

# IT314: Software Engineering

Prof. Saurabh Tiwari

## Project Title: Crowd-Powered Smart Complaint Management System

### Contributions (Group 6)

Name	Student ID
JAINIL SHAILESH JAGTAP	202301032
SHAMIT GANDHI	202301041
MEHTA DHIRUVIL VIMALKUMAR (LEADER)	202301061
BHATT PARTH BHASKARBHAI	202301022
OM KANTILAL SANTOKI	202301019
NEEV VEGADA	202301031
TIRTH KORADIYA	202301018
KARAN MAKASANA	202301053
RASHA PARMAR	202301012
SAMARTH AGARWAL	202301040



## **Jainil Shailesh Jagtap (202301032)**

### **Technical (Coding):**

- Created the initial User Model.
- Migrated the entire auth workflow to JWT authentication with Redis + PostgreSQL setup for caching as well as persistence.
- Static analysis + Unit testing of Users app (Auth workflow) achieving 97% coverage.
- Implemented Resolution by Field Worker and Approval of Resolution by Citizen backend logic.
- Backend logic for Citizen Profile (afterwards refactored to general profile) and Password Change for Citizen.
- Implemented AI pipeline for Severity Analysis and Time Prediction using Computer Vision + GenAI.
- Addition of AI Department Suggestion feature as a secondary feature.
- Optimized the performance of the complaints and image-fetching APIs, reducing average response time from about 20s to 2s.
- Mutation Testing of Users and Notifications App achieving 100% mutation score.

### **Non-Technical:**

- Interviewed a Government Official and a Field Worker from Municipal Corporation for requirement gathering.
- Documented Sprint 5 (AI pipeline for time prediction) using Sequence and Class Diagram.
- Documented Elicitation Techniques, Functional Requirements and User Stories for Government Authority, and Proof of Concept for Sprint 1 (Before Mid-Evaluation).

## **Shamit Gandhi (202301041)**

### **Technical (Coding):**

- Initial sign-up, sign-in and login views for earlier template-based version.
- Tested user creation and deletion using Django tests for first sprint.
- Verify and send OTP functionality in Backend.
- Backend functionality of posting, viewing (as in a feed), and deleting complaints for citizens.
- Added backend functionality of taking GPS coordinates from device and finding address using geolocation.
- Created admin user functionality using Django admin (via superuser) and provided access to manage all given apps.
- DB hosted on Railway and Image hosting on Cloudinary.
- Backend functionality for search complaints.
- Backend for Field Worker home page.
- Generalised profile backend for all users; generalised My Complaints (past complaints) view for all users.
- CSRF token to ensure that requests modifying server data originate from the user's legitimate session.
- Backend unit testing until sprint 3 for complaint and its resolution.
- To reduce joins while calling certain functionalities, added user\_type in the ParentUser model.
- Backend functionality for government approval of reported fake complaints by citizens.

### **Non-Technical:**

- Survey creation and conduct.
- Elicitation Techniques for Functional Requirements of Citizens.
- Functional Requirements of Citizens.
- User Stories of Citizens.
- Epics and conflicts between them.

- Class diagrams for Sprint 1, 2, 3, 4 and 6.

## **Mehta Dhruvil Vimalkumar (202301061)**

### **Technical (Coding):**

- Initial sign-up, sign-in and login forms for earlier template-based version.
- Added serializers and URL routing for backend navigation and provided a ListView for accessing available government departments.
- Implemented backend functionality for signup OTP validation.
- Designed and created the complaints models for the system.
- Implemented forgot-password and reset-password functionality in the backend.
- Developed backend functionality for viewing past complaints for all users.
- Implemented backend functionality for the Government Home Page.
- Added backend functionality for generating and returning fake confidence values.
- Implemented detailed complaint view, anonymous complaint creation, anonymous complaint display, and notification-based redirection to complaint details, along with minor improvements.
- Backend unit testing until sprint 4 for complaint and its resolution.

### **Non-Technical:**

- Survey creation and conduct.
- Elicitation Techniques for Functional Requirements for Citizens.
- Functional Requirements for Citizens.
- User Stories for Citizens.
- Epics and conflicts between them.



## **Bhatt Parth Bhaskarbhai (202301022)**

### **Technical:**

- FIGMA design.
- Trending complaints & field worker leaderboard.
- Upvote and report functionality.
- Added AI and GPS part in raise complaint for frontend.
- Field Worker resolution submission.
- Sequence and Activity diagram.
- Help page frontend for all users.

### **Non-Technical:**

- Survey creation and conduct.
- Stakeholders and End Users.
- Elicitation Techniques for Functional Requirements for Citizens.
- Functional Requirements for Citizens.
- User Stories for Citizens.

## **Om Kantilal Santoki (202301019)**

### **Technical:**

- Black Box testing.
- User Acceptance Testing.
- Load testing.
- Unit testing for frontend.
- Integrated React rerouting on frontend for client-side URL routing for Sprint 1.



### **Non-Technical:**

- Elicitation Techniques for Functional Requirements of System Admin.
- Non-Functional Requirements.
- User Stories for System Admin.

## **Neev Vegada (202301031)**

### **Technical:**


- Frontend UI for GPS coordinates in raise-complaint modal.
- Tailwind styling for government pages and navigation.
- React frontend and CSS setup for main interfaces.
- File upload functionality in frontend.
- New frontend layout and improved routing.
- App component update based on user type.
- Reset & forget-password frontend.
- Government authentication home page (frontend + backend connect).
- Field Worker Home Page (frontend).

### **Non-Technical:**

- Elicitation Techniques for Functional Requirements of Field Workers.
- Functional Requirements of Field Workers.
- Non-Functional Requirements.
- User Stories for Field Workers.

## **Tirth Koradiya (202301018)**

### **Technical:**

- 
- Complaint detail view frontend.
  - Improved government auth home page (blur UI).
  - Home page with Tailwind CSS.
  - Forgot password frontend
  - Assigned Fieldworker functionality frontend
  - Improved notifications frontend.
  - Anonymous complaint submission frontend.
  - Unit testing of selected frontend pages.

#### **Non-Technical:**

- Elicitation Techniques for Functional Requirements of Field Workers.
- Functional Requirements of Field Workers.
- User Stories for Field Workers.


### **Karan Makasana (202301053)**

#### **Technical:**

- Improved UI with general enhancements.
- Tailwind changes to signup forms and corrected routing.
- Updated homepage UI/UX and matched with new JWT authentication.
- Frontend for field-worker assignment functionality.
- Added search functionality (frontend) with multiple image upload and display.
- Past complaints frontend for citizens.
- Added a profile page with an option for password-change request.
- Minor UI changes.
- Sorting and filtering for frontend along with login error improvement.
- Mutation testing for Complaints app.

#### **Non-Technical:**

- Elicitation Techniques for Functional Requirements of Field Workers.

- 
- Functional Requirements of Field Workers.
  - User Stories for Field Workers.

## **Rasha Parmar (202301012)**

### **Technical:**

- Static home page for citizens for the first sprint.
- GUI Testing.

### **Non-Technical:**

- Survey creation and conduct.
- Elicitation Techniques for Functional Requirements for Citizens.
- Functional Requirements for Citizens.
- User Stories for Citizens.

## **Samarth Agarwal (202301040)**

### **Technical:**

- Developed a responsive and visually appealing home page using React.js and Tailwind CSS.

### **Non-Technical:**

- Elicitation Techniques for Functional Requirements of Tech Support.