

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi-590 018



A Project Work on

## “Simple Smart Connected City App”

A Dissertation work submitted in partial fulfillment of the requirement  
for the award of the degree

**Bachelor of Engineering**  
In  
**Information Science & Engineering**

Submitted by

<b>Mr. Jay Narayan Das</b>	<b>1AY12IS034</b>
<b>Mr. Lunagariya Divyesh</b>	<b>1AY12IS045</b>
<b>Ms. Shubha Shree N</b>	<b>1AY12IS081</b>
<b>Mr. Shubham Kumar</b>	<b>1AY12IS082</b>

Under the guidance of  
**Prof. Srinivas .K**  
Assistant Professor



**DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING**  
**ACHARYA INSTITUTE OF TECHNOLOGY**

(AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI. APPROVED BY AICTE, NEW DELHI)

Acharya Dr. Sarvepalli Radhakrishnan Road, Soldevanahalli, Bengaluru-560107

**2015-16**

**DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING**  
**ACHARYA INSTITUTE OF TECHNOLOGY**

(AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI. APPROVED BY AICTE, NEW DELHI)  
Acharya Dr. Sarvepalli Radhakrishnan Road, Soldevanahalli, Bengaluru-560107



**ACHARYA  
INSTITUTES**

**Certificate**

This is to Certify that the Project work entitled “**Simple Smart Connected City App**” is a bonafide work carried out by **Mr. Jay Narayan Das (1AY12IS034)**, **Mr. Lunagariya Divyesh (1AY12IS045)**, **Ms. Shubha Shree N (1AY12IS081)** and **Mr. Shubham Kumar (1AY12IS082)** in partial fulfillment for the award of the degree of **Bachelor of Engineering in Information Science and Engineering** of the **Visvesvaraya Technological University**, Belagavi during the year 2015-16. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The Project has been approved as it satisfies the academic requirements in respect of Project work prescribed for the Bachelor of Engineering Degree.

\_\_\_\_\_  
**(Prof. Srinivas .K)**  
Guide

\_\_\_\_\_  
**(Prof. C.K Marigowda)**  
HOD

\_\_\_\_\_  
**(Dr. H. D. Maheshappa)**  
Principal

**External Viva**

**Name of the Examiners**  
**date**

**Signature with**

1. \_\_\_\_\_

\_\_\_\_\_

2. \_\_\_\_\_

\_\_\_\_\_

# ACKNOWLEDGEMENT

We express our gratitude to our institute and management for providing us with good infrastructure facilities, qualified and inspiring staff, whose guidance was of immense help in the completion of this project successfully.

We express our sincere gratitude to **Dr. H.D Maheshappa**, Principal, AIT, for providing facility inside the campus and helping in the completion of course.

We also express our gratitude to **Prof. C.K Marigowda**, HeadDepartment of Information Science and Engineering, AIT, for his support and guidance.

We thank our project guide **Prof. Srinivas .K** Assistant Professor, Department of Information Science and Engineering for helping us and giving us the guidance in doing this project.

We also thank our project coordinator **Prof. A.R Chayapathi**, Assistant Professor, Department of Information Science and Engineering for his cooperation and support in completion of the project.

We would like to extend our gratitude to our parents for their help and support rendered us in various phases of this project.

Jay Narayan Das (1AY12IS034)

Lunagariya Divyesh (1AY12IS045)

Shubha Shree N (1AY12IS081)

Shubham Kumar (1AY12IS082)

# ABSTRACT

Simple Smart Connected City App, an Android App is aimed to facilitate support to General Public who face problems and issues like failure of street lights, sparking, electric pole damage and other issues including water supply, drainage system etc. This App helps general public in direct communication with the concerned department and notifies the issues and problems. Benefits of this project include improved targeted communication there by reducing the waiting time for the public to ensure the concerns reach the right department on-time and in a simple manner.

Traditional ideas of smart cities suggest lots of data collection by using sensors for Electricity, Water & Traffic management. This data needs to be carefully compiled and integrated into a Smart grid and then fed into computers that can focus on making the city as efficient as possible. This process is cumbersome and data management has multiple challenges. In addition, end-users, Consumers & General public communication has minimal input options.

By using Simple Smart Connected City App, users will be able to notify the issues seen and alert the department staff to take necessary actions and create a better living lifestyle and better city to live-in.

Smart Cities across different countries focus on using digital technology methods for faster, simple and effective notification alerts.

# CONTENTS

	Page Number
1 INTRODUCTION-----	1
1.1 Drawbacks of Existing-----	3
1.2 Problem Statement-----	3
1.3 Applications of the project-----	3
1.4 Proposed System-----	4
2 LITERATURE SURVEY-----	6
2.1 Literature Survey on Android App “CRAMAT Karnataka” -----	6
2.2 Literature Survey on Android App “Smart 24x7” -----	7
2.3 Literature Survey on Android App “CITIZEN COP” -----	10
3 REQUIREMENT SPECIFICATION AND ANALYSIS-----	12
3.1 Feasibility Study-----	12
3.2 Requirement Specification-----	13
3.3 Functional Requirements-----	13
3.4 Non Functional Requirements-----	14
3.5 Hardware Requirements-----	14
3.6 Software Requirements-----	15
3.7 Look and feel-----	31
4 DESIGN AND IMPLEMENTATION-----	32
4.1 Implementation-----	32
4.2 Detailed Design-----	42
5 TESTING-----	43
5.1 Unit Testing-----	43
5.2 Integration Testing-----	44
5.3 System Testing-----	45
5.4 Performance Testing-----	45
6 CONCLUSION AND FUTURE ENHANCEMENT-----	46
6.1 Conclusion-----	46
6.2 Future Enhancement-----	46
REFERENCES-----	47

# LIST OF FIGURES

Figure 1.1:Street View-----	2
Figure 2.1:Screenshot of CRAMAT App-----	7
Figure 2.2:Screenshot of Smart 24x7 App-----	9
Figure 2.3: Screenshot of CITIZEN COP App-----	11
Figure 3.1: Android Architecture -----	15
Figure 3.2:Dalvik Virtual Machine-----	21
Figure 4.1:Snapshot of Registration Screen-----	33
Figure 4.2:Login Screen-----	33
Figure 4.3:Welcome Screen-----	34
Figure 4.4:Complaint Status-----	34
Figure 4.5:Module 1: Volunteers/Senior Citizens-----	35
Figure 4.6:Admin Welcome Screen-----	36
Figure 4.7:Take Action Screen-----	36
Figure 4.8:Module 2: Main Admin-----	37
Figure 4.9:Welcome Screen-----	38
Figure 4.10: Complaint Status of Solved problem-----	38
Figure 4.11: Module 3: Police Admin-----	39
Figure 4.12: Data Flow diagram-----	40
Figure 4.13: Use case Diagram-----	41
Figure 4.14: Sequence Diagram-----	42

# LIST OF TABLE

Table 5.1:Unit Testing-----	43
Table 5.2: Integration Testing-----	44
Table 5.3: System Testing-----	45

## Chapter 1

# INTRODUCTION

As the digital world is growing exponentially, the need is to be able to adapt to the technology. From the very beginning of his election campaign, Our Indian Prime Minister has spoken about building smart cities in India. The promise has always been part of the great vision to make this India's century, alongside it is not possible to develop all at a single time. But not much had been done, so to part of development we are coming with one solution and we mainly we here concern about electricity and garbage collection.

When coming to point of electricity in our country maintain of electricity management very poor. We have or facing many problems like street light whether it is working or not, transformers short circuits, wiring, sparking, fell of electric pole like etc. and when we take city cleaning i.e. Garbage collection we have problems like waste dump, overfill, water over raising from manhole etc. We have solution for all these problems but thing is it long process to solve. Hence we face lot of problems by that. To avoid that we coming with new approach by this application.

To solving these our vision of an urban space that is ecologically friendly, technologically integrated and meticulously planned, with a particular reliance on the use of information technology to improve efficiency. In most parts of the world, the idea begins with using digital technology to make a city more efficient and to improve wellbeing.

Traditional ideas of smart cities suggest lots of data collection, using sensors – electricity, gas, water, traffic and other government analytics – that can be carefully compiled and integrated into a smart grid and then fed into computers that can focus on making the city as efficient as possible.



Figure 1.1:Street View

Most of the cities adopted the electricity supply by wire connections, and problem created by this system like ( We have or facing many problems like street light whether it is working or not, transformers short circuits, wiring, sparking, fell of electric pole ) very risky so we should aware of that.

Most of the cities adopted the underground drainage system and it is the duty of managing station (Municipal Corporation) to maintain cleanliness of the cities. If the drainage maintenance is not proper the pure water gets contaminate with drainage water and infectious diseases may get spread. The drainage gets blocked during rainy season, it will create problem for routine life such as traffic may get jammed, the environment becomes dirty, and totally it upsets the public. Suppose if there should be a facility which would be there in Municipal Corporation (managing station) that the officials come to know immediately after blocking of drainage in which area and the exact place where it is blocked and it also informs if the manhole lid is open.

To overcome this problem we develop an Android application .This Application can download by public and they can inform to related department for that Problem through the Application.

## 1.1 Drawbacks of Existing

- There's no privacy provided in existing apps.
- There's no direct communication to police department in case of Emergency cases like robbery, accident etc.
- User can't give picture from there phone gallery.
- User are not able to take pictures from there camera sometimes.

## 1.2 Problem Statement

Project aims at addressing - **General public do not have simple targeted communication methods to notify the issues and problems to concerned departments for an immediate action.**

## 1.3 Applications of the project

This project has lot of applications for day-to-day needs of General public. Users will be able to communicate to the respective departments for the following list of issues and concerns –

- Manhole open
- Overflow of sewage
- No water flow that are faced during rainy season
- Wastage dumped in empty sites
- Public water taps issues
- Street light damaged issues
- Problems in electric poles,
- Transformers short circuiting etc

## 1.4 Proposed System

Project aims at creating a Simple efficient method of communication the concern and issues faced by the general public.

Project has 3 Main Modules:

- a) Registration Module
- b) Login Module
- c) Admin Module

### a) Registration Module

This registration maintains the details about the Volunteer or Senior Citizen the database. The login information provides the accessible permission to the user and restricts the unauthorized users. The system records:-

- Public details and particulars.

### b) Login Module

This Login maintains the details about App users. The username and password helps to authenticate users while login which is stored in database through Web server, after login if any problem saw by public he can click capture button in the Application if capture clicked it will fetch the current location of the user send that information to database of government through web server by using this app.

Once if the problems solved the register of compliant citizen gets credit points. That point helps him getting a discount in some government bill paying.

### **c) Admin Module**

This Admin maintains the data in the database. Whenever a new user register all data are saved in database which only admin access that and even admin is only that having the privilege to see the complaints given by volunteers/Senior Citizen.

## **1.4.1 Advantages of the Proposed System**

- Simple Smart Connected City App will help in effective communication with government departments.
- Reduce the waiting time and improve the lifestyle.
- Day-to-day problems will be communicated to the right departments.
- Improve the turnaround time.
- The over-head of data collection, infrastructure requirement will be reduced.

## Chapter 2

# LITERATURE SURVEY

The literature review is a survey of the literature on a particular subject or area of interest. It involves finding out what literature is available on the subject, what the main areas of research are, who the prominent researchers in the field are, and what the current and possible future research questions are:

### 2.1 Literature Survey on Android App “CRAMAT Karnataka”

Send complaints about Street Lights, Potholes and Garbage in your locality quickly and easily to your municipal authorities with pictures and location maps. Help to make your city clean and green.

CRAMAT is Citizens' Reporting and Mapping Tool and part of Karnataka Mobile One initiative.

You can file complaints under following categories:

1. Street Lights
2. Roads
3. Sanitation
4. Others

You will get an instant sms with your complaint number. Complaint will be resolved by the Municipal Corporation as soon as possible.

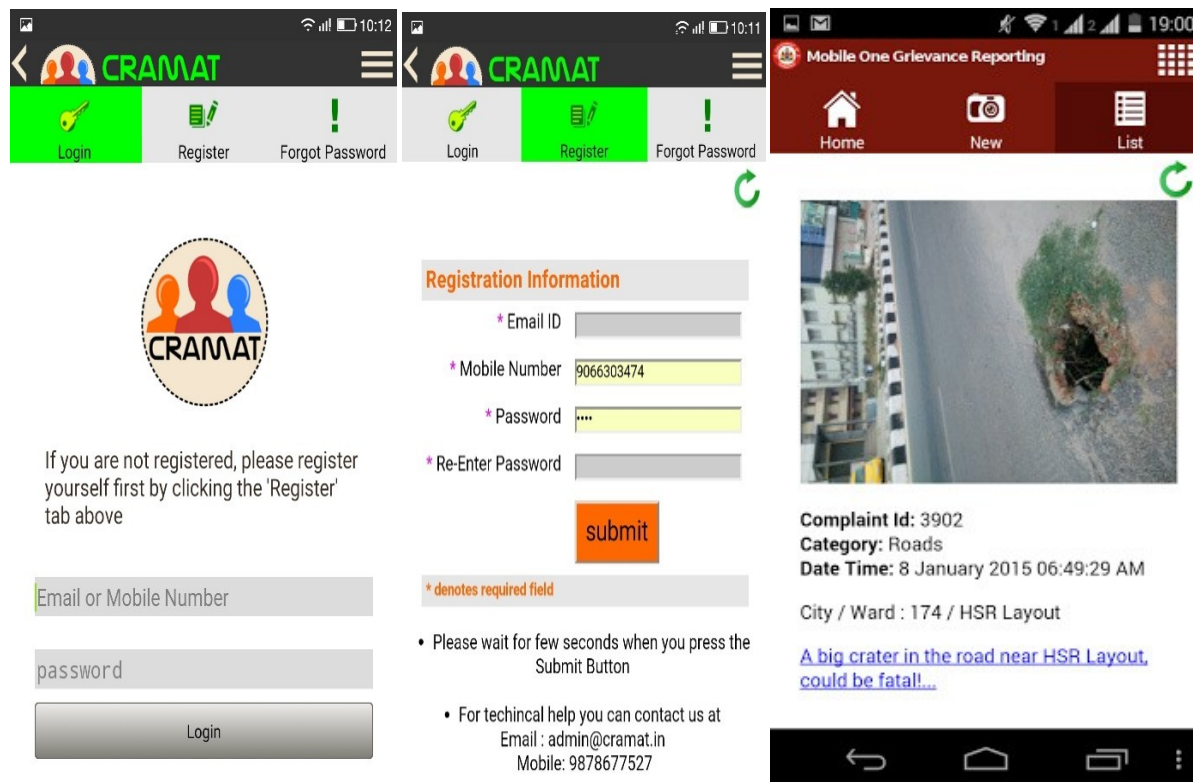


Figure 2.1: Screenshot of CRAMAT App

## 2.2 Literature Survey on Android App “Smart 24x7”

Smart24x7 has introduced a personal safety & security app for personal & business use that alerts emergency contacts with your GPS location. This is a unique approach towards strengthening citizen security, you can now also help others by pressing the button on your mobile. Our app is currently supported by Gurgaon Police, Jalandhar Police, Chandigarh Police, Jammu Police, Mohali Police, UP Fire Services (Lucknow & Noida). Smart24x7 enables the old cities to turn Smart City.

This application can also be used by Senior citizens to secure them when they are in distress by sending SOS signals to their loved ones.

At a time when personal safety apps have become a necessity for all smart phone users specially women, Smart 24X7, a personal safety app stands out in crowd.

Smart 24X7 is a unique safety app that enables a person in trouble to not only get connected with dear ones but also provides immediate assistance with the help of a 24-hour operational customer center.

The app by Smart 24X7 Response Services Private limited eyes public security, especially the safety of women. The personal safety application is useful for people of all age groups. One can send alert messages to family and friends and also get in touch with fire, police and ambulance services available nearby. The app has versions for Android, Window phone, iPhone and also Blackberry.

It allows users to feed five primary contacts once the app is installed. Irrespective of the cell phone model used by people in primary contacts, panic Alerts will be sent to them whenever user presses the PANIC Button during emergency. If GPRS is not working, the alert will be generated via SMS. The user also gets an immediate call from the Smart 24X7 Customer Care Centre.

It can be downloaded for free from the app store of the smartphone. Once downloaded a personal data form needs to be filled by the user. The user has to mention all the details including blood group, phone number and his or her location correctly to install the app in the phone.

As the app needs to track the location of the user, one can also get help from the nearest police or fire station, even hospitals. Geographic location of the person's position is also sent out to the primary contact list.

However, the important thing to know is that if the app is used without Internet connectivity, up to Rs. 30 can be deducted for each press of the PANIC button, which would be quite a dent on your phone balance.

### **\*Key Features:**

- Panic Alerts will be sent to the loved ones whenever user presses PANIC Button during emergency.
- If GPRS is not working alert will be generated via SMS.
- You can get instant help from nearest Police, Hospitals, Fire.
- Use Fake Call feature to walk out of any difficult situation.

## Simple Smart Connected City App

- Smart24x7 Application does voice recording, photographs during the panic Situation & transfers to the police.
- 24x7 Call center to assist you in emergency.
- Tracking of Primary Contact while in emergency.
- Share Travel alerts with your loved ones & Social Network.
- Basic chat to communicate with Friends.
- Tracking of Service Providers location Ambulance, Police and Fire.
- Maid/Servant Registration with Police is simplified
- Improve battery life

New features- On the way to Office/Home to inform your loved ones about your status.

Whenever you are in emergency Just follow three easy steps:-

- 1) Press the Panic button.
- 2) Select the type of services required.
- 3) Click Submit and you are done.

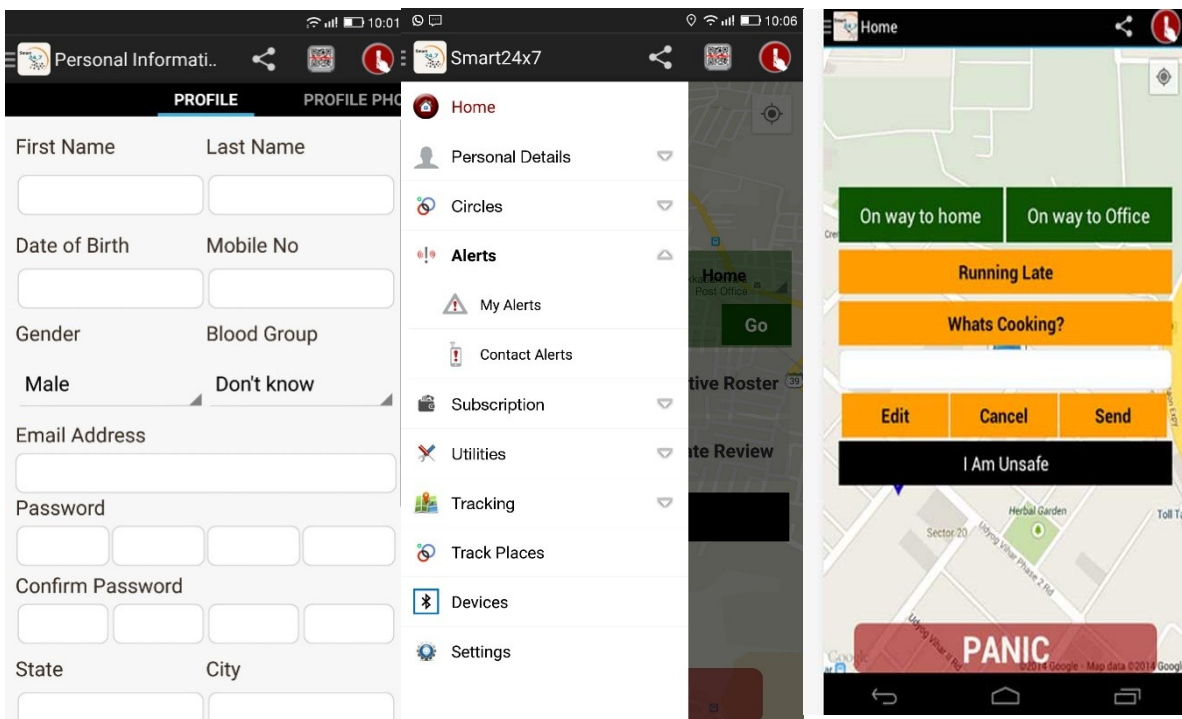


Figure 2.2: Screenshot of Smart 24x7 App

### **2.3 Literature Survey on Android App “CITIZEN COP”**

Citizen Cop provides a hassle free and easy way for crime reporting. Built with an initiative to empower common man, Citizen COP is a location based safety app. It is an initiative by Infocrats to end crime in your city and help develop a healthy living environment.

Citizen COP emphasizes on wellbeing of residents of a city, especially women. It has been successful in removing the barrier between citizens and police department. Whether you are victim of any crime or witness to an incident, Citizen COP is here to assist you. With Citizen COP crime reporting app, you can –

- Get Help in case of emergency
- Report any criminal incident or illegal activity anonymously
- Report lost or stolen articles
- Make emergency calls or send alerts
- Call Police
- Create a safe boundary with My safe zone (e-LakshmanRekha)

Citizen COP also allows you to –

- Search if your vehicle has been towed
- Allow loved ones to track your location with live tracking
- Check if a vehicle is stolen by providing vehicle registration card details
- Calculate auto-taxi fares and much more
- Know latest news and traffic updates

Citizen COP is a social initiative by INFOCRATS and was first started in Indore. It is now being used in Indore, Bhopal, Jabalpur, Ujjain, Raipur, Bengaluru, Jhansi, Navi Mumbai, Noida, Varanasi and some other major cities of India. It is a serious step taken to remove crime from the society so any unethical or irrelevant use of the application will result in disciplinary action.

Disclaimer:-

"Continued use of GPS running in the background can dramatically decrease battery life.

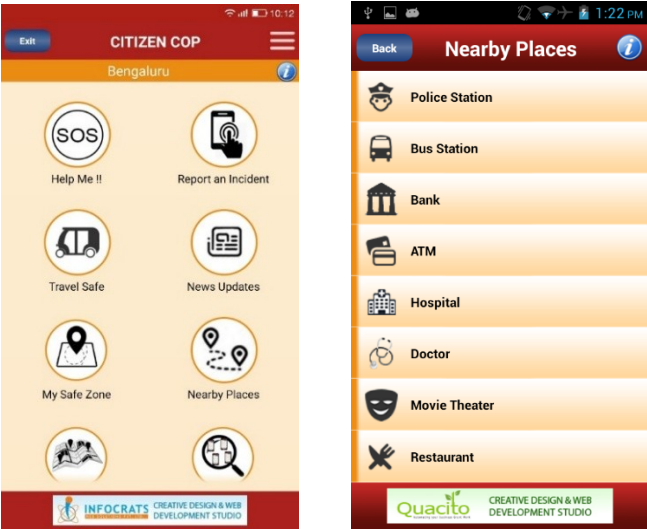


Figure 2.3: Screenshot of CITIZEN COP App

## Chapter 3

# REQUIREMENT SPECIFICATION AND ANALYSIS

Requirement in system engineering and software engineering compasses those tasks that go into determining the needs or conditions to meet for new or altered products, taking account of possibly conflicting requirements of the various stakeholders, such as beneficiaries or users. Requirement analysis is a critical to the success of a development of a project. Requirement must be actionable, measurable and testable related to identified business needs or opportunities and defined to enable a detailed sufficient for system design.

Conceptually, requirements analysis includes 3 types of activities:

- Eliciting Requirements: The task of communicating with customers and users to determine what their requirements are. This is sometimes also called requirement gathering.
- Analyzing Requirements: Determining whether the stated requirements are unclear, incomplete, ambiguous or contradictory and then resolving these issues.
- Recording Requirements: Requirements maybe documented in various forms, such as natural language documents, use-case, user stories or process specification.

### 3.1 Feasibility Study

Feasibility is a preliminary study undertaken to assess whether a planned project is likely to be practical and successful and to estimate its cost. If a project is seemed to be feasible from the result of the study, the next logical study is to proceed with it. The research and information uncovered in the feasibility study will support the detailed planning and reduce the search time.

- The main objective of the feasibility study is to determine whether a certain plan of action is feasible, that is, whether or not it will work and whether or not it is worth doing economically.

- The second part of a good feasibility study should focus on the proposed plan of action and provide a detailed estimate of its cost and benefits.

This involves questions such as whether the technology needed for the system exists, how difficult it will be to build, and whether the firm has enough experience using that technology. The assessment is based on an outline design of system requirements in terms of input, process, output, fields, programs and procedures. This can be quantified in terms of volumes of data, trends, frequency of updating etc. In order to estimate if the new system will perform adequately or not. Diagnosis of large and complex software systems is a challenging task that can highly benefit from monitoring of the high level functional requirements.

## **3.2 Requirement Specification**

The software requirement specification phase is an important step comes in software development lifecycle and it is a complete description of the behavior of the system to be developed. SRS should describe the general factors that affect the product and its requirements. The purpose of Software Requirement Specification (SRS) document is to specify the user goals and tasks that needed to be achieved. It must also include detailed description of the context and functional requirements, which are vital to the success completion of the project. Apart from this, the SRS document incorporates in itself the constraints and assumptions made during the course of the project. Requirement must be measurable, testable, related to identified needs or opportunities, and defined to a level of detail sufficient for system design.

## **3.3 Functional Requirements**

The functional requirements for a system describe the functionality or services that the system is expected to provide. These depend on the type of the software that is being developed and type of the system that is being developed. Functional requirement capture the intended behavior of the system. It is organized into 3 modules:

### 3.3.1 Admin/Certificate Authority Module

The Admin module functions as follows:

- Admin/Certificate authority is main module.
- Certificate authority is going to accept user's details.
- User has to fill the details in the register form which is given by admin.
- After registering admin will give session key to the authorized users.
- Using the session key generated by admin/certificate authority encryption and decryption takes place.

### 3.4 Non Functional Requirements

Nonfunctional requirements are the functions offered by the system. It includes time constraints and constraints on the development process and standards. The non-functional requirements are as follows:

- **Speed:** The system should process the given input into output within appropriate time.
- **Ease of use:** The software should be user friendly. Then the customers can use easily, so it doesn't require much training time.
- **Reliability:** The rate of failures should be less then only the system is more reliable.
- **Portability:** It should be easy to implement in any system.

### 3.5 Hardware Requirements

- **Processor:** P4 or more
- **Memory:** 512 MB of RAM, 1GB recommended
- **Hard Disk:** 2.5GB of hard disk space required for Android SDK, 140 MB additional hard disk space required for eclipse, java and android plug in installation.
- **Display :**1024x 768 or higher-resolution display with 16 bits colors of android mobile phone

- **Mobile phone:** Android (Version 2.3 or higher)

### 3.6 Software Requirements

- **Operating System:** The Android SDK can be installed on the following platforms:  
Microsoft Windows®7, 8, 10
- **Other:**
  - Eclipse
  - Android-sdk r15-windows for Windows.
  - ADT (Android Development Tool)
  - Android Emulator
- **Language:** Android, JAVA, SOAP or Restful web service.
- **Database:** Mysql
- **Web Server:** Apache Tomcat 7.0 or higher.
- SOAP web service

#### 3.6.1 Definition



Figure 3.1: Android Architecture

## ➤ **Android**

**Android** is a complete set of software for mobile devices such as tablet computers, notebooks, smart phones, electronic book readers, set-top boxes etc. It contains a Linux-based Operating System, middleware and key mobile applications.

It can be thought of as a mobile operating system. But it is not limited to mobile only. It is currently used in various devices such as mobiles, tablets, televisions etc.

It is developed by Google and later the OHA (Open Handset Alliance). Java language is mainly used to write the android code even though other languages can be used. The goal of the android project is to create a successful real-world product that improves the mobile experience for end users.

### **3.6.2 What is Open Handset Alliance?**

It's a consortium (association) of 84 companies such as google, Samsung, AKM, Synaptics, KDDI, Garmin, Teleca, Ebay, Intel etc.

It was established on 5th November 2007, led by Google. It is committed to advance open standards, provide services and deploy handsets using the Android Platform.

### **3.6.3 Features of Android**

- It is open-source.
- Anyone can customize (modify (something) to suit a particular individual or task.) the Android Platform.
- There are a lot of mobile applications that can be chosen by the consumer.
- It provides many interesting features like weather details, opening screen, live RSS (Really Simple Syndication) feeds etc.
- It provides support for messaging services (SMS and MMS), web browser, storage (SQLite), connectivity (GSM, CDMA, Blue Tooth, and Wi-Fi ), media, handset layout etc.

### 3.6.4 Android Version History

The first commercial version, Android 1.0, was released in September 2008. Since **2008**, Android has seen numerous updates which have incrementally improved the operating system, adding new features and fixing bugs in previous releases. Each major release is named in **alphabetical order** (the exceptions are versions 1.0 and 1.1) after a **dessert or sugary treat**; for example, version 1.5 Cupcake was followed by 1.6 Donut. The latest released version, 4.4.4 KitKat, appeared as a security-only update; it was released on June 19, 2014, shortly after the release of 4.4.3

- Android alpha (1.0)
- Android beta (1.1)
- Cupcake (1.5)
- Donut (1.6)
- Eclair (2.0–2.1)
- Froyo (2.2–2.2.3)
- Gingerbread (2.3–2.3.7)
- Honeycomb (3.0–3.2.6)
- Ice Cream Sandwich (4.0–4.0.4)
- Jelly Bean (4.1–4.3.1)
- KitKat (4.4–4.4.4)
- "L" release (developer preview)

There were at least two internal releases (Android Alpha) inside Google and the OHA before the Android beta was released in November 2007.

The **Android beta was released on November 5, 2007**, while the software development kit (SDK) was released on November 12, 2007. The November 5 date is popularly celebrated as Android's "birthday".

- Android version 1.0 (API level 1) - September 23, 2008
- Android version 1.1 (API level 2) – February 9, 2009

- Android version 1.5 (API level 3) – CUPCAKE – April 27, 2009
- Android version 1.6 (API level 4) – DONUT – September 15, 2009
- Android version 2.0 (API level 5) – ECLAIR – October 26, 2009
- Android version 2.0.1 (API level 6) – ECLAIR – December 3, 2009
- Android version 2.1 (API level 7) – ECLAIR – January 12, 2010
- Android version 2.2-2.2.3 (API level 8) – FROYO – May20, 2010
- Android version 2.3-2.3.2 (API level 9) – GINGERBREAD – December 6, 2010
- Android version 2.3.3-2.3.7 (API level 10) – GINGERBREAD – February9, 2011
- Android version 3.0 (API level 11) – HONEYCOMB – February 22, 2011
- Android version 3.1 (API level 12) – HONEYCOMB – May10, 2011
- Android version 3.2 (API level 13) – HONEYCOMB – July15, 2011
- Android version 4.0-4.0.2 (API level 14) – ICE CREAM SANDWICH – October 18, 2011
- Android version 4.0.3-4.0.4 (API level 15) – ICE CREAM SANDWICH – December 16, 2011
- Android version 4.1 (API level 16) – JELLY BEAN – July 09, 2012
- Android version 4.2 (API level 17) – JELLY BEAN – November13, 2012
- Android version 4.3 (API level 18) – JELLY BEAN – July24, 2013
- Android version 4.4 (API level 19) – KITKAT – October 31, 2013
- Android version 4.4 (API level 20) – KITKAT with wearable extensions – July 22, 2014

### **3.6.5 API Meaning**

In computer programming, an application programming interface (API) specifies a software component in terms of its operations, their inputs and outputs and underlying types. Its main purpose is to define a set of functionalities that are independent of their respective implementation, allowing both definition and implementation to vary without compromising each other.

In addition to accessing databases or computer hardware, such as hard disk drives or video cards, an API can be used to ease the work of programming graphical user interface components, to allow integration of new features into existing applications (a so-called "plug-in API"), or to share data between otherwise distinct applications. In practice, many times an API comes in the form of a library that includes specifications for routines, data structures, object classes, and variables. In some other cases, notably for SOAP and RESTservices, an API comes as just a specification of remote calls exposed to the API consumers.

### 3.6.6 Building Blocks of Android

- **Activity**
  - Present a visual user interface for one focused endeavor the user can undertake
  - Example: a list of menu items users can choose from
- **Services**
  - Run in the background for an indefinite period of time
  - Example: calculate and provide the result to activities that need it
- **Broadcast Receivers**
  - Receive and react to broadcast announcements
  - Example: announcements that the time zone has changed
- **Content Providers**
  - Store and retrieve data and make it accessible to all applications
  - Example: Android ships with a number of content providers for common data types (e.g., audio, video, images, personal contact information, etc.)
- **Intents**
  - Hold the content of a message
  - Example: convey a request for an activity to present an image to the user or let the user edit some text

**Fragments:** are like parts of the activity. An activity can display one or more fragments on the screen at the same time.

**AndroidManifest.xml:** contains information about activities, content providers, permissions etc. It also **declares the android API** that the application is going to use. It is like the web.xml file in Java EE.

**<intent-filter>:** is the sub-element of activity that describes the type of intent to which activity, service or broadcast receiver can respond to.

**<Action>:** adds an action for the intent-filter. The intent-filter must have at least one action element.

**Android Virtual Device (AVD):** is used to test the android application without the need for mobile or tablet etc. It can be created in different configurations to emulate different types of real devices.

**Android Emulator:** is used to run, debug and test the android application. If you don't have the real device, it can be the best way to run, debug and test the application. It uses an open source processor emulator technology called QEMU (Quick Emulator).

**SetContentView (R.layout.activity\_main):** gives information about our layout resource.

**R.java:** file is an auto-generated file that contains IDs for all the resources of res directory. It is generated by **aapt (Android Asset Packaging Tool)**. Whenever we create any component on the layout, a corresponding ID is created in the R.java file which can be used in the Java Source file later.

**Apk (Android application Package):** file is created by the framework automatically. If we want to run the android application on the mobile, transfer and install it.

#### **Manifest file:**

It contains information about the package including components such as activities, services, content providers, broadcast receivers etc.

It performs some other tasks also:

- It is responsible for protecting the application to access any protected parts by providing the permissions.
- It also declares the android API that the application is going to use.
- It lists the instrumentation classes. The instrumentation classes provide profiling and other information. This information is removed just before the application is published etc.

This is the required XML file for all the android application and located inside the root directory.

### 3.6.7 Dalvik Virtual Machine

Dalvik is a name of a town in Iceland. The Dalvik VM was written by Dan Bornstein. The Dex compiler converts the class files into the .dex files that run on the DalvikVM. Let's see

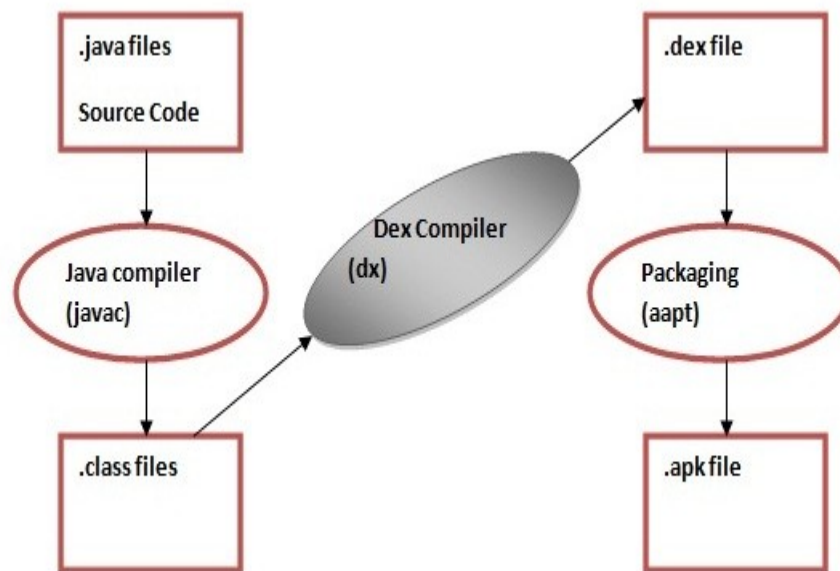


Figure 3.2: Dalvik Virtual Machine

the compiling and packaging process from the source file:

The **javac tool** compiles the java source file into the class file. The **dx tool** takes all the class files of our application and generates a single **.dex (Dalvik Executable)** file. It is a platform-specific tool. The **Android Assets Packaging Tool (aapt)** handles the packaging process.

- The Android system uses a special virtual machine, Dalvik, to run Java-based applications. Dalvik uses a custom bytecode format which is different from Java bytecode.
- Therefore, we cannot run Java class files on Android directly; they need to be converted into the Dalvik bytecode format.
- Dalvik virtual machine is register-based whereas JVM is stack based.
- The Dalvik Virtual Machine (DVM) is optimized for mobile devices. It optimizes the JVM for memory, battery life and performance.
- In standard Java environments, Java source code is compiled into Java bytecode, which is stored within .class files. The .class files are read by the JVM at runtime.
- On the Android platform, The Android SDK contains a tool called dx (platform specific) which converts Java class files into a .dex (Dalvik Executable) file.
- During this conversion process, redundant information in the class files is optimized in the .dex file.
- These .dex files are much smaller in size than the corresponding class files. The .dex file and the resources of an Android project, e.g., the images and XML files, are packed into a .apk (Android Package) file. The program aapt (Android Asset Packaging Tool) performs this step.
- The resulting .apk file contains all necessary data to run the Android application and can be deployed to an Android device via the ADB (Android Debug Bridge) tool.

### 3.6.8 Android Widgets

There are given a lot of android widgets with simplified examples such as Button, Edit Text, Auto Complete Text View, Toggle Button, Date Picker, Time Picker and Progress Bar etc. Android widgets are easy to learn. The widely used android widgets with examples are given below:

**Android Button:** Let's learn how to perform event handling on button click.

**Android Toast:** Displays information for the short duration of time.

**Custom Toast:** We are able to customize the toast, such as we can display image on the toast

**Toggle Button:** It has two states ON/OFF.

**Check Box:** Let's see the application of simple food ordering.

**Alert Dialog:** Alert Dialog displays an alert dialog containing the message with OK and Cancel buttons.

**Spinner:** Spinner displays the multiple options, but only one can be selected at a time.

**AutoCompleteTextView:** Let's see the simple example of AutoCompleteTextView.

**Rating Bar:** Rating Bar displays the rating bar.

**Date Picker:** Date picker displays the date picker dialog that can be used to pick the date.

**Time Picker:** Time Picker displays the time picker dialog that can be used to pick the time.

**Progress Bar:** Progress displays progress task

### 3.6.9 Software's used:

- **Overview of Java**

Java is loosely based on C++ syntax, and is meant to be Object-Oriented Structure of java is midway between an interpreted and a compiled language. The java compiler into Byte Codes, which are secure and portable across different platforms, compiles Java programs. These byte codes are essentially instructions encapsulated in single type, to what is known as java virtual machine (JVM), which resides in standard browser.

JVM is available for almost all OS. JVM converts these byte codes into machine specific instructions at runtime. Java is actually a platform consisting of three components:

- Java programming language.
- Java library of classes and interfaces.
- Java Virtual Machine

- **Features of Java**

- Java is a simple language. It does not make use of pointers, function overloading etc...

- Java is object-oriented language and supports encapsulation, inheritance, Polymorphism and dynamic binding, but does not support multiple inheritances.
- Everything in java is an object except some primitive data types.
- Java is portable.
- It is an architecture neutral that is java programs once compiled can be executed on any machine that is enabled.
- Java is distributed in its approach and used for Internet programming.
- Java is robust, secured, high performing and dynamic in nature.
- Java supports multithreading. Therefore different parts of the program can be executed at the same time.

- **JDK Versions**

- **JDK Alpha and Beta**

Alpha and Beta Java public releases had highly unstable APIs and ABIs. The supplied Java web browser was named Web Runner.

- **JDK 1.0**

Originally called Oak. Initial release [3] [4] the first stable version, JDK 1.0.2, is called Java 1.

Note: In versions of Java and the JDK up to 1.0.1, private and protected keywords could be used together to create yet another form of protection that would restrict access to methods or variables solely to subclasses of a given class. As of 1.0.2, this capability has been removed from the language.

## ➤ **JDK 1.1**

Major additions included: [5]

- An extensive retooling of the AWT event model.
- Inner classes added to the language.
- JavaBeans.
- JDBC.
- RMI.
- Reflection which supported Introspection only, no modification at runtime was possible.
- JIT (Just in Time) compiler on Microsoft Windows platforms, produced for Java Soft by Symantec.3.7.4 J2SE 1.2

Codename Playground. This and subsequent releases through J2SE 5.0 were rebranded retrospectively Java 2 and the version name "J2SE" (Java 2 Platform, Standard Edition) replaced JDK to distinguish the base platform from J2EE (Java 2 Platform, Enterprise Edition) and J2ME (Java 2 Platform, Micro Edition). This was a very significant release of Java as it tripled the size of the Java platform to 1520 classes in 59 packages. Major additions included:[6]

- Strict fp keyword.
- The Swing graphical API was integrated into the core classes.
- Sun's JVM was equipped with a JIT compiler for the first time.
- Java plug-in.
  
- Java IDL, an IDL implementation for CORBA interoperability.
- Collections framework.
- 

## ➤ **J2SE 1.3**

Codename Kestrel. The most notable changes were:[7][8]

- HotSpot JVM included (the HotSpot JVM was first released in April 1999 for the J2SE 1.2 JVM)
- RMI was modified to support optional compatibility with CORBA
- Java Naming and Directory Interface (JNDI) included in core libraries (previously available as an extension)
- Java Platform Debugger Architecture (JPDA)
- Java Sound
- Synthetic proxy classes

### ➤ **J2SE 1.4**

Codename Merlin. This was the first release of the Java platform developed under the Java Community Process as JSR 59. Major changes included:<sup>[9][10]</sup>

#### Language changes

- Assert keyword (Specified in JSR 41.). Library improvements
- Regular expressions modeled after Perl regular expressions
- Exception chaining allows an exception to encapsulate original lower-level exception
- Internet Protocol version 6 (IPv6) support
- Non-blocking IO (named New Input/Output, NIO) (Specified in JSR 51.)
- Logging API (Specified in JSR 47.)
- Image I/O API for reading and writing images in formats like JPEG and PNG
- Integrated XML parser and XSLT processor (JAXP) (Specified in JSR 5 and JSR 63.)
- Integrated security and cryptography extensions (JCE, JSSE, JAAS)
- Java Web Start included (Java Web Start was first released in March 2001 for J2SE 1.3) (Specified in JSR 56.)
- Preferences API (java.util.prefs). Support and security updates for Java 1.4 ended in October 2008.<sup>[11]</sup>

## ➤ J2SE 5.0

Codename Tiger. Originally numbered 1.5, it is still used as the internal version number. The number was changed to "better reflect the level of maturity, stability, scalability and security of the J2SE." <sup>[12]</sup> This version was developed under JSR 176.

J2SE 5.0 entered its end-of-public-updates period on April 8, 2008; updates are no longer available to the public as of November 3, 2009. Updates will be available to Oracle Customers until May 2014. <sup>[13]</sup>

Tiger added a number of significant new language features. <sup>[14][15]</sup>

- Generics: Provides compile-time (static) type safety for collections and eliminates the need for most typecasts (type conversion). (Specified by JSR 14.)
- Metadata: Also called annotations; allows language constructs such as classes and methods to be tagged with additional data, which can then be processed by metadata-aware utilities. (Specified by JSR 175.)
- Autoboxing/unboxing: Automatic conversions between primitive types (such as `int`) and primitive wrapper classes (such as `Integer`). (Specified by JSR 201.)
- Enumerations: The `enum` keyword creates a typesafe, ordered list of values (such as `Day.MONDAY`, `Day.TUESDAY`, etc.). Previously this could only be achieved by non-typesafe constant integers or manually constructed classes (typesafe enum pattern). (Specified by JSR 201.)
- Varargs: The last parameter of a method can now be declared using a type name followed by three dots (e.g. `void draw text (String... lines)`). In the calling code any number of parameters of that type can be used and they are then placed in an array to be passed to the method, or alternatively the calling code can pass an array of that type.
- Enhanced for each loop: The `for` loop syntax is extended with special syntax for iterating over each member of either an array or any `Iterable`, such as the standard `Collection` classes. (Specified by JSR 201.)
- Fix the previously broken semantics of the Java Memory Model, which defines how threads interact through memory.

- Static imports

There were also the following improvements to the standard libraries:

- Automatic stub generation for RMI objects.
- Swing: New skinnable look and feel, called synth.
- The concurrency utilities in package `java.util.concurrent`.<sup>[16]</sup>
- Scanner class for parsing data from various input streams and buffers.

Java 5 is the last release of Java to officially support the Microsoft Windows 9x line (Windows 95, Windows 98, Windows ME),<sup>[17]</sup> while Windows Vista is the newest version of Windows that J2SE 5 was supported on prior to Java 5 going end of life in October 2009.<sup>[11]</sup>

Java 5 is the default version of Java installed on Apple Mac OS X 10.5 (Leopard). Java 6 can be installed and set as the default to be used on 64-bit (Core 2 Duo and higher) processor machines.<sup>[18]</sup> Java 6 is also supported by 32-bit machines running Mac OS X 10.6 (Snow Leopard).

### ➤ **JAVA SE**

Codename Mustang. As of this version, Sun replaced the name "J2SE" with Java SE and dropped the ".0" from the version number.<sup>[19]</sup> Internal numbering for developers remains 1.6.0.<sup>[20]</sup> This version was developed under JSR 270.

During the development phase, new builds including enhancements and bug fixes were released approximately weekly. Beta versions were released in February and June 2006, leading up to a final release that occurred on December 11, 2006.

Major changes included in this version:

- Support for older Win9x versions dropped; unofficially, Java 6 Update 7 was the last release of Java shown to work on these versions of Windows.<sup>[citation needed]</sup> This is believed <sup>[by whom?]</sup> to be due to the major changes in Update 10.
- Scripting Language Support (JSR 223): Generic API for tight integration with scripting languages, and built-in Mozilla JavaScript Rhino integration
- Dramatic performance improvements for the core platform,<sup>[23][24]</sup> and Swing.
- Improved Web Service support through JAX-WS (JSR 224)
- JDBC 4.0 support (JSR 221).
- Java Compiler API (JSR 199): an API allowing a Java program to select and invoke a Java Compiler programmatically.
- Upgrade of JAXB to version 2.0: Including integration of a StAX parser.
- Support for pluggable annotations (JSR 269)<sup>[25]</sup>
- Many GUI improvements, such as integration of Swing Worker in the API, table sorting and filtering, and true Swing double-buffering (eliminating the gray-area effect).
- JVM improvements include: synchronization and compiler performance optimizations, new algorithms and upgrades to existing garbage collection algorithms, and application start-up performance.<sup>[26]</sup>

Java 6 reached the end of its supported life in February 2013, at which time all updates, including security updates, were scheduled to be stopped.<sup>[27][28]</sup> Oracle released one more update to Java 6 in March 2013, which patched some security vulnerabilities.<sup>[29]</sup>

- **Eclipse:**

Eclipse is a platform that has been designed from the ground up for building integrated web and application development tooling. By design, the platform does not provide a great deal of end-user functionality by itself.

- **Android SDK**

It is the software development kit used for developing android apps. This kit includes-

- Debugger
- Libraries
- Quick emulator
- Documentation
- Sample code
- Tutorials

Android virtual device is used to get a view of an app. This SDK will be included in the android in the bundle called as ADT bundle. This ADT bundle is then extracted to get the above facilities.

- **ADT(Android Development Tool)**

It is the plug-in for the eclipse with the help of which following tasks can be performed-

- Set up new Android projects
- Creating a new application UI

In total it provides an environment in which the android projects can be imported, build and run successfully.

- **Web Service**

A Web service is a software which can connect any device that is active on the internet to another and establish communication between them. It uses HTTP as a common communication protocol. Web service is required to establish communication between Android device and Shop's database to exchange information.

- **SOAP**

SOAP is called as Simple Object Access protocol. SOAP provides a way to communicate between applications running on different operating systems, with different technologies and programming languages.

- **MYSQL**

MySQL pronounced either "My SQL" or "My Sequel," is an open source relational database management system. It is based on the structure query language (SQL), which is used for adding, removing, and modifying information in the database. Standard SQL commands, such as ADD, DROP, INSERT, and UPDATE can be used with MySQL.

MySQL can be used for a variety of applications, but is most commonly found on Web servers. A website that uses MySQL may include Web pages that access information from a database. These pages are often referred to as "dynamic," meaning the content of each page is generated from a database as the page loads. Websites that use dynamic Web pages are often referred to as database-driven websites.

### **3.7 Look and feel**

In software design look and feel is used in respect of GUI and comprises of its design, including elements such as colors, shapes, layout and typefaces(the "LOOK") as well as the behavior of dynamic elements such as button, boxes and menus(the "FEEL"). The term look and feel is used in reference to both software and websites.

## Chapter 4

# DESIGN AND IMPLEMENTATION

### 4.1 Implementation

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new Simple Smart Connected City App and in giving the user, confidence that the new Simple Smart Connected City App will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and its constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

The Simple Smart Connected City App is implemented in three modules/three different interfaces first one for the volunteers/senior citizens which they can use to register to the app, login, give complaint, check status and get credit point. The second interface is for main admin who monitor's the whole work like see active and solved complaints and resolve the given issue by the user and giving credit point according to the complaint given. The third interface is for police admin who also act as an admin in case of emergency cases by seeing and resolving the given complaint by volunteers/senior citizen.

### 4.1.1 Module 1: Volunteers/Senior Citizens

- Download the Android based Simple Smart Connected city App on his/her mobile.
- Register to the App by providing the Customer details like Name, Gender, City, Age, Phone No, Email id etc. by selecting senior citizen or volunteers from the radio button.



Figure 4.1: Snapshot of Registration Screen

- Login with Username and password (Unique password will be generated and send to Customer Email ID/Phone Number).

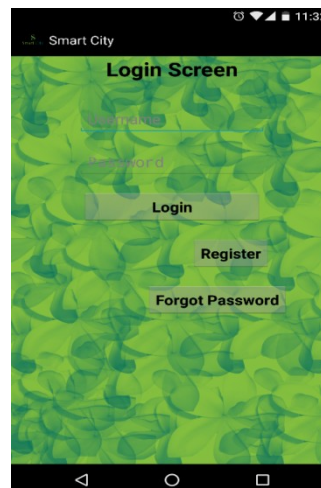


Figure 4.2: Login Screen

- Once the user logs in the welcome screen will appear then the user can select any one option from



Figure 4.3: Welcome Screen

3 options like Rise issues, Check Status and Credit points.

In rise issues here we are registering complaints regarding particular issues.

- In check status here we can see the status of issue raised like open status or work on progress and so on.
- In credit points, here Admin will give credits base on issue raised.
- All the data will be save on database server with respect GPS location.



Figure 4.4: Complaint Status

**Block Diagram:**

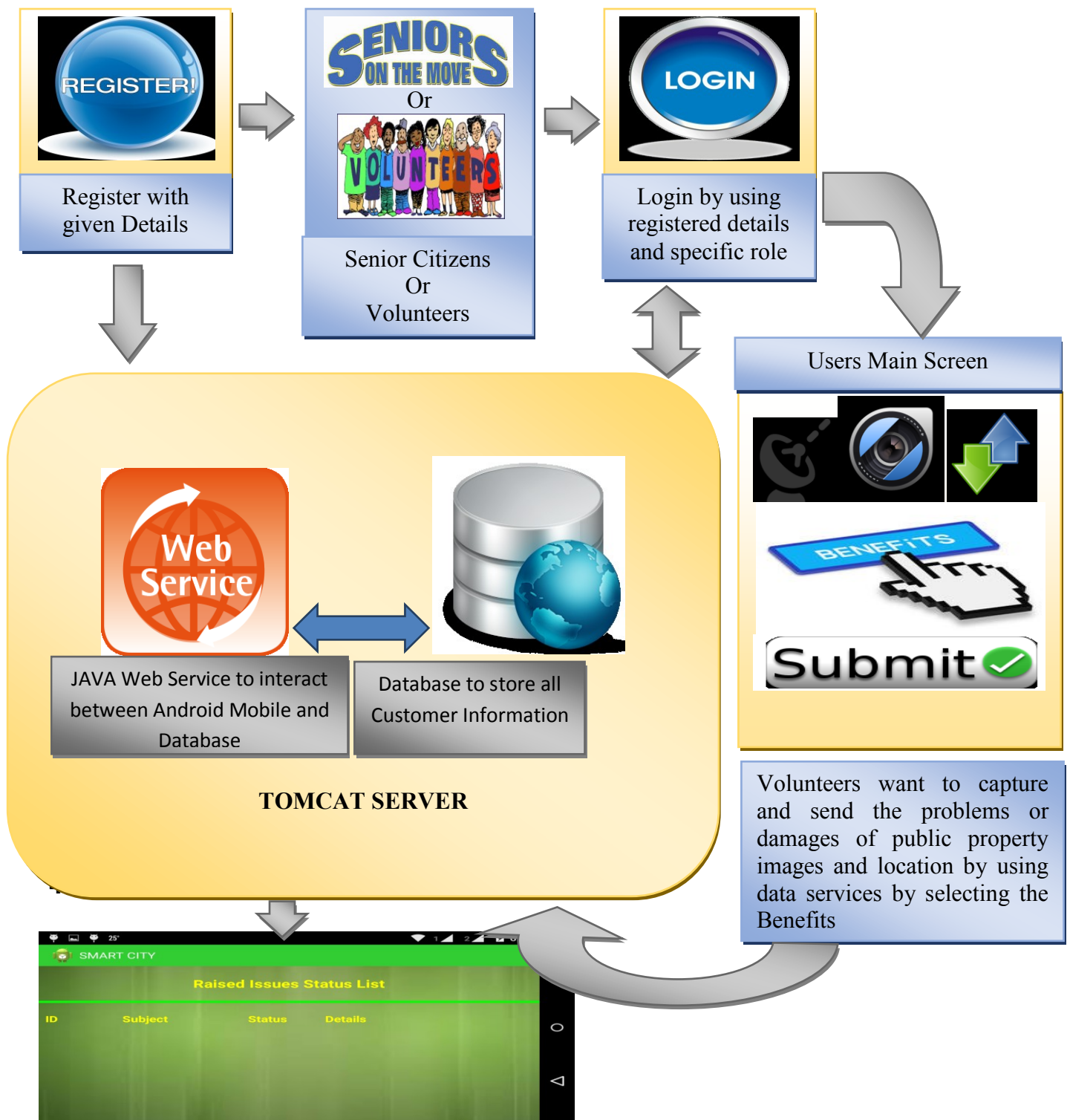


Figure 4.5: Module 1: Volunteers/Senior Citizens

### 4.1.2 Module 2: Main Admin

- Admin will login by entering their user id and password.
- Which is mapped with the information in the database.
- Database is used to store all the information.
- If the login credentials are correct it shows a main screen consists of solved problems and active problems.
- Active problems contains list of emergency issues which are uploaded by the senior citizen/volunteers with respect to name, location and other details.
- Admin take in charge immediate action like “ACCEPT/REJECT”.
- Solves the issue and provide credit points to the user who has raised an issue.
- Solved problems contains list of all the solved problems with the details and the benefits given.
- Tomcat Server is used.
- Java web service is used to interact with the android mobile and the database.

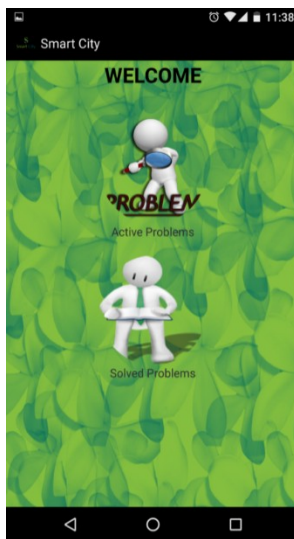


Figure 4.6: Admin Welcome Screen



Figure 4.7: Take Action Screen

**Block Diagram:**

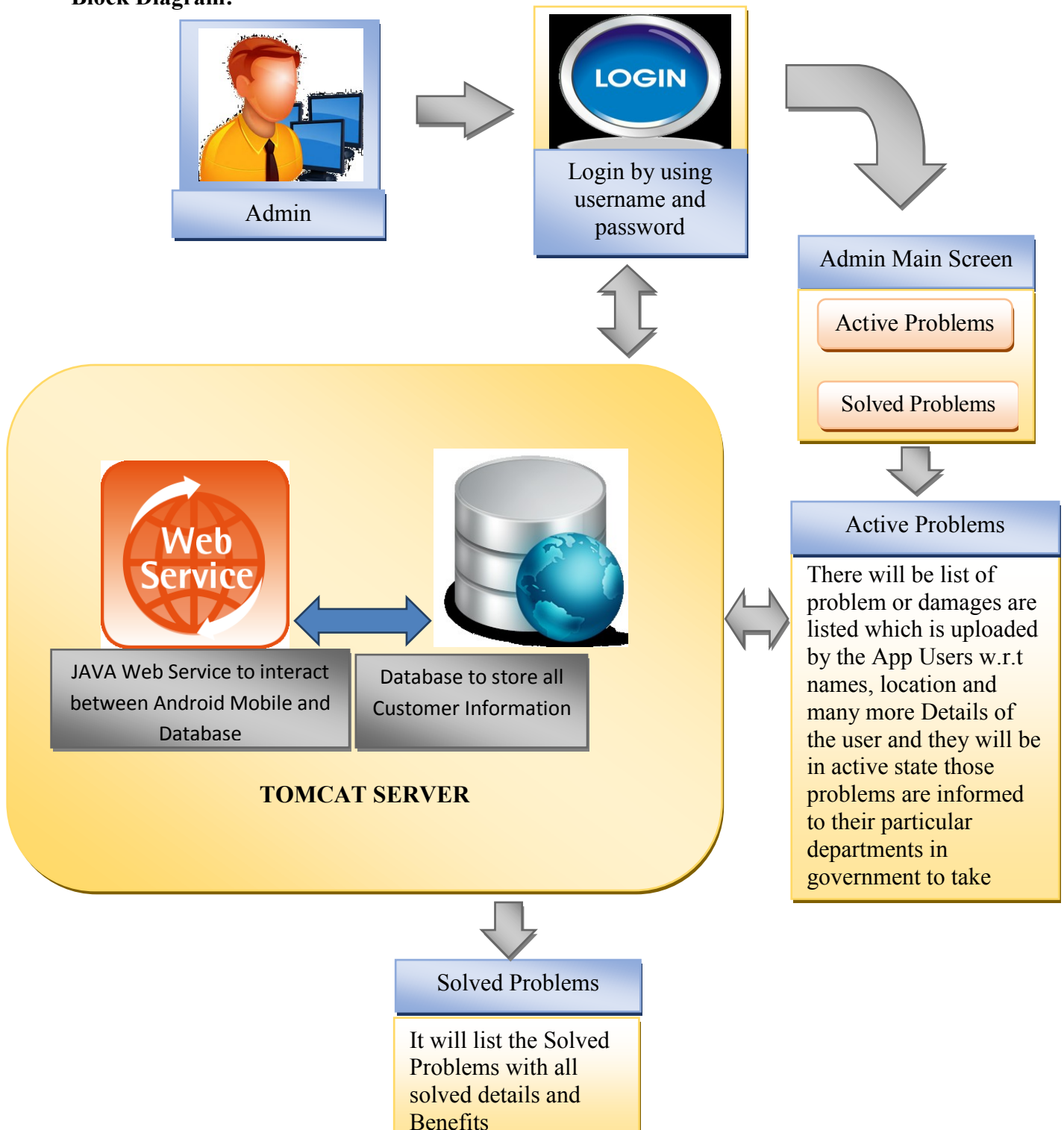


Figure 4.8:Module 2: Main Admin

### 4.1.3 Module 3: Police Station Incharge

- Police station in charge will login by entering their user id and password
- Which is mapped with the information in the database.
- Database is used to store all the information.
- If the login credentials are correct it shows a main screen consists of solved problems and active problems.
- Active problems contains list of emergency issues which are uploaded by the senior citizen and volunteers with respect to name, location and other details.
- Police station in charge take immediate action like “ACCEPT/REJECT”.
- Solves the issue and provide credit points to the user who has raised an issue.
- Solved problems contains list of all the solved problems with the details and the benefits given.
- Tomcat Server is used.
- Java web service is used to interact with the android mobile and the database.



Figure 4.9: Welcome Screen



Figure 4.10: Complaint Status of Solved problem

**Block Diagram:**

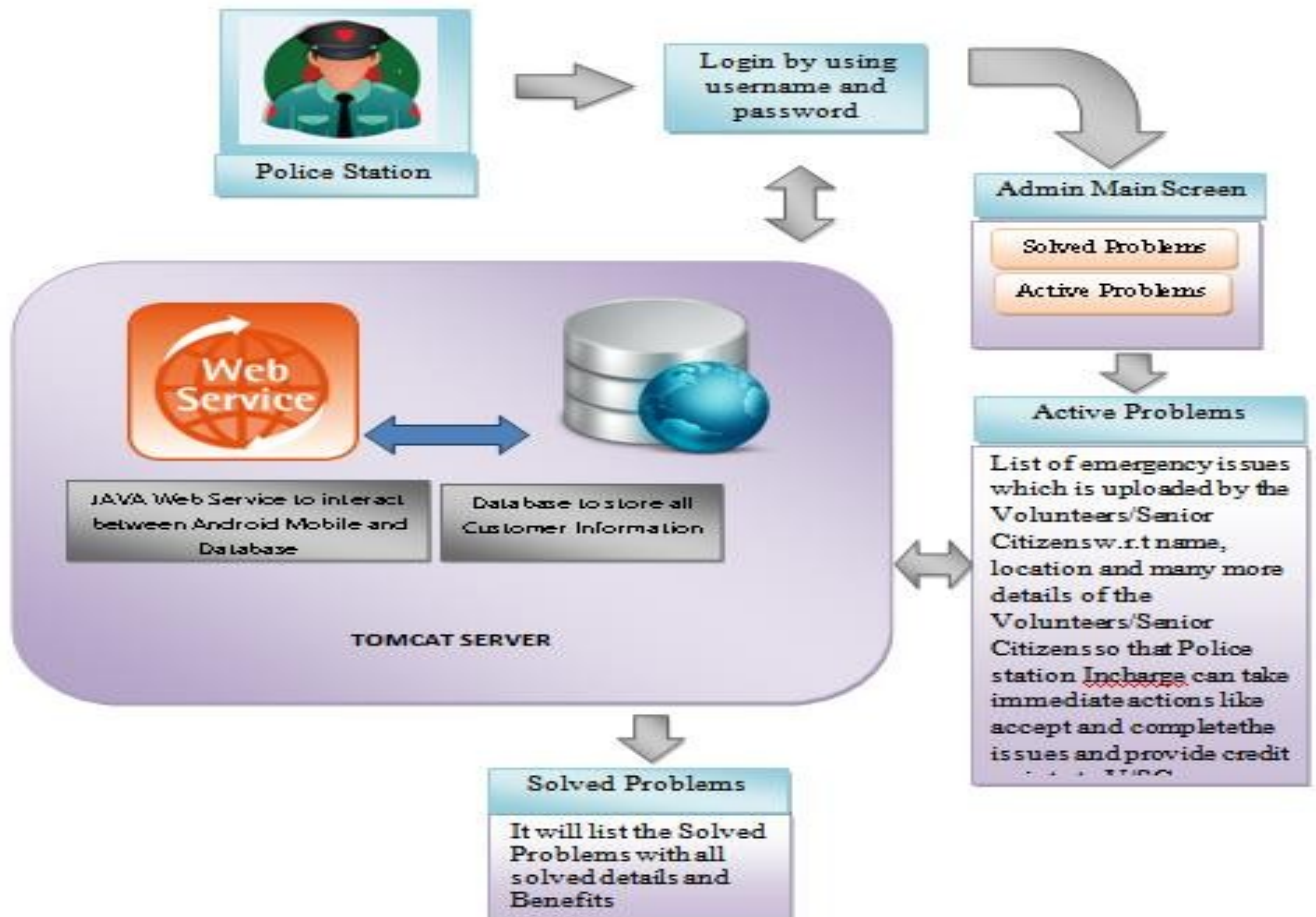
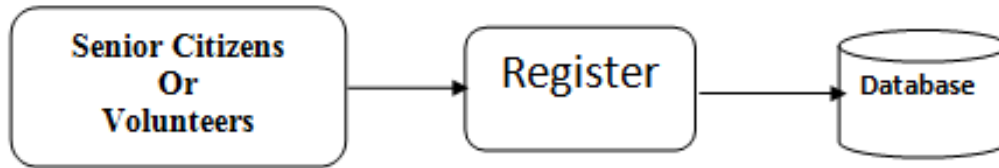


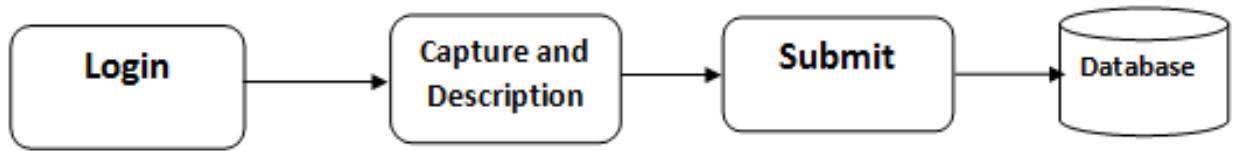
Figure 4.11:Module 3: Police Admin

### 4.1.4 Data Flow diagram

#### Data flow level 1



#### Data flow level 2



#### Data flow level 3



#### Data flow level 4

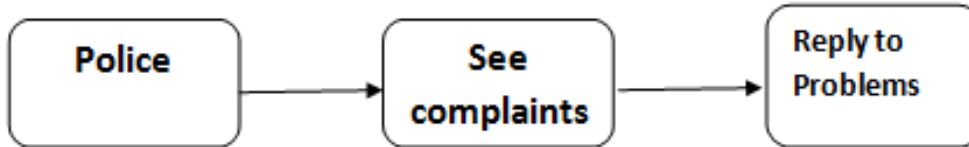


Figure 4.12:Data Flow diagram

### 4.1.5 Use case Diagram

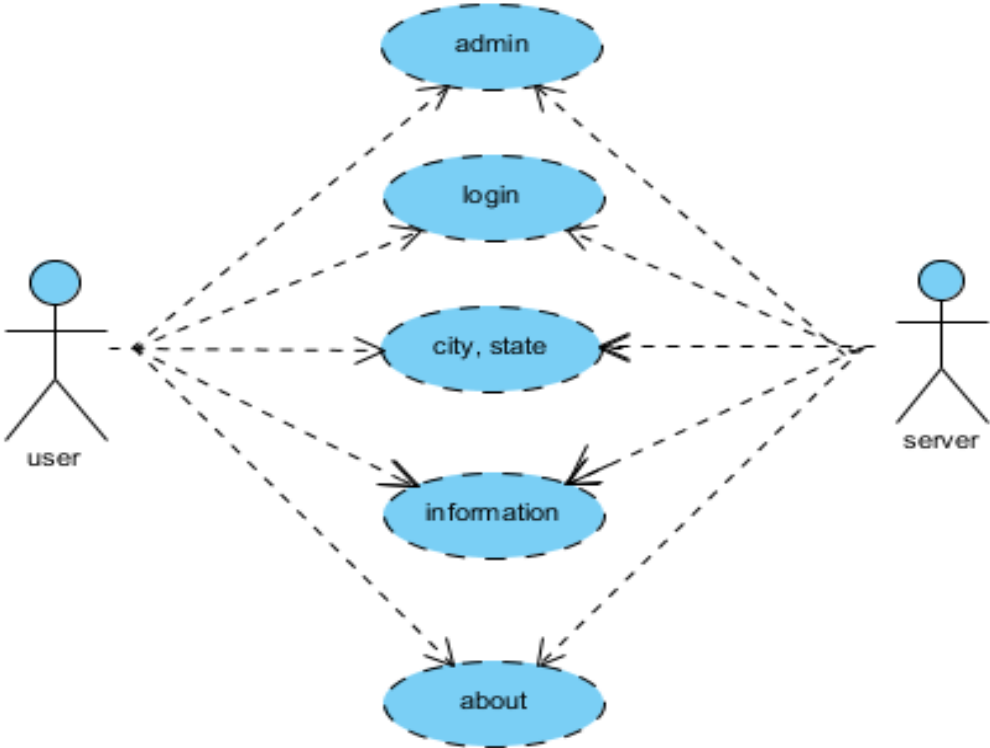


Figure 4.13:Use case Diagram

### 4.1.6 Sequence Diagram

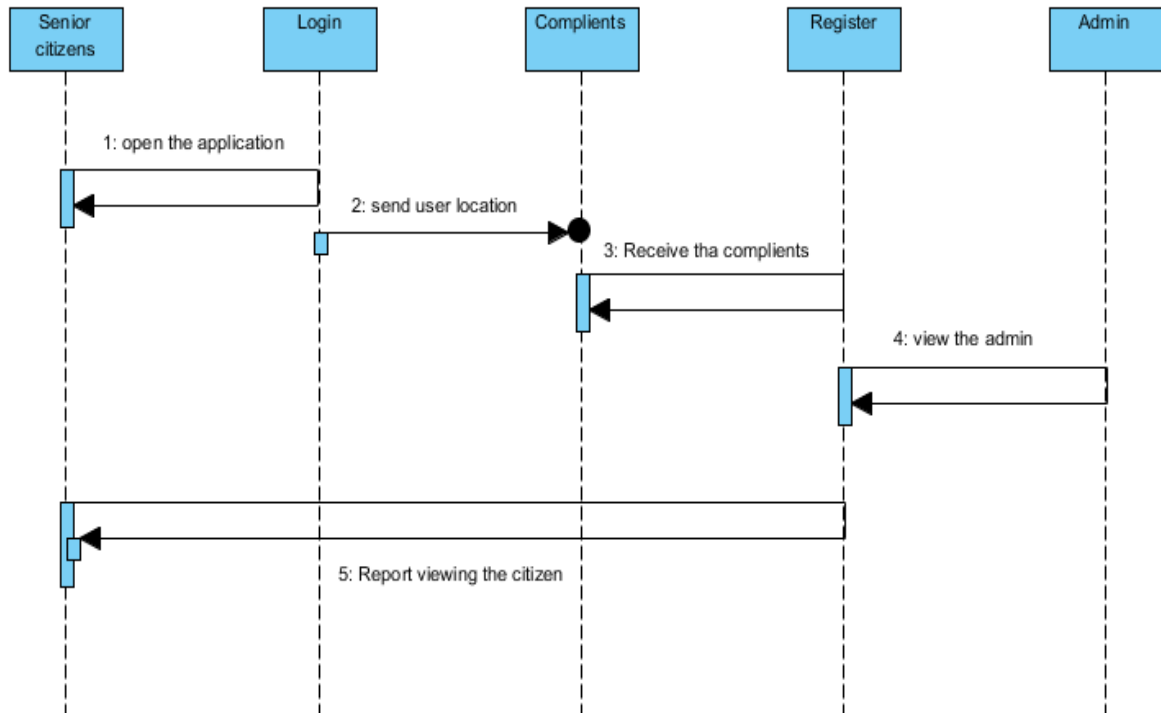


Figure 4.14:Sequence Diagram

## 4.2 Detailed Design

Software design is a process of problem solving and planning for a software solution. After the purpose and specification of software is determined, software developer will design or employ designers a plan for a solution. High level design gives an overview of the system flow. However, this describes a lot for the user to understand the logic. Here we see the basic knowledge about the system design and architecture.

## Chapter 5

# TESTING

Software testing is a process of executing a program or application with the intent of finding the software bugs. It can also be stated as the process of validating and verifying that a software program or application or product: Meets the business and technical requirements that guided its design and development.

In our testing phase we went through different testing like unit testing, integration testing, system testing and performance testing as described below:

### 5.1 Unit Testing:

Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. Unit testing is often automated but it can also be done manually.

Serial Number	Components Tested	Method used	Description	Status	Remarks
1.	Login Page	Unit Testing	Check if the login details are submitted to the server and if the server responds.	Pass	Verification takes 2-3 seconds
2.	Registration Page	Unit Testing	Check if the user is able to register to server by providing the required details.	Pass	User registers by proving the required details
3.	Forgot Password Page	Unit Testing	Check if the user is able to retrieve the forgot password from the server database by proving required details	Pass	Server send user's password to their register mobile/email
4.	Issue detail page	Unit Testing	Check if the user is able to retrieve the details of the complaint give to the concerned department	Pass	User is able to see full description of their complaint

5.	Take action page	Unit Testing	Check if the admin is able to take action on the complaint given by users	Pass	Admin is able to take action on the user's complaint
----	------------------	--------------	---	------	--

Table 5.1: Unit Testing

## 5.2 Integration Testing:

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing.

In this testing we are trying to test by combing the smallest module part and then test as a group.

Serial Number	Components Tested	Method used	Description	Status	Remarks
1.	Login Phase	Integration Testing	Check if the user is able to login or not	Pass	Server verifies the user and is able to Login
2.	Raise Issue Phase	Integration Testing	Check if the user is able to give a complaint to the concerned department	Pass	User is able to give complaint to concerned department
3.	Check Status Phase	Integration Testing	Check if the user is able to check their own complained status	Pass	User is able to check their own complained status given to the concerned department
4.	Credit Points Phase	Integration Testing	Check if the user is able to see the credit point given by the concerned department	Pass	User is able to see their credit points

Table 5.2: Integration Testing

### 5.3 System Testing:

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black-box testing, and as such, should require no knowledge of the inner design of the code or logic.

In this system testing we are trying to test each module separately.

<b>Serial Number</b>	<b>Components Tested</b>	<b>Method used</b>	<b>Description</b>	<b>Status</b>	<b>Remarks</b>
1.	Installation	System Testing	Check whether the application is installing as desired	Pass	Application is installing as desired
2.	Uninstallation	System Testing	Check whether the application is uninstalling without leaving any data behind	Pass	Application is uninstalling as desired
3.	Volunteers/Senior Citizens Application	System Testing	Check whether the Volunteers/Senior Citizen application is working as desired	Pass	Volunteers/Senior Citizen Application is working as desired
4.	Admin Application	System Testing	Check whether the Admin application is working as desired	Pass	Admin Application is working as desired
5.	Police Admin Application	System Testing	Check whether the Police admin application is working as desired	Pass	Police Admin Application is working as desired

Table 5.3: System Testing

### 5.4 Performance Testing:

Testing the stability and response time of the application by applying a load to it is called performance testing.

## Chapter 6

# CONCLUSION AND FUTURE ENHANCEMENT

## 6.1 Conclusion

Simple Smart Connected City App, an Android App will help in effective communication with government departments and reduce the waiting time and improve the lifestyle. Traditional methods of huge data collection are reduced and usage of eco-friendly digital technology will create Smart Cities. Day-to-day problems will be communicated to the right departments and improve the turnaround time.

## 6.2 Future Enhancement

- Improving the admin user interface.
- Deploying for an amazon server.
- Enhance more Emergency services like by proving near hospital number in case of accident.
- Can include feature like SOS, News Updates.
- Proving simple and easy to understand interface for the senior citizens.

## REFERENCES

- [1] Giffinger, R., Fertner, C., Kramar, H., Kalasek, R., Pichler-Milanović, N., & Meijers, E., "Smart Cities: Ranking of European Medium-Sized Cities. Vienna, Austria: Centre of Regional Science (SRF), Vienna University of Technology, (2007).
- [2.] Faisal Razzak, "Spamming the Internet of Things: A Possibility and its probable Solution", The 9th International Conference on Mobile Web information Systems, Procedia Computer Science 10(2012) 658-665.
- [3.] Hafedh Chourabi, Taewoo Nam, Shawn Walker, J. Ramon Gil-Garcia, Sehl Mellouli, Karine Nahon, Theresa A. Pardo, Hans Jochen Scholl, "Understanding Smart Cities : An Integrative Framework", 45th Hawaii International Conference on System Sciences (2012).
- [4.] Framework Agreement, kerala.gov.in, "Agreement reached on Smart City project".Thehindu.com (2011-02-02). Retrieved on 2012-09-12.
- [5.] <http://www.encyclopedia.com/searchresults.aspx?q=software%2btesting>
- [6.] <http://www.smartcitiesprojects.com/>
- [7.] <http://india.smartcitiescouncil.com/article/smart-city-apps-urban-challenges>
- [8.] <http://cis-india.org/internet-governance/blog/sustainable-smart-cities-india-conference-2015-bangalore>