Bilkent University

Department of Computer Engineering

Senior Design Project

reporTown

Project Specifications Report

Arda Akça Büyük, Elif Özer, Cemre Biltekin, Oğuz Kaan İmamoğlu, Mustafa Yaşar

Supervisor: Dr. Ayşegül Dündar

Jury Members: Dr. Shervin Arashloo, Dr. Hamdi Dibeklioğlu

Innovation Expert: Mehmet Çakır

Oct 11, 2021

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

Table of Contents

1. Introduction

Cities are home to many problems that can victimize and even sometimes endanger the citizens living in them. Road problems, garbage problems, transportation problems can be given as examples of those problems. When citizens encounter such problems, most of the time, they cannot do anything individually. Therefore, these problems can only be resolved by institutions like the municipality, governorship, or non-governmental organizations. Citizens who encounter such situations can report these situations to the authorities through various channels. However, these tools cannot provide adequate solutions both during the reporting of the situation and during the follow-up of the situation. In addition, if the relevant problem does not create enough agenda, the authorities tend not to do what is necessary about the issue or take it slow. The fact that the addressee of the problem is not known about many problems also undermines the problem-solving process.

In some cases, instead of reaching out to the authorities, people may need help from other people and may want to find a volunteer to fix a problem. The feeding of stray animals can be given as an example of such cases.

Although there are applications that promise solutions in this area, almost none of these applications focus on solving problems related to the city while having social media platform features [1]. Reporting apps designed by municipalities for citizens provide users with a one-sided communication opportunity between the municipality and the citizen. Citizens cannot interact with each other or even find a volunteer for a problem. In addition, these applications are local and do not have the opportunity to be used globally. On the other hand, although social media platforms offer such an opportunity to users, the main focus of these platforms is not solving problems in people’s environment.

It is difficult to report problems in cities, and following this process makes people desensitized about these problems. The lack of platforms where these problems can have a social impact also makes authorities insensitive. The fact that people do not have the opportunity to see the problems in the city and the locations of these problems also makes people’s daily lives difficult and wastes their time. All this causes cities to become more and more neglected and more challenging to live in. The need for a project that can offer solutions to all of these problems seems obvious.

1.2. Description

The reporTown application aims to accelerate the process of resolving problems that require collaborative or authoritative cooperation while ensuring safety and security for its users. The proposed solutions will enable people who encounter a problem to report their problem via their mobile phones to the related authorities or request help from volunteers to reach the solution as fast as possible.

A feature that makes reporTown different from similar applications and innovative is that it enables institutions, authorities, aid agencies, and volunteers to work together to solve the problems of their cities. They can see what people need from the posts reported to their accounts or search for problems that require volunteers. When they observe such problems, they could find out the details of the problems such as what it is about and where it is, after that, they could contact the person who posted the problem to solve it efficiently. After they solve the problem, they notify the post owner, and if the owner confirms that the problem is resolved, the report will be classified as resolved. Authorities that resolve more issues will have more points. This point system enables authorities to race with their counterparts, increasing their motivations to resolve more issues. The point system not only applied to authorities but also citizens. Citizens can upvote posts of other citizens, and a citizen with many upvotes will have more points and become one step closer to being a model citizen. In addition, posts of citizens with high points are more prominent than other posts.

A person who encounters a problem takes a photograph of the problem, uploads it to the application by giving detailed information such as the description and location of the problem, and tags the institution or municipality that should solve the problem, or as volunteering aid. Along with machine learning algorithms and computer vision, the application analyzes the problem to categorize it.

Users may not know which institute is responsible for the solution for some specific problems, and this situation decelerates the solution process. Again, by using machine learning, the application recommends interlocutors for problems to users.

Users and authorities will be able to see the problems on a map user interface. Depending on the frequency of problems reported in an area, nodes of different colors will appear on the map. Hot colors like red and orange mean that there are many problems reported in that area, and cold colors like blue and green mean that a few problems are reported in the area. By touching these nodes, users and authorities can view related posts. Thanks to this map, authorities can plan their actions wisely, and citizens can plan their days with these problems in their minds.

1.3. Constraints

1.3.1. Implementation

- The application must run on Android and IOS mobile platforms.
- Git will be used as a version control system, and GitHub will be used for managing Git repositories and collaborative work for project management.
- React Native will be used as a mobile application framework to build the application as a cross-platform native app (for both Android and IOS).
- The application will be written in Javascript programming language for the frontend and Java for the backend.
- Spring Boot will be used to build the backend services of the application (API), providing a RESTful service and easing the object-oriented programming (OOP) process.
- MongoDB as a NoSQL database management system will be used for storing user information (including credentials, scores, and reports (including components like photographs, location).
- Google Maps API will be used for location services and displaying the map.
- Python programming language and PyTorch, Tensorflow, and Keras libraries will be used to provide the machine learning functionality.
- i18next internationalization-framework will be used for internationalization and localization to offer smart language support in English and Turkish.

1.3.3. Economic

- Mentioned frameworks, libraries, and environments for the development are free-to-use services.
- The one-time fee of $25 must be paid to the Google Play Store to open a developer account to publish the application. For the Apple Store, the cost of opening and maintaining a developer account is $99 and must be paid yearly.
- The application must be free to download for all user types.

1.3.4. Environmental

- The diversity and complexity of environmental city problems must be appropriated for reporting to fit into eight categories recognized by computer vision technology. The categories are garbage disposal, transportation, road, electric and water shortage, traffic accident, missing report, and stray animals.

1.3.5. Social

- The user cannot link with other users by following or adding as a friend, and can only interact with them by upvoting their reports or volunteering.
- The institution and volunteers will be encouraged to solve a reported issue responsibly, which is ensured with the mutual solution check.
- A user can see the reports of another user by visiting their profiles.

1.3.6. Ethical

- The application will be developed in accordance with The National Society of Professional Engineers Code of Ethics [2] in the course of the application development.
- The immediate location data of users should only be used and shared in city problem reporting for displaying the report on the map to protect users' privacy.
- Users' personal information and account credentials will not be shared with third parties and will be protected by the system.
- Any inappropriate content (image, text) or discrimination in the city problem reporting system that is not filtered by the application can be reported by the users, and the unwanted content and its owner might be deleted from the system.

1.3.7. Health and Safety

- The safety of the users in the volunteering system should be protected by the mutual confirmation system and the display of the volunteer’s credibility for the user’s assessment.
- The volunteers and institution officials will be alerted by hazards posing health and safety risks in the reported problem before they investigate the said region to protect their wellbeing and take appropriate measures.
- Institution officials can only start using the application when they receive their account information from the registered institution to prevent fraud.

1.3.8. Sustainability

- The user should be able to send feedback (rating and comment) regarding the application experience through accessing Settings and clicking on Rate Your Experience button. The user can also rate the application in Google Play Store and Apple Store when released in these mobile markets.
- The application should be maintained for bug fixes in accordance with user feedback every two weeks. The maintenance should take at most a day.

1.4. Professional and Ethical Issues

Since the application relies on personal sensitive user data to perform certain actions like reporting a city problem with the immediate location information and a documented photograph, the user should be given a chance to allow or reject sharing and storing of certain user data like location and mobile camera access to the application. Thus, without the user's permission, none of their information will be used. Acknowledging that the user trusts the system to keep their data secure, their data will be protected against external threats by encryption algorithms and will not be shared with third parties. The source code will also be private and protected. The application will only use/track the location services when the application is open and running on the front to prevent spying on the user.

The institutions’ ratings are solely determined by their problem-solving performance with mathematical metrics like how fast and frequent they respond without a user rating system to avoid manipulation of reliability (like political attack) of the institutions as the application focuses on problem response.

Similarly, the ratings of the users who do not engage in solving problems but reporting them are determined by the accuracy and correctness of the problem that they reported to avoid spam or inappropriate usage of the application.

2. Requirements

2.1. Functional Requirements

2.1.1 Initial Screen

- All users must be presented with the main screen with the “Login” and “Register” buttons.
- A user or an institution that wants to register must be able to click the “Register” button to go to the registration page in which there will be a form that includes some credential fields (email, username, password)
- A user that wants to log in must be able to click the “Login” button to go to the login page in which there will be a form that includes some credential fields (username/email, password)

2.1.2 User Registration

- A user should be able to register to reporTown by entering the credentials (email, username, password).
- A user must verify their email from the verification link sent to their email by reporTown, to be successfully registered.

2.1.3 Institution Registration

- An institution should be able to register to reporTown by entering the credentials (email, institution name, password).
- An institution must verify their email from the verification link sent to their email by reporTown to be successfully registered.
- An institution must verify itself by filling a form that the admins will manually check, which appears on the registration page. Otherwise, their registration will not proceed.

2.1.4 Institution Official Registration

- An institution should be able to register an official account by entering the account credentials (email, username, password) that the official will have.
- An institution official should be able to log in with the credentials (email/username, password) that the institution account specified.

2.1.5 Login

- All user types should be able to log in with their accounts' credentials (username/email, password) if there are no currently logged-in users.

2.1.6 Reporting a Problem

- A user can report a problem by creating a report about the issue.
- A user can go to the "Report Problem" page by clicking the report button at the bottom of the screen.
- A user must be presented with a form with the following fields: A problem's title (*), subject (*), description (*), picture, responsible institution (*), and location information (*).
- A user should fill the required fields and upload a picture on the "Report Problem" page.
- A user can upload the report by clicking the "Upload" button at the bottom of the form.

2.1.7 User Profile Navigation

- A user can go to their profile by clicking the "My Account" button at the right bottom of the app so that the user can see their reported problem(s).
- A user should be able to see their profile picture, the numbers of resolved and unresolved reports, reliability score, and a short bio.
- A user must be presented with two sections as Resolved Problems and Unresolved Problems, including their reported problem(s).
- A user should be able to see their reports that have not been resolved yet by clicking the "Unresolved" button that appears on their profile.
- A user should be able to see their reports that have been resolved by clicking the "Resolved" button that appears on their profile.

- A user should be able to click a report and navigate to the “Report” page, where there is a single report with all of its details (title, subject, description, picture, responsible institution, location, upvotes, comments).
- A user should be able to see the solution that the institution uploaded in the “Report” page that contains a resolved problem.

2.1.8 Institution Profile Navigation

- A user should be able to search for an institution by its name from the search page after clicking the “Search” button at the bottom of the page.
- A user should be able to select an institution from the search results and go to its profile by clicking on it.
- A user should be able to see the information about the institution on the institution profile (profile picture, institution score, and institution officials’ accounts).
- A user should be able to see the reports that the institution has not resolved yet by clicking the "Unresolved" button that appears on its profile.
- A user should be able to see the reports that the institution has resolved by clicking the "Resolved" button that appears on its profile.

2.1.9 Profile Editing

- All user types should be able to change their profile picture and bio by clicking the “Edit Profile” button.
- An institution should be able to pin some resolved problems it would like to be seen to the top of its profile.

2.1.10 Marking a Report as Resolved

- An official of the institution authorized to resolve the reported problem must notify the report owner when the problem is resolved by clicking the “Problem Resolved" button.
- When the institution official clicks on the “Problem Resolved" button, they must fill out a solution form with the following sections to inform the user: an explanation, a picture showing the problem is resolved.

- An institution official must write an explanation and upload a picture showing that the problem has been resolved for the report owner to see and confirm that the problem has been resolved.
- A report owner user should receive the "Problem Resolved" notification, and they can click on this notification to see the report.
- A user who wants to make sure that the problem is resolved should check the solution uploaded by the authorized person and confirm that the problem is resolved by pressing the "Confirm" button if they are sure that the problem is resolved. If the user decides that the problem is still unresolved, they should click the “Reject” button, and the problem remains unresolved.
- A problem should be visible at the “Resolved Problems” of both the user’s account and institution account if both user and institution official approve that the problem is resolved.
- When the approval process is completed, both the scores of the user and the institution must be incremented by a certain amount according to the number of upvotes that the report got.

2.1.11 Map

- A user must be able to click on the “Map” button on the bottom left side of the screen to navigate to the Map.
- A user must be presented with a map with a heatmap structure where there will be intense colors on the regions with high report density.
- A user should be able to navigate on the map by swiping the screen.
- A user should be able to zoom out and zoom in on the map.
- A user should be able to click on a particular region on the map to see the reports from that particular region.

2.1.12 Feed

- A user must be able to see a feed that will consist of local reports from that other users posted, just after logging in or after clicking the “Home” button at the bottom of the screen.
- A user should be able to choose a location from the “Location” button at the top of the feed if they want to see the reports at that location.

- A user should be able to choose a category from the “Category” button at the top of the feed if they want to see the reports on that category (trash, transportation, road, electricity, water, accidents, missing people, volunteering, etc.).

2.1.13 Report Interactions

- A user should be able to see the reports and the number of upvotes of the reports.
- A user should be able to upvote a report by clicking the upside arrow button if they find the report as an important issue. Then, the number of upvotes of the report must be incremented by one.
- A user should be able to comment on a report.
- A user should be able to report a report that they think is spam or an unrelated issue, to the admins.

2.1.14 Search

- A user should be able to search for a specific topic about the reports by clicking the “Search” button at the bottom of the screen.
- A user must be presented with the trending topics, which they can click and see the related posts about that topic, at the top of the screen under the search bar.
- A user should be able to search with a keyword for user profiles, institution profiles, and posts, on the search page.
- After the search operation is done, a user must be presented with all results (users, institutions, posts) that contain the entered keywords.

2.1.15 Volunteering

- A user should be able to create a report with the category “Volunteering”.
- If a user or institution wants to participate in volunteering, they should be able to hit the “Joining” button to indicate that they are joining, which corresponds to the “Upvote” functionality in the other report categories.

2.1.16 Logging Out

- All types of users should be able to log out by clicking the “Logout” button on their profiles.

2.2. Nonfunctional Requirements

2.2.1 Usability

- Providing users with two different report tracking systems, a map and a feed, gives users the chance to choose the feature they want to use more, thus increasing the application's usability.
- Problems that receive high upvotes will appear at the beginning of the feed. In this way, users will be able to see more common problems in the upper ranks, and the traceability of the problems will increase.
- In the map feature, the problems will be displayed on a map, and users will be able to access the reports in the city by going to the desired location on the map, which increases the usability by increasing the intensity of colors on the map where report frequency is high.
- The feature of uploading photos to the reports in the application gives the users the opportunity to express the problems better and enables the institutions and other users to understand the problem visually. In this way, usability is increased by providing clarity of the problem and ease of use.
- While creating the report, there will be a category recommendation with the help of location information and computer vision so that users can create reports conveniently.
- A user whose English and/or Turkish proficiency level is A1 or A2 (beginner-level) should be able to understand menu headings, read report titles, categories and descriptions, read institution, institution officials’ and users’ pages, and post their own reports. They might need a dictionary for complex words for the reporting process.
- All functional screens/pages can be reached from the main (home) page within four clicks.

2.2.2 Scalability

- reporTown is an application that a large number of people will use. Therefore, there will be a massive load in the system. To avoid system failures, a NoSQL database should be used to obtain scalability [3].
- For a more scalable environment, load balancers on a cloud environment should be utilized.

2.2.3 Security

- The user should receive a confirmation mail while registering to the system to prevent creating fake accounts by using other people's email accounts.
- Institutions have to provide an official document approved by the institution so that the system admins deny the malicious and fake institution accounts.

2.2.4. Reliability

- The application should be 99% reliable while creating a single report and should not crash.
- If the application crashes or the user quits abruptly, the worst-case data loss in the application is losing the unfinished report, which is not yet published and was in the process of creating.

2.2.5 Performance

- The response time of each request should be less than 2 seconds.
- Verification emails should be sent within 60 seconds so that users will complete their registration process in 180 seconds.
- The application should load in less than 30 seconds when the number of users is greater than 2500.
- The photo upload should be done in less than 5 seconds while creating a report.

3. References

1. “Mavi Masa - Ankara Büyükşehir belediyesi,” *Mavi Masa - Ankara Büyükşehir Belediyesi*. [Online]. Available: https://mavimasa.ankara.bel.tr/#about-mavimasa. [Accessed: 10-Oct-2021].
2. “Code of ethics,” *Code of Ethics* | *National Society of Professional Engineers*. [Online]. Available: https://www.nspe.org/resources/ethics/code-ethics. [Accessed: 10-Oct-2021].
3. L. Shiff and W. Rowe, “SQL VS NoSQL databases: What's the difference?”, *BMC Blogs*, 05-Mar-2018. [Online]. Available: https://www.bmc.com/blogs/sql-vs-nosql/#:~:text=Scalability,-Another%20big%20difference&text=In%20contrast%2C%20NoSQL%20databases%20are,or%20constantly%20evolving%20data%20sets. [Accessed: 10-Oct-2021].