

Project Report

ROUTE ALERT

Contents

1. **Definition of the Problem** ........................................................................................................ 3
2. **Requirement Analysis**........................................................................................ 3
3. **Design Phase** .......................................................................................................... 11
4. **Architecture of Application** .......................................................................................................... 18
5. **Evaluation/Testing** .......................................................................................................... 24
6. **Project Tracking and Monitoring Activities**..................................................................................... 25
7. **Documentation Section** .......................................................................................................... 26
8. **Source Code** .......................................................................................................... 27
9. **User Guide** ............................................................................................................. 27

# 1. Definition of the Problem

***Problem definition***

|  |
| --- |
| Route Alert is Mobile Application and purpose of this Application document is to provide a detail information of its Feature, interface and constraints. Need of the system in urban area is to provide solution in traffic control department, Ambulance and fire brigade who cannot afford traffic problem in urban area. |

# 2. Requirement Analysis

## List of inputs to the system

1. Login

2-Notification

3-Submit Complaint

3-Emergency Help

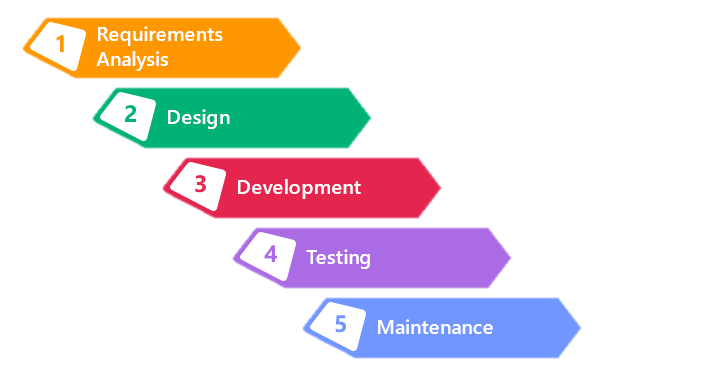
4-Feedback

## ➢ List of outputs expected from the system

1. Registration
2. Clear Area
3. Complaint
4. Emergency Help
5. Notification
6. Feedback

## ➢ Overview of processes involved in the system

The remainder of this document is two chapters, the first providing a full description of the project for the Routine Alert of the Traffic. It lists all the functions performed by the system. The final chapter concerns details of each of the system functions and actions in full for the software developers’ assistance.



## ➢ Planning: -

1. **Evaluate Process**

What is Working.

Evaluate team planning experienced? Creative? Analytics?

Evaluate Tools.

1. **Use a tool**

Setup Project

Breakdown work into tasks

Identify Dependencies

Estimate time

Assign Resources

1. **Plan Planning**

Allocate Time

Include Time

Cast a vision

Identify Roles

Set Ground rules

Simplify

## ➢ Analysis:-

The goal of analysis is to determine where the problem is, in an attempt to fix the system. This step involves breaking down the system in different pieces to analyze the situation, analyzing project goals, breaking down what needs to be created, and attempting to engage users so that definite requirements can be defined.

## ➢ Design:-

In systems design, the design functions and operations are described in detail, including screen layouts, business rules, process diagrams, and other documentation. The output of this stage will describe the new system as a collection of modules or subsystems.

The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts.

Design elements describe the desired system features in detail, and they generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo-code, and a complete entity-relationship diagram with a full data dictionary. These design elements are intended to describe the system in sufficient detail, such that skilled developers and engineers may develop and deliver the system with minimal additional input design.

## ➢ Development: -

Developers must follow the coding guidelines define by their organization and programming tools like compilers, interpreter and debuggers etc.

## ➢ Testing: -

Implementation Phase, during which the system is actually built (or purchased, in the case of packaged software design.)  The first step in implementation is systems construction, during which the system is built and tested to ensure it performs as designed.  Deliverables during this step include Programs, the Test Plan, User Documentation, and the Tested System.  Conversion is the process by which the old system is turned off and the new on turned on, during which the project team creates a Conversion plan, a Change Management Plan, a Support Plan, and a Project Assessment.  At the end of Implementation, the final system is delivered to the project sponsor and approval committee.

## ➢ Maintenance: -

The deployment of the system includes changes and enhancements before the decommissioning or sunset of the system. Maintaining the system is an important aspect of SDLC. As key personnel change positions in the organization, new changes will be implemented. There are two approaches to system development: the traditional approach (structured) and object oriented. Information engineering includes the traditional system approach, which is also called the structured analysis and design technique. The object oriented approach views information system as a collection of objects that are integrated with each other to make a full and complete information system.

## ➢ Hardware and software required for implementing the project

➢ **Hardware**

* Quad-core Max 1.40GHz Processor or higher
* 8 GB RAM or above
* Color SVGA
* 120 GB Hard Disk space
* Mouse
* Keyboard
* Android Supporting Smart Phone

➢ **Software**

Technologies to be used:

1. Technology: Android (Java)
2. IDE: Android Studio
3. Android Version 7.0
4. Database: MYSQL

➢ **Additional Feature**

1-Images

2-Notification with Firebase

3-PHP API

4-Data Exchange Jason

## ➢ Customer’s acceptance criteria for the project

|  |  |
| --- | --- |
| **S. No.** | **Customer’s Acceptance Criteria** |
| 1. | Signup credential |
| 2. | Create user login |
| 3. | Login with the created Email ID and Password |
| 4. | View notifications |
| 5. | Submit complaints |
| 6. | Check complaint status |
| 7. | Emergency help |
| 8. | Add feedback details |

## Project Plan

## 1. Project Details

● **Name of the Client:** **Aptech Techwiz.**

● **Date of the Project Plan: 25 March 2019**

**● Project Vision/Objectives:**

This project can be helpful for citizen of the cities. User can use for quickly and easily excess to own cell phone. As more and more people use smart phones, the need for high-quality Android Developers becomes more vital than ever. People in this profession and responsible for developing apps on phones so that the devices possesses functionality.

## ● Scope:

This software system will be a Mobile Application to be used by individuals as well as city traffic control department. This system will be designed to minimize traffic related issues and provide users a platform to request for emergency help as well as submit their grievances regarding traffic.

## ● Our understanding of the client organization

The app will be designed as a set of forms/pages, Navigation, and Fragments with menus representing choice of activities to be performed.

1. Registration
2. Login
3. View Notification
4. Submit Complaints
5. Check Complaint Status
6. Emergency Help
7. Feedback

## ● Project Organization with Responsibilities and Authorities

|  |  |
| --- | --- |
| **Team** **Members** | **Roles** **&** **Responsibilities** |
| 1-Syed Daniyal Ali | Full Stack Developer & Analyzer |
| 2-Muhammad Mohsin | Documentation & Project Reports |
| 3-Saad Khan | Database & API |
| 4-Subhan Ayub | XML Programmer |

## 2. Project Initiation/Requirement Documents

**Registration**: It will allow various individuals across city to get registered with Route Alert by city traffic department based on valid identity proof such as Passport, Voter ID/SSNID, or Driving Licenses. At the time of registration, users’ needs to provide First Name, Last Name, Email ID, Contact Number, Valid Identity Proof, and Address. Also, they need to choose password to get logged in into the

Route Alert system.

Appropriate error-checking must be done on the fields of the form to ensure correct data. For example, email id can be checked to see if it is of the appropriate format.Registration details will be sent to the administrator for validation and approval based on the ID proof and after successfully validation, they will be added into the table as a record. An email will be sent to the user informing him/her about success of validation. A unique registration id would be auto-generated for each user upon successful registration.

**Login**: It will allow successful registered users to get logged in into Route Alert and access various features of system through menus or other options such as View Notifications, Submit Complaints, Check Status of Complaints, Emergency Help, Feedback, and Logout.

**View Notification**: A registered user will be able to view various notifications provided

by city traffic department.

Notifications can be generated datewise by the admin (hint: you can accept them in the form of a text area or similar element) and sent to all registered users. Notification text will include location, date, time, and description. Every unique notification will have an auto-generated id.

**Submit Complaints**: A registered user is allowed to submit complaints. He/she needs to select complaint category such as Natural Calamities, Accident, Traffic Congestion, and so on. Each complaint will have an auto-generated complaint id which will be given to the user. Using this id, they can follow up on the complaint status.

**Check Complaints Status**: Users who submitted complaints earlier will be able to check the status of their complaints, whether in-process or resolved. They can do this based on the complaint id.

**Emergency Help**: It is a feature specially designed to help registered users in terms of emergency such as Accident or medical emergency. ID of the user who submits the help request and the emergency help id along with location and user details such as address and contact number will be stored in the database for each request.

**Feedback**: A registered user can submit feedback about Route Alert in terms of improvements required to the system. User id, auto-generated feedback id, date and description will be added to each feedback entry.

**Logout**: This feature will allow logged-in users to log out of the

## 3. Deliverables

* Problem Definition
* Design specifications
* Diagrams such as flowcharts for various activities, Data Flow Diagrams, and so on
* Database Design

## 4. Project Dependencies

* Android Phone
* Less Version
* Internet Connection
* Updated Browser
* Location
* include Contact Number

## 5. Major Project Milestones

|  |  |
| --- | --- |
| **Milestone** | **Status** |
| Register | Done |
| Login | Done |
| Complaint | Done |
| Notification | Done |
| Emergency | Done |
| FeedBack | Done |

## 6. Quality Plan

● **Review Activities** (Review meeting participants, frequency and so on)

|  |  |  |
| --- | --- | --- |
| **Milestone** | Activity Done by | Check By |
| Register | M. Mohsin | Saadkhan |
| Login | Syed Daniyal Ali | Daniyal |
| Complaint | Saad khan | Subhan |
| Notification | Syed Daniyal Ali | Saadkhan |
| Emergency | Saad khan | Daniyal |
| FeedBack | M.Mohsin | Subhan |

## ● Testing Activities (Final Test)

|  |  |  |
| --- | --- | --- |
| S.No | Features Tested | Remarks |
| 1 | Register | 100% |
| 2 | Login | 100% |
| 3 | Admin login | 100% |
| 4 | Complaint | 100% |
| 5 | Notification | 100% |
| 6 | Emergency | 100% |
| 7 | FeedBack | 100% |

● **Backup and Recovery Strategies** (In case of disk crash, network failures, and so on)

* The project data will be saved at hot site and cold site (Google Drive) for recovery.

|  |
| --- |
| **3. Design Phase** |

## Document Design

|  |  |
| --- | --- |
| **Property** | **Value** |
| Document Theme and Color Scheme | #5cbb89,#3bae7a,#43a047 |
| Form-Background Color | White |
| Title-Font Size | 24dp |
| Title-Font Color | Green |
| Title-Font Style | Normal |
| Title-Alignment | Center |
| Background color of Controls on the form | White |
| Foreground color of Controls on the form | #3bae7a |
| Control Caption-    Font Size | 18dp |
| Control Caption-    Font Color | Black |
| Control Caption-    Font Style | Normal |
| Control caption and controls-Alignment | - |
| Command button-Alignment | - |

## 1. List of forms to be created

|  |  |  |
| --- | --- | --- |
| **Page/Form Name** | **Description** | **Controls on the Page/Form** |
| REGISTRATION | User & admin Login | Textboxes and Buttons |
| LOGIN | User Register | Textboxes and Buttons and image view |
| COMMPLAINT | User can Send a Complaint | Textboxes and Buttons and image view |
| EMERGENCY | User can Send a Emergency Help | Textboxes and Buttons and image view |
| NOTIFICATION | User can read a notification | Textboxes and Buttons |
| FEEDBACK | User can send Feedback | Textboxes and Buttons and image view |  |

**Database design**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name of the table** | **Table Description** | **Number of**  **Fields** | **Primary Key** | **Related tables** | **Foreign Key** |
| **registration** | Manage  users | 8 | **User\_ID** | **1.**compalint  **2**.emergency  **3**. notification  **4**.readnotification | **null** |
| **complaint** | Create  Complaint | **7** | Complaint\_ID | **1.**registration | User\_ID |
| **feedback** | For use  feedback | **4** | Feedback\_ID | **1.**registration | User\_ID |
| **notification** | manage Notification | **5** | Notification\_ID | **1.** readnotification |  |
| **readnotification** | For admin read | **3** |  | **1.**Registration  2.notification | User\_ID  Notifcation\_ID |

# Module Design

**Process Modules:**

**Login:**

User

Login

Homepage

Use Services

**Registration:**

**User**

**Register**

**Submit**

**CTCD**

**Approve**

**Send Confirm**

**Email**

**Allow Use**

**Services**

**Complaint:**

**Emergency Help:**

**User**

**Complaint**

**Submit**

**CTCD View**

**CTCD Give**

•

**Status**

**User View**

**Status**

**User**

**Ask Emergency**

**Help**

**View CTCD**

**Send**

**Notification**

**User View**

**Status**

**Notification:**

**CTCD**

**Send**

**Notification**

**View Users**

1. **Designing Coding Standards**
2. **Programming Standards**

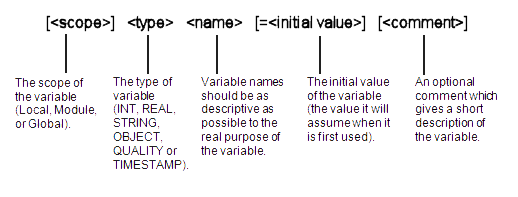
Java Coding Standards. Coding Standards for Components: It is recommended to write components name by its purpose. This approach improves the readability and maintainability of code. Coding Standards for Classes: Usually class name should be noun starting with uppercase letter.

1. **Standards for Code Writing Style**

* Every functions, loops, conditions have there well defined scopes
* Separate classes for every modules
* Variable name, package name and function name should start with initial small letter and should have a capital letter when a changing keyword comes. A package name does not contain any capital letter.

1. **Standards for Declaring Variables**

When declaring variables, you should use consistent formatting. A variable declaration has up to five parts. Each part is separated by at least one tab stop:



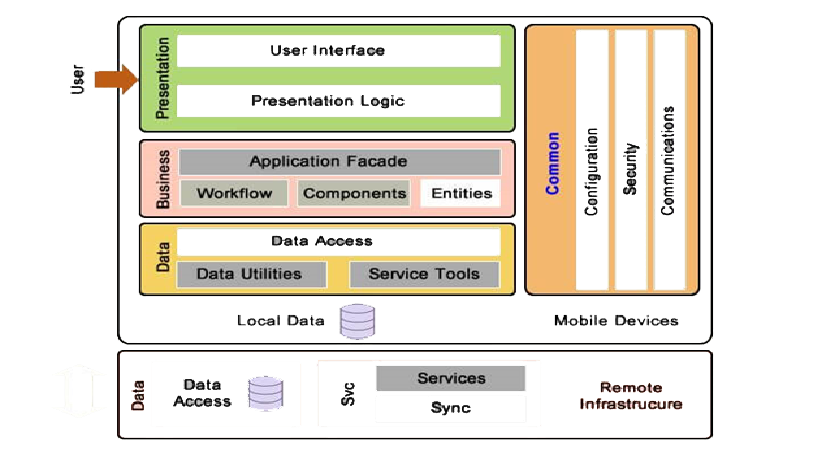
1. **Other Standards**

* **Safe:** It can be used without causing harm.
* **Secure:**It can’t be hacked.
* **Reliable:** It functions as it should, every time.
* **Testable:** It can be tested at the code level.
* **Maintainable:** It can be maintained, even as your codebase grows.
* **Portable:**It works the same in every environment.

# 4. Assigning And Monitoring Task

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project Ref. No.:** | **Project Title:** | **Activity Plan**  **Prepared By:** | | **Date of Preparation of Activity Plan:** | | |
| **Task Sub Division** | **Description** | **Planned**  **Start Date** | **Actual**  **Start Date** | **Actual**  **Days** | **Team**  **Member Names** | **Status** |
| 1. Front End | Create forms | 25 Mar | 25 Mar | 02 | Syed daniyal, & Subhan | Done |
| 2**.** Database Design | Create database & all tables | 25 Mar | 26 Mar | 01 | saadkhan | Done |
| 3.Backend Develop | All backend development | 26 Mar | 28 Mar | 02 | Syed daniyal ali | Done |
| 4. Api | Create All api | 26 Mar | 27 Mar | 01 | saadkhan | Done |
| 5. Graphic Designing and Reports | Designing Themes logos and project Reports | 28 Mar | 28 Mar | 02 | M.Mohsin | Done |

**Architecture Design**

****

**Data Flow Diagram (DFD)**

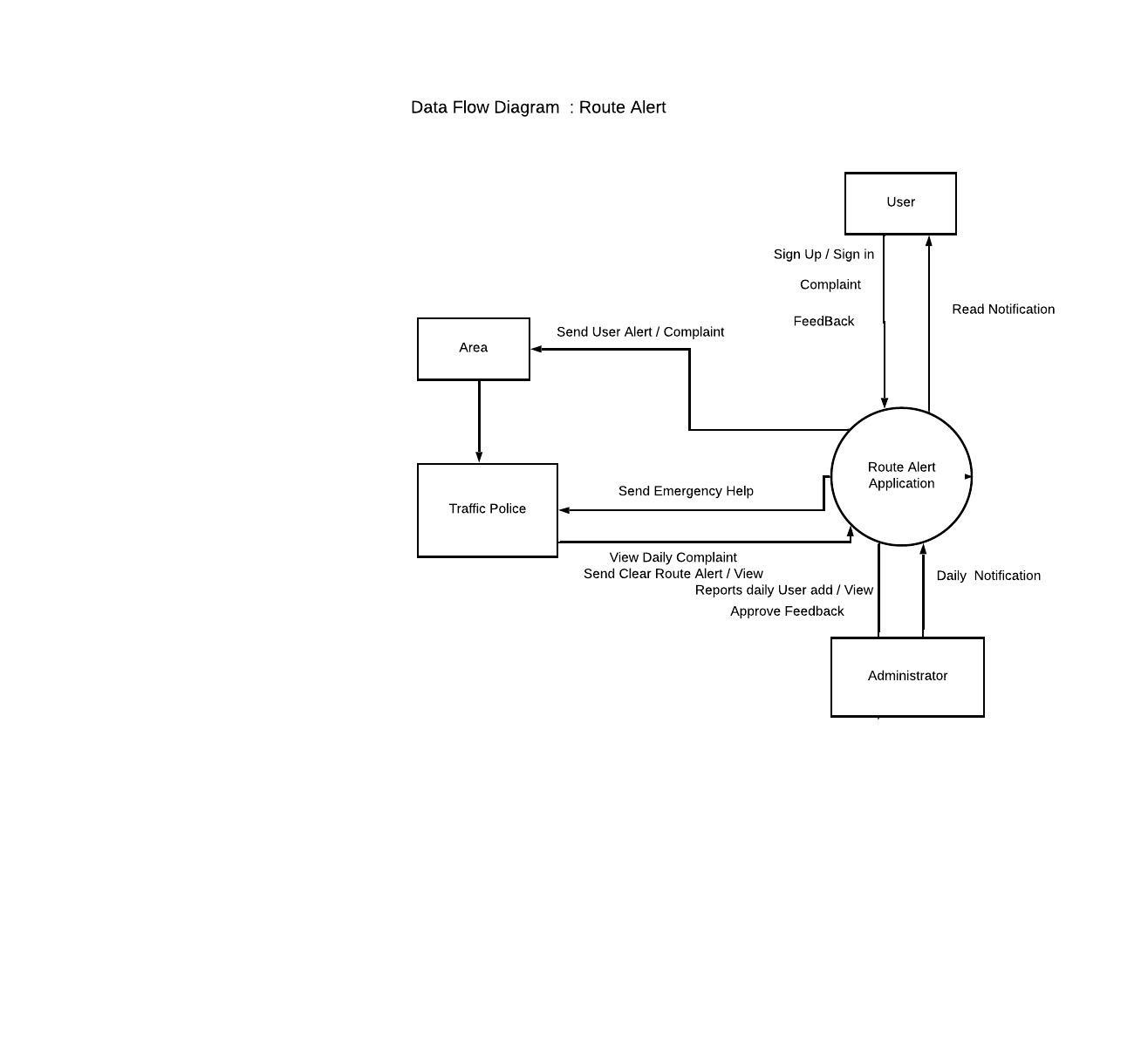
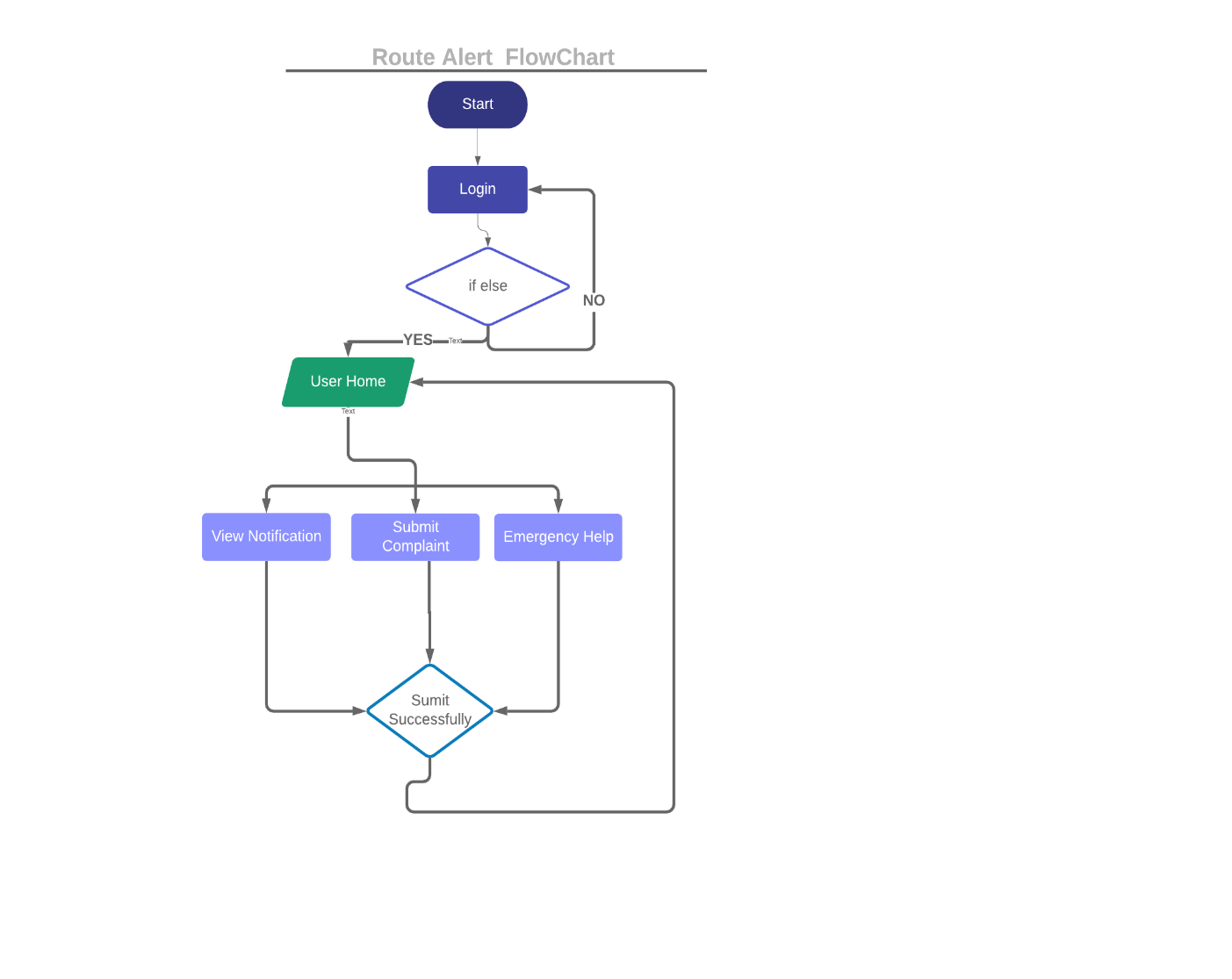
****

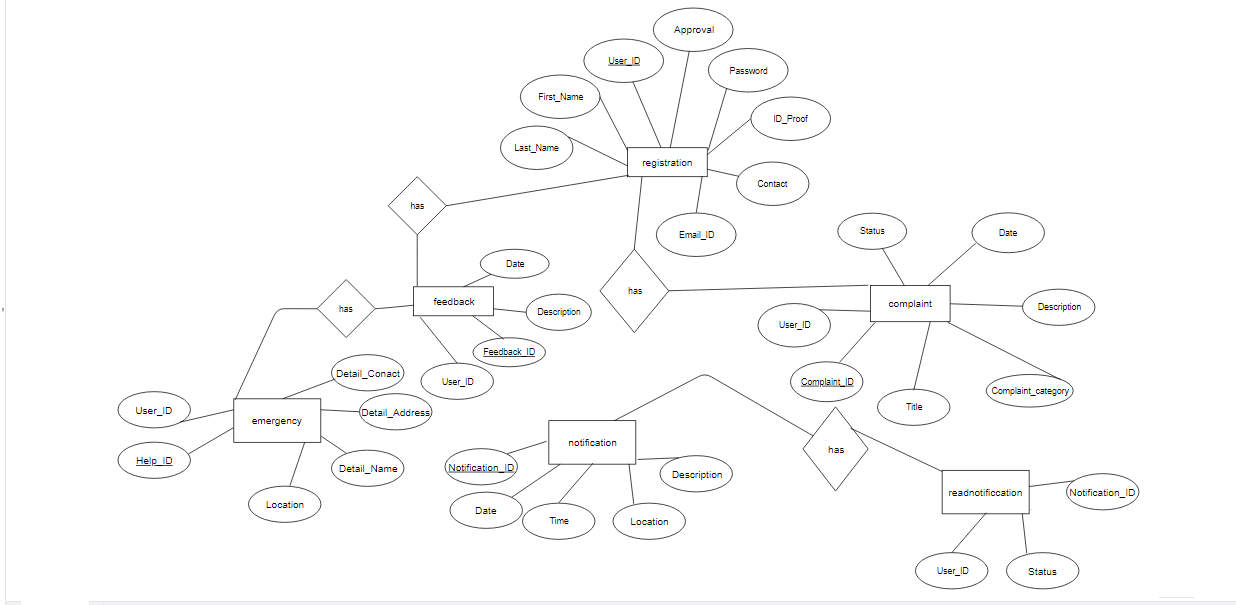
Figure 1.5 displays the second level DFD <for second activity>.

**Flowchart**

****

.

**Entity Relationship (ER) Diagram**

****

**Database Design**

**Table: <Registration>**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Key** | **Description** |
| User\_ID | bigint | PK | Stores the Unique User\_ID of the Registration (auto-generated unique identifier of this table) |
| First\_Name | varchar |  | Stores the First Name of the Registration |
| Last\_Name | varchar |  | Stores the Last Name of the Registration |
| Email\_ID | varchar |  | Stores the Email\_ID Uniq of the Registration |
| Contact | varchar |  | Stores the Contact of the Registration |
| Address | varchar |  | Stores the address of the Registration |
| Password | varchar |  | Stores the Password of the Registration |
| ID\_Proof | varchar |  | Stores the login ID\_Proof of the Registration |

**Table: <Complaint>**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Key** | **Description** |
| User\_ID | bigint | FK | Stores the Unique User\_ID of the Complaint(auto-generated unique identifier of this table) |
| Complaint\_ID | id |  | Stores the complaint\_ID of the Complaint |
| Title | varchar |  | Stores the Tittle of the Complaint |
| Complain\_category | varchar |  | Stores the complaint\_category Uniq of the Complaint |
| Date | date |  | Stores the Date of the Complaint |
| Status | varchar |  | Stores the Status of the Complaint |

**Table: <Emergency>**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Key** | **Description** |
| User\_ID | bigint | FK | Stores the Unique User\_ID of the Emergency (auto-generated unique identifier of this table) |
| Help\_ID | int |  | Stores the Help\_ID of the Emergency |
| Location | varchar |  | Stores the Location of the Emergency |
| Detail\_Name | varchar |  | Stores the Detail\_Name Uniq of the Emergency |
| Detail\_Adress | varchar |  | Stores the Detail\_Adress of the Emergency |
| Detail\_Contact | varchar |  | Stores the Detail\_Contact of the Emergency |

**Table: <Notification>**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Date Type** | **Key** | **Description** |
| Notification\_ID | int | Pk | Stores the Unique User\_ID of the Notification (auto-generated unique identifier of this table) |
| Date | date |  | Stores the Date of the Notification |
| Time | time |  | Stores the Time of the Notification |
| Location | varchar |  | Stores the Location Uniq of the Notification |
| Description | varchar |  | Stores the Description of the Notification |

**Table: <Feedback>**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Key** | **Description** |
| User\_ID | bigint | FK | Stores the Unique User\_ID of the Feedback (auto-generated unique identifier of this table) |
| Feedback\_ID | int |  | Stores the Feedback\_ID of the Feedback |
| Description | varchar |  | Stores the Description of the Feedback |
| Date |  |  | Stores the Date Uniq of the Feedback |

**Table: <Read Notification>**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Key** | **Description** |
| User\_ID | bigint | FK | Stores the Unique User\_ID of the Read Notification (auto-generated unique identifier of this table) |
| Status | varchar |  | Stores the Status of the Read Notification |
| Notification\_ID | int | FK | Stores the Notification\_ID of the Read Notification |

# 5. Evaluation/Testing

This is the most crucial phase where each unit is tested for its functionality. Test data is used to check if the module is able to process it without causing any errors. Test data may be live data

|  |  |  |
| --- | --- | --- |
| **S.No** | **Features Tested** | **Remarks** |
| 1 | Register | Working Properly |
| 2 | Login | Working Properly |
| 3 | Complaint | Working Properly |
| 4 | Notification | Working Properly |
| 5 | Emergency | Working Properly |
| 6 | FeedBack | Working Properly |

# 6. Project Tracking and Monitoring Activities

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date:** | **Project Plan/ Milestone** | **Work**  **Specification** | **Status of the Activity** | **Remarks** | **Responsibility** |
| 25 Mar, 2019 | Database | All tables | Done | Good | Saadkhan |
| 25 Mar, 2019 | Api | All php Api | Done | Good | Saadkhan |
| 26 Mar, 2019 | Front end Designing | Forms | Done | Good | M.Mohsin |
| 27 Mar, 2019 | Front end Designing | forms | Done | Good | Subhan ayub |
| 28 Mar, 2019 | Backend Designing | backend Work | Done | Good | Syed daniyal ali |
| 29 Mar, 2019 | Backend Designing | backend Work | Done | Good | Syed daniyal ali |
| 25 Mar, 2019 | Database | All tables | Done | Good | Saadkhan |
| 25 Mar, 2019 | Api | All php Api | Done | Good | Saadkhan |

# 7. Final Check List

|  |  |  |
| --- | --- | --- |
| **S.No** | **Aspect Tested** | **Suggestion/Remarks** |
| 1. | Have all the modules been properly integrated and are they completely functional? | yes |
| 2. | Does each unit meet its objective and purpose?  Are all the validations happening as specified in Process Design? | yes |
| 3. | Have all the Design and Coding standards been followed and implemented? | yes |
| 4. | Is the GUI design consistent all over? | yes |
| 5. | Are the codes working as per specification? | yes |
| 6. | Does the application’s functionality resolve the client problem, and satisfy his/her needs completely? | yes |
| 7. | Have the hardware and software been correctly chosen? | yes |
| 8. | Additional features and utilities that give value addition to the entire project. | yes |

|  |  |  |
| --- | --- | --- |
|  | **Prepared By (Student)** |  |
| **Date** | **Muhammad Mohsin** |  |

|  |
| --- |
| **8. Source Code** |

|  |
| --- |
| **9. User Guide** |

1. **System Requirements:**

* + Android Supporting Smart Phone
  + Required Android Version Minimum 5.1 Lollipop-Maximum 7.0 Nougat
  + Required Internet

1. **Install and Run Application:**

* Download Application from Google play Store and Install Application.