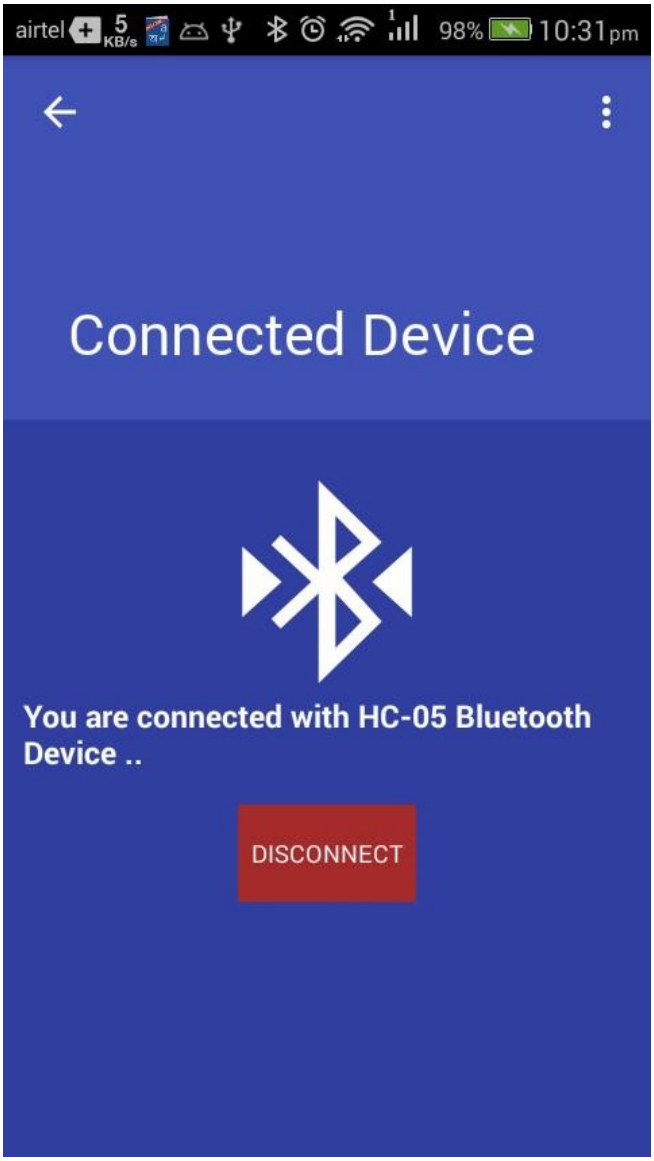

Figure 3.17: After Connecting with HC-05 (Vessel Control App)

- **Send alert and safe notification to passengers and control room:** This is the main part of this application. App gets information from Arduino. If Arduino sends an alert or save signal to our application, our application immediately sends that notification in our server and server sends push notification based on that signal type to the control room and specific passengers.

# CHAPTER 4

## ADVANTAGES & DISADVANTAGES

# 4.1 Advantages

## 4.1.1 Reduce Vessel Accident and Save Lives

There is a verse in Quran “if anyone saves a life, it shall be as though he had saved the lives of all mankind.”

So noting can be compared with a life. 20% off vessel accident happened because of vessel overloading problem and it caused the loss of many lives. Our system reduce vessel accidents and save lots of lives.

## 4.1.2 Monitoring All Vessels

In current system there is now digital system for monitoring all vessels. So many unfit and unregistered vessels are running today. In our system only registered vessel can run and all these registered vessel details are stored in our database. Control room has the access to monitoring all these vessels.

## 4.1.3 Easy to Configure

This system is so simple to configure that anyone can configure it. After putting the switches in the right position of a vessel vessel captain just need to pair the HC-05 Bluetooth device with an android phone and vessel the “Vessel Control Application” on that phone.

If we think about normal user, there configuration is also simple. They just need to create an account which is very easy. After creating an account they just need to create his journey for getting alerts of that vessel, our user friendly UI makes it very easy.

### 4.1.4 Large Scope to Extend Features

It is still a demo system and contained a very limited features. Right now user can only add their journey. There is not ticket buying or selling option right now, it can be added in near future. The possibilities are endless.

## 4.2 Disadvantages

### 4.2.1 Extreme River Waves

Sometime river gone crazy and makes extreme waves, on that time our vertical floating water level switches may not worked as we expect because of extreme waves. We try to manage some worst case scenarios, but we have no hand over nature and anything can happened then.

So when nature gone wild our system may not work perfectly.

# CHAPTER 5

## CONCLUSION

## 5.1 Summary

Completing an IOT project requires a great deal of knowledge and commitment. During this endeavor, it was necessary to gain a considerable knowledge on Microprocessors, Java, PHP, and Android Application Development, know how a network works, acquire a more comprehensive understanding of electronics and of course, obtain the skill to utilize all of these together to assemble a fully functional system. This project has been an effort to build a system that can reduce vessel accidents and save lives. This system can be implemented easily in our country and it is cost efficient. It can be used for monitoring inland vessels. As passenger also being notified by a vessel's overloaded condition, so passenger need not to depend on control room for the status of a vessel.

## 5.2 Future Improvement

Time and other resources is always factor in a better product. There is always room for improvement and this project has a lot of areas that needs work. Now our system sends vessel's position while it is overloaded. In future we want to monitor all the vessels live and create an internal network for communicating with one vessel to another. Our Vertical floating level switches are not advanced enough, we are working on it hopefully very soon we can upgrade these switches.

# APPENDIX

## MicroController:

We use Arduino as a MicroController. All the microcontroller codes are given below.

**Monitoring_vessel_floating_position.ino:**

This code is uploaded to Arduino. It check the floating position of vessel and communicate with Vessel Control Application via Bluetooth HC-05 Device.

```
//Arduino Rx is not connected for Send data in phone

byte leftpin = 5,rightpin=6,f_pin=7,b_pin=8;
int c_r=0,c_l=0,c_f=0,c_b=0;
int sec=1;
int f_o_l=0,f_l,f_o_r=0,f_r=0,f_o_f=0,f_f=0,f_o_b=0,f_b=0;
int f_alert_status=0;

void setup() {
  Serial.begin(9600);
  pinMode(leftpin, OUTPUT);
  pinMode(rightpin, OUTPUT);
  pinMode(f_pin, OUTPUT);
  pinMode(b_pin, OUTPUT);

}

void loop() {

  //Left start

  if(digitalRead(leftpin) == HIGH)
  {
    c_l=c_l+1;
   // Serial.println(1);
    if(c_l>sec)
    {
      if(c_r<sec || c_f<sec || c_b<sec )
      {
        // Only left side is effected
        if(f_o_l== 0)
```

```
      {
       // previosly not send the same alert =P .. send left side is effected Alert
       f_o_l=1;
       alert_only_launch(1, 1);
      }
    }

  }

}

 if(digitalRead(leftpin) == LOW)
 {
   //Serial.println("Disconnect Left");
  if(c_l>sec)
  {
    if(c_r<sec || c_f<sec || c_b<sec )
    {
      // Only left side was effected ... now this alert is solved ..send solved message
        f_o_l=0;
        alert_only_launch(1, 0);

    }
  }


   if(c_r>sec && c_f>sec && c_l>sec && c_b>sec)
   {
    f_alert_status=0;
    alert_solved();
   }

  c_l=0;

 }

//left end

//Right start

 if(digitalRead(rightpin) == HIGH)
 {
  c_r=c_r+1;
```

```
  Serial.println(2);
  if(c_r>sec)
  {
    if(c_l<sec || c_f<sec || c_b<sec )
    {
      // Only Right side is effected
      if(f_o_r== 0)
      {
        // previosly not send the same alert =P .. send left side is effected Alert
        f_o_r=1;
        alert_only_launch(2, 1);
      }
    }

  }

}

if(digitalRead(rightpin) == LOW)
{
   Serial.println("Disconnect Right");
  if(c_r>sec)
  {
    if(c_l<sec || c_f<sec || c_b<sec )
    {
       // Only Right side was effected ... now this alert is solved ..send solved message
         f_o_r=0;
         alert_only_launch(2, 0);

    }
  }


   if(c_r>sec && c_f>sec && c_l>sec && c_b>sec)
   {
    f_alert_status=0;
    alert_solved();
   }

  c_r=0;

}
```

```
//Right end


//FRONT start

 if(digitalRead(f_pin) == HIGH)
 {
  c_f=c_f+1;
  //Serial.println(3);
  if(c_f>sec)
  {
    if(c_r<sec || c_l<sec || c_b<sec )
    {
      // Only left side is effected
      if(f_o_f== 0)
      {
        // previosly not send the same alert =P .. send left side is effected Alert
        f_o_f=1;
        alert_only_launch(3, 1);
      }
    }

  }

 }

 if(digitalRead(f_pin) == LOW)
 {
  // Serial.println("Front Disconnect");
  if(c_f>sec)
  {
    if(c_r<sec || c_l<sec || c_b<sec )
    {
       // Only Front side was effected ... now this alert is solved ..send solved message
         f_o_f=0;
         alert_only_launch(3, 0);

    }
  }

   if(c_r>sec && c_f>sec && c_l>sec && c_b>sec)
   {
    f_alert_status=0;
```

```
    alert_solved();
   }

  c_f=0;

 }

//Front  end


//Back start

 if(digitalRead(b_pin) == HIGH)
 {
  c_b=c_b+1;
 //  Serial.println(4);
  if(c_b>sec)
  {
   if(c_r<sec || c_f<sec || c_l<sec )
   {
     // Only BAck side is effected
     if(f_o_b== 0)
     {
       // previosly not send the same alert =P .. send BAck side is effected Alert
       f_o_b=1;
       alert_only_launch(4, 1);
     }
   }

  }

 }

 if(digitalRead(b_pin) == LOW)
 {
 //  Serial.println("Back Disconnect");
  if(c_b>sec)
  {
   if(c_r<sec || c_f<sec || c_l<sec )
   {
      // Only Back side was effected ... now this alert is solved ..send solved message
       f_o_b=0;
       alert_only_launch(4, 0);
```

```
     }
   }

    if(c_r>sec && c_f>sec && c_l>sec && c_b>sec)
    {
     f_alert_status=0;
     alert_solved();
    }

   c_b=0;

 }

 //Back end
 if(c_r>sec && c_f>sec && c_l>sec && c_b>sec)
 {
 if(f_alert_status==0)
 {
   //send alert to phone via bluetooth .. this alert is not send yet ... =D
   alert();
   f_alert_status=1;
 }
 }

delay(1000);

}

  void alert_only_launch(int side, int alert_status) {
    String side_name;
    if(side==1)
    {
      side_name="left";
    }
    if(side==2)
    {
      side_name="Right";
    }
    if(side==3)
    {
      side_name="Front";
    }
```

```
  if(side==4)
  {
    side_name="Back";
  }
 //  Serial.print(side" Side" );
    Serial.print(side_name+" side ");
   Serial.println(alert_status);
}

void alert_solved() {
 // Serial.println("Alert Solved ");
 Serial.println("0");
}

void alert() {
 // Serial.println("Alert Alert");
 Serial.println("1");
}
```

# Server Scripting:

We use PHP as server scripting language.All the important php codes are given below. All codes are available in github. Github Link:https://goo.gl/B8fZ3I

**Index.php**:

It handle all json requests.

```
<?php

error_reporting(-1);
ini_set('display_errors', 'On');

require_once '../include/db_handler.php';
require '.././libs/Slim/Slim.php';

\Slim\Slim::registerAutoloader();

$app = new \Slim\Slim();

// User register
$app->post('/user/register', function () use ($app) {
```

```
    verifyRequiredParams(array('user_name', 'user_email', 'user_phone', 'user_password',
'user_gcm_token','access_level'));

    $name = $app->request->post('user_name');
    $email = $app->request->post('user_email');
    $phone = $app->request->post('user_phone');
    $password = $app->request->post('user_password');
    $gcm_token = $app->request->post('user_gcm_token');
    $access_level = $app->request->post('access_level');

    validateEmail($email);

    $db = new DbHandler();
    $response = $db->createUser($name, $email, $phone, $password, $gcm_token,$access_level);

    echoRespnse(200, $response);
});

// User login
$app->post('/user/login', function () use ($app) {
    // check for required params
    verifyRequiredParams(array('user_password', 'user_email'));

    // reading post params
    $email = $app->request->post('user_email');
    $password = $app->request->post('user_password');

    // validating email address
    validateEmail($email);

    $db = new DbHandler();
    $response = $db->checkLogin($email, $password);

    // echo json response
    echoRespnse(200, $response);
});

$app->post('/addVessel', function () use ($app) {

    verifyRequiredParams(array('vessel_name', 'vessel_registration_no', 'vessel_description'));

    $name = $app->request->post('vessel_name');
    $vessel_registration_no = $app->request->post('vessel_registration_no');
```

```
    $vessel_description = $app->request->post('vessel_description');

 //   echo $name." reg = ".$vessel_registration_no."  des = ".$vessel_description;


    $db = new DbHandler();
    $response = $db->createVessel($name, $vessel_registration_no, $vessel_description);

    echoRespnse(200, $response);
});


$app->post('/getAllVesselsByName', function () use ($app) {
    // check for required params
    verifyRequiredParams(array('input'));

    // reading post params
    $name = $app->request->post('input');

    $db = new DbHandler();
    $result = $db->getAllVesselsByName($name);

    $response["error"] = false;
    $response["vessels"] = array();

    // pushing single chat room into array
    while ($vessel = $result->fetch_assoc()) {
        $tmp = array();
        $tmp["vessel_id"] = $vessel["vessel_id"];
        $tmp["vessel_name"] = $vessel["vessel_name"];
        $tmp["vessel_registration_no"] = $vessel["vessel_registration_no"];
        $tmp["vessel_description"] = $vessel["vessel_description"];
        array_push($response["vessels"], $tmp);
    }

    // echo json response
    echoRespnse(200, $response);
});


/* * *
 * Updating user
 *  we use this url to update user's gcm registration id
```

```
 */
$app->put('/user/:id', function ($user_id) use ($app) {
   global $app;

   //verifyRequiredParams(array('user_gcm_token','user_id'));
   verifyRequiredParams(array('user_gcm_token'));

   $gcm_registration_id = $app->request->put('user_gcm_token');
   //  $user_id = $app->request->put('user_id');

   $db = new DbHandler();
   $response = $db->updateGcmID($user_id, $gcm_registration_id);

   echoRespnse(200, $response);
});


$app->put('/updateJourney/:id', function ($journey_id) use ($app) {
   global $app;

   //verifyRequiredParams(array('user_gcm_token','user_id'));
   verifyRequiredParams(array('user_id', 'vessel_id', 'start_time', 'end_time', 'is_journey_activated'));

   $user_id = $app->request->put('user_id');
   $vessel_id = $app->request->put('vessel_id');
   $start_time = $app->request->put('start_time');
   $end_time = $app->request->put('end_time');
   $is_journey_activated = $app->request->put('is_journey_activated');


   $db = new DbHandler();
   $response = $db->updateJourney($user_id, $vessel_id, $start_time, $end_time,
$is_journey_activated,$journey_id);

   echoRespnse(200, $response);
});

$app->put('/updateVesel/:id', function ($vessel_id) use ($app) {
   global $app;

   //verifyRequiredParams(array('user_gcm_token','user_id'));
   verifyRequiredParams(array('id', 'reg', 'des', 'name'));
```

```
    $id = $app->request->put('id');
    $name = $app->request->put('name');
    $des = $app->request->put('des');
    $reg = $app->request->put('reg');


    $db = new DbHandler();
    $response = $db->updateVessel($id, $name, $des, $reg);

    echoRespnse(200, $response);
});

$app->post('/user/addJourney', function () use ($app) {

    verifyRequiredParams(array('user_id', 'vessel_id', 'start_time', 'end_time', 'is_journey_activated'));

    $user_id = $app->request->post('user_id');
    $vessel_id = $app->request->post('vessel_id');
    $start_time = $app->request->post('start_time');
    $end_time = $app->request->post('end_time');
    $is_journey_activated = $app->request->post('is_journey_activated');

    $db = new DbHandler();
    $response = $db->setJourney($user_id, $vessel_id, $start_time, $end_time, $is_journey_activated);

    echoRespnse(200, $response);
});



$app->post('/user/add', function () use ($app) {

     verifyRequiredParams(array('aa'));
    // verifyRequiredParams(array('vessel_id', 'alert_location', 'alert_time', 'alert_is_resolved'));


    $vessel_id = $app->request->post('aa');
   /* $alert_time = $app->request->post('alert_time');
    $alert_location = $app->request->post('alert_location');
    $alert_is_resolved = $app->request->post('alert_is_resolved');*/

  //  $db = new DbHandler();
```

```
  //  $response = $db->createAlert($vessel_id, $alert_location, $alert_time, $alert_is_resolved);

  echo $vessel_id;


    //echoRespnse(200, $response);
});


$app->post('/addAlert', function () use ($app) {

    verifyRequiredParams(array('vessel_id', 'alert_location', 'alert_time', 'alert_is_resolved','lat','lng'));


    $vessel_id = $app->request->post('vessel_id');
    $alert_time = $app->request->post('alert_time');
    $alert_location = $app->request->post('alert_location');
    $alert_is_resolved = $app->request->post('alert_is_resolved');
    $lat = $app->request->post('lat');
    $lng = $app->request->post('lng');

    $db = new DbHandler();
    $response = $db->createAlert($vessel_id, $alert_location, $alert_time, $alert_is_resolved,$lat,$lng);

    $response_update_status = $db->updateStatus($vessel_id,
$alert_location,$lat,$lng,$alert_time,$alert_is_resolved);


   //   if (true) {
    if ($response['error'] == false) {
        require_once __DIR__ . '/../libs/gcm/gcm.php';
        require_once __DIR__ . '/../libs/gcm/push.php';
        $gcm = new GCM();
        $push = new Push();

        $registration_ids = array();
        $vessel = $db->getVesselById($vessel_id);

        $users = $db->getAllAlertedUsersByVesselId($vessel_id, $alert_time);
        // preparing gcm registration ids array
        foreach ($users as $u) {

            $user_id = $u['user_id'];
```

```
        $gcm_token = $db->getUserGcmTokenByid($user_id);
        array_push($registration_ids, $gcm_token['user_gcm_token']);
    }

    $access_level=1;
    $users = $db->getControlRoomUsers($access_level);

    foreach ($users as $u) {

        $user_id = $u['user_id'];
        $gcm_token = $db->getUserGcmTokenByid($user_id);
        array_push($registration_ids, $gcm_token['user_gcm_token']);
    }

    $data = array();
    $data['vessel'] = $vessel;
    $data['alert_location'] = $alert_location;
    $data['alert_time'] = $alert_time;
    $data['alert_is_resolved'] = $alert_is_resolved;

    if(  $data['alert_is_resolved']==1){
        $push->setTitle("Vessel Overloaded !! ");
    }
     if(  $data['alert_is_resolved']==0){
        $push->setTitle("Relax . Vessel is Not Overloaded now ");
    }



    $push->setIsBackground(FALSE);
    $push->setFlag(2);
    $push->setData($data);

    // echo json_encode($push->getPush());exit;
    // sending push message to a topic
    $gcm->sendMultiple($registration_ids, $push->getPush());
    $response['Gcm_token'] = $registration_ids;
    $response['data'] = $data;
    $response['error'] = false;
}


echoRespnse(200, $response);
```

```
});


$app->get('/updateStatus', function () {
   $response = array();
   $db = new DbHandler();

   $vessel_id='21';
   $alert_location="Dhaka";
   $lat="lat 10125";
   $lng="lng 124";
   $alert_time="";
   $alert_is_resolved='0';

   // fetching all user tasks
    $response_update_status = $db->updateStatus($vessel_id,
$alert_location,$lat,$lng,$alert_time,$alert_is_resolved);;

  //  $response["error"] = false;


   echoRespnse(200, $response_update_status);
});
/*$app->get('/getControlRoomUser', function () {
   $response = array();
   $db = new DbHandler();

    $access_level=1;
   // fetching all user tasks
   $users = $db->getControlRoomUsers($access_level);
  //  echo $users;

     foreach ($users as $u) {
         $user_id = $u['user_id'];
         echo "User id in index ".$user_id;
         //$gcm_token = $db->getUserGcmTokenByid($user_id);
        //  array_push($registration_ids, $gcm_token['user_gcm_token']);
       }
     $response["error"] = false;
     $response["vessels"] = array();
```

```
    echoRespnse(200, $response);
});*/

/* * *
 * fetching all chat rooms
 */
$app->get('/getAllVessels', function () {
    $response = array();
    $db = new DbHandler();

    // fetching all user tasks
    $result = $db->getAllVessels();

    $response["error"] = false;
    $response["vessels"] = array();


    // pushing single chat room into array
    while ($vessel = $result->fetch_assoc()) {
        $tmp = array();
        $tmp["vessel_id"] = $vessel["vessel_id"];
        $tmp["vessel_name"] = $vessel["vessel_name"];
        $tmp["vessel_registration_no"] = $vessel["vessel_registration_no"];
        $tmp["vessel_description"] = $vessel["vessel_description"];


        $result_status = $db->getStatusByVesselId($vessel["vessel_id"]);

        while ($s = $result_status->fetch_assoc()) {

            $tmp["vessel_status_id"] = $s["status_id"];

            $tmp["vessel_status"] = $s["is_resolved"];
            $tmp["vessel_location"] = $s["status_place"];
            $tmp["vessel_status_time"] = $s["status_time"];
            $tmp["vessel_lat"] = $s["latitude"];
            $tmp["vessel_lng"] = $s["longitude"];

        }



        //$tmp["vessel_status"] = $is_resolved ;
```

```
    array_push($response["vessels"], $tmp);

  }

  echoRespnse(200, $response);
});


$app->get('/getAllAlertedStatus', function () {
  $response = array();
  $db = new DbHandler();

  // fetching all user tasks
  $result = $db->getAllAlertedStatus();

  $response["error"] = false;
  $response["status"] = array();

  // pushing single chat room into array
  while ($status = $result->fetch_assoc()) {
    $tmp = array();
    $tmp["vessel_id"] = $status["vessel_id"];
    $tmp["status_time"] = $status["status_time"];
    $tmp["status_place"] = $status["status_place"];
    $tmp["is_resolved"] = $status["is_resolved"];
    $tmp["latitude"] = $status["latitude"];
    $tmp["longitude"] = $status["longitude"];

    $vessel = $db->getVesselById( $status["vessel_id"]);

    $tmp["vessel"]=$vessel;

    array_push($response["status"], $tmp);
  }

  echoRespnse(200, $response);
});
```

```
$app->post('/getAllAlertedStatusByTime', function () use ($app) {
//$app->get('/getAllAlertedStatus', function () {

   verifyRequiredParams(array('start_time', 'end_time'));

   $response = array();
   $db = new DbHandler();

   $start_time = $app->request->post('start_time');
   $end_time = $app->request->post('end_time');


   // fetching all user tasks
   $result = $db->getAllAlertedStatusByTime($start_time,$end_time);

   $response["error"] = false;
   $response["status"] = array();

   // pushing single chat room into array
   while ($status = $result->fetch_assoc()) {
      $tmp = array();
      $tmp["vessel_id"] = $status["vessel_id"];
      $tmp["status_time"] = $status["status_time"];
      $tmp["status_place"] = $status["status_place"];
      $tmp["is_resolved"] = $status["is_resolved"];
      $tmp["latitude"] = $status["latitude"];
      $tmp["longitude"] = $status["longitude"];

      $vessel = $db->getVesselById( $status["vessel_id"]);

      $tmp["vessel"]=$vessel;

      array_push($response["status"], $tmp);
   }

   echoRespnse(200, $response);
});
$app->post('/test', function () use ($app) {
//$app->get('/getAllAlertedStatus', function () {
   echo "hello";
  verifyRequiredParams(array('vessel_id','start_time', 'end_time'));
});
$app->post('/getAllCurrentUserAertByVessel_id', function () use ($app) {
```

```
//$app->get('/getAllAlertedStatus', function () {

   verifyRequiredParams(array('vessel_id','start_time', 'end_time'));
   $response = array();
   $db = new DbHandler();
   $alert_array = array();
  /* $start_time = "2016-08-29 00:11:36";
   $end_time = "2016-08-29 16:37:00";
   $vessel_id ='24';*/

   $vessel_id = $app->request->post('vessel_id');
   $start_time = $app->request->post('start_time');
   $end_time = $app->request->post('end_time');
   $result = $db->getAllCurrentUserAlertByVesselId($vessel_id,$start_time,$end_time);
    $response["error"] = false;
   $temp_alert_id=-10;
   while ($alert = $result->fetch_assoc()) {
      $tmp = array();
      $tmp["vessel_id"] = $alert["vessel_id"];
      $tmp["alert_id"] = $alert["alert_id"];
      $tmp["alert_time"] = $alert["alert_time"];
      $tmp["alert_location"] = $alert["alert_location"];
      $tmp["alert_is_resolved"] = $alert["alert_is_resolved"];

     //  array_push($alert_array, $tmp);

        if ($temp_alert_id != $tmp["alert_id"] ) {
             $temp_alert_id= $tmp["alert_id"] ;
            array_push($alert_array, $tmp);
          }

   }
   $response["alerts"] = $alert_array;
   echoRespnse(200, $response);
});


//$app->get('/getAllUserJourneyAlertByVessel_id/:id/:start_time/:end_time',                    function
($vessel_id,$start_time,$end_time) {
   /*$app->get('/getAllCurrentUserAertByVessel_id/:id', function ($vessel_id) {

   global $app;
```

```
    $alert_array = array();
    $db = new DbHandler();

    $start_time = "2016-08-29 00:11:36";
    $end_time = "2016-08-29 16:37:00";


     $result = $db->getAllCurrentUserAlertByVesselId($vessel_id,$start_time,$end_time);



   // $result = $db->getJourneyByVesselId($vessel_id,$start_time,$end_time);
    // $result = $db->getAllCurrentUserAlertByVesselId($vessel_id);

     $response["error"] = false;

   // echo "Id ". $vessel_id;
     $temp_alert_id=-10;

    while ($alert = $result->fetch_assoc()) {

       $tmp = array();
       $tmp["vessel_id"] = $alert["vessel_id"];
       $tmp["alert_id"] = $alert["alert_id"];
       $tmp["alert_time"] = $alert["alert_time"];
       $tmp["alert_location"] = $alert["alert_location"];
       $tmp["alert_is_resolved"] = $alert["alert_is_resolved"];

      //  array_push($alert_array, $tmp);

          if ($temp_alert_id != $tmp["alert_id"] ) {
               $temp_alert_id= $tmp["alert_id"] ;
              array_push($alert_array, $tmp);
           }

       //$temp_alert_id = $tmp["alert_id"] ;



    }
    $response["alerts"] = $alert_array;
    echoRespnse(200, $response);
});*/
```

```
$app->post('/getAllAlertForControlRoom', function () use ($app) {

  // $app->get('/getAllAlertForControlRoom', function () {

   global $app;

   verifyRequiredParams(array('start_time', 'end_time'));

   $start_time = $app->request->post('start_time');
   $end_time = $app->request->post('end_time');

   $alert_array = array();
   $db = new DbHandler();

  // $result = $db->getJourneyByVesselId($vessel_id,$start_time,$end_time);
    $result = $db->getAllCurrentUserAlertForControlRoomInBetweenTime($start_time,$end_time);

    $response["error"] = false;

   $temp_alert_id=-10;

   while ($alert = $result->fetch_assoc()) {

      $tmp = array();
      $tmp["vessel_id"] = $alert["vessel_id"];
      $tmp["alert_id"] = $alert["alert_id"];
      $tmp["alert_time"] = $alert["alert_time"];
      $tmp["alert_location"] = $alert["alert_location"];
      $tmp["alert_is_resolved"] = $alert["alert_is_resolved"];


      if ($temp_alert_id != $tmp["alert_id"] ) {
         array_push($alert_array, $tmp);
      }

      $temp_alert_id = $tmp["alert_id"] ;

   }
   $response["alerts"] = $alert_array;
   echoRespnse(200, $response);
});
```

```
$app->get('/getAllJourneyByUser_id/:id', function ($user_id) {
   global $app;
   $vessel_array = array();
   $journey_array = array();


   $db = new DbHandler();

   date_default_timezone_set('Asia/Dhaka');
   $current_time = date('Y-m-d H:i:s');

   $result = $db->getAllRegisteredJourniesByUserId($user_id, $current_time);

   $response["error"] = false;


   while ($journey = $result->fetch_assoc()) {
      $tmp = array();
      $tmp["vessel_id"] = $journey["vessel_id"];
      $vessel = $db->getVesselById($journey["vessel_id"]);
      //$r["vessels"] = $vessel;
     // $r["journey"] = $journey;
     //  echoRespnse(200, $r);
      array_push($vessel_array, $vessel);
      array_push($journey_array, $journey);
   }

   $response["vessels"] = $vessel_array;
    $response["journey"] = $journey_array;
   echoRespnse(200, $response);
});


$app->post('/users/push_test', function () {
   global $app;

   verifyRequiredParams(array('message', 'api_key', 'token'));

   $message = $app->request->post('message');
   $apiKey = $app->request->post('api_key');
```

```
$token = $app->request->post('token');
$image = $app->request->post('include_image');

$data = array();
$data['title'] = 'Google Cloud Messaging';
$data['message'] = $message;
if ($image == 'true') {
   $data['image'] = 'http://api.androidhive.info/gcm/panda.jpg';
} else {
   $data['image'] = '';
}
$data['created_at'] = date('Y-m-d G:i:s');

$fields = array(
   'to' => $token,
   'data' => $data,
);

// Set POST variables
$url = 'https://gcm-http.googleapis.com/gcm/send';

$headers = array(
   'Authorization: key=' . $apiKey,
   'Content-Type: application/json'
);
// Open connection
$ch = curl_init();

// Set the url, number of POST vars, POST data
curl_setopt($ch, CURLOPT_URL, $url);

curl_setopt($ch, CURLOPT_POST, true);
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);

// Disabling SSL Certificate support temporarly
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);

curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($fields));

$response = array();

// Execute post
```

```
    $result = curl_exec($ch);
    if ($result === FALSE) {
        $response['error'] = TRUE;
        $response['message'] = 'Unable to send test push notification';
        echoRespnse(200, $response);
        exit;
    }

    // Close connection
    curl_close($ch);

    $response['error'] = FALSE;
    $response['message'] = 'Test push message sent successfully!';

    echoRespnse(200, $response);
});



/**
 * Verifying required params posted or not
 */
function verifyRequiredParams($required_fields)
{
    $error = false;
    $error_fields = "";
    $request_params = array();
    $request_params = $_REQUEST;
    // Handling PUT request params
    if ($_SERVER['REQUEST_METHOD'] == 'PUT') {
        $app = \Slim\Slim::getInstance();
        parse_str($app->request()->getBody(), $request_params);
    }
    foreach ($required_fields as $field) {
        if (!isset($request_params[$field]) || strlen(trim($request_params[$field])) <= 0) {
            $error = true;
            $error_fields .= $field . ', ';
        }
    }

    if ($error) {
        // Required field(s) are missing or empty
        // echo error json and stop the app
```

```
        $response = array();
        $app = \Slim\Slim::getInstance();
        $response["error"] = true;
        $response["message"] = 'Required field(s) ' . substr($error_fields, 0, -2) . ' is missing or empty';
        echoRespnse(400, $response);
        $app->stop();
    }
}

/**
 * Validating email address
 */
function validateEmail($email)
{
    $app = \Slim\Slim::getInstance();
    if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        $response["error"] = true;
        $response["message"] = 'Email address is not valid';
        echoRespnse(400, $response);
        $app->stop();
    }
}

function IsNullOrEmptyString($str)
{
    return (!isset($str) || trim($str) === '');
}

/**
 * Echoing json response to client
 * @param String $status_code Http response code
 * @param Int $response Json response
 */
function echoRespnse($status_code, $response)
{
    $app = \Slim\Slim::getInstance();
    // Http response code
    $app->status($status_code);

    // setting response content type to json
    $app->contentType('application/json');

    echo json_encode($response);
```

```
}

$app->run();
?>
```

**DbHandler.php:**

It handle all database operations in server.

```
<?php


class DbHandler
{

   private $conn;

   function __construct()
   {
      require_once dirname(__FILE__) . '/db_connect.php';
      // opening db connection
      $db = new DbConnect();
      $this->conn = $db->connect();
   }

   // creating new user if not existed
   public function createUser($name, $email, $phone, $password, $gcm_token,$access_level)
   {
      require_once 'PassHash.php';
      $response = array();

      // First check if user already existed in db
      if (!$this->isUserExists($email)) {
         // insert query
        //  $password_hash = PassHash::hash($password);
         $password_hash = $password;
         $stmt = $this->conn->prepare("INSERT INTO tbl_user(user_email,
user_name,user_phone,user_password,user_gcm_token,access_level) values(?,?,?,?,?,?)");
         $stmt->bind_param("sssssi", $email, $name, $phone, $password, $gcm_token,$access_level);

         $result = $stmt->execute();
         $stmt->close();
         // Check for successful insertion
```

```
        if ($result) {
            // User successfully inserted
            $response["error"] = false;
            $response["user"] = $this->getUserByEmail($email);
        } else {
            // Failed to create user
            $response["error"] = true;
            $response["message"] = "Oops! An error occurred while registereing";
        }
    } else {
        // User with same email already existed in the db
        $response["error"] = false;
        $response["user"] = $this->getUserByEmail($email);
    }

    return $response;
  }

  public function createVessel( $vessel_name, $vessel_registration_no, $vessel_description)
  {
    $response = array();

   //      $stmt      =      $this->conn->prepare("INSERT      INTO      tbl_vessel(vessel_name,
vessel_registration_no,vessel_description) values(?,?,?");
    $stmt      =      $this->conn->prepare("INSERT      INTO      tbl_vessel(vessel_name,
vessel_registration_no,vessel_description) values(?,?,?)");
    $stmt->bind_param("sss", $vessel_name, $vessel_registration_no, $vessel_description);

    $result = $stmt->execute();


    // Check for successful insertion
    if ($result) {
        // User successfully inserted
        $vessel_id = $this->conn->insert_id;
       // echo "Vessel id : ".$vessel_id;
        $this->createVesselStatus($vessel_id);

        $response["error"] = false;
        $response["message"] = "Vessel added successfuully";

    } else {
        // Failed to create user
```

```
        $response["error"] = true;
        $response["message"] = "Oops! An error occurred while set your journey";
      }

       $stmt->close();
      return $response;
   }
   public function createVesselStatus($vessel_id)
   {
      $response = array();
      $status=0;
     // echo "Before  == vessel id : ".$vessel_id."  status : ".$status;
      $stmt = $this->conn->prepare("INSERT INTO tbl_status(vessel_id) values(?)");
      $stmt->bind_param("i", $vessel_id);

    //  echo "after == vessel id : ".$vessel_id."  status : ".$status;
     $result = $stmt->execute() ;
      $stmt->close();

      // Check for successful insertion
      if ($result) {

      } else {

      }
     // return $response;
   }

   public function setJourney($user_id, $vessel_id, $start_time, $end_time, $is_journey_activated)
   {
      $response = array();

      $stmt = $this->conn->prepare("INSERT INTO tbl_journey(user_id,
vessel_id,start_time,end_time,is_journey_activated) values(?,?,?,?,?)");
      $stmt->bind_param("iissi", $user_id, $vessel_id, $start_time, $end_time, $is_journey_activated);

      $result = $stmt->execute();

      $stmt->close();

      // Check for successful insertion
      if ($result) {
        // User successfully inserted
```

```
            $response["error"] = false;
            $response["vessel"] = $this->getVesselById($vessel_id);
            $response["user"] = $this->getUserById($user_id);
        } else {
            // Failed to create user
            $response["error"] = true;
            $response["message"] = "Oops! An error occurred while set your journey";
        }
        return $response;
    }
    public function createAlert($vessel_id, $alert_location, $alert_time, $alert_is_resolved,$lat,$lng)
    {
   $response = array();
        $stmt = $this->conn->prepare("INSERT INTO tbl_alert(vessel_id,
alert_location,alert_time,alert_is_resolved,latitude,longitude) values(?,?,?,?,?,?)");
        $stmt->bind_param("ississ", $vessel_id, $alert_location, $alert_time, $alert_is_resolved,$lat,$lng);
        $result = $stmt->execute();
        $stmt->close();
        // Check for successful insertion
        if ($result) {
            // User successfully inserted
          //  $alert_id = $this->conn->insert_id;
            $response["error"] = false;
            $response["alert_time"] = $alert_time;
            $response["alert_location"] = $alert_location;
            $response["alert_is_resolved"] = $alert_is_resolved;
            $response["lat"] = $lat;
            $response["lng"] = $lng;
        } else {
            // Failed to create user
            $response["error"] = true;
            $response["message"] = "Oops! An error occurred while creating Alert";
        }
        return $response;
    }
    public function getVesselById($id)
    {
        $stmt = $this->conn->prepare("SELECT vessel_id, vessel_name, vessel_registration_no,
vessel_description FROM tbl_vessel WHERE vessel_id = ?");
        $stmt->bind_param("i", $id);
        if ($stmt->execute()) {
            // $user = $stmt->get_result()->fetch_assoc();
            $stmt->bind_result($id, $name, $reg, $des);
```

```
        $stmt->fetch();
        $vessel = array();
        $vessel["vessel_id"] = $id;
        $vessel["vessel_name"] = $name;
        $vessel["vessel_registration_no"] = $reg;
        $vessel["vessel_description"] = $des;
        $stmt->close();
        return $vessel;
    } else {
        return NULL;
    }

public function checkLogin($email, $password)
{
    require_once 'PassHash.php';
    $response = array();
    // fetching user by email
    $stmt = $this->conn->prepare("SELECT user_password FROM tbl_user WHERE user_email = ?");
    $stmt->bind_param("s", $email);
    $stmt->execute();
    $stmt->bind_result($password_hash);
    $stmt->store_result();
    if ($stmt->num_rows > 0) {
        $stmt->fetch();
        $stmt->close();
      //  if (PassHash::check_password($password_hash, $password)) {
            if ($password_hash== $password) {
            // User password is correct
            $response["error"] = false;
            $response["user"] = $this->getUserByEmail($email);
            return $response;
        } else {
            // user password is incorrect
            $response["error"] = true;
            $response["message"] = "OOPS ! Password Does not Match ! \n Please Try Again";
            return $response;
        }
    } else {
        $stmt->close();

        $response["error"] = true;
        $response["message"] = "OOPS ! Email Does not Exist ! \n Please Try Again";;
        return $response;
```

```
        }
    }
    // updating user GCM registration ID
    public function updateGcmID($user_id, $gcm_registration_id)
    {
        $response = array();
        $stmt = $this->conn->prepare("UPDATE tbl_user SET user_gcm_token = ? WHERE user_id = ?");
        $stmt->bind_param("si", $gcm_registration_id, $user_id);

        if ($stmt->execute()) {
            // User successfully updated
            $response["error"] = false;
            $response["message"] = 'GCM registration ID updated successfully'.$gcm_registration_id;
        } else {
            // Failed to update user
            $response["error"] = true;
            $response["message"] = "Failed to update GCM registration ID";
            $stmt->error;
        }
        $stmt->close();
        return $response;
    }
    public function updateStatus($vessel_id, $status_location,$lat,$lng,$status_time,$is_resolved)
    {
        $response = array();
        $stmt = $this->conn->prepare("UPDATE tbl_status SET status_time = ? ,status_place = ?
,is_resolved = ? ,longitude = ? ,latitude = ? WHERE vessel_id = ?");
        $stmt->bind_param("ssissi", $status_time, $status_location,$is_resolved,$lng,$lat,$vessel_id);

        if ($stmt->execute()) {
            // User successfully updated
            $response["error"] = false;
            $response["message"] = 'Vessel Status updated successfully';
        } else {
            // Failed to update user
            $response["error"] = true;
            $response["message"] = "Failed to update Vessel Status";
            $stmt->error;
        }
        $stmt->close();

        return $response;
    }
```

```
 public function updateJourney($user_id, $vessel_id, $start_time, $end_time, $is_journey_activated,$journey_id)
   {
      $response = array();

      $stmt = $this->conn->prepare("UPDATE tbl_journey SET user_id = ? ,vessel_id = ? ,start_time = ? ,end_time = ? ,is_journey_activated = ? WHERE journey_id = ?");
      $stmt->bind_param("iissii", $user_id, $vessel_id, $start_time, $end_time, $is_journey_activated,$journey_id);

      // Check for successful insertion
      if ($stmt->execute()) {
         // User successfully inserted
         $response["error"] = false;
        $response["message"] = 'Journey Updated Successfully ';

      } else {
         // Failed to create user
         $response["error"] = true;
        $response["message"] = 'Journey Update Failed ';
      }
       $stmt->close();
      return $response;
   }

   public function updateVessel($id, $name, $des, $reg)
   {
      $response = array();

      $stmt = $this->conn->prepare("UPDATE tbl_vessel SET vessel_name = ? ,vessel_description = ? ,vessel_registration_no = ?  WHERE vessel_id = ?");
      $stmt->bind_param("sssi", $name, $des, $reg, $id);

      // Check for successful insertion
      if ($stmt->execute()) {
         // User successfully inserted
         $response["error"] = false;
        $response["message"] = 'Vessel Updated Successfully ';

      } else {
         // Failed to create user
         $response["error"] = true;
        $response["message"] = 'Vessel Update Failed ';
```

```
      }
       $stmt->close();
      return $response;
   }
    public function getAllCurrentUserAlertByVesselId($vessel_id,$start_time,$end_time)
  {

     $stmt = $this->conn->prepare("SELECT tbl_alert.alert_id, tbl_alert.vessel_id,tbl_alert.alert_location,
      tbl_alert.alert_time,tbl_alert.alert_is_resolved FROM tbl_alert,tbl_journey WHERE
tbl_alert.vessel_id = ?
      AND tbl_alert.alert_time BETWEEN ? AND ?");

    $stmt->bind_param("iss", $vessel_id,$start_time,$end_time);
    $stmt->execute();
    $alerts = $stmt->get_result();
    $stmt->close();
    return $alerts;

   }

   public function getAllCurrentUserAlertForControlRoomInBetweenTime($start_time,$end_time)
  {

     $stmt = $this->conn->prepare("SELECT tbl_alert.alert_id, tbl_alert.vessel_id,tbl_alert.alert_location,
      tbl_alert.alert_time,tbl_alert.alert_is_resolved FROM tbl_alert WHERE tbl_alert.alert_time
BETWEEN ? AND ?");

    $stmt->bind_param("ss", $start_time,$end_time);
    $stmt->execute();
    $alerts = $stmt->get_result();
    $stmt->close();
    return $alerts;

   }
public function getControlRoomUsers($access_level)
   {

      $users = array();
      $query = "SELECT user_id FROM tbl_user WHERE access_level = '$access_level'";
      $stmt = $this->conn->prepare($query);
      $stmt->execute();
      $result = $stmt->get_result();
```

```
      while ($user = $result->fetch_assoc()) {
         $tmp = array();
         $tmp["user_id"] = $user['user_id'];
        // $tmp["gcm_registration_id"] = $user['gcm_registration_id'];
         array_push($users, $tmp);
      }

return $users;
   }

   public function getUserById($id)
   {
      $stmt = $this->conn->prepare("SELECT user_id, user_name,user_email,
user_phone,user_gcm_token,access_level FROM tbl_user WHERE user_id = ?");
      $stmt->bind_param("i", $id);
      if ($stmt->execute()) {
         // $user = $stmt->get_result()->fetch_assoc();
         $stmt->bind_result($user_id, $name, $email, $phone, $gcm_token,$access_level);
         $stmt->fetch();
         $user = array();
         $user["user_id"] = $user_id;
         $user["user_name"] = $name;
         $user["user_email"] = $email;
         $user["user_phone"] = $phone;
         $user["user_gcm_token"] = $gcm_token;
         $user["access_level"] = $access_level;
         $stmt->close();
         return $user;
      } else {
         return NULL;
      }
   }

   public function getUserByEmail($email)
   {
      $stmt = $this->conn->prepare("SELECT user_id, user_name, user_email,
user_phone,user_gcm_token,access_level FROM tbl_user WHERE user_email = ?");
      $stmt->bind_param("s", $email);
      if ($stmt->execute()) {
         // $user = $stmt->get_result()->fetch_assoc();
         $stmt->bind_result($user_id, $name, $email, $phone, $gcm_token,$access_level);
         $stmt->fetch();
         $user = array();
```

```
            $user["user_id"] = $user_id;
            $user["user_name"] = $name;
            $user["user_email"] = $email;
            $user["user_phone"] = $phone;
            $user["user_gcm_token"] = $gcm_token;
             $user["access_level"] = $access_level;
            $stmt->close();
            return $user;
        } else {
            return NULL;
        }
    }
    public function getUserGcmTokenByid($id)
    {
        $stmt = $this->conn->prepare("SELECT user_gcm_token FROM tbl_user WHERE user_id = ?");
        $stmt->bind_param("i", $id);
        if ($stmt->execute()) {
            // $user = $stmt->get_result()->fetch_assoc();
            $stmt->bind_result($gcm_token);
            $stmt->fetch();
            $user = array();
            $user["user_gcm_token"] = $gcm_token;
            $stmt->close();
            return $user;
        } else {
            return NULL;
        }
    }

    // fetching single user by id
    public function getUser($user_id)
    {
        $stmt = $this->conn->prepare("SELECT user_id, name, email, gcm_registration_id, created_at
FROM users WHERE user_id = ?");
        $stmt->bind_param("s", $user_id);
        if ($stmt->execute()) {
            // $user = $stmt->get_result()->fetch_assoc();
            $stmt->bind_result($user_id, $name, $email, $gcm_registration_id, $created_at);
            $stmt->fetch();
            $user = array();
            $user["user_id"] = $user_id;
            $user["name"] = $name;
            $user["email"] = $email;
```

```
            $user["gcm_registration_id"] = $gcm_registration_id;
            $user["created_at"] = $created_at;
            $stmt->close();
            return $user;
        } else {
            return NULL;
        }
    }

    public function getAllRegisteredJourniesByUserId($user_id,$alert_time)
    {
        $journies = array();
       //  $query = "SELECT * FROM tbl_journey WHERE user_id = '$user_id' AND '$alert_time' BETWEEN
start_time AND end_time";
         $query = "SELECT * FROM tbl_journey WHERE user_id = '$user_id'";
       //  echo $query;
        $stmt = $this->conn->prepare($query);

        $stmt->execute();
        $journey = $stmt->get_result();
        $stmt->close();
        return $journey;

    }

    public function getAllAlertedUsersByVesselId($vessel_id,$alert_time)
    {
        $users = array();
        $query = "SELECT user_id FROM tbl_journey WHERE vessel_id = '$vessel_id' AND '$alert_time'
BETWEEN start_time AND end_time";
        $stmt = $this->conn->prepare($query);
        $stmt->execute();
        $result = $stmt->get_result();

        while ($user = $result->fetch_assoc()) {
            $tmp = array();
            $tmp["user_id"] = $user['user_id'];
           // $tmp["gcm_registration_id"] = $user['gcm_registration_id'];
            array_push($users, $tmp);
        }

return $users;
}
```

```
  public
  function getAllVessels()
  {
     $stmt = $this->conn->prepare("SELECT * FROM tbl_vessel");
     $stmt->execute();
     $vessel = $stmt->get_result();
     $stmt->close();
     return $vessel;
  }

  public function getAllAlertedStatus()
  {
     $stmt = $this->conn->prepare("SELECT * FROM tbl_status WHERE is_resolved = 1");
     $stmt->execute();
     $status = $stmt->get_result();
     $stmt->close();
     return $status;
  }

  public function getAllAlertedStatusByTime($start_time,$end_time)
  {
     $stmt = $this->conn->prepare("SELECT * FROM tbl_status WHERE is_resolved = 1 AND status_time
BETWEEN '$start_time' AND '$end_time'");
     $stmt->execute();
     $status = $stmt->get_result();
     $stmt->close();
     return $status;
  }
   public function getStatusByVesselId($vessel_id)
  {
     $stmt = $this->conn->prepare("SELECT * FROM tbl_status WHERE vessel_id = ?");
     $stmt->bind_param("i", $vessel_id);

     $stmt->execute();
     $status = $stmt->get_result();
     $stmt->close();
     return $status;
  }
  public function getAllVesselsByName($name)
  {
    // $stmt = $this->conn->prepare("SELECT * FROM tbl_vessel WHERE vessel_name = ?");
     $stmt = $this->conn->prepare("SELECT * FROM tbl_vessel WHERE vessel_name LIKE '%$name%'");
```

```
      // $stmt->bind_param("s", $name);
       $stmt->execute();
       $vessel = $stmt->get_result();
       $stmt->close();
       return $vessel;
    }

// fetching multiple users by ids
public
function getUsers($user_ids)
{

    $users = array();
    if (sizeof($user_ids) > 0) {
       $query = "SELECT user_id, name, email, gcm_registration_id, created_at FROM users WHERE
user_id IN (";

       foreach ($user_ids as $user_id) {
          $query .= $user_id . ',';
       }
       $query = substr($query, 0, strlen($query) - 1);
       $query .= ')';

       $stmt = $this->conn->prepare($query);
       $stmt->execute();
       $result = $stmt->get_result();

       while ($user = $result->fetch_assoc()) {
          $tmp = array();
          $tmp["user_id"] = $user['user_id'];
          $tmp["name"] = $user['name'];
          $tmp["email"] = $user['email'];
          $tmp["gcm_registration_id"] = $user['gcm_registration_id'];
          $tmp["created_at"] = $user['created_at'];
          array_push($users, $tmp);
       }
    }

    return $users;
}

// messaging in a chat room / to persional message
public
```

```
function addMessage($user_id, $chat_room_id, $message)
{
   $response = array();

   $stmt = $this->conn->prepare("INSERT INTO messages (chat_room_id, user_id, message) values(?, ?,
?)");
   $stmt->bind_param("iis", $chat_room_id, $user_id, $message);

   if ($result = $stmt->execute()) {
      $response['error'] = false;
      // get the message
      $message_id = $this->conn->insert_id;
      $stmt = $this->conn->prepare("SELECT message_id, user_id, chat_room_id, message, created_at
FROM messages WHERE message_id = ?");
      $stmt->bind_param("i", $message_id);
      if ($stmt->execute()) {
         $stmt->bind_result($message_id, $user_id, $chat_room_id, $message, $created_at);
         $stmt->fetch();
         $tmp = array();
         $tmp['message_id'] = $message_id;
         $tmp['chat_room_id'] = $chat_room_id;
         $tmp['message'] = $message;
         $tmp['created_at'] = $created_at;
         $response['message'] = $tmp;
      }
   } else {
      $response['error'] = true;
      $response['message'] = 'Failed send message ' . $stmt->error;
   }

   return $response;
}

// fetching all chat rooms
public
function getAllChatrooms()
{
   $stmt = $this->conn->prepare("SELECT * FROM chat_rooms");
   $stmt->execute();
   $tasks = $stmt->get_result();
   $stmt->close();
   return $tasks;
}
```

```
// fetching single chat room by id
function getChatRoom($chat_room_id)
{
   $stmt = $this->conn->prepare("SELECT cr.chat_room_id, cr.name, cr.created_at as chat_room_created_at, u.name as username, c.* FROM chat_rooms cr LEFT JOIN messages c ON c.chat_room_id = cr.chat_room_id LEFT JOIN users u ON u.user_id = c.user_id WHERE cr.chat_room_id = ?");
   $stmt->bind_param("i", $chat_room_id);
   $stmt->execute();
   $tasks = $stmt->get_result();
   $stmt->close();
   return $tasks;
}

/**
 * Checking for duplicate user by email address
 * @param String $email email to check in db
 * @return boolean
 */
private
function isUserExists($email)
{
   $stmt = $this->conn->prepare("SELECT user_id from tbl_user WHERE user_email = ?");
   $stmt->bind_param("s", $email);
   $stmt->execute();
   $stmt->store_result();
   $num_rows = $stmt->num_rows;
   $stmt->close();
   return $num_rows > 0;
}
}

?>
```

**Android Application:** Some important code for developing the android applicatrions for this project are given below

**1. Vessel Control Application:** All the codes for this application is available in Github. Github link:https://goo.gl/UKqRkb . Some important codes are only given below.

**BluetoothDeviceListActivity.java:** All blouetooth devices that are paired on that phone are shown in this activity, so that user can connect with HC-05 Bluetooth Device easily.

```
package uniqueid_1010.gcm_chat.ovie_codecirrus.vesselcontrlapp.activity;

import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.TextView;
import android.widget.Toast;

import java.util.ArrayList;
import java.util.Set;

import uniqueid_1010.gcm_chat.ovie_codecirrus.vesselcontrlapp.R;
import
uniqueid_1010.gcm_chat.ovie_codecirrus.vesselcontrlapp.adapter.BluetoothDeviceAdapter;
import uniqueid_1010.gcm_chat.ovie_codecirrus.vesselcontrlapp.helper.LocationClass;
import
uniqueid_1010.gcm_chat.ovie_codecirrus.vesselcontrlapp.helper.MyPreferenceManager;
import
uniqueid_1010.gcm_chat.ovie_codecirrus.vesselcontrlapp.model.BluetoothDeviceClass;
public class BluetoothDeviceListActivity extends AppCompatActivity {

private String TAG = BluetoothDeviceListActivity.class.getSimpleName();
    RecyclerView recyclerView;
    Context context;
//Bluetooth
private BluetoothAdapter myBluetooth = null;
private Set<BluetoothDevice>pairedDevices;
public static String EXTRA_ADDRESS = "device_address";
    TextView tv_lat, tv_lang;
    ArrayList<BluetoothDeviceClass>pairedDeviceClassesArrayList = new ArrayList<>();

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_bluetooth_device_list);
```

```
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
context = this;
recyclerView = (RecyclerView) findViewById(R.id.recyclearList_device_list);

//if the device has bluetooth
myBluetooth = BluetoothAdapter.getDefaultAdapter();
if (myBluetooth == null) {       Toast.makeText(getApplicationContext(), "Bluetooth
Device Not Available", Toast.LENGTH_LONG).show();
// finish();
} else if (!myBluetooth.isEnabled()) {
//Ask to the user turn the bluetooth on
Intent turnBTon = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
startActivityForResult(turnBTon, 1);
        }
    }
public void paired_devices(View view) {
    pairedDevicesList();
    }
private void pairedDevicesList() {
pairedDevices = myBluetooth.getBondedDevices();
        ArrayList list = new ArrayList();

if (pairedDevices.size() >0) {
for (BluetoothDevice bt : pairedDevices) {
                String name = bt.getName();
                String address = bt.getAddress();
                BluetoothDeviceClass pairedDeviceClass = new
BluetoothDeviceClass(name, address);
pairedDeviceClassesArrayList.add(pairedDeviceClass);
            }
        } else {
        }

        BluetoothDeviceAdapter deviceAdapter = new BluetoothDeviceAdapter(this,
pairedDeviceClassesArrayList);
recyclerView.setAdapter(deviceAdapter);
recyclerView.setLayoutManager(new LinearLayoutManager(this));
    }
@Override
public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu_bluetooth_device_list, menu);
return true;
    }
@Override
public boolean onOptionsItemSelected(MenuItem item) {
int id = item.getItemId();

if (id == R.id.go_device_activity) {
            Intent intent = new Intent(BluetoothDeviceListActivity.this,
DeviceActivity.class);
            startActivity(intent);
return true;
        }
if (id == R.id.change_vessel_bluetooth_device_list) {

            MyPreferenceManager myPreferenceManager = new
```

```
MyPreferenceManager(context);
            myPreferenceManager.clear();
            Intent intent = new Intent(BluetoothDeviceListActivity.this,
VesselChoosingActivity.class);
            startActivity(intent);
return true;
        }
if (id == R.id.action_add_location) {
            LocationClass locationClass=new LocationClass(context);
            String lat = locationClass.getLatitude();
            String lng = locationClass.getLongitude();
return true;
        }

return super.onOptionsItemSelected(item);
    }

}
```

**DeviceActivity.java:**

After connecting with HC-05 Bluetooth device, this activity handle communication between server and Arduino.

```
package uniqueid_1010.gcm_chat.ovie_codecirrus.vesselcontrlapp.activity;

import android.app.Activity;
import android.app.ProgressDialog;
import android.bluetooth.BluetoothA2dp;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothProfile;
import android.bluetooth.BluetoothSocket;
import android.content.Context;
import android.content.Intent;
import android.location.Geocoder;
import android.location.Location;
import android.os.Bundle;
import android.os.Handler;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
```

```
import com.android.volley.NetworkResponse;
import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.GooglePlayServicesUtil;
import com.google.android.gms.common.api.GoogleApiClient;
import com.google.android.gms.common.api.PendingResult;
import com.google.android.gms.common.api.Status;
import com.google.android.gms.location.LocationListener;
import com.google.android.gms.location.LocationRequest;
import com.google.android.gms.location.LocationServices;

import org.json.JSONException;
import org.json.JSONObject;

import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.HashMap;
import java.util.List;
import java.util.Locale;
import java.util.Map;
import java.util.Set;
import java.util.UUID;

import uniqueid_1010.gcm_chat.ovie_codecirrus.vesselcontrlapp.R;
import uniqueid_1010.gcm_chat.ovie_codecirrus.vesselcontrlapp.app.EndPoints;
import uniqueid_1010.gcm_chat.ovie_codecirrus.vesselcontrlapp.app.MyApplication;
import
uniqueid_1010.gcm_chat.ovie_codecirrus.vesselcontrlapp.helper.MyPreferenceManager;
import
uniqueid_1010.gcm_chat.ovie_codecirrus.vesselcontrlapp.model.BluetoothDeviceClass;
import uniqueid_1010.gcm_chat.ovie_codecirrus.vesselcontrlapp.model.VesselClass;

public class DeviceActivity extends AppCompatActivity implements
LocationListener,
        GoogleApiClient.ConnectionCallbacks,
        GoogleApiClient.OnConnectionFailedListener {

private String TAG = DeviceActivity.class.getSimpleName();
//String DEVICE_ADDRESS = "20:15:05:21:05:42";
String DEVICE_ADDRESS;
    String Device_name;
private final UUID PORT_UUID = UUID.fromString("00001101-0000-1000-8000-
00805f9b34fb");//Serial Port Service ID
private BluetoothDevice device;
private BluetoothSocket socket;
private OutputStream outputStream;
private InputStream inputStream;
    Button btn_connection;
```

```
    TextView tv_connection_status;

boolean deviceConnected = false;
    Thread thread;
byte buffer[];
int bufferPosition;
boolean stopThread;
    Context context;

private static final long INTERVAL = 1000 * 10;
private static final long FASTEST_INTERVAL = 1000 * 50;
private static String latitude_global = "23.482879";
private static String longitude_global = "90.251619";
private static String alert_location_global = "Moucak";

    LocationRequest mLocationRequest;
    GoogleApiClient mGoogleApiClient;
    Location mCurrentLocation;
    String mLastUpdateTime;

    ImageView img_connection;
    BluetoothDeviceClass bluetoothDeviceClass;

    ProgressDialog progressDialog;

protected void createLocationRequest() {
mLocationRequest = new LocationRequest();
mLocationRequest.setInterval(INTERVAL);
mLocationRequest.setFastestInterval(FASTEST_INTERVAL);
mLocationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
    }


@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_device);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
context = this;
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

progressDialog = new ProgressDialog(DeviceActivity.this);
Intent intent = getIntent();
bluetoothDeviceClass = (BluetoothDeviceClass) intent.getSerializableExtra("obj");
DEVICE_ADDRESS = bluetoothDeviceClass.getDevice_address();
Device_name = bluetoothDeviceClass.getDevice_name();
btn_connection = (Button) findViewById(R.id.btn_connect);
img_connection = (ImageView) findViewById(R.id.img_connection_image);
tv_connection_status = (TextView) findViewById(R.id.tv_connection_text);


// setUiEnabled(false);
setUIConnectionData(false);


if (!isGooglePlayServicesAvailable()) {
```

```
            finish();
        } else {

            createLocationRequest();
mGoogleApiClient = new GoogleApiClient.Builder(context)
                    .addApi(LocationServices.API)
                    .addConnectionCallbacks(this)
                    .addOnConnectionFailedListener(this)
                    .build();
}


    }


private boolean isGooglePlayServicesAvailable() {
int status = GooglePlayServicesUtil.isGooglePlayServicesAvailable(context);
if (ConnectionResult.SUCCESS == status) {
return true;
        } else {
            GooglePlayServicesUtil.getErrorDialog(status, (Activity) context,
0).show();
return false;
        }
    }

@Override
public void onConnected(Bundle bundle) {
mGoogleApiClient.isConnected());
        startLocationUpdates();
    }

protected void startLocationUpdates() {
        PendingResult<Status> pendingResult =
LocationServices.FusedLocationApi.requestLocationUpdates(
mGoogleApiClient, mLocationRequest, this);
    }

@Override
public void onConnectionSuspended(int i) {

    }

@Override
public void onConnectionFailed(ConnectionResult connectionResult) {
            }

@Override
public void onLocationChanged(Location location) {
mCurrentLocation = location;
mLastUpdateTime = DateFormat.getTimeInstance().format(new Date());
//  getLocation();
}

private void getLocation() {
double lat_double;
double lng_double;
```

```
if (null != mCurrentLocation) {
            String lat = String.valueOf(mCurrentLocation.getLatitude());
            String lng = String.valueOf(mCurrentLocation.getLongitude());

latitude_global = lat;
longitude_global = lng;
            lat_double = mCurrentLocation.getLatitude();
            lng_double = mCurrentLocation.getLongitude();
            Geocoder geocoder;
            List<android.location.Address> addresses = null;

            geocoder = new
Geocoder(this,
                    Locale.getDefault());
try {
                addresses =
                        geocoder.getFromLocation
                                (lat_double, lng_double, 1);
                String address = addresses.get
                        (0).getAddressLine(0);
alert_location_global = address;
                String city = addresses.
                        get(0).getAddressLine(1);
                String state = addresses.get(0).getAdminArea();
                String country = addresses.get(0).getCountryName();
                String postalCode = addresses.get(0).getPostalCode();
                String knownName = addresses.get(0).getFeatureName()

            } catch (IOException e) {
                e.printStackTrace();
            }
} else {
        }
    }

public void location_start(View view) {
        getLocation();
    }

public boolean IsDeviceConnected() {
final BluetoothAdapter mBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
        BluetoothProfile.ServiceListener mProfileListener = new
BluetoothProfile.ServiceListener() {
public void onServiceConnected(int profile, BluetoothProfile proxy) {
if (profile == BluetoothProfile.A2DP) {
boolean deviceConnected = false;
                    BluetoothA2dp btA2dp = (BluetoothA2dp) proxy;
                    List<BluetoothDevice> a2dpConnectedDevices =
btA2dp.getConnectedDevices();
if (a2dpConnectedDevices.size() != 0) {
for (BluetoothDevice device : a2dpConnectedDevices) {
if (device.getName().contains(Device_name)) {
                                deviceConnected = true;
                            }
                        }
                    }
if (!deviceConnected) {
```

```
}
mBluetoothAdapter.closeProfileProxy(BluetoothProfile.A2DP, btA2dp);
                }
            }

public void onServiceDisconnected(int profile) {
}
        };
        mBluetoothAdapter.getProfileProxy(context, mProfileListener,
BluetoothProfile.A2DP);

return deviceConnected;
    }


public boolean BTinit() {
boolean found = false;
        BluetoothAdapter bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
if (bluetoothAdapter == null) {
                    }
if (!bluetoothAdapter.isEnabled()) {
            Intent enableAdapter = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
            startActivityForResult(enableAdapter, 0);
try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
        Set<BluetoothDevice> bondedDevices = bluetoothAdapter.getBondedDevices();
if (bondedDevices.isEmpty()) {
            Toast.LENGTH_SHORT).show();
        } else {
for (BluetoothDevice iterator : bondedDevices) {
if (iterator.getAddress().equals(DEVICE_ADDRESS)) {
device = iterator;
                    found = true;
break;
                }
            }
        }
return found;
    }

public boolean BTconnect() {
boolean connected = true;
try {
socket = device.createRfcommSocketToServiceRecord(PORT_UUID);
socket.connect();
        } catch (IOException e) {
            e.printStackTrace();
            connected = false;
        }
if (connected) {
try {
outputStream = socket.getOutputStream();
            } catch (IOException e) {
```

```
                e.printStackTrace();
            }
try {
inputStream = socket.getInputStream();
            } catch (IOException e) {
                e.printStackTrace();
            }

        }


return connected;
    }

public void onClickStart(View view) throws IOException {

if (deviceConnected == false) {
tv_connection_status.setText("Trying To Connect .. Please wait for connecting .... ");


if (BTinit()) {
if (BTconnect()) {
//   setUiEnabled(true);
deviceConnected = true;

                    setUIConnectionData(true);
deviceConnected = true;beginListenForData();
                }
            } else {

                setUIConnectionData(false);
deviceConnected = false;
}


        } else {

tv_connection_status.setText("Trying To DisConnect .. Please wait for Disconnecting
.... ");
            setUIConnectionData(false);
            Log.e(TAG, "In Stop connection ");
stopThread = true;
outputStream.close();
inputStream.close();
socket.close();
deviceConnected = false;
        }
}
void setUIConnectionData(boolean is_bt_device_connected) {
if (is_bt_device_connected) {

            String device_name = bluetoothDeviceClass.getDevice_name();
if (device_name.isEmpty()) {
                device_name = "This";
            }
tv_connection_status.setText("You are connected with " + device_name + " Bluetooth
Device .. ");
```

```
btn_connection.setBackgroundColor(getResources().getColor(R.color.red));
btn_connection.setText("Disconnect");
img_connection.setImageResource(R.drawable.bl_connectyed);
        } else {

            String device_name = bluetoothDeviceClass.getDevice_name();
if (device_name.isEmpty()) {
                device_name = "This";
            }
tv_connection_status.setText("You are not connected with " + device_name + " Bluetooth
Device .. ");
btn_connection.setBackgroundColor(getResources().getColor(R.color.green));
btn_connection.setText("Connect");
img_connection.setImageResource(R.drawable.bl_disconnected);
        }
    }
void beginListenForData() {
final Handler handler = new Handler();
stopThread = false;
buffer = new byte[1024];
        Thread thread = new Thread(new Runnable() {
public void run() {
while (!Thread.currentThread().isInterrupted() && !stopThread) {
//                      try {
int byteCount = inputStream.available();
if (byteCount >0) {
                                Log.e("Data coming", "data is coming");
byte[] rawBytes = new byte[byteCount];
inputStream.read(rawBytes);
                                String input = new String(rawBytes, "UTF-8");
final String finalInput = input;


final String finalInput1 = input;
handler.post(new Runnable() {
public void run() {

if (finalInput1.equals("1")) {
                                        AddAlert(1);
                                    } if (finalInput1.equals("0")) {
AddAlert(0);                                        }
                                    }
                                });
                            }
                        } catch (IOException ex) {
stopThread = true;
                        }
                    }
                }
        });
        thread.start();
    }

@Override
public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu_device, menu);
```

```
return true;
    }
@Override
public boolean onOptionsItemSelected(MenuItem item) {
int id = item.getItemId();

if (id == R.id.action_add_alert) {
            AddAlert(1);
return true;
        }
if (id == R.id.action_resolved_alert) {
            AddAlert(0);
return true;
        }
if (id == R.id.action_vessel_change) {
            MyPreferenceManager myPreferenceManager = new
MyPreferenceManager(context);
            myPreferenceManager.clear();
            Intent intent = new Intent(DeviceActivity.this,
VesselChoosingActivity.class);
            startActivity(intent);
return true;
        }
return super.onOptionsItemSelected(item);
    }
@Override
public void onStart() {
super.onStart();
mGoogleApiClient.connect();
    }
@Override
public void onStop() {
super.onStop();
mGoogleApiClient.disconnect();
mGoogleApiClient.isConnected());
    }
@Override
protected void onPause() {
super.onPause();
        stopLocationUpdates();
    }
protected void stopLocationUpdates() {
        LocationServices.FusedLocationApi.removeLocationUpdates(
mGoogleApiClient, this);
    }
@Override
public void onResume() {
super.onResume();
if (mGoogleApiClient.isConnected()) {
            startLocationUpdates();
        }
    }
private void AddAlert(final int alert_status) {
        getLocation();
        StringRequest strReq = new StringRequest(Request.Method.POST,
                EndPoints.Add_Alert, new Response.Listener<String>() {
@Override
```

```
public void onResponse(String response) {
try {
                    JSONObject obj = new JSONObject(response);
if (obj.getBoolean("error") == false) {
                    } else {
                    }
                    } catch (JSONException e) {
                    Toast.makeText(getApplicationContext(), "Json parse error: " +
e.getMessage(), Toast.LENGTH_SHORT).show();
                }
            }
        }, new Response.ErrorListener() {

@Override
public void onErrorResponse(VolleyError error) {
                NetworkResponse networkResponse = error.networkResponse;
                Log.e(TAG, "Volley error: " + error.getMessage() + ", code: " +
networkResponse);
                Toast.makeText(getApplicationContext(), "Volley error: " +
error.getMessage(), Toast.LENGTH_SHORT).show();
            }
        }) {
@Override
protected Map<String, String> getParams() {
                Map<String, String> params = new HashMap<>();
                VesselClass vessel =
MyApplication.getInstance().getPrefManager().getVessel();
                Log.e(TAG, "In Map");
                DateFormat df = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
                String alert_time_string =
df.format(Calendar.getInstance().getTime());

                Log.e(TAG, "Current Date " + alert_time_string);
                String alert_time = alert_time_string;
                String alert_location = alert_location_global;
                String vessel_id = vessel.getId();
int alert_is_resolved = alert_status;
params.put("vessel_id", vessel_id);
                params.put("alert_location", alert_location);
                params.put("alert_time", alert_time);
                params.put("alert_is_resolved", alert_is_resolved + "");
                params.put("lat", latitude_global);
                params.put("lng", longitude_global);
                Log.e(TAG, "params: " + params.toString());
return params;
            }
        };
MyApplication.getInstance().addToRequestQueue(strReq);

    }
}
```

**2. User Application:** All the codes for this application is available in Github. Github link: https://goo.gl/fxp66Y. Some important codes are only given below.

**LoginActivity.java:**
User needs to login first. This activity handle Login portion.

```
package com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.activity;
import android.content.Intent;
import android.os.Bundle;
import android.support.design.widget.TextInputLayout;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.text.Editable;
import android.text.TextUtils;
import android.text.TextWatcher;
import android.util.Log;
import android.view.View;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast
import com.android.volley.NetworkResponse;
import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.R;
import com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.app.EndPoints;
import com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.app.MyApplication;
import com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.model.User;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.HashMap;
import java.util.Map;
public class LoginActivity extends AppCompatActivity {
private String TAG = LoginActivity.class.getSimpleName();
private EditText inputPassword, inputEmail;
private Button btnEnter;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
if (MyApplication.getInstance().getPrefManager().getUser() != null) {
            User user = MyApplication.getInstance().getPrefManager().getUser();
            String access_level = user.getAccess_level();
if (access_level.equals("0")) {
startActivity(new Intent(this, MainActivity.class));
          finish();}
if (access_level.equals("1")) {
startActivity(new Intent(this, ControlRoomMain.class));
       finish() }     }       setContentView(R.layout.activity_login);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
  setSupportActionBar(toolbar);
inputEmail = (EditText) findViewById(R.id.input_email);
```

```
inputPassword = (EditText) findViewById(R.id.et_password);
btnEnter = (Button) findViewById(R.id.btn_enter);
btnEnter.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
        login();
           }
        });
    }
private void login() {
final String password = inputPassword.getText().toString();
final String email = inputEmail.getText().toString();

        StringRequest strReq = new StringRequest(Request.Method.POST,
                EndPoints.LOGIN, new Response.Listener<String>() {
@Override
public void onResponse(String response) {
                Log.e(TAG, "response: " + response);
try {
                    JSONObject obj = new JSONObject(response);
if (obj.getBoolean("error") == false) {
JSONObject userObj = obj.getJSONObject("user");
                        User user = new User(userObj.getString("user_id"),
                                userObj.getString("user_name"),
                                userObj.getString("user_email"),
                                userObj.getString("user_phone"),
                                userObj.getString("access_level"),
                                userObj.getString("user_gcm_token"));
                        String access_level = userObj.getString("access_level");
user.setAccess_level(access_level);MyApplication.getInstance().getPrefManager().storeUser(user);
if (access_level.equals("0")) {
                            startActivity(new Intent(getApplicationContext(),
MainActivity.class));
                    finish();
                      }
if (access_level.equals("1")) {                          startActivity(new
Intent(getApplicationContext(), ControlRoomMain.class));
                            finish();         }
                    } else {String login_rmessage4=obj.getString("message");
                        Toast.makeText(getApplicationContext(), "" + login_rmessage4,
Toast.LENGTH_LONG).show();
                    }

                } catch (JSONException e) {
                    Log.e(TAG, "json parsing error: " + e.getMessage());
                    Toast.makeText(getApplicationContext(), "Json parse error: " +
e.getMessage(), Toast.LENGTH_SHORT).show();
                }
            }
        }, new Response.ErrorListener() {
@Override
public void onErrorResponse(VolleyError error) {
                NetworkResponse networkResponse = error.networkResponse;
                        }
        }) {
@Override
protected Map<String, String> getParams() {
```

```
                Map<String, String> params = new HashMap<>();
                params.put("user_email", email);
                params.put("user_password", password);
                Log.e(TAG, "params: " + params.toString());
return params;
            }
        };
public void register_button(View view) {
        Intent intent = new Intent(LoginActivity.this, SignUpActivity.class);
        startActivity(intent); }
}
```

**MainActivity.java:**
This is for passengers. After Login a passenger can see his journey.

```
package com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.activity;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v4.content.LocalBroadcastManager;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.DefaultItemAnimator;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.support.v7.widget.Toolbar;
import android.util.Log;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.Toast;

import com.android.volley.NetworkResponse;
import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.R;
import com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.adapter.JourneyAdapter;
import com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.app.Config;
import com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.app.EndPoints;
import com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.app.MyApplication;
import com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.gcm.GcmIntentService;
import com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.model.JourneyClass;
import com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.model.User;
import com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.model.VesselClass;
import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.GoogleApiAvailability;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {

private String TAG = MainActivity.class.getSimpleName();
private static final int PLAY_SERVICES_RESOLUTION_REQUEST = 9000;
private BroadcastReceiver mRegistrationBroadcastReceiver;
private ArrayList<VesselClass>vesselClassArrayList;
private ArrayList<JourneyClass>journeyClassArrayList;
private JourneyAdapter journeyAdapter;
```

```
private RecyclerView recyclerView;

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);

if (MyApplication.getInstance().getPrefManager().getUser() == null) {
            launchLoginActivity();
        }


        User user=MyApplication.getInstance().getPrefManager().getUser();
        setContentView(R.layout.activity_main);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        getSupportActionBar().setTitle("Welcome "+ user.getName());
recyclerView = (RecyclerView) findViewById(R.id.recycler_view);
vesselClassArrayList = new ArrayList<>();
journeyClassArrayList = new ArrayList<>();

        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
                Snackbar.make(view, "Replace with your own action", Snackbar.LENGTH_LONG)
                        .setAction("Action", null).show();

                Intent intent = new Intent(MainActivity.this, SetJourneyActivity.class);
                startActivity(intent);
            }
        });


mRegistrationBroadcastReceiver = new BroadcastReceiver() {
@Override
public void onReceive(Context context, Intent intent) {

// checking for type intent filter
if (intent.getAction().equals(Config.REGISTRATION_COMPLETE)) {
String token = intent.getStringExtra("token"


                } else if (intent.getAction().equals(Config.SENT_TOKEN_TO_SERVER)) {
Log.e(TAG, "Gcm token is send to server: ");
} else if (intent.getAction().equals(Config.PUSH_NOTIFICATION)) {
Log.e(TAG, "Push Notification Created before fnc: ");
                    handlePushNotification(intent);
                    Log.e(TAG, "Push Notification Created  after func: ");
}
            }
        };

journeyAdapter = new JourneyAdapter(this, journeyClassArrayList);
        LinearLayoutManager layoutManager = new LinearLayoutManager(this);
recyclerView.setLayoutManager(layoutManager);

recyclerView.setItemAnimator(new DefaultItemAnimator());
```

```
recyclerView.setAdapter(journeyAdapter);
fetchAllJourney();
if (checkPlayServices()) {
       registerGCM();
        }
    }

private void handlePushNotification(Intent intent) {
int type = intent.getIntExtra("type", -1);

}

public void fetchAllJourney() {
        User user = MyApplication.getInstance().getPrefManager().getUser();
if (user == null) {
            Toast.makeText(getApplicationContext(), "User not Avilable ..", Toast.LENGTH_SHORT).sho
return;
        }
        String endPoint = EndPoints.GET_ALL_JOURNEY_BY_USER_ID.replace("_ID_", user.getId());
        StringRequest strReq = new StringRequest(Request.Method.GET,
                endPoint, new Response.Listener<String>() {

@Override
public void onResponse(String response) {
                Log.e(TAG, "response: " + response);

try {
                    JSONObject obj = new JSONObject(response);

                    Log.e(TAG, "Response of all chatrooms : " + obj.toString());
// check for error flag
if (obj.getBoolean("error") == false) {
                        JSONArray VesselArray = obj.getJSONArray("vessels");
                        JSONArray JourneyArray = obj.getJSONArray("journey");

for (int i = 0; i < VesselArray.length(); i++) {
                            JSONObject VesselObj = (JSONObject) VesselArray.get(i);
                            JSONObject journey_obj = (JSONObject) JourneyArray.get(i);
JourneyClass journeyClass = new JourneyClass();

                            journeyClass.setVesel_name(VesselObj.getString("vessel_name"));
                            journeyClass.setVessel_id(VesselObj.getString("vessel_id") + "");
                            journeyClass.setVessel_details(VesselObj.getString("vessel_description"
                            journeyClass.setId(journey_obj.getString("journey_id") + "");
                            journeyClass.setStart_time(journey_obj.getString("start_time"));
                            journeyClass.setEnd_time(journey_obj.getString("end_time"));

                            String is_journey_activated_string =
journey_obj.getString("is_journey_activated");
int is_activated = Integer.parseInt(is_journey_activated_string);
                            journeyClass.setIs_journey_activated(is_activated);

                            Log.e(TAG, "Journey class "+ journeyClass.toString());

journeyClassArrayList.add(journeyClass);
```

```
                        }

                    } else {
Toast.makeText(getApplicationContext(), "" + obj.getJSONObject("error").getString("message"),
Toast.LENGTH_LONG).show();
                    }

                } catch (JSONException e) {
                    Log.e(TAG, "json parsing error: " + e.getMessage());
                    Toast.makeText(getApplicationContext(), "Json parse error: " + e.getMessage(),
Toast.LENGTH_LONG).show();
                }journeyAdapter.notifyDataSetChanged();
            }
        }, new Response.ErrorListener() {

@Override
public void onErrorResponse(VolleyError error) {
                NetworkResponse networkResponse = error.networkResponse;
                Log.e(TAG, "Volley error: " + error.getMessage() + ", code: " + networkResponse);
                Toast.makeText(getApplicationContext(), "Volley error: " + error.getMessage(),
Toast.LENGTH_SHORT).show();
            }
        });MyApplication.getInstance().addToRequestQueue(strReq);
    }
public void fetchAllVessels() {
        StringRequest strReq = new StringRequest(Request.Method.GET,
                EndPoints.GET_VESSELS, new Response.Listener<String>() {

@Override
public void onResponse(String response) {
                Log.e(TAG, "response: " + response);

try {
                    JSONObject obj = new JSONObject(response);
                    Log.e(TAG, "Response of all chatrooms : " + obj.toString());
// check for error flag
if (obj.getBoolean("error") == false) {
                        JSONArray chatRoomsArray = obj.getJSONArray("vessels");
for (int i = 0; i < chatRoomsArray.length(); i++) {
                            JSONObject VesselObj = (JSONObject) chatRoomsArray.get(i);
                            VesselClass vesselClass = new VesselClass();
vesselClass.setId(VesselObj.getString("vessel_id") + "");
                            vesselClass.setName(VesselObj.getString("vessel_name"));
                            vesselClass.setRegistration(VesselObj.getString("vessel_registration_no

vesselClass.setDescription(VesselObj.getString("vessel_description"));vesselClassArrayList.add(vesse
                        }

                    } else {
Toast.makeText(getApplicationContext(), "" + obj.getJSONObject("error").getString("message"),
Toast.LENGTH_LONG).show();
                    }
                } catch (JSONException e) {
                    Log.e(TAG, "json parsing error: " + e.getMessage());
Toast.makeText(getApplicationContext(), "Json parse error: " + e.getMessage(), Toast.LENGTH_LONG).sh
            }       journeyAdapter.notifyDataSetChanged();
            }
```

```
        }, new Response.ErrorListener() {

@Override
public void onErrorResponse(VolleyError error) {
                NetworkResponse networkResponse = error.networkResponse;
                Log.e(TAG, "Volley error: " + error.getMessage() + ", code: " + networkResponse);
                Toast.makeText(getApplicationContext(), "Volley error: " + error.getMessage(),
Toast.LENGTH_SHORT).show();
            }
        });
MyApplication.getInstance().addToRequestQueue(strReq);
    }
private void launchLoginActivity() {
        Intent intent = new Intent(MainActivity.this, LoginActivity.class);
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK);
        startActivity(intent);
        finish();
    }
private void registerGCM() {
        Intent intent = new Intent(this, GcmIntentService.class);
startService(intent);
    }
private boolean checkPlayServices() {
        GoogleApiAvailability apiAvailability = GoogleApiAvailability.getInstance();
int resultCode = apiAvailability.isGooglePlayServicesAvailable(this);
if (resultCode != ConnectionResult.SUCCESS) {
if (apiAvailability.isUserResolvableError(resultCode)) {
                apiAvailability.getErrorDialog(this, resultCode, PLAY_SERVICES_RESOLUTION_REQUEST)
                        .show();
            } else {
                finish();
            }
return false;
        }
return true;
    }
@Override
protected void onResume() {
super.onResume();
LocalBroadcastManager.getInstance(this).registerReceiver(mRegistrationBroadcastReceiver,
new IntentFilter(Config.REGISTRATION_COMPLETE));
LocalBroadcastManager.getInstance(this).registerReceiver(mRegistrationBroadcastReceiver,
new IntentFilter(Config.PUSH_NOTIFICATION));
    }
@Override
protected void onPause() {
LocalBroadcastManager.getInstance(this).unregisterReceiver(mRegistrationBroadcastReceiver);
super.onPause();
    }
@Override
public boolean onCreateOptionsMenu(Menu menu) {
MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.menu_main, menu);
return true;
    }
public boolean onOptionsItemSelected(MenuItem menuItem) {
switch (menuItem.getItemId()) {
```

```
case R.id.action_logout:
                MyApplication.getInstance().logout();
break;
        }
return super.onOptionsItemSelected(menuItem);
    }
}
```

**SetJourneyActivity:**

Passenger can set a journey in this activity.

```
package com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.activity;

import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.AutoCompleteTextView;
import android.widget.Button;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.NetworkResponse;
import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.R;
import com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.adapter.SearchAutoCompleteAdapter;
import com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.app.EndPoints;
import com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.app.MyApplication;
import com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.model.JourneyClass;
import com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.model.User;
import com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.model.VesselClass;
import com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.model.Vessel_Status_Class;
import com.wdullaer.materialdatetimepicker.date.DatePickerDialog;
import com.wdullaer.materialdatetimepicker.time.RadialPickerLayout;
import com.wdullaer.materialdatetimepicker.time.TimePickerDialog;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
```

```
import java.util.ArrayList;
import java.util.Calendar;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

public class SetJourneyActivity extends AppCompatActivity implements AdapterView.OnItemSelectedListe
TimePickerDialog.OnTimeSetListener,
        DatePickerDialog.OnDateSetListener {

    VesselClass vesselClass;
    JourneyClass journeyClass_global = new JourneyClass();
public int date_flag = 0;
public int time_flag = 0;
    String d_start, d_end, t_start, t_end;
private String TAG = LoginActivity.class.getSimpleName();
    TextView tv_s_date, tv_s_time, tv_e_date, tv_e_time, tv_des;
AutoCompleteTextView auto_complete_vessel_name;
    String[] months_array = {" ", "Jan", "Fab", "March", "April", "May", "June", "July", "Aug", "Sep

final List<String>IDtList = new ArrayList<String>();
final List<String>NametList = new ArrayList<String>();
    ArrayList<Vessel_Status_Class>vessel_status_class_ArrayList;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_set_journey);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
tv_des = (TextView) findViewById(R.id.tv_vessel_description_set_journey);
tv_s_date = (TextView) findViewById(R.id.tv_s_date_set_journey);
tv_e_date = (TextView) findViewById(R.id.tv_e_date_set_journey);
tv_s_time = (TextView) findViewById(R.id.tv_s_time_set_journey);
tv_e_time = (TextView) findViewById(R.id.tv_e_time_set_journey);
auto_complete_vessel_name = (AutoCompleteTextView) findViewById(R.id.searchView_vessel_name_set_jou
// spinner = (Spinner) findViewById(R.id.spinner);


getSupportActionBar().setDisplayHomeAsUpEnabled(true);

auto_complete_vessel_name.setAdapter(new SearchAutoCompleteAdapter(this, R.layout.vessel_search_row
auto_complete_vessel_name.setThreshold(1);

auto_complete_vessel_name.setOnItemClickListener(new AdapterView.OnItemClickListener() {
@Override
public void onItemClick(AdapterView<?> parent, View view, int position, long id) {

                VesselClass vesselClass = (VesselClass) parent.getItemAtPosition(position);
                String v_id = vesselClass.getId();
if (v_id != "NULL") {

journeyClass_global.setVessel_id(v_id);
                    String v_name = vesselClass.getName();
auto_complete_vessel_name.setText(v_name);
// Log.e(TAG, v_id + "   id " + v_id);
tv_des.setText(vesselClass.getDescription());
auto_complete_vessel_name.setFocusable(false);
```

```
  }

            }
        });vessel_status_class_ArrayList = new ArrayList<>();
}
public void ShowSpinner() {

IDtList.clear();
NametList.clear();
for (int i = 0; i <vessel_status_class_ArrayList.size(); i++)
{NametList.add(vessel_status_class_ArrayList.get(i).getVessel_name());
            String id = vessel_status_class_ArrayList.get(i).getVessel_id();
IDtList.add(id);

            Log.e(TAG, "from db " + toString());
        }

        Log.e(TAG, "After loop");

        ArrayAdapter<String> vesselListAdapter = new ArrayAdapter<String>(this, android.R.layout.si
NametList);vesselListAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item)
    }
public void set_journey(View view) {
if (journeyClass_global.getVessel_id() == null || d_start == null || d_end == null || t_start == nu
            Toast.makeText(getApplicationContext(), "Please fill all fields", Toast.LENGTH_SHORT).s

        } else {
  addJourney();
        }
    }
public void date_start(View view) {
date_flag = 1;

        Calendar now = Calendar.getInstance();
        DatePickerDialog dpd = DatePickerDialog.newInstance(
                (DatePickerDialog.OnDateSetListener) SetJourneyActivity.this,
                now.get(Calendar.YEAR),
                now.get(Calendar.MONTH),
                now.get(Calendar.DAY_OF_MONTH)
        );      dpd.show(getFragmentManager(), "date_dialog_start");
    }
public void date_end(View view) {
date_flag = 2;

        Calendar now = Calendar.getInstance();
        DatePickerDialog dpd = DatePickerDialog.newInstance(
                (DatePickerDialog.OnDateSetListener) SetJourneyActivity.this,
                now.get(Calendar.YEAR),
                now.get(Calendar.MONTH),
                now.get(Calendar.DAY_OF_MONTH)
        );
        dpd.show(getFragmentManager(), "date_dialog_start");
    }

public void time_start(View view) {
time_flag = 1;
```

```
        Calendar now = Calendar.getInstance();
        TimePickerDialog dpd = TimePickerDialog.newInstance(
                (TimePickerDialog.OnTimeSetListener) SetJourneyActivity.this,
                now.get(Calendar.HOUR_OF_DAY),
                now.get(Calendar.MINUTE),
true
);
    dpd.show(getFragmentManager(), "time_dialog_start");
    }
public void time_end(View view) {
time_flag = 2;
        Calendar now = Calendar.getInstance();
        TimePickerDialog dpd = TimePickerDialog.newInstance(
                (TimePickerDialog.OnTimeSetListener) SetJourneyActivity.this,
                now.get(Calendar.HOUR_OF_DAY),
                now.get(Calendar.MINUTE),
true
);        dpd.show(getFragmentManager(), "time_dialog_start");
    }
@Override
public void onDateSet(DatePickerDialog view, int year, int monthOfYear, int dayOfMonth) {
        String date = year + "-" + (monthOfYear + 1) + "-" + dayOfMonth;
if (date_flag == 1) {
d_start = date;
tv_s_date.setText(dayOfMonth + ", " + months_array[monthOfYear + 1] + ", " + year % 100);
        }
if (date_flag == 2) {
d_end = date;
tv_e_date.setText(dayOfMonth + ", " + months_array[monthOfYear + 1] + ", " + year % 100);
        }
    }
@Override
public void onTimeSet(RadialPickerLayout view, int hourOfDay, int minute, int second) {
        String time = hourOfDay + ":" + minute + ":" + "00";
if (time_flag == 1) {
t_start = time;tv_s_time.setText(t_start);
        }
if (time_flag == 2) {
t_end = time;
tv_e_time.setText(t_end);        }
    }
public void fetchAllVessels() {
// from =0 for this activity
        // from =1  for Set JorneyActivity
String endPoint = EndPoints.GET_VESSELS;
        StringRequest strReq = new StringRequest(Request.Method.GET,
                endPoint, new Response.Listener<String>() {

@Override
public void onResponse(String response) {
try {
                    JSONObject obj = new JSONObject(response);
if (obj.getBoolean("error") == false) {
                        JSONArray VesselArray = obj.getJSONArray("vessels");

for (int i = 0; i < VesselArray.length(); i++) {
                            JSONObject VesselObj = (JSONObject) VesselArray.get(i);
```

```
vessel_status_class = new Vessel_Status_Class();
vessel_status_class.setVessel_id(VesselObj.getString("vessel_id").toString());
                            vessel_status_class.setVessel_name(VesselObj.getString("vessel_name").t
                            vessel_status_class.setVessel_des(VesselObj.getString("vessel_descripti
                            vessel_status_class.setVessel_reg(VesselObj.getString("vessel_registrat
                            vessel_status_class.setStatus_time(VesselObj.getString("vessel_status_t
                            vessel_status_class.setStatus_location(VesselObj.getString("vessel_loca
                            vessel_status_class.setIs_resolved(VesselObj.getString("vessel_status")
                            vessel_status_class.setLatitude(VesselObj.getString("vessel_lat"));

vessel_status_class.setLongitude(VesselObj.getString("vessel_lng"));());vessel_status_class_ArrayLi
                        }
                    } else {
}
                } catch (JSONException e) {
            }
        }, new Response.ErrorListener() {

@Override
public void onErrorResponse(VolleyError error) {
                NetworkResponse networkResponse = error.networkResponse;
                Log.e(TAG, "Volley error: " + error.getMessage() + ", code: " + networkResponse);
                MyApplication.getInstance().addToRequestQueue(strReq);
    }
private void addJourney() {
        StringRequest strReq = new StringRequest(Request.Method.POST,             EndPoints.Add_Jou
Response.Listener<String>() {
@Override
public void onResponse(String response) {
try {
                    JSONObject obj = new JSONObject(response);
if (obj.getBoolean("error") == false) {
                        Intent intent = new Intent(SetJourneyActivity.this, MainActivity.class);
         startActivity(intent);
                    } else {                    }
                } catch (JSONException e) {
                }
            }
        }, new Response.ErrorListener() {
@Override
public void onErrorResponse(VolleyError error) {
                NetworkResponse networkResponse = error.networkResponse;
            }
        }) {
@Override
protected Map<String, String> getParams() {
                Map<String, String> params = new HashMap<>();
                User user = MyApplication.getInstance().getPrefManager().getUser();
                String start_time = d_start + " " + t_start;
                String end_time = d_end + " " + t_end;
                String user_id = user.getId();
                String vessel_id = journeyClass_global.getVessel_id();
int is_activated = 1;
             params.put("user_id", user_id);
           params.put("vessel_id", vessel_id);             params.put("start_time", start_time);
params.put("end_time", end_time);
                params.put("is_journey_activated", is_activated + "");
```

```
return params;
            }
        };
MyApplication.getInstance().addToRequestQueue(strReq);
    }
@Override
public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l)
{journeyClass_global.setVessel_id(IDtList.get(i));
        String v_name =NametList.get(i);auto_complete_vessel_name.setText(v_name);
auto_complete_vessel_name.setFocusable(false);
@Override
public void onNothingSelected(AdapterView<?>adapterView) {
    }
}
```

**EditJourneyActivity.Java:**

After set a journey passenger can update his journey here.

```
package com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.activity;

import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.AutoCompleteTextView;
import android.widget.Button;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.NetworkResponse;
import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.R;
import
com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.adapter.SearchAutoCompleteAdapter;
import com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.app.EndPoints;
import com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.app.MyApplication;
import com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.model.JourneyClass;
import com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.model.User;
import com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.model.VesselClass;
import com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.model.Vessel_Status_Class;
import com.wdullaer.materialdatetimepicker.date.DatePickerDialog;
import com.wdullaer.materialdatetimepicker.time.RadialPickerLayout;
import com.wdullaer.materialdatetimepicker.time.TimePickerDialog;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
```

```
import java.util.Calendar;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

public class SetJourneyActivity extends AppCompatActivity implements
AdapterView.OnItemSelectedListener, TimePickerDialog.OnTimeSetListener,
DatePickerDialog.OnDateSetListener {
    VesselClass vesselClass;
    JourneyClass journeyClass_global = new JourneyClass();
public int date_flag = 0;
public int time_flag = 0;
    String d_start, d_end, t_start, t_end;
private String TAG = LoginActivity.class.getSimpleName();
    TextView tv_s_date, tv_s_time, tv_e_date, tv_e_time, tv_des;
//  Spinner spinner;
AutoCompleteTextView auto_complete_vessel_name;
    String[] months_array = {" ", "Jan", "Fab", "March", "April", "May", "June",
"July", "Aug", "Sept", "Oct", "Nov", "Dec"};

final List<String>IDtList = new ArrayList<String>();
final List<String>NametList = new ArrayList<String>();

    ArrayList<Vessel_Status_Class>vessel_status_class_ArrayList;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState); setContentView(R.layout.activity_set_journey);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
tv_des = (TextView) findViewById(R.id.tv_vessel_description_set_journey);
tv_s_date = (TextView) findViewById(R.id.tv_s_date_set_journey);
tv_e_date = (TextView) findViewById(R.id.tv_e_date_set_journey);
tv_s_time = (TextView) findViewById(R.id.tv_s_time_set_journey);
tv_e_time = (TextView) findViewById(R.id.tv_e_time_set_journey);
auto_complete_vessel_name = (AutoCompleteTextView)
findViewById(R.id.searchView_vessel_name_set_journey);
// spinner = (Spinner) findViewById(R.id.spinner);


getSupportActionBar().setDisplayHomeAsUpEnabled(true);

auto_complete_vessel_name.setAdapter(new SearchAutoCompleteAdapter(this,
R.layout.vessel_search_row));
auto_complete_vessel_name.setThreshold(1);

auto_complete_vessel_name.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
@Override
public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
                VesselClass vesselClass = (VesselClass)
parent.getItemAtPosition(position);
                String v_id = vesselClass.getId();
if (v_id != "NULL") {journeyClass_global.setVessel_id(v_id);
                    String v_name = vesselClass.getName();
auto_complete_vessel_name.setText(v_name);
tv_des.setText(vesselClass.getDescription());
auto_complete_vessel_name.setFocusable(false);

                }
```

```
            }
        });
vessel_status_class_ArrayList = new ArrayList<>();
}
public void ShowSpinner() {

IDtList.clear();
NametList.clear();

for (int i = 0; i <vessel_status_class_ArrayList.size(); i++) {

NametList.add(vessel_status_class_ArrayList.get(i).getVessel_name());
            String id = vessel_status_class_ArrayList.get(i).getVessel_id();
IDtList.add(id);

            Log.e(TAG, "from db " + toString());
        }

        Log.e(TAG, "After loop");

        ArrayAdapter<String> vesselListAdapter = new ArrayAdapter<String>(this,
android.R.layout.simple_spinner_item,
NametList);vesselListAdapter.setDropDownViewResource(android.R.layout.simple_spinner
_dropdown_item);

    }
public void set_journey(View view) {
if (journeyClass_global.getVessel_id() == null || d_start == null || d_end == null
|| t_start == null || t_end == null) {
            Toast.makeText(getApplicationContext(), "Please fill all fields",
Toast.LENGTH_SHORT).show();

        } else {
            addJourney();
        }
    }

public void date_start(View view) {
date_flag = 1;

        Calendar now = Calendar.getInstance();
        DatePickerDialog dpd = DatePickerDialog.newInstance(
                (DatePickerDialog.OnDateSetListener) SetJourneyActivity.this,
                now.get(Calendar.YEAR),
                now.get(Calendar.MONTH),
                now.get(Calendar.DAY_OF_MONTH)
        );
        dpd.show(getFragmentManager(), "date_dialog_start");
    }


public void date_end(View view) {
date_flag = 2;

        Calendar now = Calendar.getInstance();
        DatePickerDialog dpd = DatePickerDialog.newInstance(
                (DatePickerDialog.OnDateSetListener) SetJourneyActivity.this,
                now.get(Calendar.YEAR),
```

```
                now.get(Calendar.MONTH),
                now.get(Calendar.DAY_OF_MONTH)
        );
        dpd.show(getFragmentManager(), "date_dialog_start");
    }

public void time_start(View view) {
time_flag = 1;


        Calendar now = Calendar.getInstance();
        TimePickerDialog dpd = TimePickerDialog.newInstance(
                (TimePickerDialog.OnTimeSetListener) SetJourneyActivity.this,
                now.get(Calendar.HOUR_OF_DAY),
                now.get(Calendar.MINUTE),
true
);
        dpd.show(getFragmentManager(), "time_dialog_start");
    }

public void time_end(View view) {
time_flag = 2;
        Calendar now = Calendar.getInstance();
        TimePickerDialog dpd = TimePickerDialog.newInstance(
                (TimePickerDialog.OnTimeSetListener) SetJourneyActivity.this,
                now.get(Calendar.HOUR_OF_DAY),
                now.get(Calendar.MINUTE),
true
);
        dpd.show(getFragmentManager(), "time_dialog_start");
    }
@Override
public void onDateSet(DatePickerDialog view, int year, int monthOfYear, int
dayOfMonth) {
        String date = year + "-" + (monthOfYear + 1) + "-" + dayOfMonth;
if (date_flag == 1) {
d_start = date;
tv_s_date.setText(dayOfMonth + ", " + months_array[monthOfYear + 1] + ", " + year %
100);
            Log.e("Set journey", "start date is  " + d_start + "");
        }
if (date_flag == 2) {
d_end = date;
tv_e_date.setText(dayOfMonth + ", " + months_array[monthOfYear + 1] + ", " + year %
100);
            Log.e("Set journey", "end date is  " + d_end + "");
        }

    }

@Override
public void onTimeSet(RadialPickerLayout view, int hourOfDay, int minute, int
second) {
        String time = hourOfDay + ":" + minute + ":" + "00";
if (time_flag == 1) {
t_start = time;
tv_s_time.setText(t_start);

        }
```

```
if (time_flag == 2) {
t_end = time;
tv_e_time.setText(t_end);
            Log.e("Set journey", "end time is  " + t_end + "");
        }
    }
public void fetchAllVessels() {
// from =0 for this activity
        // from =1  for Set JorneyActivity
String endPoint = EndPoints.GET_VESSELS;
        StringRequest strReq = new StringRequest(Request.Method.GET,
                endPoint, new Response.Listener<String>() {
@Override
public void onResponse(String response) {
                Log.e(TAG, "response: " + response);
try {
                    JSONObject obj = new JSONObject(response);

                    Log.e(TAG, "Response of all chatrooms : " + obj.toString());
if (obj.getBoolean("error") == false) {
                        JSONArray VesselArray = obj.getJSONArray("vessels");

for (int i = 0; i < VesselArray.length(); i++) {
                            JSONObject VesselObj = (JSONObject) VesselArray.get(i);

                            Vessel_Status_Class vessel_status_class = new
Vessel_Status_Class();


vessel_status_class.setVessel_id(VesselObj.getString("vessel_id").toString());

vessel_status_class.setVessel_name(VesselObj.getString("vessel_name").toString());

vessel_status_class.setVessel_des(VesselObj.getString("vessel_description").toString
());

vessel_status_class.setVessel_reg(VesselObj.getString("vessel_registration_no").toSt
ring());

vessel_status_class.setStatus_time(VesselObj.getString("vessel_status_time"));

vessel_status_class.setStatus_location(VesselObj.getString("vessel_location"));

vessel_status_class.setIs_resolved(VesselObj.getString("vessel_status"));

vessel_status_class.setLatitude(VesselObj.getString("vessel_lat"));

vessel_status_class.setLongitude(VesselObj.getString("vessel_lng"));
Log.e(TAG, "Vessel class " + vessel_status_class.toString());
vessel_status_class_ArrayList.add(vessel_status_class);
                        }
                    } else {
}
                } catch (JSONException e) {
                    Log.e(TAG, "json parsing error: " + e.getMessage());
ShowSpinner();
            }
        }, new Response.ErrorListener() {
@Override
```

```
public void onErrorResponse(VolleyError error) {
                NetworkResponse networkResponse = error.networkResponse;
                Log.e(TAG, "Volley error: " + error.getMessage() + ", code: " +
networkResponse);
                MyApplication.getInstance().addToRequestQueue(strReq);
    }

private void addJourney() {
        StringRequest strReq = new StringRequest(Request.Method.POST,
                EndPoints.Add_Journey, new Response.Listener<String>() {

@Override
public void onResponse(String response) {
                Log.e(TAG, "response: " + response);

try {
                    JSONObject obj = new JSONObject(response);
if (obj.getBoolean("error") == false) {
                        Intent intent = new Intent(SetJourneyActivity.this,
MainActivity.class);                    startActivity(intent);
                                            } else
{Toast.makeText(getApplicationContext(), "" +
obj.getJSONObject("error").getString("message"), Toast.LENGTH_LONG).show();
                    }
                } catch (JSONException e) {
                    Log.e(TAG, "json parsing error: " + e.getMessage());
Toast.makeText(getApplicationContext(), "Json parse error: " + e.getMessage(),
Toast.LENGTH_SHORT).show();
                }
            }
        }, new Response.ErrorListener() {

@Override
public void onErrorResponse(VolleyError error) {
                NetworkResponse networkResponse = error.networkResponse;

            }
        }) {

@Override
protected Map<String, String> getParams() {
                Map<String, String> params = new HashMap<>();
                User user = MyApplication.getInstance().getPrefManager().getUser();

                String start_time = d_start + " " + t_start;
                String end_time = d_end + " " + t_end;
                String user_id = user.getId();
                String vessel_id = journeyClass_global.getVessel_id();
int is_activated = 1;             params.put("user_id", user_id);
                params.put("vessel_id", vessel_id);
                params.put("start_time", start_time);
                params.put("end_time", end_time);
                params.put("is_journey_activated", is_activated + "");

                Log.e(TAG, "params: " + params.toString());
return params;
            }
        };MyApplication.getInstance().addToRequestQueue(strReq);
    }
```

```
@Override
public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l) {
journeyClass_global.setVessel_id(IDtList.get(i));
        String v_name =NametList.get(i);
auto_complete_vessel_name.setText(v_name);
auto_complete_vessel_name.setFocusable(false);
    }
@Override
public void onNothingSelected(AdapterView<?> adapterView) {
    }
}
```

**ControlRoomMain.Java:**

It is only for control Room authority. Here they can see all the overloaded vessels list live. They can also sort the time limit for showing the overloaded vessels on that time frame.

```
package com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.activity;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.os.Bundle;
import android.support.v4.content.LocalBroadcastManager;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.DefaultItemAnimator;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.support.v7.widget.Toolbar;
import android.util.Log;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.AutoCompleteTextView;
import android.widget.TextView;
import android.widget.Toast;


import com.android.volley.NetworkResponse;
import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.R;
import
com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.adapter.AlertAdapter_CurrentUser;
import com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.adapter.StatusAdapter;
import com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.app.Config;
import com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.app.EndPoints;
import com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.app.MyApplication;
import com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.gcm.GcmIntentService;
import com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.helper.MyPreferenceManager;
import com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.model.AlertClass;
```

```
import com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.model.User;
import com.gcmandroid_uniqueid1010.nemo.NirapodNouvromon.model.Vessel_Status_Class;
import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.GoogleApiAvailability;
import com.wdullaer.materialdatetimepicker.date.DatePickerDialog;
import com.wdullaer.materialdatetimepicker.time.RadialPickerLayout;
import com.wdullaer.materialdatetimepicker.time.TimePickerDialog;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.Calendar;
import java.util.HashMap;
import java.util.Map;


public class ControlRoomMain extends AppCompatActivity implements
TimePickerDialog.OnTimeSetListener,
        DatePickerDialog.OnDateSetListener {

private String TAG = ControlRoomMain.class.getSimpleName();
private static final int PLAY_SERVICES_RESOLUTION_REQUEST = 9000;
private BroadcastReceiver mRegistrationBroadcastReceiver;
private ArrayList<Vessel_Status_Class>status_class_array_list = new ArrayList<>();
private RecyclerView recyclerView;
private StatusAdapter statusAdapter;
    Context context;
    TextView tv_status_upper_title;

public int date_flag = 0;
public int time_flag = 0;
    String d_start, d_end, t_start, t_end;
    TextView tv_s_date, tv_s_time, tv_e_date, tv_e_time, tv_des;
    String[] months_array = {" ", "Jan", "Fab", "March", "April", "May", "June",
"July", "Aug", "Sept", "Oct", "Nov", "Dec"};

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_control_room_main);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
context = this;

recyclerView = (RecyclerView) findViewById(R.id.recycler_view);
tv_status_upper_title = (TextView)
findViewById(R.id.tv_alet_upper_title_control_room);

statusAdapter = new StatusAdapter(this, status_class_array_list, 1);
        LinearLayoutManager layoutManager = new LinearLayoutManager(this);
recyclerView.setLayoutManager(layoutManager);

recyclerView.setItemAnimator(new DefaultItemAnimator());
recyclerView.setAdapter(statusAdapter);

// fetchAllVessels();
        // FetchAllAlertForControlRoom();
```

```
mRegistrationBroadcastReceiver = new BroadcastReceiver() {


@Override
public void onReceive(Context context, Intent intent) {

                Log.e(TAG, "Receoiver in Control Room");
if (intent.getAction().equals(Config.PUSH_NOTIFICATION)) {
// new push message is received
Log.e(TAG, "Condition matched");
                    handlePushNotification(intent);
                }
            }
        };


tv_s_date = (TextView) findViewById(R.id.tv_s_date_set_journey);
tv_e_date = (TextView) findViewById(R.id.tv_e_date_set_journey);
tv_s_time = (TextView) findViewById(R.id.tv_s_time_set_journey);
tv_e_time = (TextView) findViewById(R.id.tv_e_time_set_journey);

        Calendar now = Calendar.getInstance();
int year = now.get(Calendar.YEAR);
int month = now.get(Calendar.MONTH);
int day = now.get(Calendar.DAY_OF_MONTH);
int hour = now.get(Calendar.HOUR_OF_DAY);
int minute = now.get(Calendar.MINUTE);

d_start = year + "-" + (month + 1) + "-" + day;
d_end=d_start;
t_start  = (hour-1) + ":" + minute + ":" + "00";
t_end = hour + ":" + minute + ":" + "00";

tv_s_date.setText(d_start);
tv_e_date.setText("Current date");
tv_s_time.setText(t_start);
tv_e_time.setText("Current Time");
tv_status_upper_title.setText("Current Alerts : ");

        FetchAllAlertedVesselStatusBYTimePostMethod();

if (checkPlayServices()) {
            registerGCM();
        }

    }


private boolean checkPlayServices() {
        GoogleApiAvailability apiAvailability = GoogleApiAvailability.getInstance();
int resultCode = apiAvailability.isGooglePlayServicesAvailable(this);
if (resultCode != ConnectionResult.SUCCESS) {
if (apiAvailability.isUserResolvableError(resultCode)) {
                apiAvailability.getErrorDialog(this, resultCode,
PLAY_SERVICES_RESOLUTION_REQUEST)
                        .show();
```

```
            } else {
                Log.i(TAG, "This device is not supported. Google Play Services not
installed!");
                Toast.makeText(getApplicationContext(), "This device is not
supported. Google Play Services not installed!", Toast.LENGTH_LONG).show();
                finish();
            }
return false;
        }
return true;
    }
private void registerGCM() {
        Intent intent = new Intent(this, GcmIntentService.class);
// intent.putExtra("key", "register");
startService(intent);
    }
public void date_start(View view) {
date_flag = 1;

        Calendar now = Calendar.getInstance();
        DatePickerDialog dpd = DatePickerDialog.newInstance(
                (DatePickerDialog.OnDateSetListener) ControlRoomMain.this,
                now.get(Calendar.YEAR),
                now.get(Calendar.MONTH),
                now.get(Calendar.DAY_OF_MONTH)
        );
        dpd.show(getFragmentManager(), "date_dialog_start");
    }


public void date_end(View view) {
date_flag = 2;

        Calendar now = Calendar.getInstance();
        DatePickerDialog dpd = DatePickerDialog.newInstance(
                (DatePickerDialog.OnDateSetListener) ControlRoomMain.this,
                now.get(Calendar.YEAR),
                now.get(Calendar.MONTH),
                now.get(Calendar.DAY_OF_MONTH)
        );
        dpd.show(getFragmentManager(), "date_dialog_start");
    }

public void time_start(View view) {
time_flag = 1;


        Calendar now = Calendar.getInstance();
        TimePickerDialog dpd = TimePickerDialog.newInstance(
                (TimePickerDialog.OnTimeSetListener) ControlRoomMain.this,
                now.get(Calendar.HOUR_OF_DAY),
                now.get(Calendar.MINUTE),
true
);
        dpd.show(getFragmentManager(), "time_dialog_start");
    }

public void time_end(View view) {
time_flag = 2;
```

```
        Calendar now = Calendar.getInstance();
        TimePickerDialog dpd = TimePickerDialog.newInstance(
                (TimePickerDialog.OnTimeSetListener) ControlRoomMain.this,
                now.get(Calendar.HOUR_OF_DAY),
                now.get(Calendar.MINUTE),
true
);
        dpd.show(getFragmentManager(), "time_dialog_start");
    }

public void all_status(View view) {

status_class_array_list.clear();
statusAdapter.notifyDataSetChanged();

tv_s_date.setText("From the begining");
tv_e_date.setText("Current date");
tv_s_time.setText("From the begining time");
tv_e_time.setText("Current Time");
tv_status_upper_title.setText("All Alerts : ");

        FetchAllAlertedVesselStatus();
    }

public void status_by_time(View view) {

status_class_array_list.clear();
statusAdapter.notifyDataSetChanged();

/* tv_s_date.setText(d_start);
        tv_e_date.setText("Current date");
        tv_s_time.setText(t_start);
        tv_e_time.setText("Current Time");*/
tv_status_upper_title.setText(" Alerts : ");

        FetchAllAlertedVesselStatusBYTimePostMethod();

    }


@Override
public void onDateSet(DatePickerDialog view, int year, int monthOfYear, int
dayOfMonth) {
        String date = year + "-" + (monthOfYear + 1) + "-" + dayOfMonth;
if (date_flag == 1) {
d_start = date;
tv_s_date.setText(dayOfMonth + ", " + months_array[monthOfYear + 1] + ", " + year %
100);
            Log.e("Set journey", "start date is  " + d_start + "");
        }
if (date_flag == 2) {
d_end = date;
tv_e_date.setText(dayOfMonth + ", " + months_array[monthOfYear + 1] + ", " + year %
100);
            Log.e("Set journey", "end date is  " + d_end + "");
        }

    }
```

```
@Override
public void onTimeSet(RadialPickerLayout view, int hourOfDay, int minute, int
second) {
        String time = hourOfDay + ":" + minute + ":" + "00";
if (time_flag == 1) {
t_start = time;
tv_s_time.setText(t_start);
            Log.e("Set journey", "start time is  " + t_start + "");
        }
if (time_flag == 2) {
t_end = time;
tv_e_time.setText(t_end);
            Log.e("Set journey", "end time is  " + t_end + "");
        }
    }
private void handlePushNotification(Intent intent) {
        AlertClass alertClass = (AlertClass) intent.getSerializableExtra("alert");
        Log.e(TAG, "Alert class in push handler : " + alertClass.toString());

        String v_id = alertClass.getVessel_id();
if (alertClass != null) {
if (alertClass.getIs_resolved() == 0) {
for (int i = 0; i <status_class_array_list.size(); i++) {

                    Vessel_Status_Class status_class =
status_class_array_list.get(i);
//  Log.e(TAG, "in loop status clkass " + status_class);
if (v_id.equals(status_class.getVessel_id())) {
status_class_array_list.remove(i);
// break;
}
                }

            }

if (alertClass.getIs_resolved() == 1) {

//  Log.e(TAG, "alert class in push handler " + alertClass.toString());
if (alertClass.getIs_resolved() == 0) {
for (int i = 0; i <status_class_array_list.size(); i++) {

                        Vessel_Status_Class status_class =
status_class_array_list.get(i);
//  Log.e(TAG, "in loop status clkass " + status_class);
if (v_id.equals(status_class.getVessel_id())) {
status_class_array_list.remove(i);
// break;
}
            }

                }
                Vessel_Status_Class vessel_status_class = new Vessel_Status_Class();

                vessel_status_class.setIs_resolved("1");
                vessel_status_class.setStatus_location(alertClass.getLocation());
                vessel_status_class.setStatus_time(alertClass.getAlert_time());
                vessel_status_class.setVessel_name(alertClass.getVessel_name());
                vessel_status_class.setVessel_id(alertClass.getVessel_id());
```

```
status_class_array_list.add(vessel_status_class);
            }

statusAdapter.notifyDataSetChanged();}
    }
private void FetchAllAlertedVesselStatus() {
        StringRequest strReq = new StringRequest(Request.Method.GET,
                EndPoints.GET_VESSEL_STATUS_ALERTED, new Response.Listener<String>()
{
@Override
public void onResponse(String response) {
                Log.e(TAG, "response: " + response);

try {
                    JSONObject obj = new JSONObject(response);

// check for error flag
if (obj.getBoolean("error") == false) {

                        JSONArray StatusJsonArray = obj.getJSONArray("status");

for (int i = 0; i < StatusJsonArray.length(); i++) {
                            JSONObject statusObj = (JSONObject)
StatusJsonArray.get(i);
                            Vessel_Status_Class vessel_status_class = new
Vessel_Status_Class();

                            JSONObject vessel_obj =
statusObj.getJSONObject("vessel");


vessel_status_class.setVessel_id(statusObj.getString("vessel_id").toString());

vessel_status_class.setStatus_time(statusObj.getString("status_time"));

vessel_status_class.setStatus_location(statusObj.getString("status_place"));

vessel_status_class.setIs_resolved(statusObj.getString("is_resolved"));

vessel_status_class.setLatitude(statusObj.getString("latitude"));
vessel_status_class.setLongitude(statusObj.getString("longitude"));
                            String v_name = vessel_obj.getString("vessel_name");
                            vessel_status_class.setVessel_name(v_name);


                            Log.e(TAG, "Vessel class " +
vessel_status_class.toString());
status_class_array_list.add(vessel_status_class);
                        }


                    } else {
// login error - simply toast the message
Toast.makeText(getApplicationContext(), "" +
obj.getJSONObject("error").getString("message"), Toast.LENGTH_LONG).show();
                    }

                } catch (JSONException e) {
                    Log.e(TAG, "json parsing error: " + e.getMessage());
```

```
                    Toast.makeText(getApplicationContext(), "Json parse error: " +
e.getMessage(), Toast.LENGTH_SHORT).show();
                }statusAdapter.notifyDataSetChanged();
      }, new Response.ErrorListener() {
@Override
public void onErrorResponse(VolleyError error) {
                NetworkResponse networkResponse = error.networkResponse;
                Log.e(TAG, "Volley error: " + error.getMessage() + ", code: " +
networkResponse);
                Toast.makeText(getApplicationContext(), "Volley error: " +
error.getMessage(), Toast.LENGTH_SHORT).show();
            }
        });MyApplication.getInstance().addToRequestQueue(strReq);
    }
private void FetchAllAlertedVesselStatusBYTimePostMethod() {
        StringRequest strReq = new StringRequest(Request.Method.POST,
                EndPoints.GET_VESSEL_STATUS_ALERTED_BY_TIME_POST_METHOD, new
Response.Listener<String>() {

@Override
public void onResponse(String response) {
                Log.e(TAG, "response: " + response);

try {
                    JSONObject obj = new JSONObject(response);

// check for error flag
if (obj.getBoolean("error") == false) {

                        JSONArray StatusJsonArray = obj.getJSONArray("status");

for (int i = 0; i < StatusJsonArray.length(); i++) {
                            JSONObject statusObj = (JSONObject)
StatusJsonArray.get(i);
                            Vessel_Status_Class vessel_status_class = new
Vessel_Status_Class();
                            JSONObject vessel_obj =
statusObj.getJSONObject("vessel");


vessel_status_class.setVessel_id(statusObj.getString("vessel_id").toString());

vessel_status_class.setStatus_time(statusObj.getString("status_time"));

vessel_status_class.setStatus_location(statusObj.getString("status_place"));

vessel_status_class.setIs_resolved(statusObj.getString("is_resolved"));

vessel_status_class.setLatitude(statusObj.getString("latitude"));

vessel_status_class.setLongitude(statusObj.getString("longitude"));
                            String v_name = vessel_obj.getString("vessel_name");
                            vessel_status_class.setVessel_name(v_name);


                            Log.e(TAG, "Vessel class " +
vessel_status_class.toString());
status_class_array_list.add(vessel_status_class);
                        }
```

```
                    } else {
Toast.makeText(getApplicationContext(), "" +
obj.getJSONObject("error").getString("message"), Toast.LENGTH_LONG).show();
                    }

                } catch (JSONException e) {
                    Log.e(TAG, "json parsing error: " + e.getMessage());
                    Toast.makeText(getApplicationContext(), "Json parse error: " +
e.getMessage(), Toast.LENGTH_SHORT).show();
                    }statusAdapter.notifyDataSetChanged();
            }
        }, new Response.ErrorListener() {
@Override
public void onErrorResponse(VolleyError error) {
                NetworkResponse networkResponse = error.networkResponse;
                Log.e(TAG, "Volley error: " + error.getMessage() + ", code: " +
networkResponse);
                Toast.makeText(getApplicationContext(), "Volley error: " +
error.getMessage(), Toast.LENGTH_SHORT).show();
            }
        }) {

@Override
protected Map<String, String> getParams() {
                Map<String, String> params = new HashMap<>();

                String start_time = d_start + " " + t_start;
                String end_time = d_end + " " + t_end;

                params.put("start_time", start_time);
                params.put("end_time", end_time);

                Log.e(TAG, "params: " + params.toString());
return params;
            }
        };
MyApplication.getInstance().addToRequestQueue(strReq);

    }
@Override
public boolean onCreateOptionsMenu(Menu menu) {
MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.menu_control_room_main, menu);
return true;
    }

public boolean onOptionsItemSelected(MenuItem menuItem) {
switch (menuItem.getItemId()) {

case R.id.action_vessel_list:
                Intent intent0 = new Intent(ControlRoomMain.this,
VesselsListActivity.class);
        startActivity(intent0);
break;
case R.id.action_add_vessel:
                Intent intent = new Intent(ControlRoomMain.this,
AddVesselActivity.class);
                startActivity(intent);
```

```
break;

case R.id.log_out:          MyPreferenceManager myPreferenceManager = new
MyPreferenceManager(context); myPreferenceManager.clear();
                Intent intent1 = new Intent(ControlRoomMain.this,
LoginActivity.class);
                startActivity(intent1);
break;
        }
return super.onOptionsItemSelected(menuItem);
    }
@Override
protected void onResume() {
super.onResume();
LocalBroadcastManager.getInstance(this).registerReceiver(mRegistrationBroadcastRecei
ver,
new
IntentFilter(Config.REGISTRATION_COMPLETE));LocalBroadcastManager.getInstance(this).
registerReceiver(mRegistrationBroadcastReceiver,
new IntentFilter(Config.PUSH_NOTIFICATION));
    }

@Override
protected void onPause() {

LocalBroadcastManager.getInstance(this).unregisterReceiver(mRegistrationBroadcastRec
eiver);
super.onPause();
    }


}
```

# REFERENCES


[1] "Above 4,000 deaths from launch accidents in 38 years," [Online]. Available: http://archive.dhakatribune.com/bangladesh/2014/may/05/above-4000-deaths-launch-accidents-38-years. [Accessed 1 March 2016].

[2] Naznin Afrose Huq and Ashraf Mahmood Dewan, "Launch disasters in Bangladesh: A geographical study." GEOGRAFIA Online Malaysian journal of society and space2 (19-30) ISSN 2180-2491 on "Magnitude of motor launch disasters ", January 2006.

[3] "Satellite-Based AIS System Provides Continuous Tracking at Sea," [Online]. Available: http://www.sea-technology.com/features/2011/0311/ais_system.php. [Accessed 22 February 2017].

[4] "Vessel Monitoring System Program," [Online]. Available: http://www.nmfs.noaa.gov/ole/about/our_programs/vessel_monitoring.html. [Accessed 22 February 2017].

[5] "ORBCOMM Vessel Tracking," [Online]. Available: https://www.orbcomm.com/en/industries/maritime/vessel-tracking. [Accessed 22 February 2017].

[6] "SkyHawk Vessel Monitoring System (VMS)," [Online]. Available: http://www.skyhawk.co/solutions/vessel-monitoring/. [Accessed 22 February 2017].

[7] "IT-based tracking system launched for BIWTC vessels," [Online]. Available: http://www.newstoday.com.bd/index.php?option=details&news_id=2345686&date=2013-05-23. [Accessed 23 February 2017].

[8] ShadmanSakib andMohammadSayemBinAbdullah, "GPS-GSM based Inland Vessel Tracking System for Automatic Emergency Detection and Position Notification." 10th International Conference on Intelligent Systems and Control (ISCO), At Coimbatore, Tamil Nadu, India, DOI: 10.1109/ISCO.2016.7727018"

[9] Major Muhammad Rabiul Islam, PhD and Cdr Kaosar Rashid, psc, BN, "A Brief Interpretation of Accidents in Bangladesh Inland River Routes." Military Institute of Science and Technology on " DATA ANALYSIS", Num. 3.0, March 2015.

[10] "Arduino Overview," [Online]. Available: https://en.wikipedia.org/wiki/Arduino. [Accessed 1 May 2016].

[11] "Arduino Technical Specs," [Online]. Available:

https://www.arduino.cc/en/Main/arduinoBoardMega2560. [Accessed 1 May 2016].

[12] "Arduino Power," [Online]. Available:
https://www.arduino.cc/en/Main/arduinoBoardMega2560. [Accessed 1 May 2016].

[13] "Arduino Memory," [Online]. Available:
https://www.arduino.cc/en/Main/arduinoBoardMega2560. [Accessed 1 May 2016].

[14] "Arduino Input and Output," [Online]. Available:
https://www.arduino.cc/en/Main/arduinoBoardMega2560. [Accessed 1 May 2016].

[15] "Arduino Communication," [Online]. Available:
https://www.arduino.cc/en/Main/arduinoBoardMega2560. [Accessed 1 May 2016].

[16] "Arduino Programming," [Online]. Available:
https://www.arduino.cc/en/Main/arduinoBoardMega2560. [Accessed 1 May 2016].

[17] "Arduino Automatic Software Reset," [Online]. Available:
https://www.arduino.cc/en/Main/arduinoBoardMega2560. [Accessed 1 May 2016].

[18] "Arduino Physical Characteristics and shield Compability," [Online]. Available:
https://www.arduino.cc/en/Main/arduinoBoardMega2560. [Accessed 1 May 2016].

[19] "HC-05 Bluetooth Device Overview," [Online]. Available:

https://www.itead.cc/wiki/Serial_Port_Bluetooth_Module_(Master/Slave)_:_HC-05. [Accessed1 May 2016].

[20] "Breadboard Overview," [Online]. Available: https://en.wikipedia.org/wiki/Breadboard [Accessed 1 May 2016].

[21] "Android Studio History,"[Online].Available:
https://en.wikipedia.org/wiki/Android_Studio. [Accessed 2 April 2016].

[22] "Android Sdk,"[Online]. Available:
https://en.wikipedia.org/wiki/Android_softwaredevelopment [Accessed 2 April 2016].

[23] "Gradle Overview," [Online]. Available: https://en.wikipedia.org/wiki/Gradle [Accessed 2 April 2016].

[24] "XML Overview,"[Online]. Available: http://stackoverflow.comtags/android-xml/info [Accessed 2 April 2016].

[25] "Material Design Documentation," [Online]. Available:
https://developer.android.com/design/material/index.html. [Accessed 10 November 2016].

[26] "Java Overview." [Online]. Available: https://en.wikipedia.org/wiki/Java_(programming_language). [Accessed 10 November 2016].

[27] “Google Play Service Overview.”[Online]. Available: https://developers.google.com/android/guides/overview . [Accessed 10 November 2016].

[28] “Google Map Service Overview.” [Online]. Available: https://en.wikipedia.org/wiki/Google_Maps. [Accessed 10 November 2016].

[29] “Transmitting Network Data Using Volley.” [Online]. Available: https://developer.android.com/training/volley/index.html. [Accessed 10 November 2016].

[30] “Material DateTime Picker.” [Online].Available: https://github.com/wdullaer/MaterialDateTimePicker. [Accessed 10 November 2016].

[31] “What exactly is RESTful programming?”[Online].Available: http://stackoverflow.com/questions/671118/what-exactly-is-restful-programming. [Accessed 25 November 2016].

[32] “Sublime Text Overview.” [Online].Available: https://en.wikipedia.org/wiki/Sublime_Text. [Accessed 25 November 2016].

[33] “PHP.” [Online].Available: https://en.wikipedia.org/wiki/PHP. [Accessed 25 November 2016].

[34] “Slim Framework Documentation.” [Online].Available: https://www.slimframework.com/. [Accessed 25 November 2016].

[35] “JSON Description.”[Online] Available: https://pear.php.net/package/Services_JSON. [Accessed 25 November 2016].

[36] “What is a Plimsoll line?” [Online]. Available: http://oceanservice.noaa.gov/facts/plimsoll-line.html. [Accessed 02Devember 2016].