Public Grievance Addressal System (PGAS)

Software Requirement Specification - Ver.1.0

Prepared by Group 7

Team Members:

1. A.Dinesh Reddy
2. S.Sunil Kumar
3. M.Bharath
4. D.Pavan
5. R.Vamshi Krishna
6. K.V.Sai Vineeth
7. K.Shanmukh
8. Nagaraju

Prepared as part of:

Information Technology Systems Course

[Date: 17th Feb, 2017]

Table of Contents

1 Introduction

1.1 Overview

Introduction: Provides overall glimpse in the system and its modules, describes the details and individual objectives of each subsystem.

Overall description: Provides the specifications of the overall system, class models,sequence model

Specific Requirements: Analysing the requirements by feature, visualisation and requirements that are related with software,networking and other constraints.

1.2 Purpose

The purpose of the document is to specify the details and the overall view of the PGAS system .Moreover this document acts as a reference manual for future users,developers and authoritative representatives of the basic flow of the system.

1.3 Scope of the project

The PGAS system offers an easy and scalable solution for the growing urban citizens' problems and a compact way of managing the problems for the municipal authorities. Any citizen with a minimal knowledge of using a smartphone can easily file a complaint/grievance using the android application. The maintenance activity is recorded in the PGAS system which also gives the user a view of the involvement of the government authority regarding a particular problem/grievance. Since this system involves both citizens and governments, it involves active participants in the system and thus ensures transparency in government organisations.

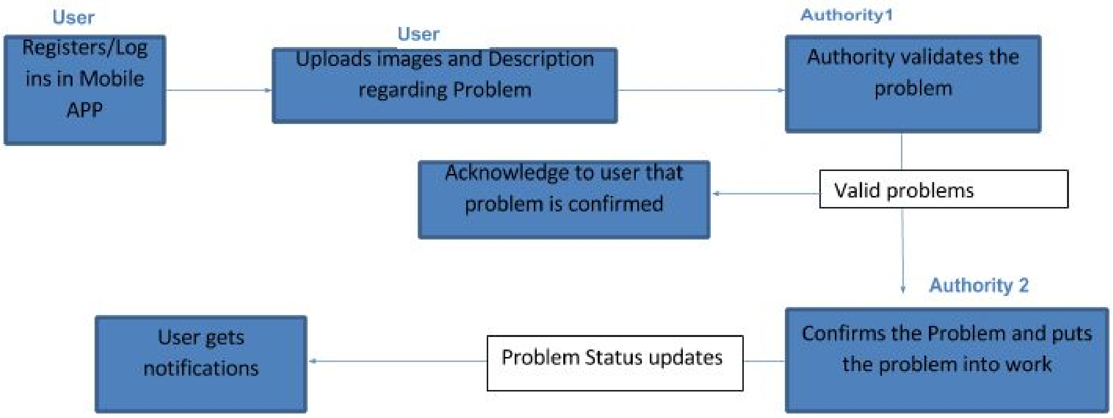
1.4 Intended Audience

This document is intended for any individual user,developer,tester,project manager or documentation writer that needs to understand the basic system architecture and its specifications.Here are the potential uses for each one of the reader types:

1. Developer : The developer who wants to read,change,modify or add new requirements into the existing program,must firstly consult this document and update the requirements with appropriate manner so as not to destroy the actual meaning of the and pass the information correctly to the next phases of the development process.
2. User : The user of this program reviews the diagrams and the specifications presented in this document and determines if the software has all the suitable requirements and if the software developer has implemented all of them.
3. Tester : The tester needs this document to validate that the initial requirements of this programs actually corresponds to the executable program correctly.

2.1

2.1 Product Perspective


User
Registers/Log ins in Mobile APP
User
Uploads images and Description regarding Problem
Authority1
Authority validates the problem
Valid problems
Acknowledge to user that problem is confirmed
Authority 2
Confirms the Problem and puts the problem into work
Problem Status updates
User gets notifications

Figure(a): Public Grievance Process flow

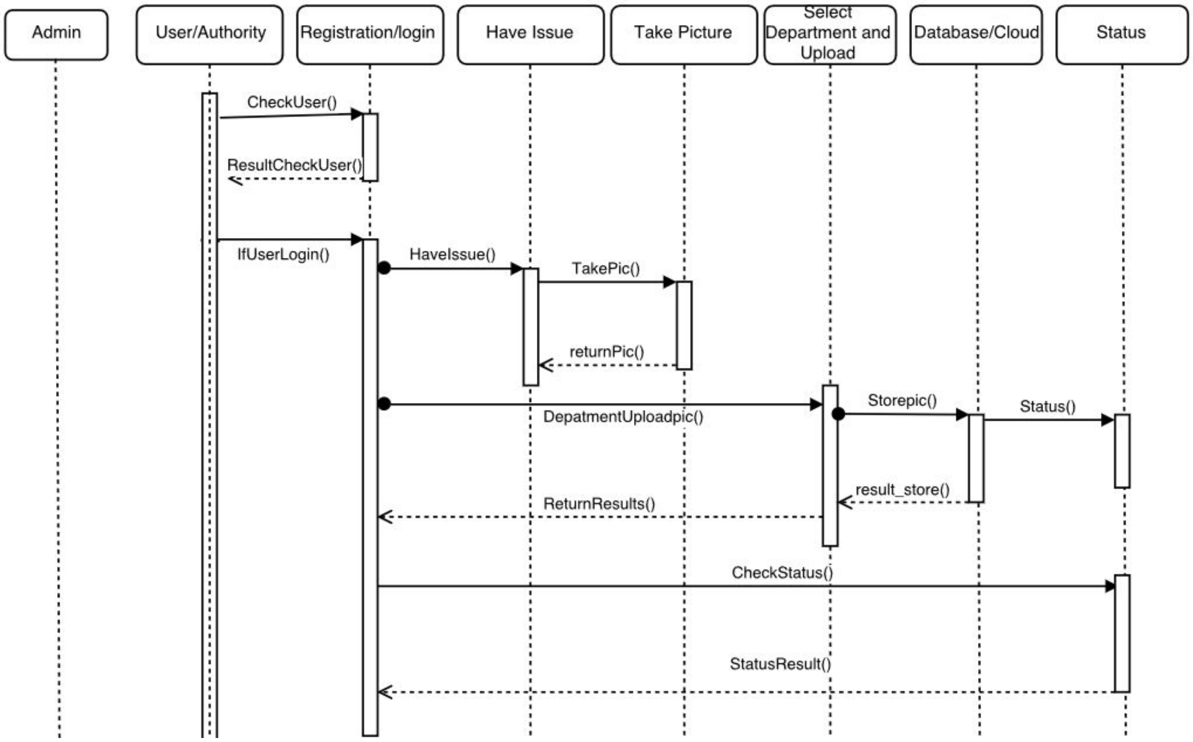
2.1.1 System Interfaces

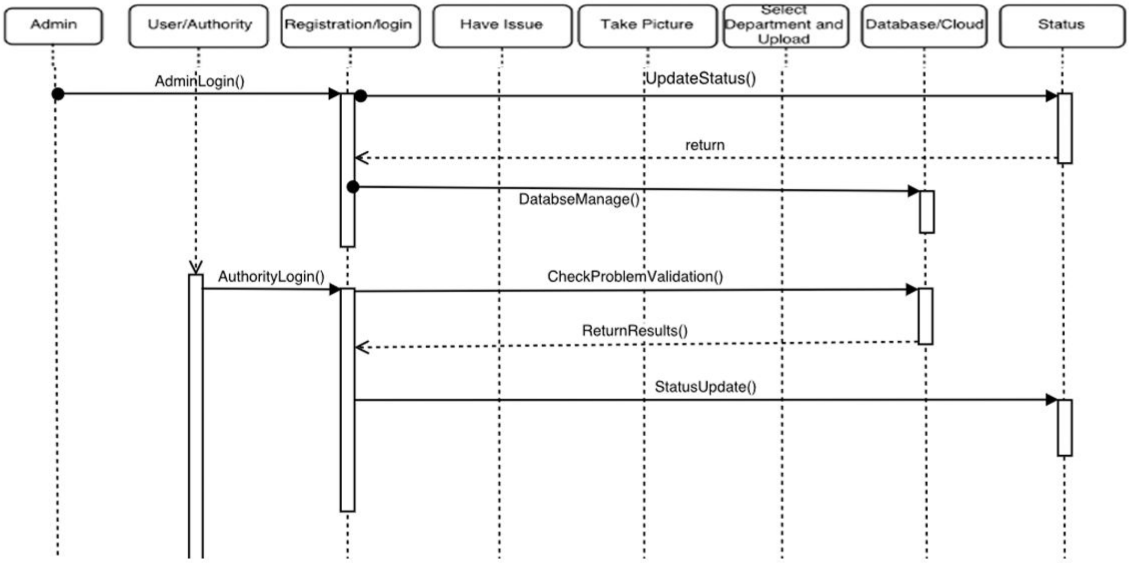
For web Portal and Android application central cloud database server is used. That is Google cloud server.

2.1.2 User Interfaces

- The user interface for the software shall be compatible to any browser such as Internet Explorer, Mozilla or Netscape Navigator by which user can access to the system.
- Simple and easy to understand
- Attractive GUI
- Easy Registration
- Simple complaint procedure

BASIC TRANSACTION FLOW OF PUBLIC GRIEVANCE





2.1.3 Software Interfaces

- For synchronization of web portal and android app there must be a external central database server.
- The communication between the database and the web portal consists of operation concerning both reading and modifying the data, while the communication between the database and the mobile application consists of only reading operations.
- To know the location of complaint, it is needed to use GIS in mobile application to get longitude and latitude.

2.1.4 Communication Interfaces

● Communication with the database is through ODBC connections

● General web application use HTTP /1.0 & HTTP/1.1 protocols which are

both TCP connections for server client connectivity.

2.1.5 Memory Constraints

Since this is a web and android application available online, there are no specific memory constraints for the device on which is being used. But in this case there is a minimum requirement of 100TB cloud storage to handle the data of citizens. As the number goes up, the cloud storage size should also be increased.

2.1.6 Operations

● Modes of Operation: Can be operated on web and mobile web browsers

● Interactive Operations: Almost all the user operation like loading a complaint, uploading the images and checking the complaint status are the interactive operations.

● Unattended Operations: System's ability to recover from system errors, ability of the system to auto save the entered data are the unattended operations.

● Backup & recovery: If the system incurs a problem, the system must be able to backup the entire data available till the time before the error occurred and hence must be able to recover automatically without any human intervention.

2.1.7 Site Adaptation Requirements

● New data tables created for this system must be installed on the company's existing DB server and populated prior to system activation.

● User Interface must be available in English.

● Any changes done in the system must be updated in the SRS and FRS before actually launching the system along with the brief description of the change and also the version number, author name.

2.2 User Characteristics

Physical Actors

Admin (Technical In charge):

- Creates Authorities.
- Handles the data in server.
- Checks and resolves errors in system.

Authority (Govt. officer):

- Validates problems of users.
- Prioritizes the problems.
- Updates the problem status of user.

User (Public):

- Registers and creates a account for himself/herself.
- Lodges a complaint by uploading images and description about the project.
- Checks the status of the complaints uploaded by him/her.

System Actors:

Client

● All the users of the systems are the clients

● These must be able to connect to the server (safely establish live network connection between client & server)

● Must be able to generate/download refined results based on user filters

Server

● Accept all the client connections

- Setup the logged in session accurately
- Maintain the download history with the corresponding user login

Database Editor/Manager

- Must offer all the types of filters to the users in order to obtain the required and expected information
- Must offer complete set of tools to manipulate and store the data into the database tables and fields

2.3 Operating Environment

This program will operate in the following operating environment for the client and the server

GUI:

- Apple Mac OS X (Safari, Chrome & Mozilla are available)
- Linux/Unix (Mozilla is available)
- Microsoft Windows (Internet Explorer, Chrome & Mozilla are available)

Mobile Platforms:

- Android Application
- Android (Chrome available)
- Apple IOS (Safari available)
- Windows Phone (Internet explorer available)

Browser:

- Internet Explorer (ver. 9 and above)
- Google Chrome (ver. 42 and above)
- Mozilla Firefox (ver. 38 and above)
- Safari (ver. 7 and above)

2.4 Design & Implementation Constraints

The PGAS being developed must have both web browser version and an equally compatible mobile browser version. The program must be developed using python, Java (Highly Scalable Application

Architecture). For better User experience, the device used by the user must have an internet speed of at least 512Kbps. The web page must be css customisable (for example, the user must be able to select his language and from then, the instructions must be available in the user opted language, the user must be able to customize the size of the text (tiny, small, normal, large, huge) and the background colour (black, white etc. for better visibility). The system must be able to handle large no.of user database (almost 5 lac farmer details and season wise record maintenance). Under internet connection failures, the system must be able to retain the information which is not saved but entered till the loss of internet connectivity. Any internet connectivity issues must logout the user account on security grounds. To handle large populated citizen`s data and season wise reports, there must be an availability of 100TB cloud memory initially and expansion whenever required.

2.5 Assumptions & Dependencies

In the entire PGAS, one of the main goals is to track the complaint status. For this, the respective authority must update the status. Other dependencies are complaint will be on work if only validation authority officer validates the complaint.

3 Specific Requirements

3.1 Functional Requirements

3.1.1 Citizen requirements

1. Register with correct Email and Aadhaar and other credentials.
2. After successful registration he can login.
3. View Problems of other Users.
4. Recommend & Claim for incidental charges/each bag produced.
5. See his personal problems uploaded by him in the Android app.
6. Track the problems of his own based on the token number.
7. Check the history of all the problems which were completed.
8. He is notified when the status of the problem is validated/prioritized followed by updation by the authority.

3.1.2 Employee requirements

1. Register with correct Email and Aadhaar and other credentials.
2. After successful registration he can login.
3. Verify the problems that are assigned to him are valid or not.
4. Updating the status of the problem once done at every stage.
5. Assigning Priority to the problems based on Importance of the problem.
6. If the Problem is invalid he can delete it immediately by verifying with the location he get from the problem.

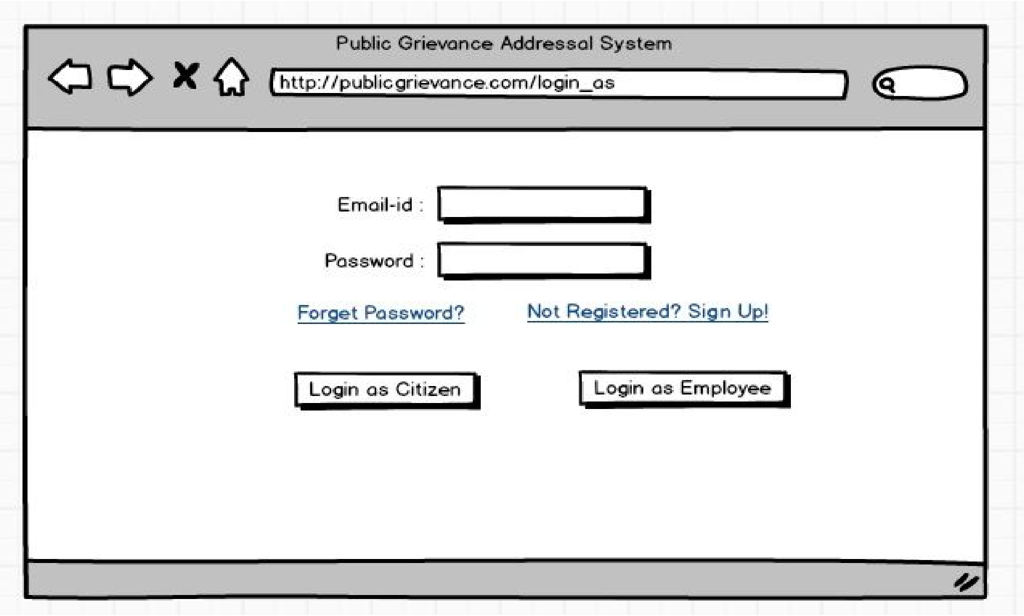
3.1.3 Admin requirements

1. Register/Create new farmer profile (Farmer belongs to the area of that MVK)
2. Allocation of problems to employees which are uploaded by Citizens.
3. Backup the data to ensure no loss.
4. Check whether authorized employees or not.
5. He validates the registration Process for Employees
6. Secured transactions between farmers & MVK

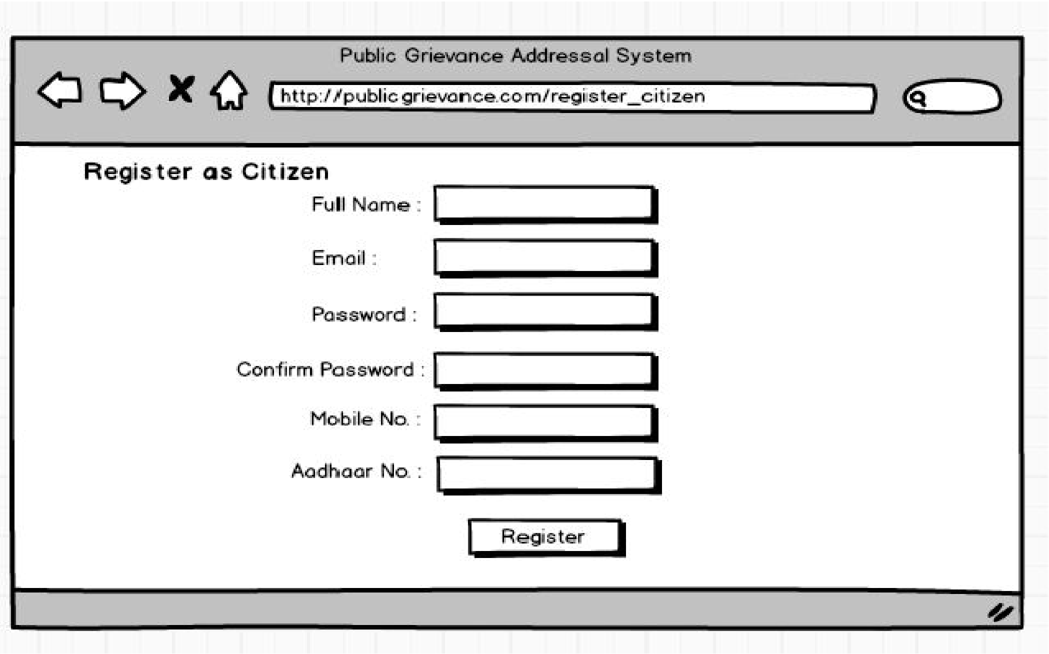
7.The administrator maintains the list of the dtaa of the problems solved and is the person who provides detailed analytical report of the data based on various aspects that can be used both on user point of view and the authority point of view.

3.2 External Interface Requirements

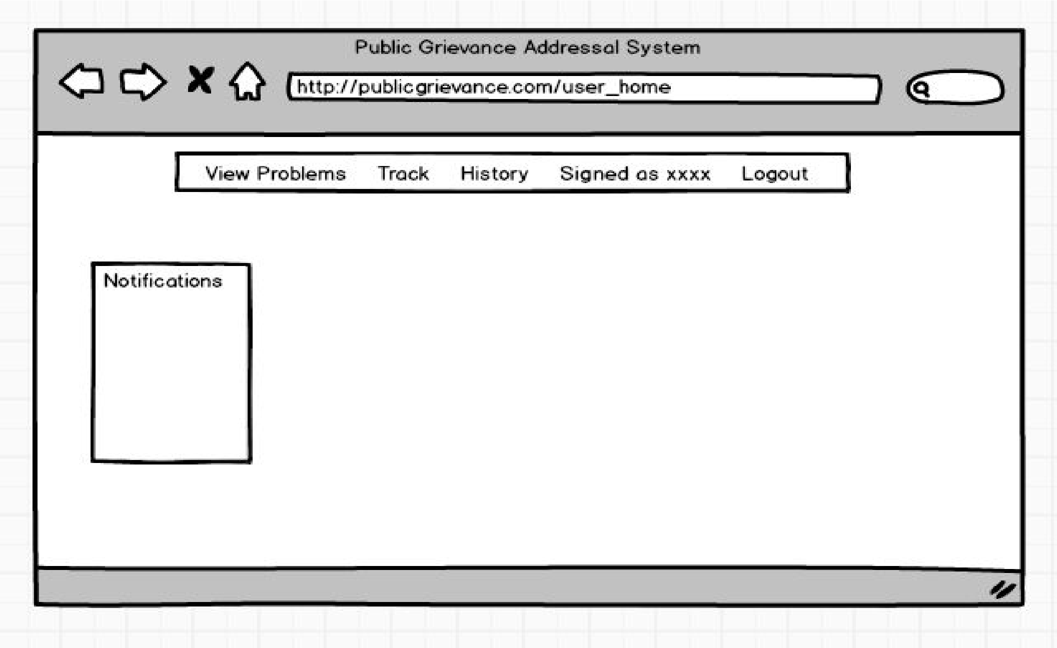
3.2.1 User Interface



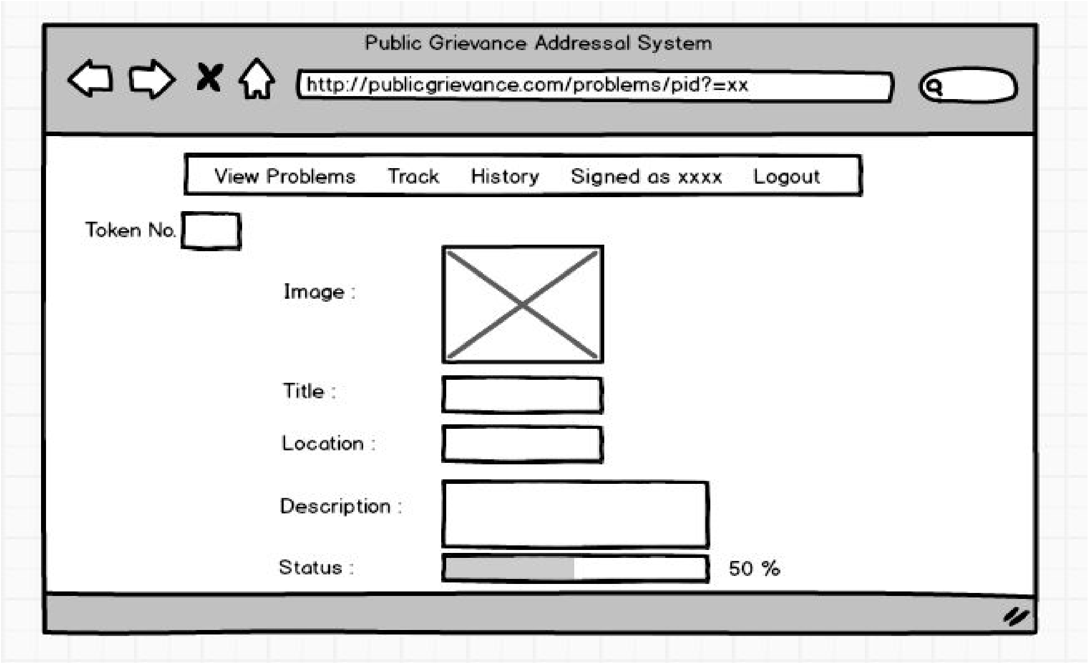
Login as viewed by the user



Register page as viewed by the user


Public Grievance Addressal System
http://publicgrievance.com/user_home
View Problems
Track
History
Signed as xxxx
Logout
Notifications

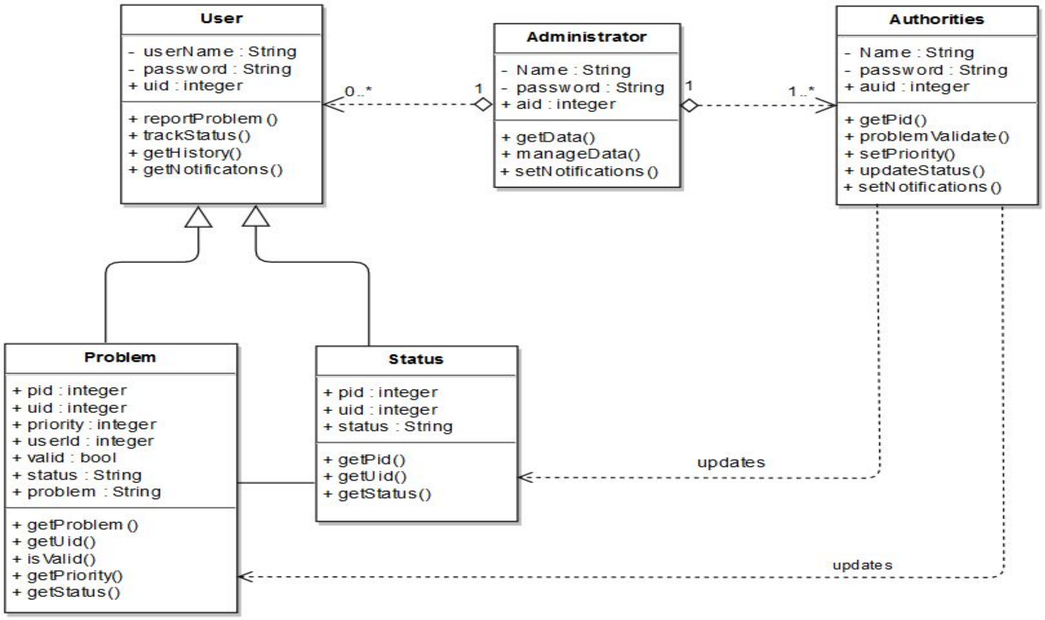
Home page as viewed by the user
after login



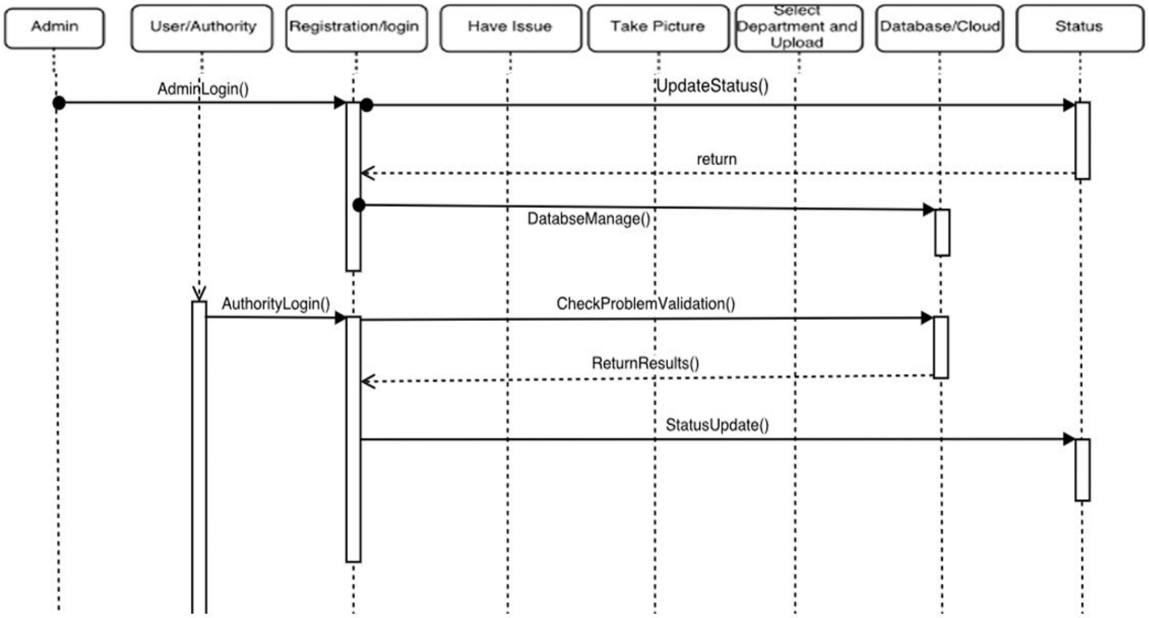
Tracking page as seen by the user

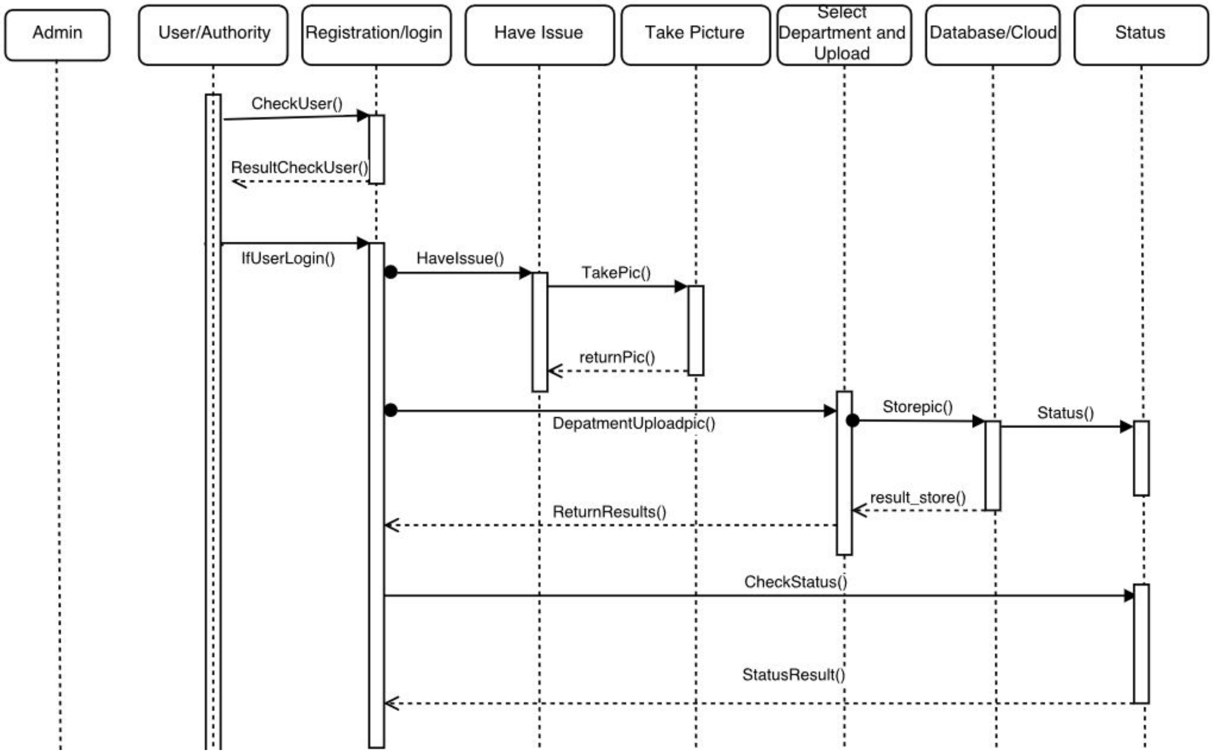
3.3 Specific Features

3.3.1 Class Diagram



3.3.2 Sequence Diagram





3.4 Other Non Functional Requirements

1.Performance requirements:

(i) The app should be able to support 1,000 users simultaneously.

(ii) It should be able to handle the 99% users complaints at the same time and run them in fraction of seconds.

(iii) Cloud should receive every user details and complaints details and store them in database without any errors.

(vi)Cloud should receive huge data and storing should happen in less than 1 sec.

2.Security Requirements:

(i) Cloud data should be protected under the password and limited number of systems should access cloud data.

(ii) Authority employees should be registered only by the admin and limited authorities should be present.

(iii) There must be validation of authority and the citizen.If the authority doesn't exists ,it should be allotted to solve any problem uploaded by the citizen.

(iv) Verification of the registered users(citizens) is necessary.

(v) Validation of problem is necessary.If the problem is found to be invalid ,the problem should be rejected and should not stored in the database.

(vi)There should not be any anonymous complaint upload.

(vii) Duplicate problems should not be present in the database.

3.Maintainability requirements:

(i)Maintain proper databases of the problems uploaded for providing the details like problem progress,authority handling the problem and for sending notifications of the problem updates to the citizens.

(ii) Data backup is necessary and backup will be taken once in a day and store in a offline system.In case of any failures data will be recovered.

(iii)Problem validation and problem solving by the authority will be operated from 10:00 AM to 4:00 PM in week days.

4.Software Quality Attributes:

(i) The system should be easy to upload complaints and the interface should be understandable.

(ii)The user documentation should be complete and it should be able to tell the type of problems that can be uploaded.

(iii) Error message should explain how to recover from the error.

(iv) A style guide should be provided.

(v)The purpose of the system should be meaningful.

(vi)The system should be available from 4:00 AM to 11:59 PM.

(vii) After three successive invalid upload of problems,the citizen will be blocked for 1 week.

APPENDICES

Appendix A. Database Design

Records to be maintained for Citizen:

(i)Citizen Profile

<<Name of the Citizen>> *string data* *max 32 characters*
<<User id>>-*Integer* Auto incremented and unique for each user.
<<Email>> *string data
<<Password>> -*string*
<<Aadhaar Number>> *integer* in 12 digits
<<Phone Number>> *integer* in 10 digits

(ii) Pictures Uploaded By citizens

<<Token number>>-*String data*
<<User id>>-*Integer* User Id point to which user has created the picture.
<<Blob Data>>-*Medium Blob* Encoded Format of the Picture
<<Title>>-*String data* Title of the Picture
<<Description>>-*string data* Brief Description About Picture
<<Formation date>>-*DD/MM/YYYY*
<<Address>>-*String data* Based on the latitude and longitude address that we get.

(iii)Status of the Problem of the Citizen(Picture Current Status)

<<Token Number>> *string data*
<<Status>> *Integer*
<< Last Updated>> *DD/MM/YYYY*

Records to be maintained For Employees

(i)Employee Profile

<<Name of the Employee>> *string data* *max 32 characters*

<<Employee id>>-*Integer* Auto incremented and unique for each user.

<<Email>> *string data*

<<Password>> -*string*

<<Department id>>-*Integer*

(ii)Employee Options

<<Token no>> *string data*

<<Formation Date>>-*DD/MM/YYYY*