

Public Grievance Addressal System (PGAS)

Functional 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 Executive Summary

1.1 Project Overview

1.2 Purpose and Scope of this Specification

2 Product/Service Description

2.1 Product Context

2.2 Actors & their tasks/roles

2.3 Assumptions

3 Requirements

3.1 Functional Requirements

3.2 Usability

3.3 Performance

3.4 Manageability/Maintainability

3.4.1 Maintenance

3.4.2 Operations

3.5 Data Management

3.6 Security

3.6.1 Protection

3.6.2 Authorization and Authentication

Appendices

Appendix A. Non Functional Requirements

Appendix B. Mockup Screens for the system

1.Executive summary

1.1 Project Overview

Public Grievance is an initiative which provide an environment to all the citizens of society to give a complaint about any problem like road damage , blocking of roads, garbage on the road and other problems.The whole process of solving the problems is managed by the admin who contacts the authority to solve the problem.According to this project the common problems in the society in a particular region can be solved by the respective authority of that region by receiving a single complaint from a single citizen.

The main goal of the program is to solve the common problems in the society to achieve a user friendly environment.Hence in cases of rainfall, storms,earthquakes etc., public grievance will take part to solve the problems caused.

1.2 Purpose and scope of this specification

There are large number of backward areas in India.In those areas there is no neatness and standard of living is very low.The people in those areas are suffering from lot of problems.There are some down areas where the water in the rivers are flowing through those areas and disturbing the environment .During rainy season because of storms, because of earthquakes, because of heavy rains ,the whole environment will get disturbed.The authorities will not take care of any problems until and unless a complaint was given. So citizens can take a picture of the problem and they can upload it in the public grievance app which takes care of solving the problems.

The biggest challenge is to make the authority to solve the problems uploaded by the citizens and should ensure that the problem is solved .Another challenge is that there should not be any duplicate or invalid problems uploaded by the citizen.

SCOPE

In Scope:

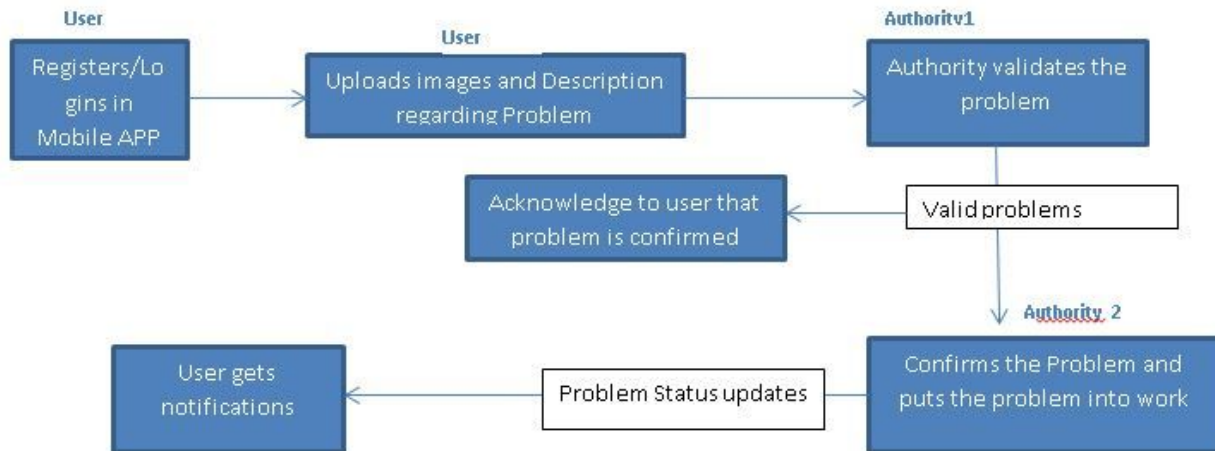
- Establishment of a authority to solve a problem
- Coding to the approved wireframe
- Graphics development for the website theme
- Testing and debugging prior to making the site public
- User Interface design

2. Product/Service Description

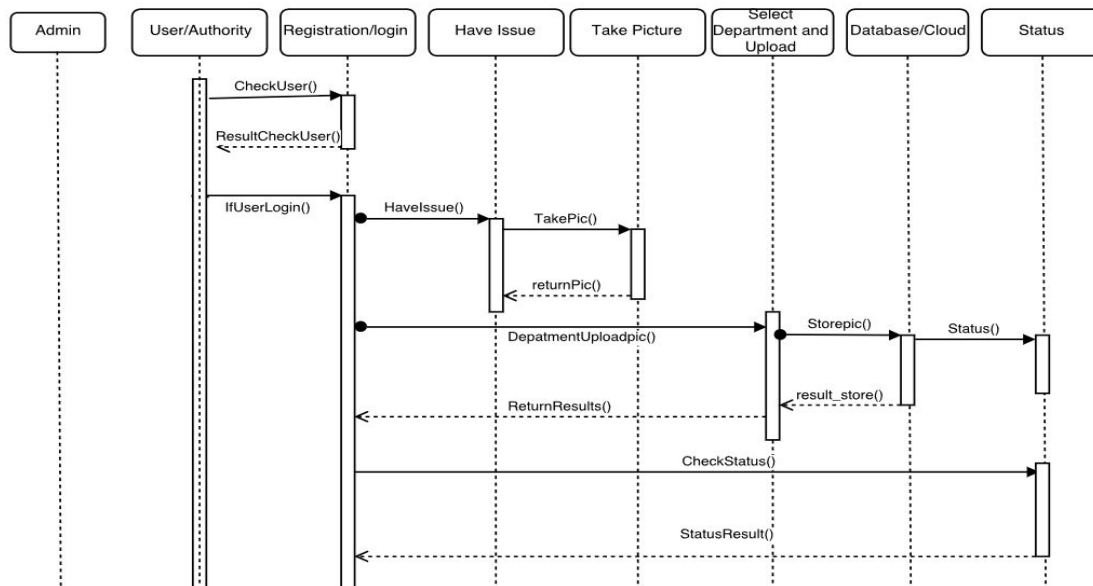
The PGAS system offers an easy and scalable solution for the growing urban citizens' problems and a compact way of managing the problems for the municipal authorities. Any citizen with a minimal knowledge of using a smartphone can easily file a

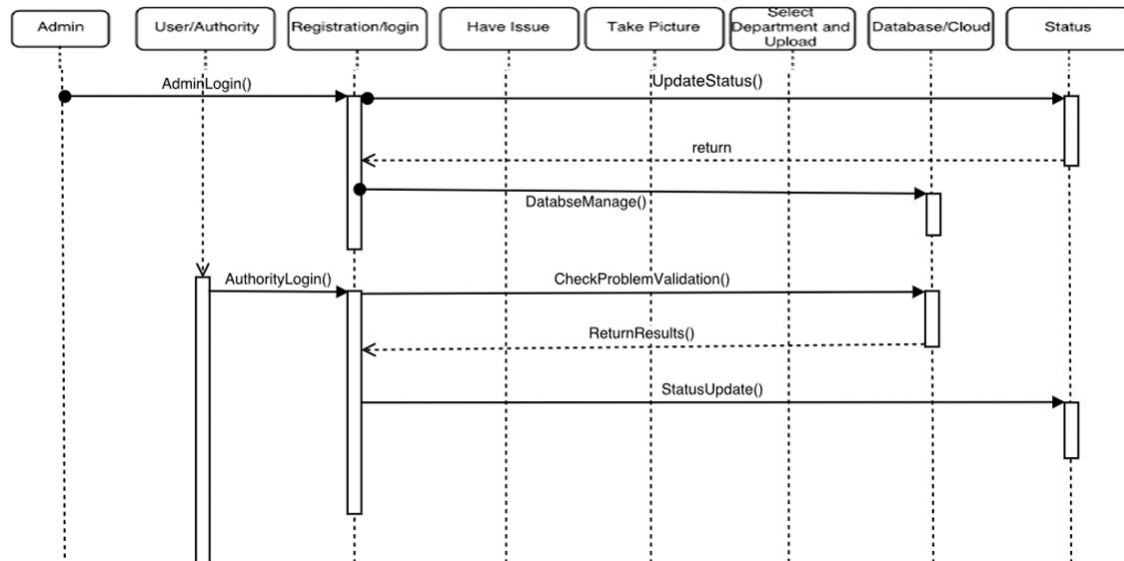
complaint/grievance using the android application. The maintenance activity is recorded in the PGAS system which also gives the user a view of the involvement of the government authority regarding a particular problem/grievance. Since this system involves both citizens and governments, it involves active participants in the system and thus ensures transparency in government organisations.

2.1 Product Context



BASIC TRANSACTION FLOW OF PUBLIC GRIEVANCE





2.2 Actors & their tasks/roles

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.

2.3 Assumptions

- All users should have basic knowledge of using mobile applications and web portal..
- Authority validates problems genuinely.

3. 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. Allocation of problems to employees which are uploaded by Citizens.
2. Backup the data to ensure no loss.
3. Check whether authorized employees or not.
4. He validates the registration Process for Employees
5. Secured transactions between farmers & MVK
6. The administrator maintains the list of the data 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. Usability:

- (i) Error message should explain how to recover from the error.
- (ii) A style guide should be provided.
- (iii) The purpose of the system should be meaningful.
- (iv) The system should be available from 4:00 AM to 11:59 PM.
- (v) After three successive invalid upload of problems, the citizen will be blocked for 1 week.

3.3.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.
- (iv) Cloud should receive huge data and storing should happen in less than 1 sec.

3.4 Manageability/Maintainability

3.4.1 Maintenance

Maintain proper data bases of the transactions done for every season for claiming the incidental charges and also for sending the receipts for clearance of the subsidy amount by the government.

3.4.2 Operations

- (i) Periods of interactive operations should be between 10:00 AM to 5:30 PM during Monday to Saturday, and periods of unattended operations are the timing left unmentioned above.
- (ii) Backup and recovery operations .There must be a daily backup stored into an offline system and under any error situations, the system must recover without human intervention and function as expected.

3.5 Data Management

Records to be maintained for Citizen:

(i)Citizen Profile

<<Name of the Citizen>> *string data* *max 32 characters*
<<User id>>-*Integer*àAuto 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* à User Id point to which user has created the picture.
<<Blob Data>>-*Medium Blob*àEncoded Format of the Picture

<<Title>>-*String data*àTitle of the Picture
<<Description>>-*string data*àBrief Description About Picture
<<Formation date>>-*DD/MM/YYYY*
<<Address>>-*String data*àBased 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*àAuto 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*

3.6 Security

3.6.1 Protection

- (i) The system must be password protected with different user privileges
- (ii) All the receipts of transactions must be displayed with encryption to undesired users.
- (iii)Activity logging, historical data must be available to avoid misuse
- (iv)There must be restrictions on inter module communications
- (v)Regular data integrity checks must be made

3.6.2 Authorization and Authentication

(i) The system shall authenticate all of its users before allowing them to update their user information

(ii) When the system detects the use of forged authentication data, then the system shall block the user's privilege and auto generate a mail to the authorities in this regard

(iii) The data center shall verify the identity of all personnel before permitting them to enter.

Appendices

Appendix A. 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 exist, 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 be 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:

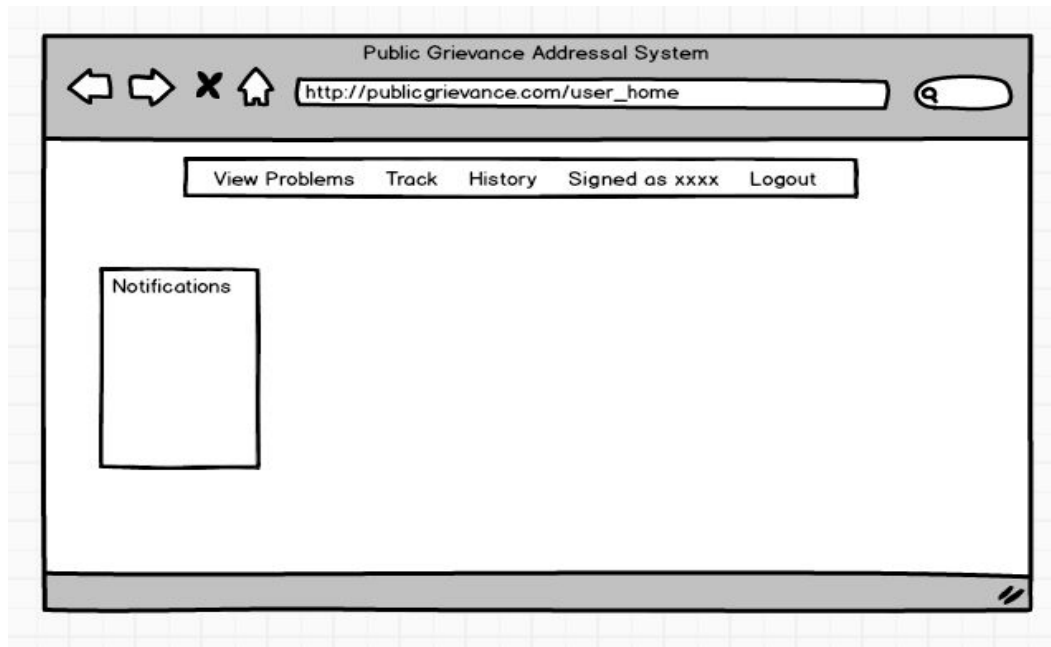
- (i) The system should be easy to upload complaints and the interface should be understandable.
- (ii)The user documentation should be complete and it should be able to tell the type of problems that can be uploaded.
- (iii) Error message should explain how to recover from the error.
- (iv) A style 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.

Appendix B. Mockup Screens for the system

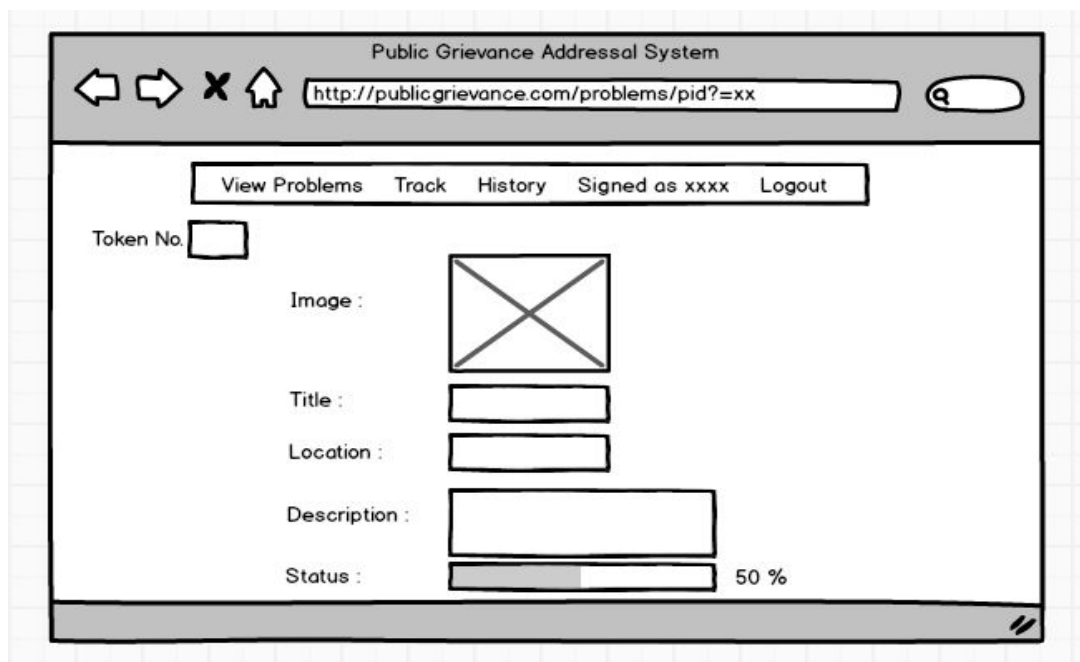
The mockup shows a web browser window titled "Public Grievance Addressal System". The address bar contains "http://publicgrievance.com/login_as". The main content area has the following elements:

- Email-id :
- Password :
- [Forget Password?](#) [Not Registered? Sign Up!](#)
-

Login as viewed by the user



Home page as viewed by the user after login



Tracking page as seen by the user