ROBUST NAVIGATION SYSTEM (RNS)

By

Muhammad Zain Ul Abedin

MirzaMuneebBaig

Syed MuhammadTayyab

Submitted to the Faculty of Computer Science, Military College of Signals
National University of Sciences and Technology, Islamabad in partial fulfillment for
the requirements of a B.E Degree in Computer Software Engineering
JUNE 2019

CERTIFICATE OF CORRECTNESS AND APPROVAL

This is to officially state that the thesis work contained in this report

"Robust Navigation System (RNS)"

is carried out by

Muhammad Zain Ul Abedin, MirzaMuneebBaig, and Syed Tayyab Zaidi

under my supervision and that in my judgment, it is fully ample, in scope and excellence, for the degree of Bachelors of Computer Software Engineering from National University of Sciences and Technology (NUST), Islamabad.

Approved By:

Signature: ________________________________

Supervisor: **APMobeenaShehzad**

MCS, Rawalpindi

ABSTRACT

As every day passes, finding precise location has become a great concern. People try to find their location using different Apps but the existing solutions are either expensive or inaccurate.

Considering the above requirements we have developed an Android App, Robust Navigation System to simplify the way to find the location in an efficient manner. Data is received from GNSS receiver and inertial sensor, and is passed through 2D Kalman Filtering algorithm to find the exact location. The system can be easily used and can be implemented with platforms like operational areas with no cellular coverage and tunnels etc. With a simple and elegant user interface, RNS provides precise location in different environment’s.

DECLARATION OF ORIGINALITY

We hereby declare that the work contained in this report and the intellectual content of this report are the product of the sole effort of our group, comprising of Muhammad Zain Ul Abedin, MirzaMuneebBaig, and Syed Tayyab Zaidi. No portion of the work presented in this dissertation has been submitted in support of another award or qualification either at this institution or elsewhere, nor does it include any verbatim of the published resources which could be treated as a violation of the international copyright decree. We also affirm that we do recognize the terms ‘plagiarism’ and ‘copyright’ and that in case of any copyright infringement or plagiarism established in this thesis, we will be held fully accountable of the consequences of any such violation.

Dedicated to all those

who lead us on the journey

from ignorance to knowledge.

ACKNOWLEDGEMENTS

All of our gratitude is extended to Almighty Allah, the most beneficent, whose blessings empowered us to tread this journey.

We feel indebted to the benign faculty of Department of Computer Software Engineering, who raised us to these standards of knowledge; and especially to AP MobeenaShehzad, whose unending support, motivation and guidance made us capable of turning the concept of this project into reality.

We also extend our thankfulness to our families and friends, who kept our moral spirits high in the times of need.

TABLE OF CONTENTS

1. Introduction 2

1.1 Motivation..2

1.2 Problem Statement 2

1.3 Project Scope 2

1.4 Project Objectives 3

1.5 Document Organization 3

1.6 Intended Audience and Reading Suggestions 4

1.7 Summary 4

2. Literature Review 6

2.1 Introduction..6

2.2 Product Perspective..6

2.3 Product Features...7

2.4 User Classes and Characteristics..7

2.5 Operating Environment..8

2.6 Design and Implementation CO...8

2.7 User Documentation 8

2.8 Assumptions and Dependencies...9

3. Requirements 11

3.1 Introduction..11

3.2 External Interface Requirements..11

3.3 Functional Requirements 12

3.4 Other Nonfunctional Requirements 16

4. Design... 20

4.1 Introduction..20

4.2 Overview of the Modules...20

4.3 Structure and Relationships 22

4.4 UI Issues 30

4.5 Detailed Description of Components..39

4.6 Summary 42

5. Implementation 44

5.1 Introduction 44

5.2 Tools & Technologies 45

5.3 Summary 49

6. Testing 51

6.1 Introduction 51

6.2 Test Items 51

6.3 Tested Features 51

6.4 Features not to be tested 52

6.5 Approach 52

6.6 Suspension Criteria and Resumption Requirements 52

6.7 Testing Tasks 53

6.8 Staffing and Training Needs 53

6.9 Test Cases 53

6.10 Summary 62

7. Conclusion& Future Work 64

7.1 Conclusion 64

7.2 Design Decisions & Tradeoffs 64

7.3 Future Work 65

Bibliography 66

LIST OF TABLES

Table 4-11 - Logout Use Case 29
Table 4-12 - Class Diagram Description.. 30
Table 6-16- Logout Testing 62

LIST OF FIGURES

Figure 2-1 - Client-Server Architecture 7
Figure 4-1 - Abstract Diagram 20
Figure 4-2 - Architecture Diagram 23
Figure 4-3 - Use Case Diagram 26
Figure 4-4 - Class Diagram 29
Figure 4-5 - Interface Diagram 30
Figure 4-6 - User Management – Activity Diagram 32
Figure 7 – User Registry – Activity Diagram 32
Figure 8 - Viewing Help – Activity Diagram 33
Figure 9 - Calculating Location – Activity Diagram 34
Figure 10 - *Registry – Sequence Diagram*........ 35
Figure 11–Getting location– Sequence Diagram 36
Figure 12 –View Help– Sequence Diagram 37
Figure 13 –Administrator rights– Sequence Diagram 38
Figure 14 - Welcome Screen........ 46
Figure 17 - Map Screen........ 49

KEY TO SYMBOLS OR ABBREVIATIONS

RNS	Robust Navigation System
Apps	Applications
CO	constraints
D	Dependency
OE	Operating Environment
REQ	Functional Requirement
SDS	Software Design Specification
SE	Security Requirements
SF	Safety Requirements
SQL	Structured Query Language
SRS	Software requirements specification
STP	Software Test Plan
UD	User documentation
UI	User Interface
INT	Inertial

CHAPTER 1

Introduction

1. Introduction

1.1 Motivation

The motivation behind making this system is that we have seen army personnel in severe need of a good aid to help them find precise location as they have to direct missiles etc.Other than this there’s the need of finding a location in tunnels etc.Trekkers also find this very useful as they also need to stay alert about their location.

1.2 Problem Statement

To find the most precise location and error free

1.3 Project Scope

- We will get a stream of data through inertial sensors
- By combining the data obtained from the GNSS receiver and the inertial sensors the data will be passed through 2D Kalman Filtering
- Using Raspberry pie the location can be exactly found by implementing the Kalman Filtering algorithm
- The system can be given to any ordinary person for normal use

1.3.1 External Scope

The scope of the project can be increased to be implemented for PG and PhD courses as well. Similarly, it can be used by other departments after slight modifications according to their own requirements.

1.4 Project Objectives

1.4.1 Academic Objectives

- To understand the core concept of programming.
- To understand the concepts of GPS and location
- To understand the concept of android development
- To go through the process of professional software development

1.4.2 Application/End-Goal Objectives

The Application Objectives are as under:

- Portability
- Easy to use
- Cheap
- Giving the precise location

1.5 Document Organization

The document looks at all the aspects of development and usage of RNS since its initial stage to its completion. Starting with Chapter 1 and 2 covering an introduction and literature review of the topic, Chapter 3 and 4 further discusses the process of requirement elicitation and design development. After completion of these stages, Chapter 5 and 6 of the document covers details regarding implementation and testing of the system.

1.6 Intended Audience and Reading Suggestions

The document is meant for the following stakeholders.

- **Project Supervisor:** to assist in project supervision and guiding the team in a better way.
- **Development Team:** to help in the development of product and trace-back of functional requirements.
- **Testing Team:** to help the testers to understand the applicable CO.
- **Users:** The potential stakeholders of the system, including the UG Students and faculty of MCS CSE Dept.
- **UG Project Evaluation Team:** to assist the evaluation committee in evaluating theprogress of UG Projects.
- **Staff:** to understand the process of data entry into the system's database.

1.7 Summary

This chapter gives a brief overview of the system and its functionalities, as well as this document, which covers all of the essential details regarding RNS. This chapter also covers details regarding the scope and objectives of the system.

CHAPTER 2

Literature Review

2. Literature Review

2.1 Introduction

Robust Navigation System is projected as a platform for aiding an ordinary person for his normal day work. The system will advance the way we look at precise location which is previously being done at a high tech level as it is being used in rockets etc. This would prove to befit all the categories of individuals that interact with the Department for academic reasons as well as for corporate reasons. In todays world when a person wants to find his location he can but that location is approximate .He needs something through which he can find his exact location and that too at a low price.

2.2 Product Perspective

Currently, when a person wants to see his location, the individual opens an application and finds his location which is in verbal sense a beneficial thing but that location is actually not precise but accurate.This is where the RNS comes in to play. It aims to find the exact location for the ease of the user. Furthermore, it will be a new, standalone product for those looking to find their location at a cheaper price too. In addition to this the location found will be with minimal instrumental error and when in tunnels or in those places where the signal can be lost we can get the location using the inertial sensors easily.

We will follow architecture technique as mentioned in the diagram below:

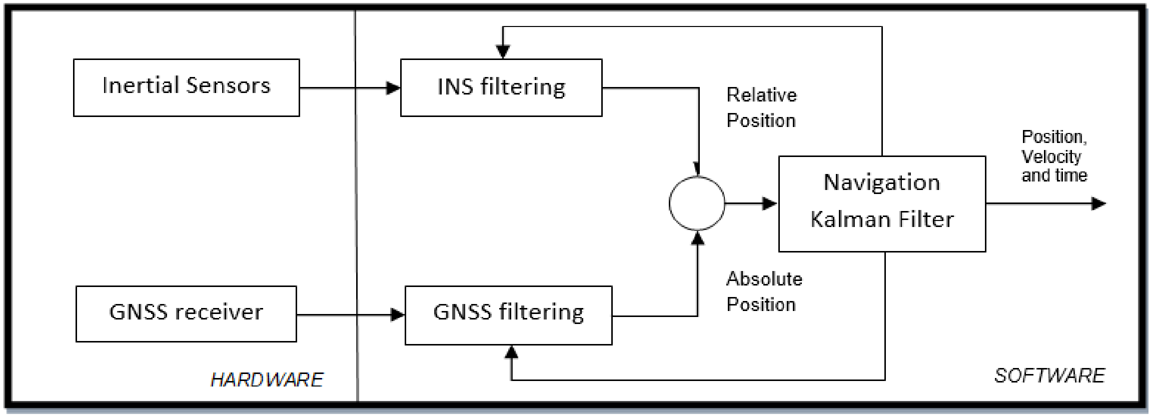


Figure 2-1 - Client-Server Architecture

2.3 Product Features

Following are the major functions of the Robust Navigation System (RNS):

- The precise location will be found
- With minimal instrumental error
- When in those locations where the signals weak using inertial sensors location found will be maintained
- The GUI will be very user friendly and even a person having minimal knowledge about the application can use it.

2.4 User Classes and Characteristics

The software has multiple types of users: Normal public, military personnel, trekkers and travelers etc. All the of users can perform functions in their respective fields.

- **Military personnel: -**When the Military personnel go to operational areas to perform a mission etc. who require the knowledge of their exact location this can be used.
- **Trekkers: -**When trekkers go to trekking they also need to find their exact location keeping in view to determine the path to their destination.

2.5 Operating Environment

OE-1: The RNS shall run on Raspberry Pie with following specifications:

- 1 GB Ram
- It has peripherals added to it for further tasking
- The inertial sensors will be connected to this
- SD card will also be needed for this

OE-2: Mobile will be used for output

OE-3: The input will be given to raspberry pie using inertial sensors

2.6 Design and Implementation CO

CO-1: Python based coding

CO-2: Android app on which Google map API will be installed

CO-3: Lack of user-expertise in using the applications on internet.

CO-4: Internet connection needed.

CO-5: Use of English language as the only means of communication in the system.

2.7 User Documentation

UD-1: Final release will be accompanied with an on app user guide to inform users how to use Robust Navigation System (RNS).

2.8 Assumptions and Dependencies

AS-1: Once there is a requirement to find the location only then this system is used.

AS-2: User has no authority in changing of the code but only developer has that.

AS-3:.This system will be able to give user the real time coordinates.

AS-4: User will get the exact coordinates.

AS-5: Once we are using the system all the sensors and necessary subjects must be in an active state.

D-1: There will be a permanent D on the Kalman filter algorithm, as without this, the errors cannot be minimized.

D-2: System is dependent on the inertial sensors as in those places where signals are hard to reach it cannot work without them.

CHAPTER 3

Requirements

3. Requirements

3.1 Introduction

The purpose of this chapter is to present a detailed description of the Robust Navigation System. It will explain the purpose, features, interfaces, functionality, entire process, CO and the application's reaction to external stimuli. It is intended for stakeholders and the system developers.

3.2 External Interface Requirements

3.2.1 User Interfaces

RNS will have an android app with which the user will be able to communicate with the system. The app will communicate using Bluetooth connection after the connection a map will open through which the precise coordinates will be achieved.

3.2.2 Hardware Interfaces

Computer System

- System shall have keyboard input.
- System shall have mouse input.
- System shall have a monitor.

Mobile

- To process requests and retrieve data

3.2.3 Software Interfaces

User-Based

- The user will interact through android app
- The mobile Android version must be above 5.1

Developer Based

- Raspberry pie 3 is a must
- Python coding will be used in compiler

3.3 Functional Requirements

Finding the location:

Description and Priority

The developer will enable the users to easily find their location and in case of developers they can modify the app as well. The priority of this system to find the location is high.

Stimulus/Response Sequences

Input: Input will the coordinates of the location gained by the GNSS receiver and Int sensors and input coordinates will be processed through Kalman Filter.

Output: The output will be precise location displayed on an android mobile app.

Functional Requirements

REQ-1: The RNS shall be able to gain coordinates from GNSS receiver and Int sensors.

REQ-2: The RNS shall be able to process the coordinates through the Kalman filter algorithm.

Login / access rights

Description and Priority

The RNS will consist of open source database. The purpose of the login is maintain the record of people who have accessed it. That person who has not registered their self can sign up.

Stimulus/Response Sequences

Input:	User will enter login credentials to the system or sign up.
Output:	The system will grant the access to the user and maintain the log book.

Functional Requirements

REQ-3:	The RNS shall allow all the users to login to their account.
REQ-4:	The RNS shall grant the new user to sign up for new registry.

Output through the RNS Android App

Description and Priority

The Google Map API will be attached with the Android App. The coordinates received after processing through the Kalman Filter from the Raspberry Pi 3 via Bluetooth in real time will be displayed in the App.

Stimulus/Response Sequences

Input:	Received coordinates from Raspberry Pi3.
Output:	Pointer on the Google Map App showing the precise location.

Functional Requirements

REQ-5: The RNS will show the accurate pointer on the Map showing the location.

REQ-6: The RNS will display the numerical coordinates.

Wifi Connectivity

Description and Priority

The Android mobile will be connected to Raspberry Pi3 through Wifi. The data must be shared in real time.

Stimulus/Response Sequences

Input: Processed coordinates from RNS.

Output: The data will be delivered to RNS Android APP

Functional Requirements

REQ-7: The RNS shall deliver the data in real time.

Processing within the Raspberry Pi3

Description and Priority

The processing of Kalman filtering will be done within the Raspberry Pi3 which will produce the minimal error location and when the GNSS receiver signals are down the location will be calculated through INS sensors.

Stimulus/Response Sequence

Input: X, Y axis coordinates through GNSS recover and INS seniors.
Output: The processed location.

Functional Requirements

REQ-8: The RNS will take the X,Y coordinates from GNSS receiver.
REQ-9: The RNS will apply Kalman filter on GNSS received coordinates.
REQ-10: The RNS will automatically shift to INS sensors processing.

User Guide

Description and Priority

The RNS Android App will consist of a user guide that will consist of the description of RNS working and how to use the App.

Stimulus/Response Sequences

Input: The user have to click on the user guide button to open the description.
Output: User Guide will be displayed.

Functional Requirements

REQ-11: The user can access the user guide without even logging in to the Android App.

3.4 Other Nonfunctional Requirements

3.4.1 Performance Requirements

Performance Requirements

Certain functionalities will be required, based on the performance and response of RNS. RNS will operate in real time.

SF

SF-1: Database will be used to maintain the registry of user logging in.

SF-2: Database can be checked by the developer.

SF-3: If system not working the user can contact the developer.

SF-4: Identity of user accessing the RNS shall not be disclosed to the public.

SR

SE-1: Users shall be required to log in to the RNS for their own credential information.

SE-2: The system shall allow new users to sign up.

SE-3: The system shall maintain the time record of user using the RNS.

SE-4: The system must perform an encoding technique such as hashing to save all passwords securely.

SE-5: The System will provide confidentiality and integrity.

3.4.2 Software Quality Attributes

Quality attributes of RNS are described below. By following these attributes, the quality of RNS will be improved.

Runtime System Qualities

At runtime RNS has to provide its users with functionalities so that they can get the precise location. Some of the runtime qualities that should be considered in the development of RNS are described here.

Functionality

RNS must provide functions to access the location. RNS must provide the functions of authentication of user.

Availability

RNS should be available 24/7 since the user can get in need of finding the location at any time.

Usability

Usability is an important criterion in the development of RNS. The system should present all functionalities in such a way that nothing is missed by the user. The Graphical UI of RNS app must be easy to be used and is the prime focus. It will help the user to navigate easily without any difficulties.

Non-Runtime System Qualities

These are qualities of RNS which are required to make this software useful for further enhancements. It will also be helpful in future development as well as extending system to different environments.

Modifiability

RNS must support modifiability so any further improvements or features are easy to incorporate.

Portability

RNS should be able to run in different computer environments. The RNS server should be a platform-independent and should support interoperability.

Testability

Different quality tests should be performed so that RNS is free from faults and perform according to requirements.

CHAPTER 4

Design

4. Design

4.1 Introduction

This design chapter covers all our functional requirements and demonstrates how they interrelate with each other abstractly. The low level design also illustrates as to how we have been implementing and how we are going to implement all of these requirements. This low level design for the time being does not address any non-functional requirements that our system has and that have been mentioned in the SRS Document.

4.2 Overview of the Modules

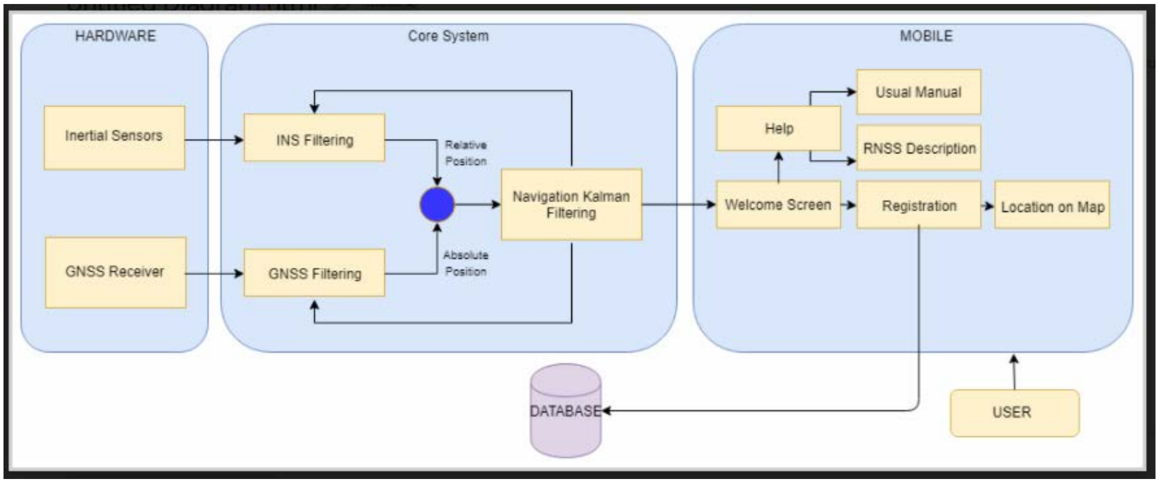


Figure 4-1 - Abstract Diagram

4.2.1 Explanation of Abstract Diagram

The system will be architected mainly in four fundamental modules “User”, “Core System”, “Mobile”, and the “Database”. It will further be having sub modules as shown in the abstract diagram above. Abstract diagram provides an overview of the system, from users accessing the system till the processing in databases. The sub modules of the Abstract diagram are further elaborated below.

User

The user is basically the person who will use the system

Inertial Sensors

Inertial Sensors consists of Accelerometer which gives us relative positioning values which will be used further if GNSS fails

GNSS receiver

The GNSS receiver is basically the GPS receiver which gets us the absolute location coordinates based on longitudes and latitudes.

INS Filtering

The GPS gives NMEA strings that contains the relative position values that are used to future calculate and help the RNS to provide base. The precise direction of the unit can be deduced from the arrangement of position refreshes from the GPS. The stings provided by the GPS can be used to make them more précised and near to accuracy.

GNSS Filtering

In GNSS we take the values from the receiver and pass it into kalman filter.

Navigation Kalman Filter

In this the main thing is basically being done that is the values of the INS receiver and the GNSS receiver are being mathematically being combined and one exact and more accurate location is being received

Database

The database stores all the entry's made and information about the person that made the entry.

Mobile

It contains the main interface as well as the user manual in which the details to use the application are given.

4.3 Structure andRelationships

3-Tiers Architecture will be utilized to execute RNSS. Presentation Tier: Is available at the top levels and it is used show data that is accessible with the administrative privileges. This level speaks with different levels by sending results. Application Tier: Also known as middle tier, logic tier, business logic or logic tier. This layer is accessed by Presentation Tier. It is responsible for application functionality and takes care od its responsibilities by performing detailed processing. Data Tier: This tier manges all database servers. All information is stored and retrieved from Database through this tier. Data in this layer is not dependent on both the other layers.

Tiers Details

Presentation Layer

It gives a stage to the association of the client with the framework. It shows information to the client and acknowledges contribution from the client. This is the part which gets the solicitation and returns the reaction. The Presentation layer can just get demands from, and return reactions to, an outside specialist. This is normally an individual, however might be another bit of programming.Can only send requests to, and receive responses from, the Application Tier. No direct link is available with the database or even the access layer.

Application Layer

It is accessed by the presentation layer.It is responsible for all type of application functionality and it is done by performing extensive processing. An applicationmust provide services to other applications.

Data Layer

This layer receives request from the Application Tier and sends back data after querying it from the database server.

4.3.1 Architecture Diagram

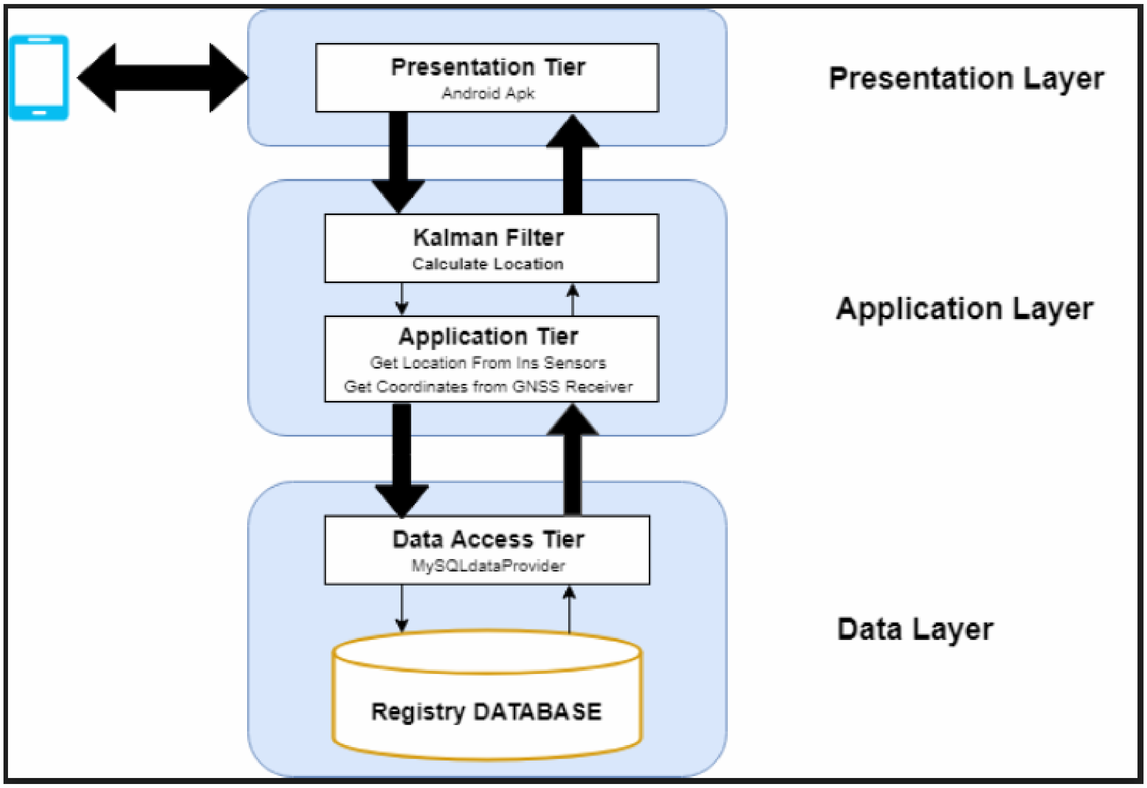


Figure 4-2 - Architecture Diagram

Once a user registers the system, he is granted access to the RNS Map. Where he can see the location on the map and get the coordinates. The registry is sent to raspberry Pie where the data is saved in Database.

Following is a detail of the tiers and their components.

Calculate Location

Kalman filter takes in data from sensors (RNSS receivers and INS seniors). Absolute coordinates from RNSS and relative from INS sensors. These two values to taken in account for calculating the location which is sent to android app.

Get Coordinates from GNSS Receiver

The GNSS Receiver is used to get absolute location that is sent to Raspberry Pie that is used to calculate Location.

Get Location from INS Receivers

Location from INS receiver is used to calculate the location when the RNSS Receiver's signals are too weak and are unable to maintain the signals. At that time the x, y-coordinates values are used to calculate the location.

Database

Database is used to maintain the registry of the login that are made this registry consists of Name, email and mobile number of the person with the time spam

4.3.2 Use Cases

A use case is a procedure whose purpose is to perform system analysis. This is done by identifying, clarifying, and organizing system requirements.A use case consist of all the sequences that are possible while communication and performing the actions and transaction between different stakeholder and the system.

The various user classes identified the following use cases and primary actors for the RNS:

Actors	**Use Cases**
User	• Register • View Help • Location
Administrators	• Access Database

The aforementioned use cases can be represented in the form of the following Use Case Diagram.

4.3.2.1 Use Case Diagram

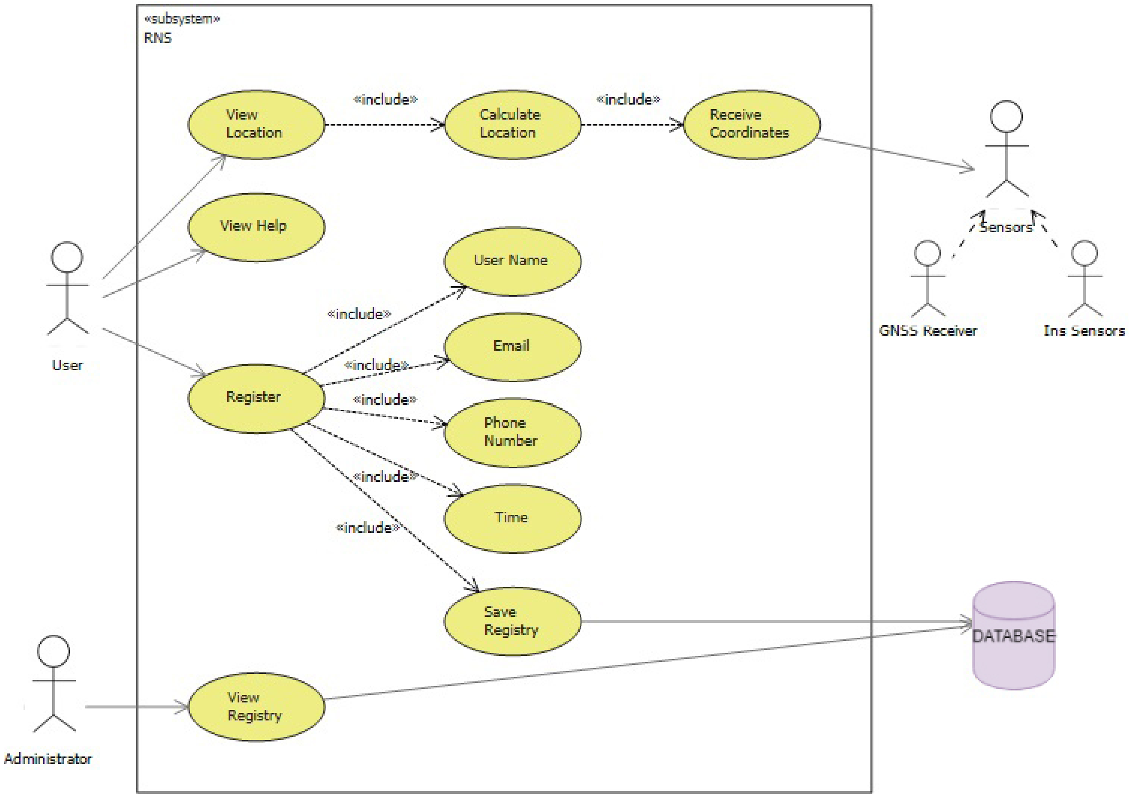


Figure 4-3 - Use Case Diagram

Use Cases Description Manage Accounts

Use Case ID:	1		
Use Case Name:	Location		
Actors:	User		
Created by:	Zain	Last Updated by:	Zain
Date Created:	27/12/2018	Date Last Updated:	27/12/2018
Description:	The user will get the location which will include getting the coordinates and calculating the location		
Preconditions:	User has to login		

Post conditions:	-
Normal Flow (Primary Scenario):	1.The user will login 2.The user will find the location
Alternative Flows:	1.The location cannot be found as the coordinates were not received 2.Find the coordinates again

Login

Use Case ID:	2		
Use Case Name:	Register		
Actors:	User		
Created by:	Zain	Last Updated by:	Zain
Date Created:	27/12/2018	Date Last Updated:	27/12/2018
Description:	The user will login which will include his Name, Email, Phone No, and Time at which he logged in and an entry in the registry will be made.		
Preconditions:	The user will open the app		
Post conditions:	After logging in the user will find his location		
Normal Flow (Primary Scenario):	The user will login and an entry will be made that the system was logged in at that particular time.		
Alternative Flows:	-		

View Help

Use Case ID:	3

Use Case Name:	View Help		
Actors:	User		
Created by:	Zain	Last Updated by:	Zain
Date Created:	27/12/2018	Date Last Updated:	27/12/2018
Description:	The user will check the way to use the application and find the location		
Preconditions:	The app has to opened first		
Post conditions:	The help menu will open 22		
Normal Flow (Primary Scenario):	The user will open the application and check the help menu		
Alternative Flows:	-		

Access Database

Use Case ID:	4		
Use Case Name:	Access Database		
Actors:	Administrator		
Created by:	Zain	Last Updated by:	Zain
Date Created:	27/12/2018	Date Last Updated:	24/12/2018
Description:	The administrator will access the database to check the registry		
Preconditions:	-		
Post conditions:	-		
Normal Flow (Primary Scenario):	The administrator will access the database through the desktop and will view the entries in the registry.		

Alternative Flows:	-

4.3.2.2 Table 4-1 - Logout Use Case

4.3.3 Class Diagram with Description

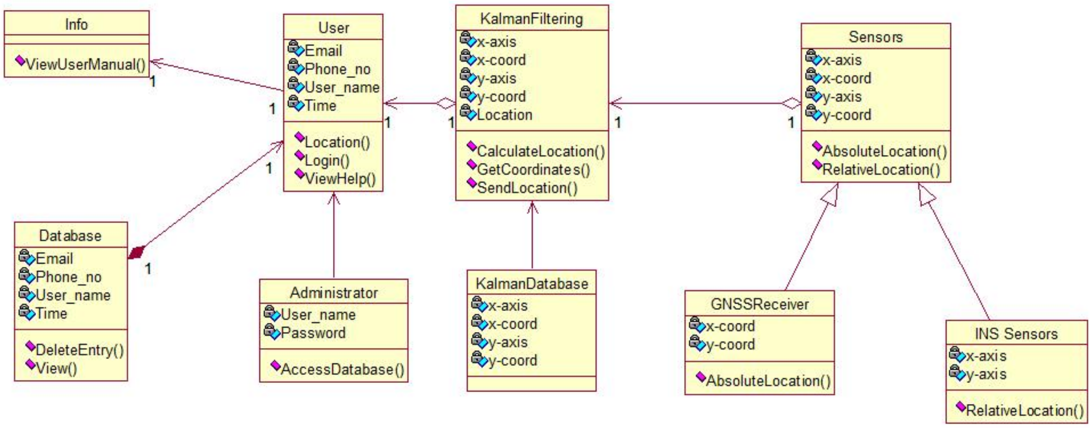


Figure 4-4 - Class Diagram

Class Name	Description
Info	Info class contains the information on how to use the application as well as the system to find the location.
User	User class contains information which is entered by the user at the time when he is registering and when he wants to get the location so basically an entry is made in the database whenever someone uses the system.
Database	Database class contains information which has been entered by the user and entries are made in the database whenever someone uses the system.

Administrator	The administrator class contains the login items which are entered by the admin when he wants to login to the system in admin mode.
Kalman Filtering	Kalman filter algorithm class is that which integrates the coordinates to find the exact location.
Sensors	The sensors class is which gives absolute and relative position by using the GNSS receiver and the INS receiver after this the values go’s into the kalman filter.

Table 4-2 - Class Diagram Description

4.4 UI Issues

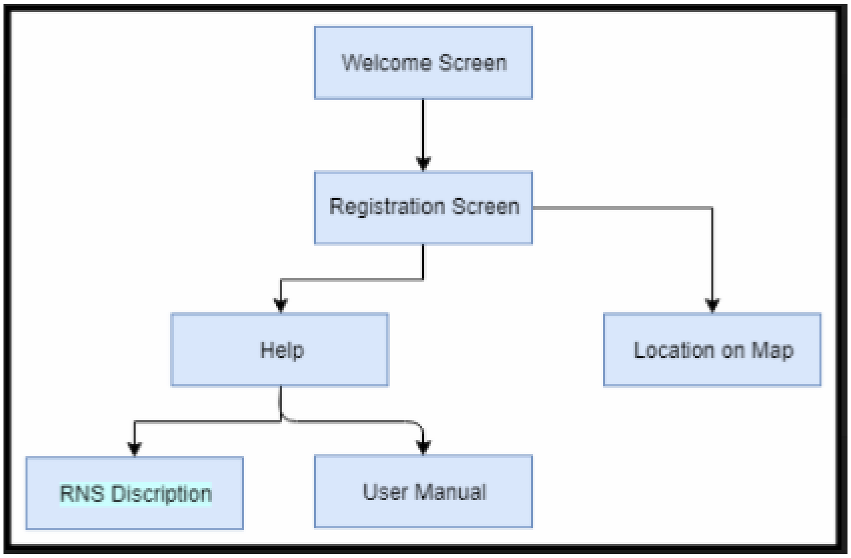


Figure 4-5 - Interface Diagram

4.4.1 Description of the Diagram

Welcome Screen

This is the main welcome screen.

Registration Screen

The registration screen requires the person to enter his details ie Email, Phone, Time, username.

Help Screen

This screen contains the details through which we come to know how to use the Application as well as the system.

Location on Map

This Screen is basically showing us the location on the map in real time.

RNS Description

The RNS description page contains the Description of the project and its members.

User Manual

This screen actually contains the User manual which will contain the step wise procedure to use the app.

4.4.2 Activity Diagrams

4.4.2.1 User Management

The diagram below displays how the users are logged into the system and displayed specific interfaces according to their category.

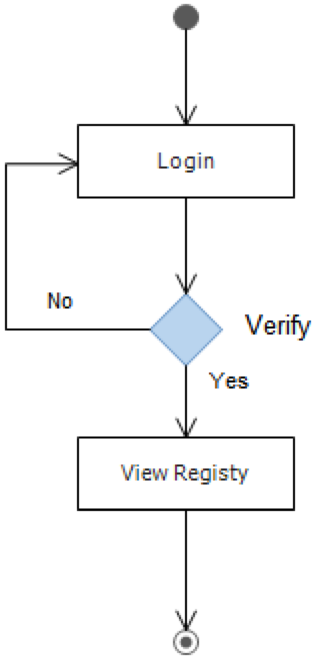


Figure 4-6 - User Management – Activity Diagram

User Registry

Shows how the user logs in to the system which is basically just for the entry sake.

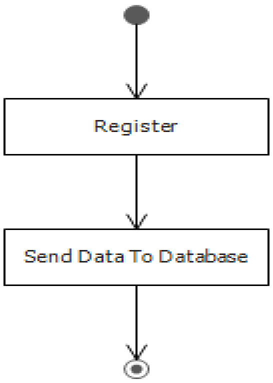


Figure 7 – User Registry – Activity Diagram

Viewing Help

The diagram below describes the feature of viewing help and getting to know the system.

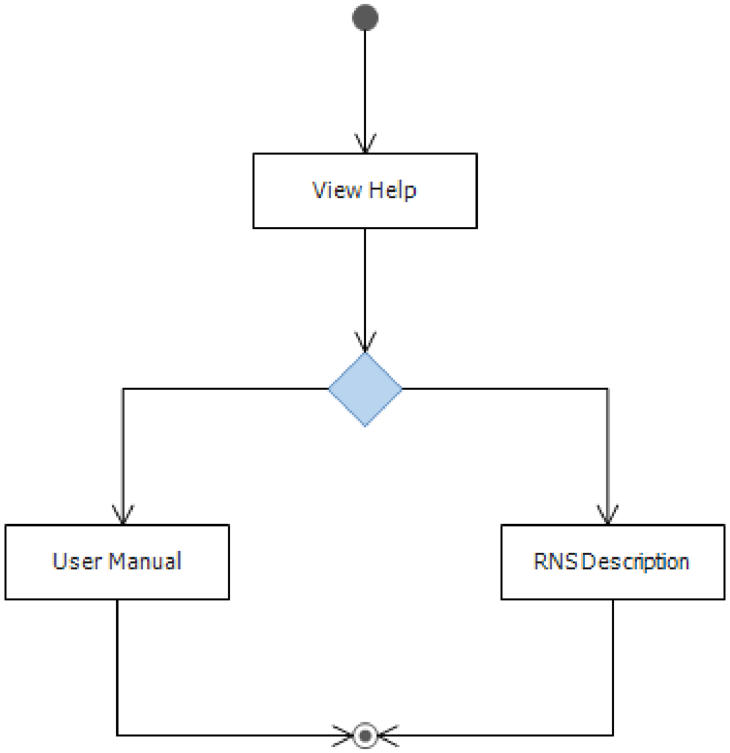


Figure 8 - Viewing Help – Activity Diagram

Calculating Location

This Diagram shows us how to the location is being calculated.

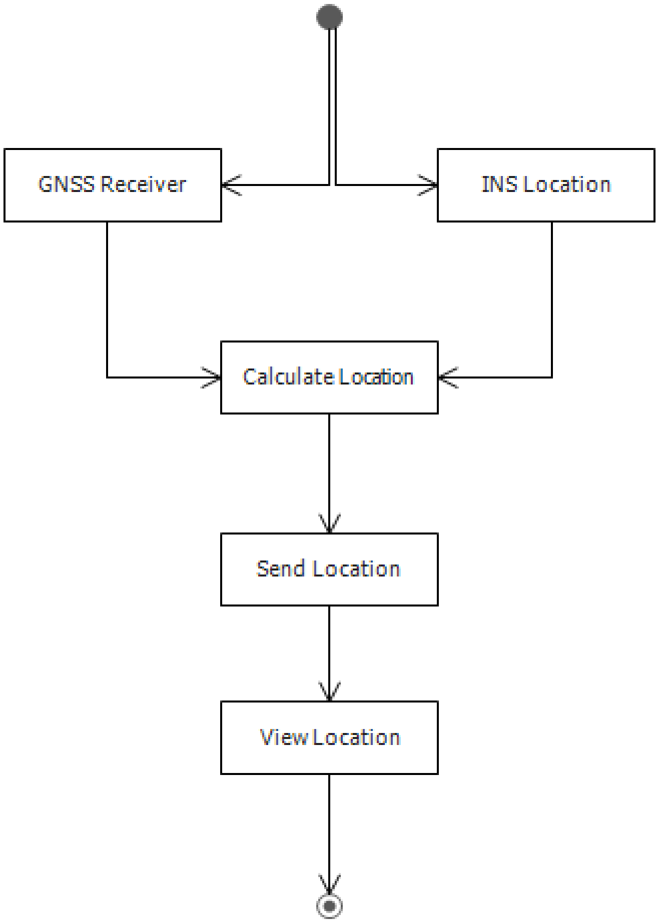


Figure 9 - Calculating Location – Activity Diagram

Sequence Diagrams

Registry

The following diagram shows the sequence of events as a person tries to login into the system

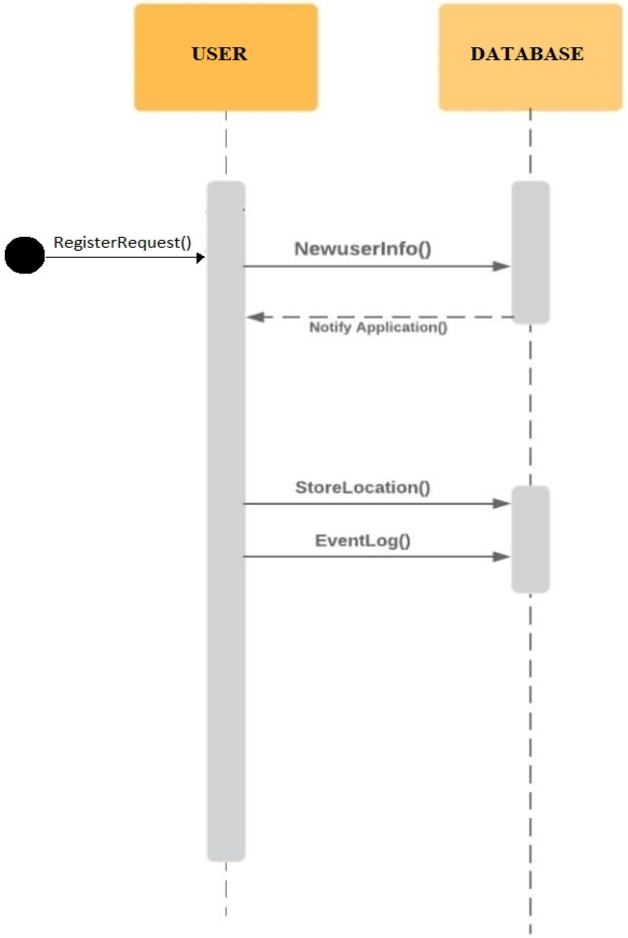


Figure 10 - Registry – Sequence Diagram

Getting Location

This sequence diagram displays the procedure of getting the location using the kalman filter.

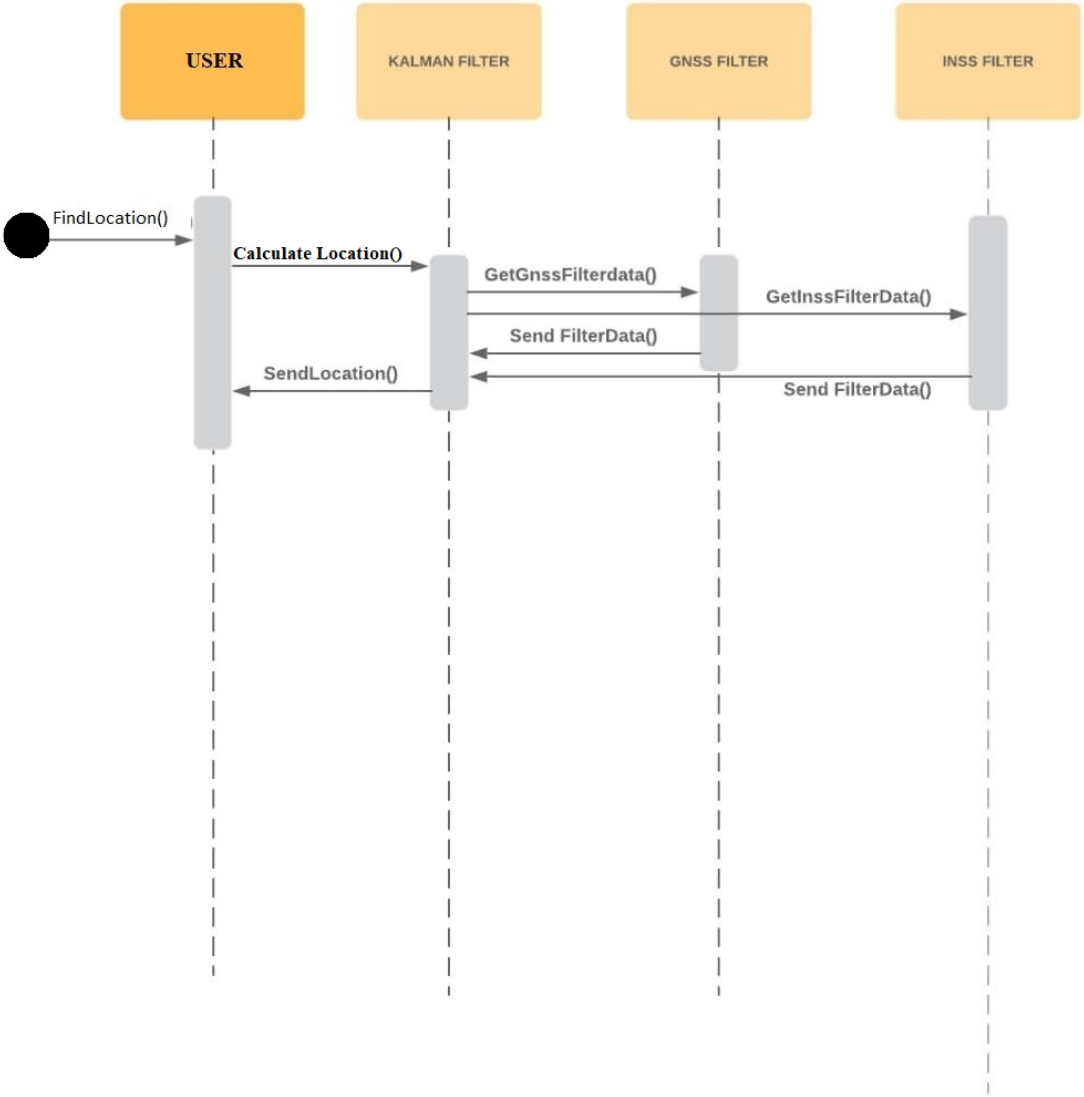


Figure 11–Getting location– Sequence Diagram

View Help

This sequence diagram is to view help.

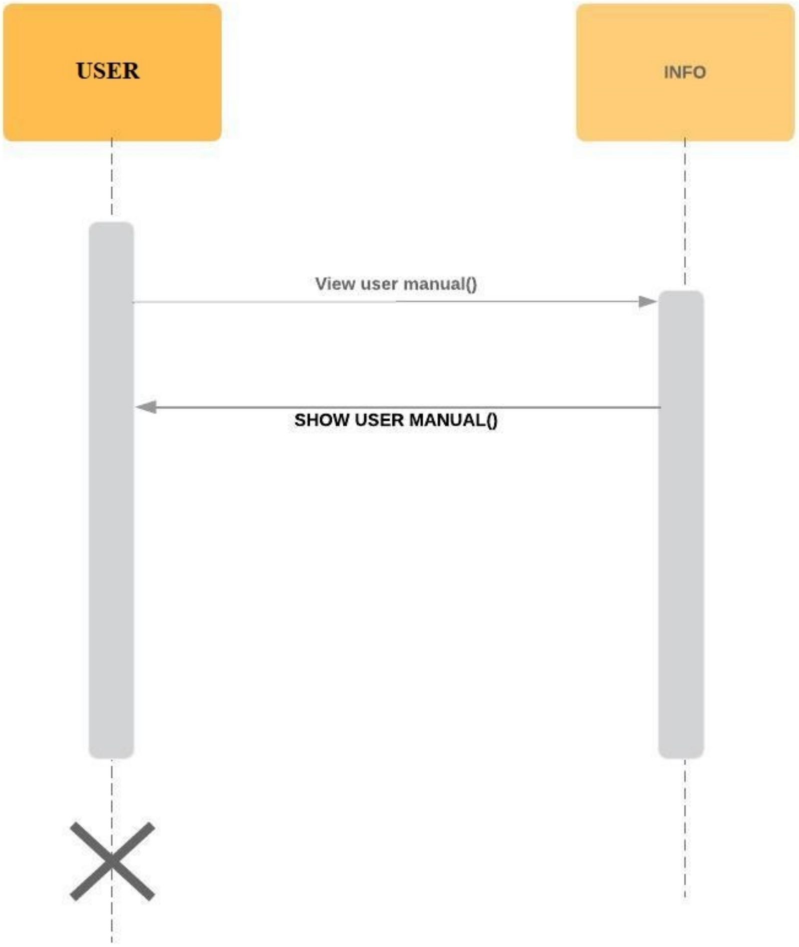


Figure 12 –View Help– Sequence Diagram

Administrator rights

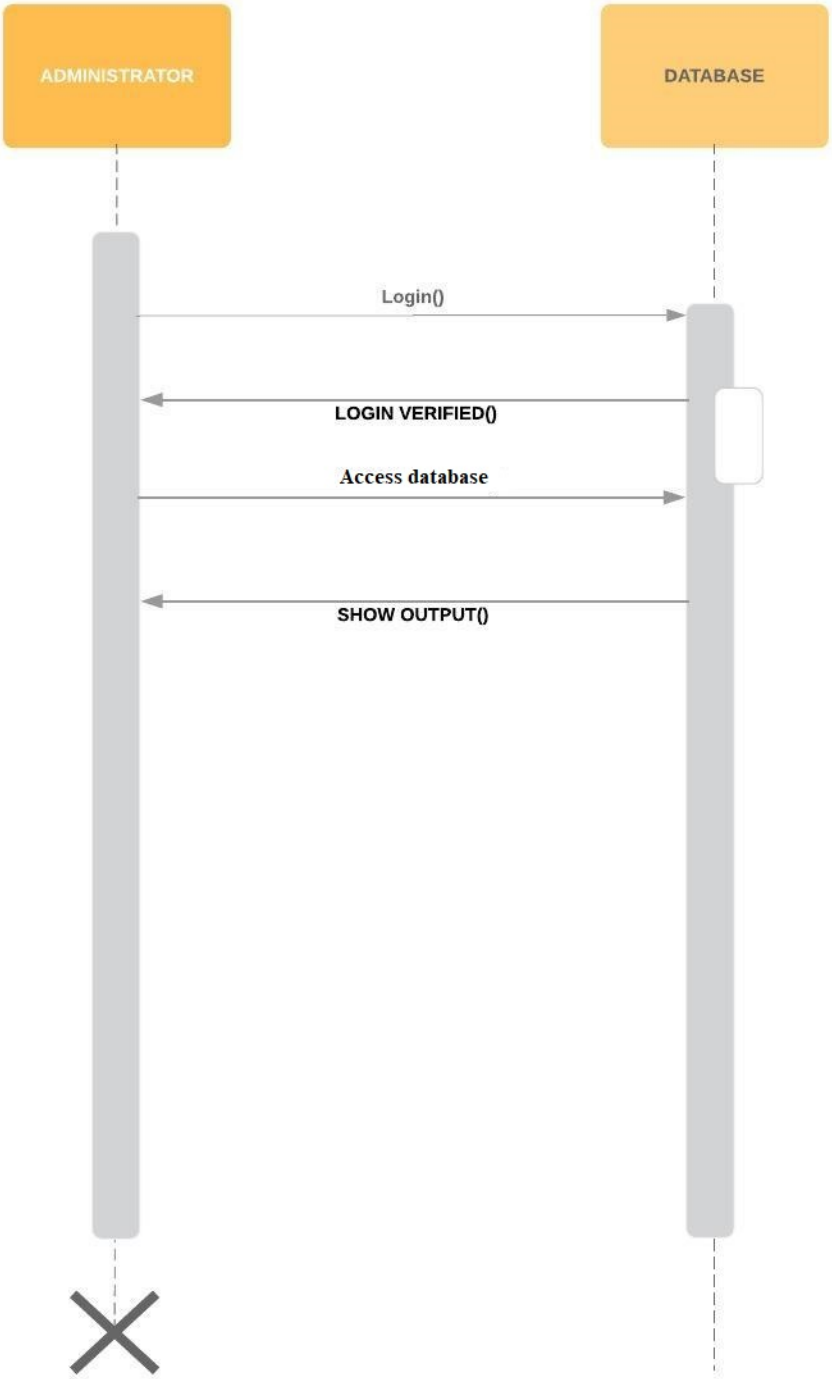


Figure 13 –Administrator rights– Sequence Diagram

4.5 Detailed Description of Components

UI Module

Identification	**UI Module**
Type	UIComponent
Purpose	• Gathering configuration related information from the user • Displaying processed location data such as coordinates etc.
Function	• Login(Username, Password) • Logout() • Getconfiguration() • Getinput()
Subordinates	Backend will be necessary to serve UI layer.
Dependencies	Data from Backend is necessary
Interfaces	• Login(button) • Logout(button)
Resources	User Database Android 5.0 onwards
Processing	• Login(username, password) logs in the user • Logout(user) logs out the user • VerifyDetails() verify user credentials from the user database • GrantsAccess() grants user access in the user database
Data	Coordinate information. User information

Backend

Identification	**Backend**
Type	Server
Purpose	• To serve UIdata
Function	• Getcoordinates()

	• GetUI() • Getuserdata()
Subordinates	Data should be preprocessed in desired form
Dependencies	Preprocessed data Formatted Coordinate data
Interfaces	• View Map from Google Api • View help
Resources	TCP capable computational node
Processing	• Build User interface • Format data
Data	Coordinates User configuration

Data preprocessor

Identification	**Data preprocessor**
Type	Component
Purpose	• Convert serial data to desirable formatted byte stream. • Apply kalman filter
Function	• ConvertSerialData() • KF()
Subordinates	Serial Microcontroller
Dependencies	Serial data stream.
Interfaces	• Access serial data stream
Resources	X86 RISC compatible computational node
Processing	• Gives calculated location
Data	Database repository for maintaining the user record, based in their individual access rights.

Serial Microcontroller

Identification	**Serial microcontroller**
Type	Hardware microcontroller
Purpose	• Gather raw coordinate data from hardware modules GNSS and INS • Convert raw data into process able serial stream
Function	• Get INS data • Get GNSS data • Convert to process able serial stream
Subordinates	Hardware modules
Dependencies	Depends on raw data from GNSS and INS
Interfaces	• button->text() [text overlay on the button] • button->bitmap() [graphic for the button] • button->screenX(), button->screenY() [screen coordinates for the button]
Resources	RISC compatible microcontroller
Processing	• GNSS data is converted into formatted strings • INS Data is converted into formatted strings
Data	INS raw coordinate stream

INS Module

Identification	**INS Module**
Type	Hardware Component
Purpose	• Gathers data from accelerometer and gyroscope
Function	•
Subordinates	Accelerometer, gyroscope.
Dependencies	-
Interfaces	GetINSDataStream()
Resources	Hardware Module

Processing	Accelerometer and gyroscope data are converted into serial
Data	X,Y,Z accelerations

GNSS module

Identification	**GNSS Module**
Type	Hardware Module
Purpose	• Find Location
Function	• Gather GPS data from satellite
Subordinates	Satellite
Dependencies	Satellite Transmission Area must have Line of Sight Connection
Interfaces	GetSatelliteLock(Min no of Satellites)
Resources	Satellite
Processing	GetGNSSData()
Data	Radio Receiver

4.6 Summary

The purpose of this chapter was to deliver a portrayal of the blueprint of the system reasonable enough for understanding of what is built and how it is developed. It provides information essential to getting a description of the details for the software and the system to be built. It presents a design view and detailed description of RNS. It explains the purpose, features, interfaces, what the system does, complete procedures in detail, the CO under the conditions and how the system reacts to inputs and what will be its outputs.

CHAPTER 5

Implementation

5. Implementation

5.1 Introduction

The preceding chapter discoursed comprehensive design of the RNS. This design is converted into an application by utilizing numerous technologies and tools. The implementation details are conferred in the subsequent divisions providing minutiae of the system's inner functioning.

5.2 Tools & Technologies

5.2.1 Python

Python is a deciphered, abnormal state, broadly useful programming language. Python’s plan reasoning accentuates code meaningfulness with its eminent utilization of huge whitespace.

5.2.2 Android Studio

Android Studio is the authority coordinated improvement condition for Google's Android working framework, based on JetBrains' Intelligent IDEA programming and structured explicitly for Android advancement and utility for extensive development. It is readily accessible for download on Windows, macOS and Linux based OS.

5.2.3 MySQL

MySQL is an open-source social database of the board framework. Its name is a mixture of "My", the name of prime supporters Michael Wideness' little girl, and "SQL", the curtailfor Structured Query Language.

5.2.4 Flask

Flask is a smaller scale web structure written in Python as a facilitator. It is delegated a small scale system since it doesn't require specific apparatuses or libraries. It has no database deliberation layer, structure approval, or whatever other segments where previous outsider libraries give normal capacities.

5.3 UI Design

The system under development shall support an intuitive and easy to use UI that will have an extremely shallow learning curve and require minimum training to be operated at maximum efficiency. In this way, users of the Android application won't feel reluctant to use the application to view his location.

Welcome Screen

Following are the sketches of a possible UI implementation for RNS. This will be the first screen that the user sees upon opening the RNS. The user shall be presented with a welcome screen that consists of RNS logo and its name. It will be displayed for 3 seconds. As visible, the interface is simple and self-explanatory.

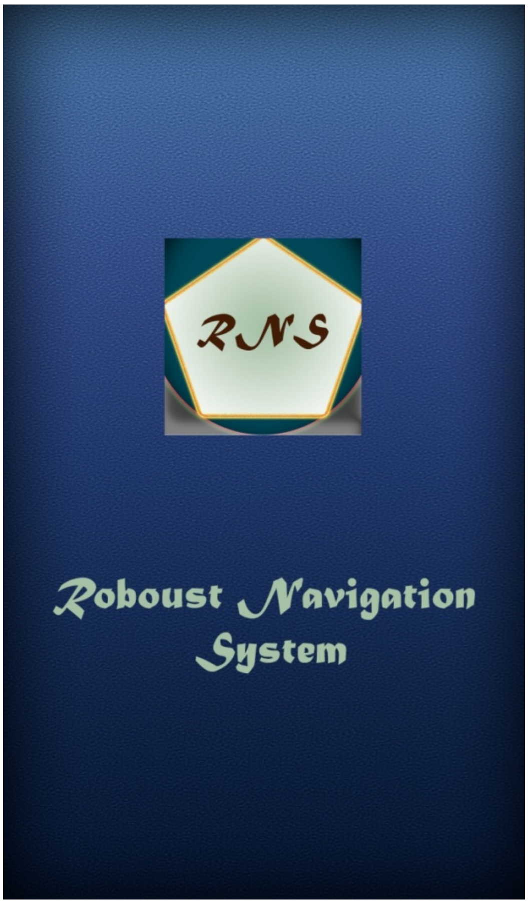


Figure 14 - Welcome Screen

Registration Screen

After the welcome screen registration screen is showed. It consists of RNS logo and multiple input fields. Input fields consist of user name, email and phone number. It also consist of a registration button. Once the user have entered the data and clicks the registration button data is sent to The Database for registration along with the time span.

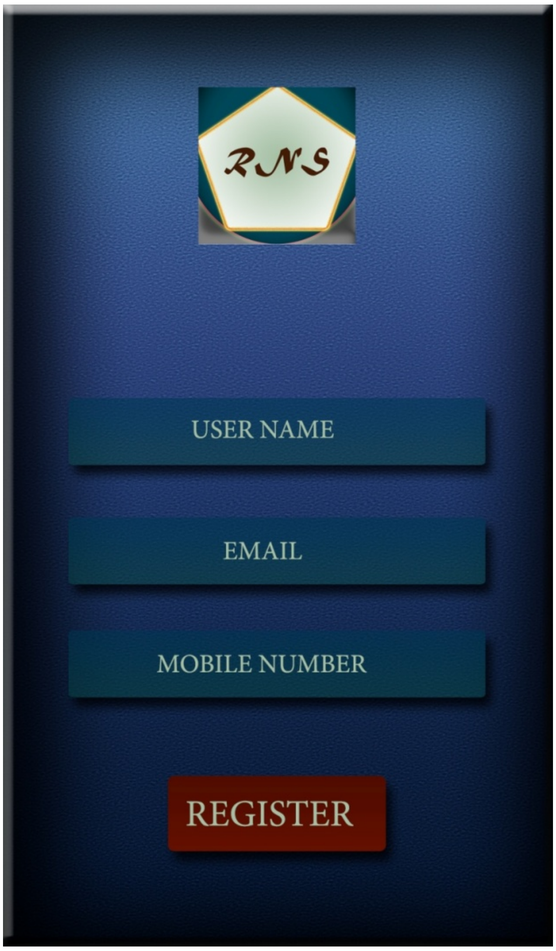


Figure 15 - Registration Screen

Help Screen

This is a help screen in which there are two options available. First, one can click on the user manual to get access to how this system works, stepwise for a person who is new to the system basic level technical terms are available which will guide you about the working of the system. Second, RNS description is an option that must be clicked or selected once a person wants to get know how of the RNS system in general and its components with working. Specifically RNS is described here.

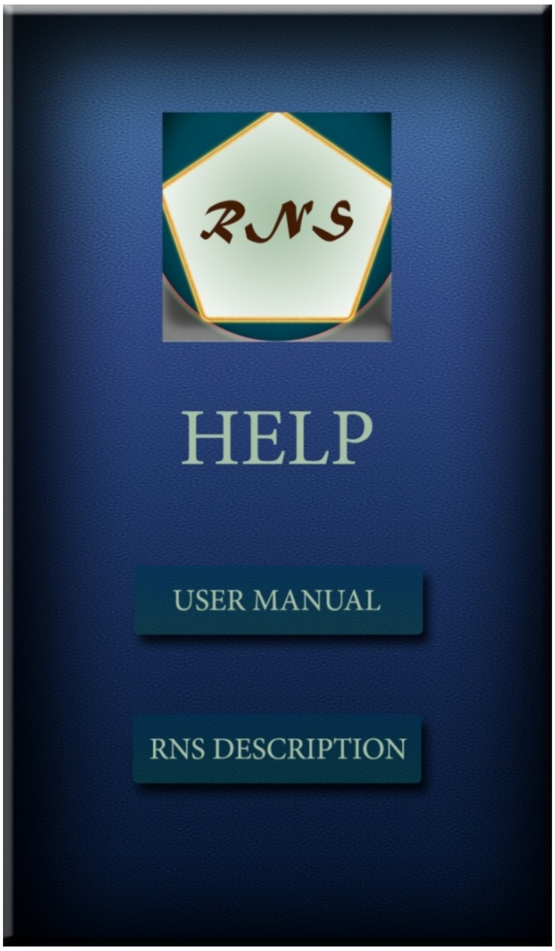


Figure 16 - Help Screen

Map Screen

This screen will appear once we have registered into the system with respective credentials. It will show us the precise coordinates in the form of pointer on the map which furthermore helps in finding the precise location, these coordinates are collected by the tedious process of kalman filtering.

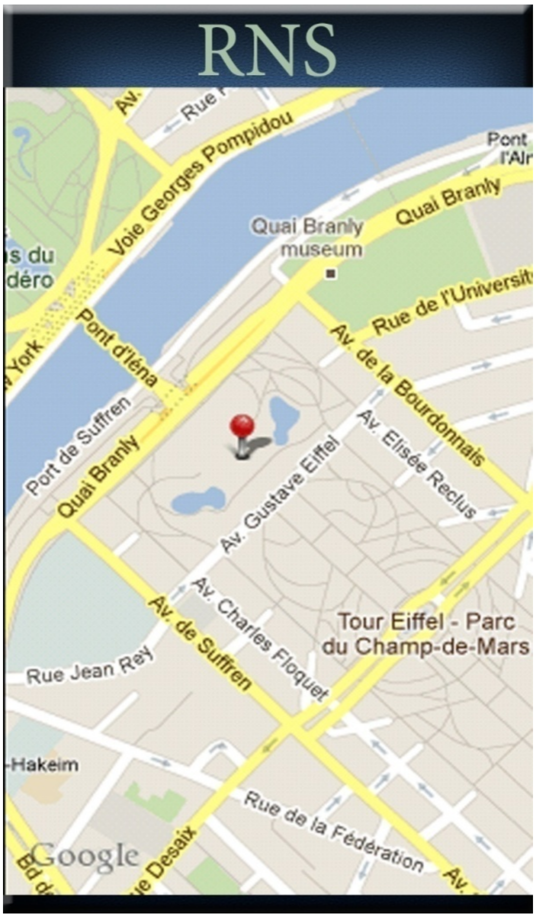


Figure 157 - Map Screen

5.4 Summary

Implementation details of RNS are discussed in this chapter. Different functionalities and strategies to develop the system have also been pondered upon.A brief introduction to different tools and technologies employed is also given.

CHAPTER 6

Testing

6. Testing

6.1 Introduction

The purpose of this chapter is to elicit all material that is essential to plan and control the test efforts for the development of this project. It specifies the test plan for RNS application during the development phase and provides rationale behind necessity of these tests. This document provides an overview of the tests that were implemented, the items that were targeted by the tests, along with the testing approach that was deployed. This testing is being done according to the elicited requirements in SRS Document for RNS.

6.2 Test Items

- Application Startup testing
- Login Feature Testing
- Google Map API Testing
- Login as Administrator
- Registration
- Accessing Map

6.3 Tested Features

- Login
- Precise Location
- Database update

6.4 Features not to be tested

Following tests are not done on the system since they are out of the scope of this project.

- Security Testing

6.5 Approach

The framework is working in modules so the testing stage was started by testing every module independently for example unit testing, and afterward well ordered coordinating modules to test them with one another for example coordination testing, trailed by the testing of the total application in general.

Item Data update Criteria

- Items will pass the test if the actual output of each of the test case is same as the desired output of the system.
- Any transfer of data between any modules is updated in the database.

6.6 Suspension Criteria and Resumption Requirements

Suspension Criteria

- The build contains many serious defects which seriously limit testing progress.
- Software/ Hardware problem.
- Assigned resources are not available when needed to be tested.

Resumption Requirements

- Resumption will only occur when the problems that caused the suspension have been resolved.

6.7 Testing Tasks

- Development of test cases
- Execution of tests based on the developed test cases
- Report defects from the executed test cases, if any
- Provision of complete test report
- Incorporate changes later in the stage of the project development

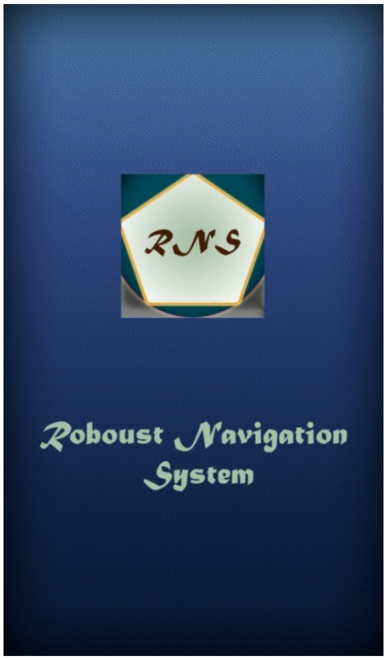
6.8 Staffing and Training Needs

- The interface for users is self-explanatory, and doesn't require any special training.

6.9 Test Cases

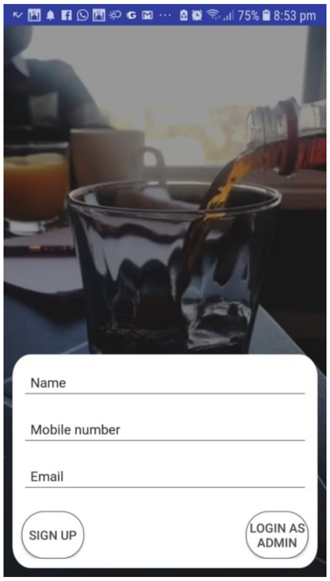
Test Case Name	Application Startup Testing
Test Case ID	1
Description	This feature sends the user to the login screen of the Android application when he/she opens the app through the Cell phone
Testing Technique Used	Black Box Testing-White Box Testing
Preconditions	The Cell phone should be turned on and well integrated.
Input Values	-
Valid Inputs	The specified URL address
Steps	1. Open Menu on Cell phone. 2. Touch the App.

Expected Output	The user will be sent to the login screen of RNS.
Actual Output	Successful opening of the application
Status	PASS

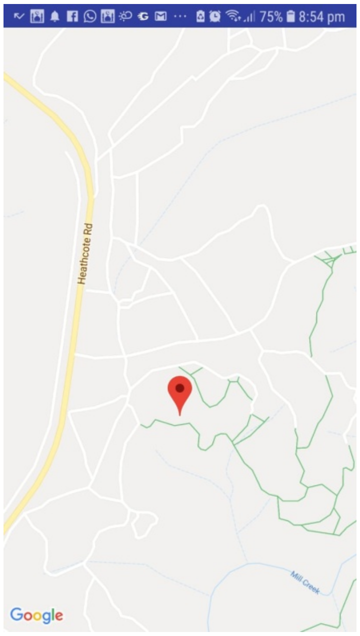


Test Case Name	Login Feature Testing
Test Case ID	2
Description	This feature asks the user to enter his/her credentials for login. This test case is aimed to check that feature works according to user requirement.
Testing Technique Used	Black Box Testing,White Box Testing
Preconditions	Application is running and connected to database. User has opened the login Screen.

Input Values	1. Username 2. Password
Valid Inputs	1. Valid and authorized username 2. Valid and authorized password
Steps	1. Enter username. 2. Enter password. 3. Click “Login”.
Expected Output	The user credentials will be passed to the server for verification. The valid users will be directed to their dashboard after login.
Actual Output	Successful login. User is directed to the profile dashboard.
Status	PASS

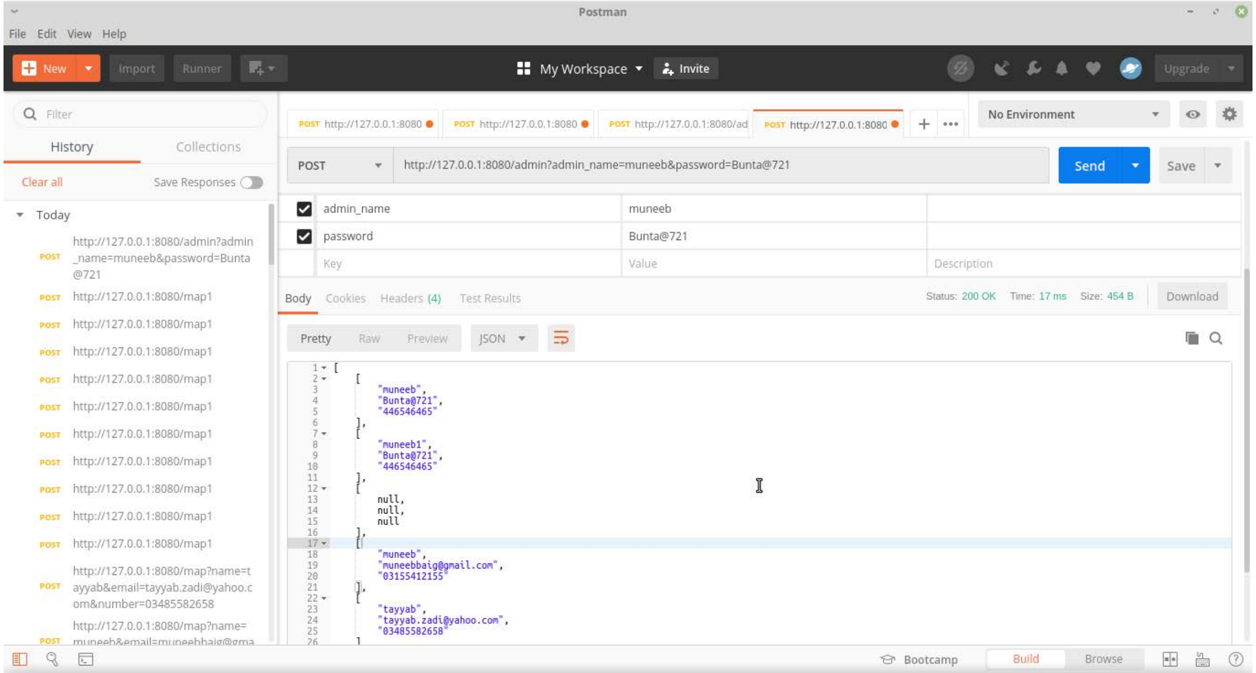


Test Case Name	Google Map API Testing
Test Case ID	3
Description	This test case checks the location with the help of a pointer on the map
Testing Technique Used	Black Box Testing,White Box Testing
Preconditions	Staff is logged into the account, and is currently viewing the dashboard.
Input Values	Latitude and Longitude
Valid Inputs	Alphanumeric values for the fields stated above
Steps	On the dashboard enter the location button .
Expected Output	The expected output is the precise location
Actual Output	The output is the exact location.
Status	PASS



Test Case Name	Login as administrator (successful)
Test Case ID	4
Description	This test case checks that what happens when a person login as administrator and how database is shown,
Testing Technique Used	Black Box Testing, White Box Testing
Preconditions	Clicking on the login as admin and giving right credentials
Input Values	1. Admin name 2. Password
Valid Inputs	Alphanumeric characters
Steps	1. Click on the login as administrator it will ask for credentials 2. It will require credentials

	3. Enter the correct credentials 4. Click login again
Expected Output	Data of users is shown as in database.
Actual Output	Basically the database is shown on the screen.
Status	PASS



POST http://127.0.0.1:8080/admin?admin_name=muneeb&password=Bunta@721
POST http://127.0.0.1:8080/map1
POST http://127.0.0.1:8080/map1
POST http://127.0.0.1:8080/map1
POST http://127.0.0.1:8080/map1
POST http://127.0.0.1:8080/map1
POST http://127.0.0.1:8080/map1
POST http://127.0.0.1:8080/map1
POST http://127.0.0.1:8080/map1
POST http://127.0.0.1:8080/map1
POST http://127.0.0.1:8080/map?name=tayyab&email=tayyab.zadi@yahoo.com&number=03485582658
POST http://127.0.0.1:8080/map?name=muneeb&email=muneebbaig@gma

POST http://127.0.0.1:8080/admin?admin_name=muneeb&password=Bunta@721

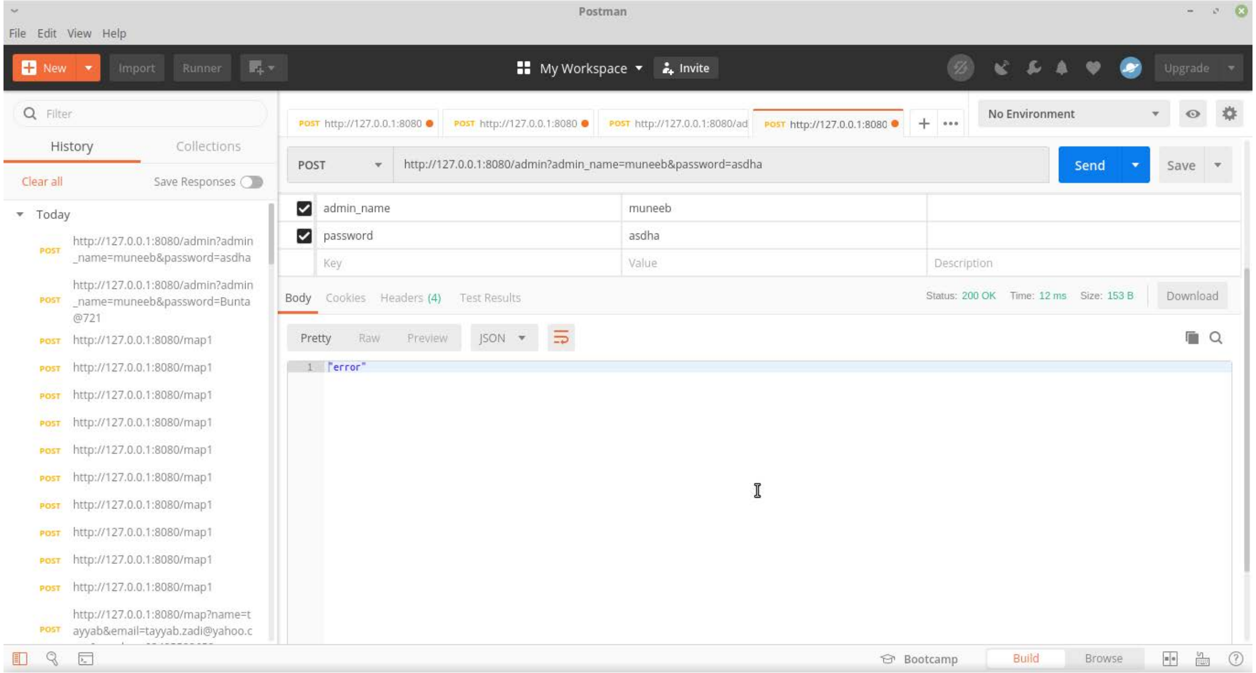
☑ admin_name	muneeb	
☑ password	Bunta@721	
Key	Value	Description

Status: 200 OK Time: 17 ms Size: 454 B

```
[
    [
        "muneeb",
        "Bunta@721",
        "446546465"
    ],
    [
        "muneeb1",
        "Bunta@721",
        "446546465"
    ],
    [
        null,
        null,
        null
    ],
    [
        "muneeb",
        "muneebbaig@gmail.com",
        "03155412155"
    ],
    [
        "tayyab",
        "tayyab.zadi@yahoo.com",
        "03485582658"
    ]
```

Test Case Name	Login as administrator
Test Case ID	5
Description	This test case checks that what happens if a person enters wrong credentials
Testing Technique Used	Black Box Testing, White Box Testing
Preconditions	Clicking on the login as admin and giving wrong credentials

Input Values	1.Admin 2.Password
Valid Inputs	Alphanumeric characters
Steps	1. Click on the login as administrator it will ask for credentials 2. It will require credentials 3. Enter the wrong credentials
Expected Output	That database of the people or users will be shown
Actual Output	This step will produce output as “error”.
Status	PASS



History:

- POST http://127.0.0.1:8080/admin?admin_name=muneeb&password=asdha
- POST http://127.0.0.1:8080/admin?admin_name=muneeb&password=Bunta@721
- POST http://127.0.0.1:8080/map1
- POST http://127.0.0.1:8080/map1
- POST http://127.0.0.1:8080/map1
- POST http://127.0.0.1:8080/map1
- POST http://127.0.0.1:8080/map1
- POST http://127.0.0.1:8080/map1
- POST http://127.0.0.1:8080/map1
- POST http://127.0.0.1:8080/map1
- POST http://127.0.0.1:8080/map1
- POST http://127.0.0.1:8080/map1
- POST http://127.0.0.1:8080/map?name=tayyab&email=tayyab.zadi@yahoo.c

Tabs: POST http://127.0.0.1:8080 · POST http://127.0.0.1:8080 · POST http://127.0.0.1:8080/ad · POST http://127.0.0.1:8080

POST http://127.0.0.1:8080/admin?admin_name=muneeb&password=asdha

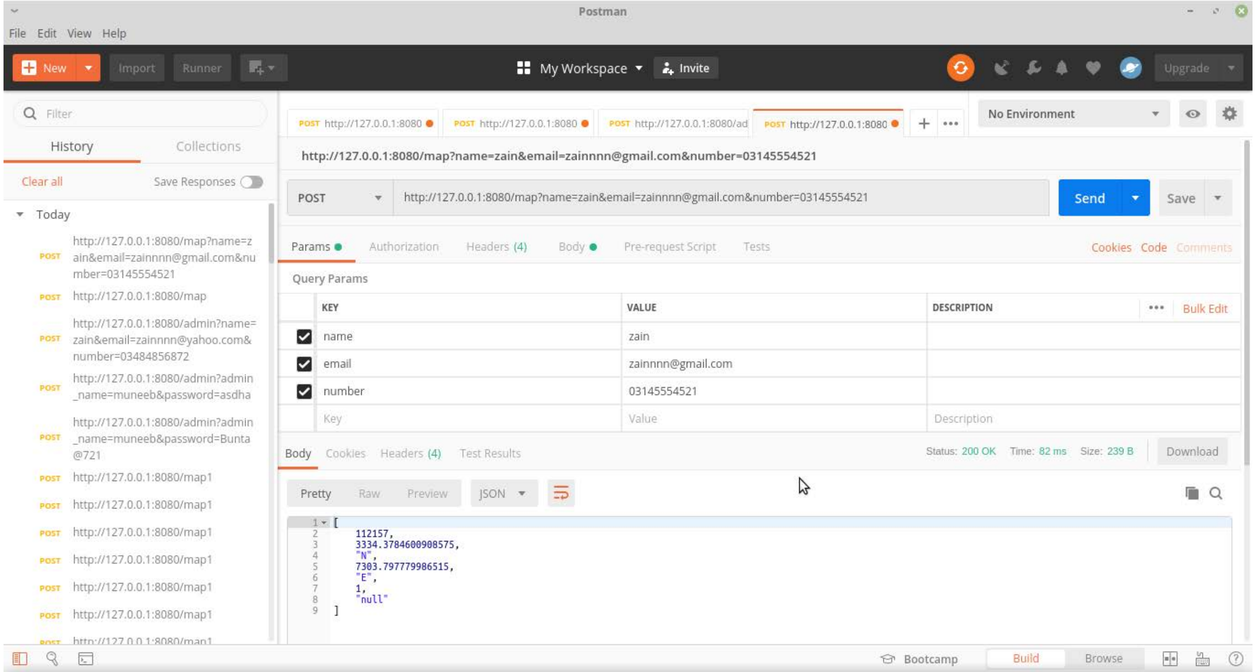
	Key	Value	Description
☑	admin_name	muneeb	
☑	password	asdha	

Status: 200 OK Time: 12 ms Size: 153 B

```
"error"
```

Test Case Name	Registration

Test Case ID	6
Description	This test checks that how the database of the people is updated once the admin name, number and email are added.
Testing Technique Used	Black Box, White Box Testing
Preconditions	Database must be in place to be updated once the specifics are entered.
Input Values	Admin name , number and email ID
Valid Inputs	Alphanumeric Characters
Steps	Click on the login which basically registers the user who have entered his/her specifics
Expected Output	New user will be updated in the database of the system.
Actual Output	Database will be updated accordingly the amount of users registered will be shown.
Status	PASS



POST http://127.0.0.1:8080/map?name=zain&email=zainnnn@gmail.com&number=03145554521

POST http://127.0.0.1:8080/map

POST http://127.0.0.1:8080/admin?name=zain&email=zainnnn@yahoo.com&number=03484856872

POST http://127.0.0.1:8080/admin?admin_name=muneeb&password=asdha

POST http://127.0.0.1:8080/admin?admin_name=muneeb&password=Bunta@721

POST http://127.0.0.1:8080/map1

POST http://127.0.0.1:8080/map1

POST http://127.0.0.1:8080/map1

POST http://127.0.0.1:8080/map1

POST http://127.0.0.1:8080/map1

POST http://127.0.0.1:8080/map1

POST http://127.0.0.1:8080/map1

http://127.0.0.1:8080/map?name=zain&email=zainnnn@gmail.com&number=03145554521

POST http://127.0.0.1:8080/map?name=zain&email=zainnnn@gmail.com&number=03145554521

Query Params

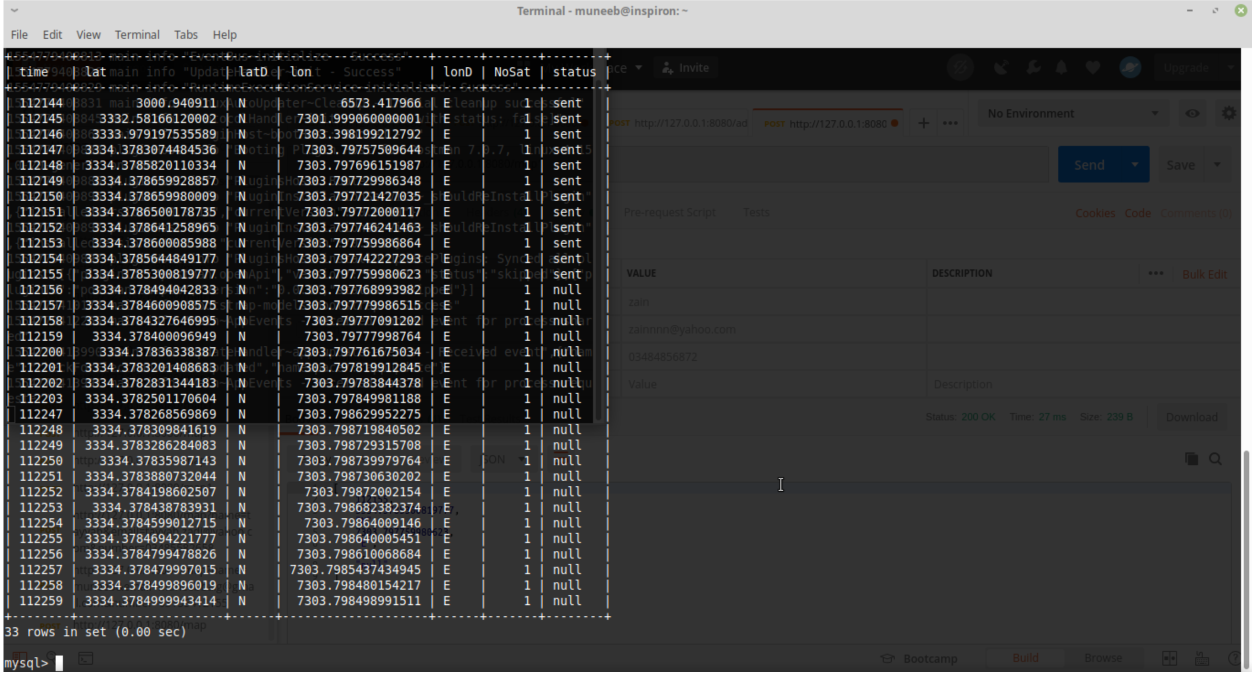
	KEY	VALUE	DESCRIPTION
☑	name	zain	
☑	email	zainnnn@gmail.com	
☑	number	03145554521	
	Key	Value	Description

Status: 200 OK Time: 82 ms Size: 239 B

```
[
    112157,
    3334.3784600908575,
    "N",
    7303.797779986515,
    "E",
    1,
    "null"
]
```

Test Case Name	Accessing map
Test Case ID	7
Description	This test case checks that what happens and how location is retrieved once map is called
Testing Technique Used	Black Box ,White Box Testing
Preconditions	It must be registered
Input Values	It must be registered as everything is done at backend
Valid Inputs	Decimal and discrete numbers as well
Steps	Just as a successful registration is required and integration of system with the software.
Expected Output	The values of Lat and Long is updated and changed with respect to location and time

Actual Output	The values of location are updated subsequently on constant basis.
Status	PASS



```
+--------+--------------------+------+--------------------+------+-------+--------+
| time   | lat                | latD | lon                | lonD | NoSat | status |
+--------+--------------------+------+--------------------+------+-------+--------+
| 112144 |        3000.940911 | N    |        6573.417966 | E    |     1 | sent   |
| 112145 |   3332.58166120002 | N    |  7301.999060000001 | E    |     1 | sent   |
| 112146 |  3333.979197535589 | N    |  7303.398199212792 | E    |     1 | sent   |
| 112147 | 3334.3783074484536 | N    |   7303.79757509644 | E    |     1 | sent   |
| 112148 | 3334.3785820110334 | N    |  7303.797696151987 | E    |     1 | sent   |
| 112149 |  3334.378659928857 | N    |  7303.797729986348 | E    |     1 | sent   |
| 112150 |  3334.378659980009 | N    |  7303.797721427035 | E    |     1 | sent   |
| 112151 | 3334.3786500178735 | N    |   7303.79772000117 | E    |     1 | sent   |
| 112152 |  3334.378641258965 | N    |  7303.797746241463 | E    |     1 | sent   |
| 112153 |  3334.378600085988 | N    |  7303.797759986864 | E    |     1 | sent   |
| 112154 | 3334.3785644849177 | N    |  7303.797742227293 | E    |     1 | sent   |
| 112155 | 3334.3785300819777 | N    |  7303.797759980623 | E    |     1 | sent   |
| 112156 |  3334.378494042833 | N    |  7303.797768993982 | E    |     1 | null   |
| 112157 | 3334.3784600908575 | N    |  7303.797779986515 | E    |     1 | null   |
| 112158 | 3334.3784327646995 | N    |   7303.79777091202 | E    |     1 | null   |
| 112159 |  3334.378400096949 | N    |   7303.79777998764 | E    |     1 | null   |
| 112200 |   3334.37836338387 | N    |  7303.797761675034 | E    |     1 | null   |
| 112201 | 3334.3783201408683 | N    |  7303.797819912845 | E    |     1 | null   |
| 112202 | 3334.3782831344183 | N    |   7303.79783844378 | E    |     1 | null   |
| 112203 | 3334.3782501170604 | N    |  7303.797849981188 | E    |     1 | null   |
| 112247 |  3334.378268569869 | N    |  7303.798629952275 | E    |     1 | null   |
| 112248 |  3334.378309841619 | N    |  7303.798719840502 | E    |     1 | null   |
| 112249 | 3334.3783286284083 | N    |  7303.798729315708 | E    |     1 | null   |
| 112250 |   3334.37835987143 | N    |  7303.798739979764 | E    |     1 | null   |
| 112251 | 3334.3783880732044 | N    |  7303.798730630202 | E    |     1 | null   |
| 112252 | 3334.3784198602507 | N    |   7303.79872002154 | E    |     1 | null   |
| 112253 |  3334.378438783931 | N    |  7303.798682382374 | E    |     1 | null   |
| 112254 | 3334.3784599012715 | N    |   7303.79864009146 | E    |     1 | null   |
| 112255 | 3334.3784694221777 | N    |  7303.798640005451 | E    |     1 | null   |
| 112256 | 3334.3784799478826 | N    |  7303.798610068684 | E    |     1 | null   |
| 112257 |  3334.378479997015 | N    | 7303.7985437434945 | E    |     1 | null   |
| 112258 |  3334.378499896019 | N    |  7303.798480154217 | E    |     1 | null   |
| 112259 | 3334.3784999943414 | N    |  7303.798498991511 | E    |     1 | null   |
+--------+--------------------+------+--------------------+------+-------+--------+
33 rows in set (0.00 sec)

mysql>
```

Table 0-1- Logout Testing

6.10 Summary

From the extensive black box testing, it can be concluded that the system is working fine and in accordance with the functional requirements stated in the SRS document. The system can be further improved and enhanced during maintenance and upgrade phase. A detailed white box testing can also highlight the errors and bugs, if present, in the code of the application.

CHAPTER 7

Conclusion and Future Work

7. Conclusion& Future Work

7.1 Conclusion

Various applications exist that perform the functions of finding the location, RNS has been developed to provide precise and accurate location using Kalman Filtering. It will gather the complete x-y complete coordinates through raspberry pie and after the thorough process of Kalman filtering it will produce precise location in places even like tunnels and other obstructions.

In world, there are many systems that provide us location e.g GPS but these softwares used to give us not so accurate location which was deviated from the actual position but this system of RNS will give us a little more preciseness, modifiability and portability to find the precise location. Concluding it will provide us the most nearest location to the actual position.

7.2 Design Decisions & Tradeoffs

This section highlights various design decisions and the ideas behind those. It enables the reader to understand the important crux of the design that is being used while excavating a bit more about the motivations behind those decisions.

7.3 Future Work

As specified in the external scope of the product, in the coming years, this software can be modified and adjusted for implementation for Post Graduate and PhD students, as well as implementation in other departments like MCS EE Dept with slight changes. A survey can be conducted among the users of the application, and considering their reviews, changes can be made in the application on annual basis.

Bibliography

[1]Kalman Filtering and Neural Networks,by Simon S 2001.

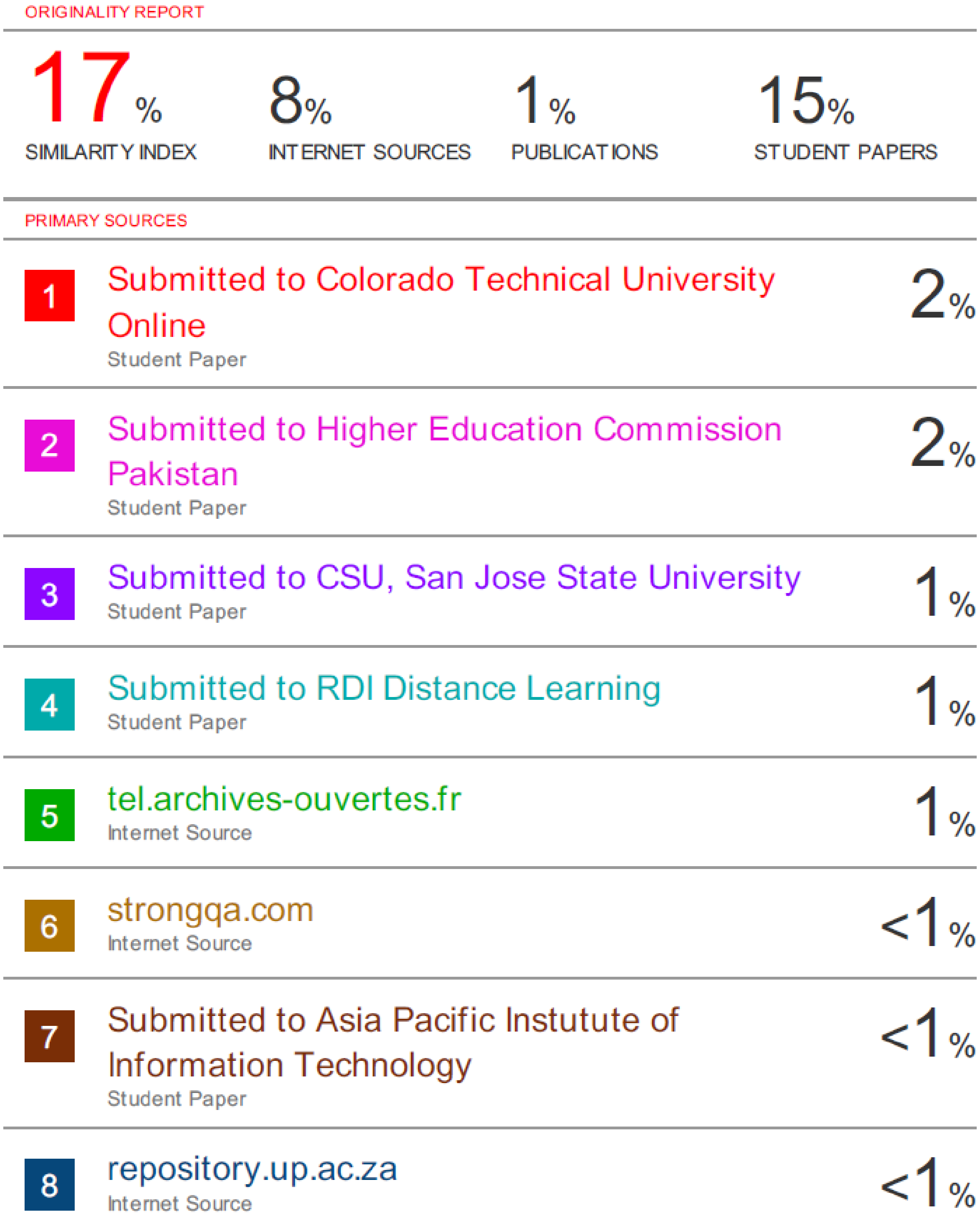
[2]Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches,by Dan Simon,2006

[3]Fluent Python: Clear, Concise, and Effective Programming,by Luciano Ramalho,2015

[4]Global Positioning System: Theory and Practice by B.Hoffman ,1992

[5]Java Programming for Android Developers For Dummies by Barry A Burd 2013

Plagiarism Report

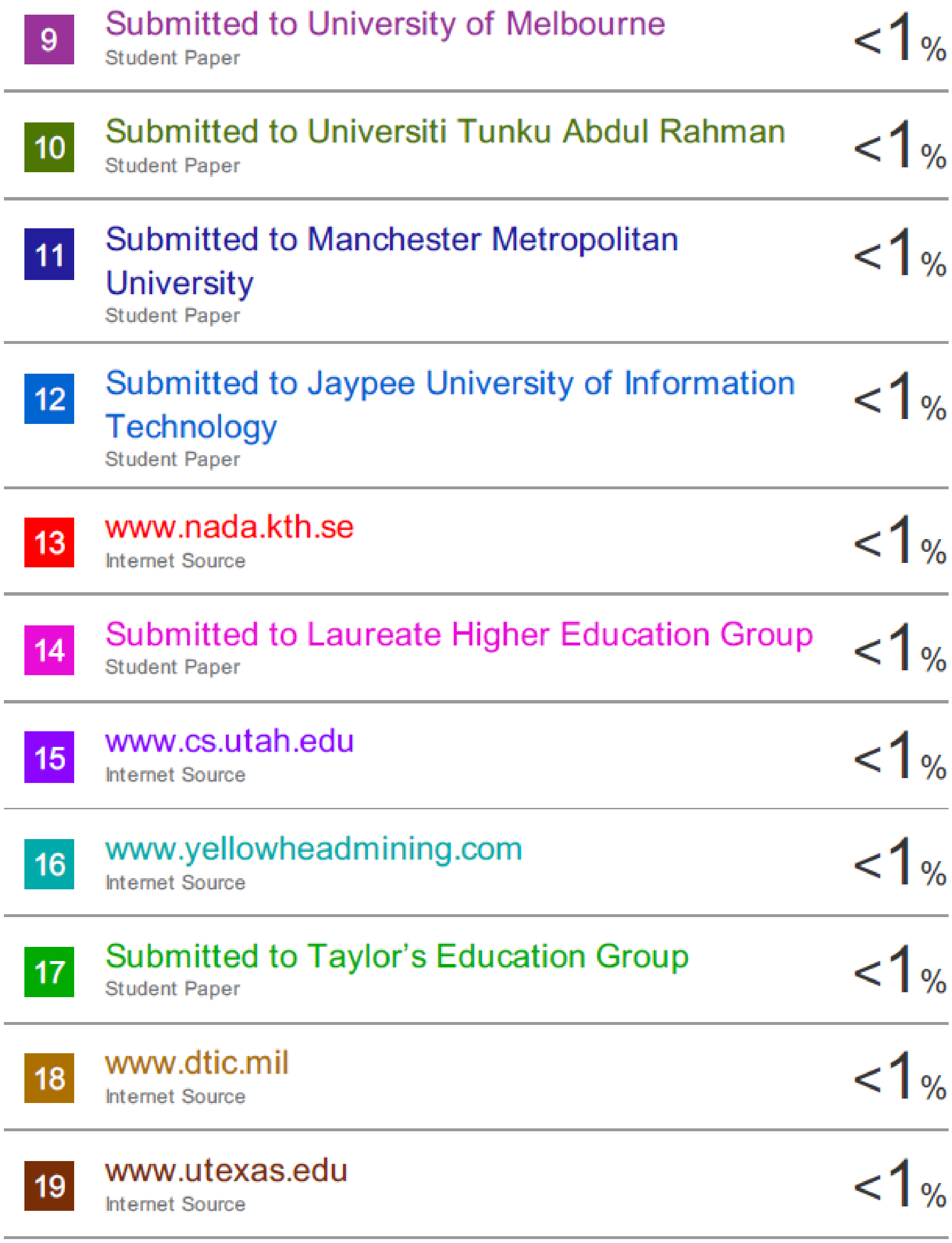


ORIGINALITY REPORT

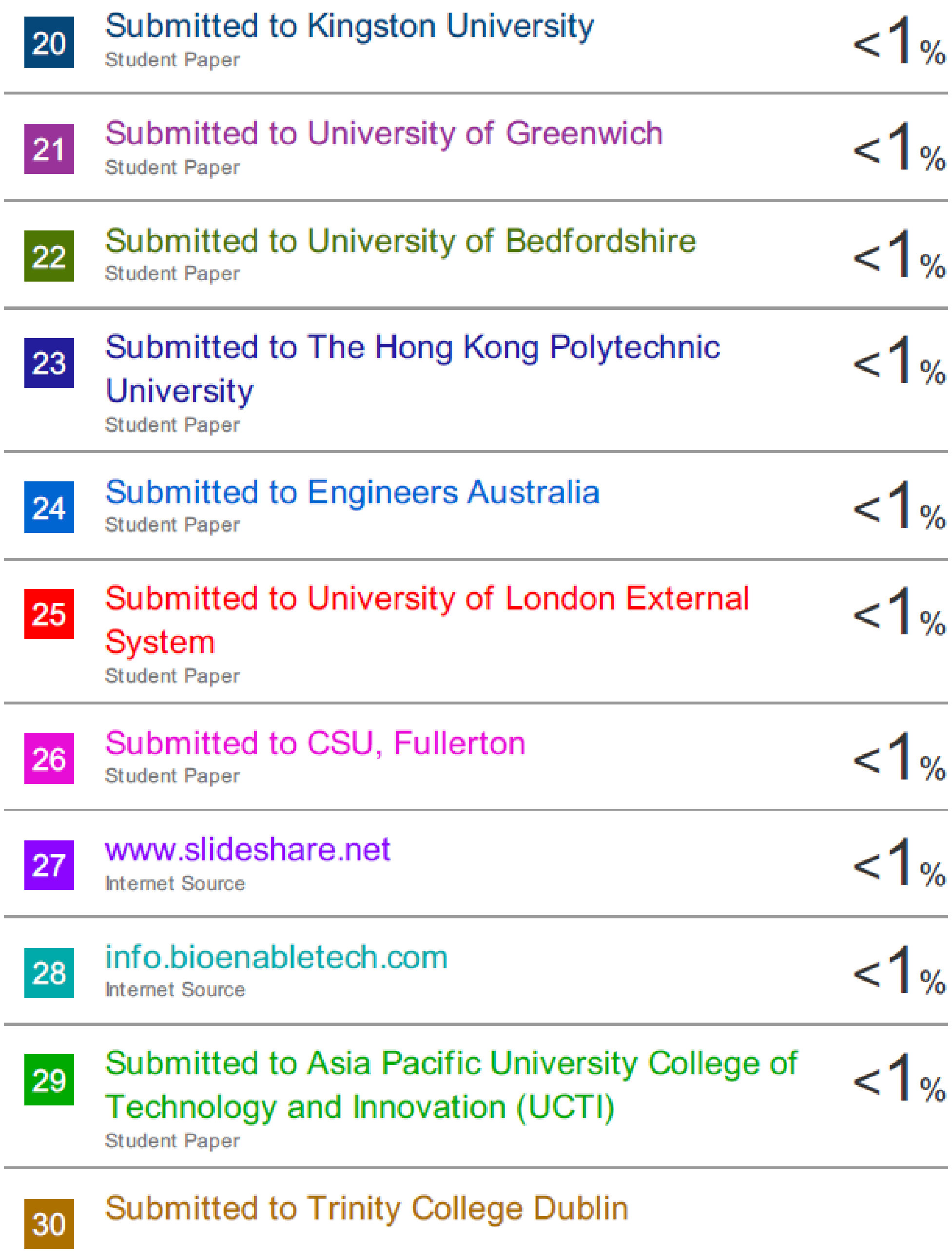
17%	8%	1%	15%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

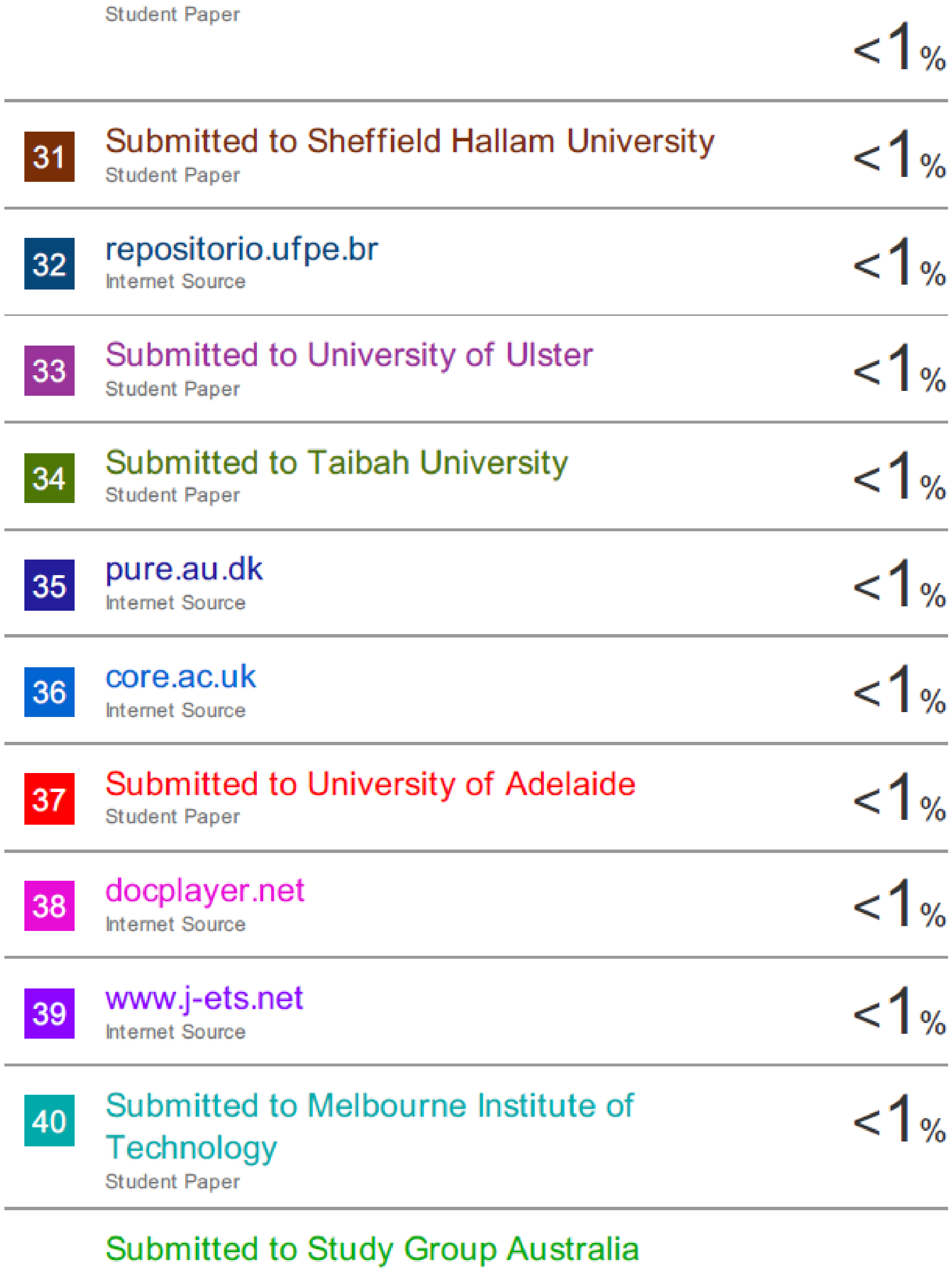
1	Submitted to Colorado Technical University Online Student Paper	2%
2	Submitted to Higher Education Commission Pakistan Student Paper	2%
3	Submitted to CSU, San Jose State University Student Paper	1%
4	Submitted to RDI Distance Learning Student Paper	1%
5	tel.archives-ouvertes.fr Internet Source	1%
6	strongqa.com Internet Source	<1%
7	Submitted to Asia Pacific Instutute of Information Technology Student Paper	<1%
8	repository.up.ac.za Internet Source	<1%



9	Submitted to University of Melbourne Student Paper	<1%
10	Submitted to Universiti Tunku Abdul Rahman Student Paper	<1%
11	Submitted to Manchester Metropolitan University Student Paper	<1%
12	Submitted to Jaypee University of Information Technology Student Paper	<1%
13	www.nada.kth.se Internet Source	<1%
14	Submitted to Laureate Higher Education Group Student Paper	<1%
15	www.cs.utah.edu Internet Source	<1%
16	www.yellowheadmining.com Internet Source	<1%
17	Submitted to Taylor's Education Group Student Paper	<1%
18	www.dtic.mil Internet Source	<1%
19	www.utexas.edu Internet Source	<1%



20	Submitted to Kingston University Student Paper	<1%
21	Submitted to University of Greenwich Student Paper	<1%
22	Submitted to University of Bedfordshire Student Paper	<1%
23	Submitted to The Hong Kong Polytechnic University Student Paper	<1%
24	Submitted to Engineers Australia Student Paper	<1%
25	Submitted to University of London External System Student Paper	<1%
26	Submitted to CSU, Fullerton Student Paper	<1%
27	www.slideshare.net Internet Source	<1%
28	info.bioenabletech.com Internet Source	<1%
29	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	<1%
30	Submitted to Trinity College Dublin	



	Student Paper	<1%
31	Submitted to Sheffield Hallam University Student Paper	<1%
32	repositorio.ufpe.br Internet Source	<1%
33	Submitted to University of Ulster Student Paper	<1%
34	Submitted to Taibah University Student Paper	<1%
35	pure.au.dk Internet Source	<1%
36	core.ac.uk Internet Source	<1%
37	Submitted to University of Adelaide Student Paper	<1%
38	docplayer.net Internet Source	<1%
39	www.j-ets.net Internet Source	<1%
40	Submitted to Melbourne Institute of Technology Student Paper	<1%
	Submitted to Study Group Australia	