



Bilkent University
Department of Computer Engineering

Senior Design Project

reporTown

Final 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

May 6, 2022

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

Table of Contents	1
1. Introduction	3
2. Requirements Details	4
2.1 Functional Requirements	4
2.1.1 Initial Screen	4
2.1.2 Register Screen	4
2.1.3 Citizen Registration	4
2.1.4 Institution Registration	5
2.1.5 Institution Official Registration	5
2.1.6 Login	5
2.1.7 Changing Password	5
2.1.8 Reporting a Problem	5
2.1.9 Citizen Profile Navigation	6
2.1.10 Institution Profile Navigation	6
2.1.11 Notifications	7
2.1.12 Report	7
2.1.13 Assigning an Official	7
2.1.14 Posting a Solution	7
2.1.15 Map	8
2.1.16 Feed	9
2.1.17 Search	9
2.1.18 Volunteering	9
2.1.19 Logging Out	9
2.2 Nonfunctional Requirements	10
2.2.1 Usability	10
2.2.2 Scalability	10
2.2.3 Security	10
2.2.4 Reliability	11
2.2.5 Performance	11
2.3 Pseudo Requirements	11
2.3.1 Implementation	11
3. Final Architecture and Design Details	13
3.1 General	14
3.2 API Server	14
3.3 Continuous Integration	14
3.4 Utility Server	14
3.5 Other Services	14
4. Development/Implementation Details	15
4.1 Back-end Application	15
4.2 Front-end Application	17

5. Testing Details	17
5.1 Functional Testing	17
5.2 Non-functional Testing	18
5.3 Integration Testing	18
6. Maintenance Plan and Details	18
7. Other Project Elements	19
7.1.Consideration of Various Factors in Engineering Design	19
7.2.Ethics and Professional Responsibilities	23
7.3 Judgments and Impacts to Various Contexts	24
7.4 Teamwork Details	28
7.4.1 Contributing and functioning effectively on the team	28
7.4.2 Helping creating a collaborative and inclusive environment	28
7.4.3 Taking lead role and sharing leadership on the team	29
7.4.4 Meeting objectives	36
7.5 New Knowledge Acquired and Applied	37
8. Conclusion and Future Work	38
9. Glossary	38
10. Project Repositories	39
11. References	40
Appendix - User Manual	41
A. Installation	41
B. Login, Register, & Authentication	41
C. Citizens	44
D. Institutions	48
E. Officials	50

1. Introduction

Cities are home to many problems that can victimize and even sometimes endanger the citizens living in them. Road problems, garbage problems, and transportation problems can be given as examples of those problems. When citizens encounter such problems, most of the time, they are unable to interfere in those problems 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 of an 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.

It is difficult to report problems in cities, and following this process makes people desensitized to 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.

2. Requirements Details

2.1 Functional Requirements

2.1.1 Initial Screen

- All users must be presented with the “Login” screen.
- A citizen or an institution that wants to register must be able to click the “Register” button to go to the “Register” screen in which there will be a form that includes some informational and credential fields.
- Any type of user that wants to log in must be able to enter their credentials (username, password) and click on the “Login” button.
- Any type of user that forgot their password must be able to click the “Forgot Password” link to go to the “Change Password” screen where they can change their password.

2.1.2 Register Screen

- A citizen must be able to click on the “As Citizen” on the Register screen to go to the “Citizen Registration” screen where they can enter their information and credentials.
- An institution must be able to click on the “As Institution” on the “Register” screen to go to the “Institution Registration” screen where they can enter their information and credentials and upload their registration document.

2.1.3 Citizen Registration

- A citizen must be able to register to reporTown by entering the credentials (name, surname, email, username, password) on the “Citizen Registration” screen.
- A citizen must be able to accept the terms and conditions through a checkbox.
- A citizen must be able to verify their email by entering the authentication code sent to their email, to be successfully registered.

2.1.4 Institution Registration

- An institution must be able to register to reporTown by entering the credentials (institution name, username, email, password) on the “Institution Registration” screen.
- An institution must be able to accept the terms and conditions through a checkbox.
- An institution must verify their email by entering the authentication code sent to their email, to be successfully registered.

2.1.5 Institution Official Registration

- An institution must be able to register an official account by entering the account information and credentials (name, surname, email, username, password, position) that the official will have, on the “Add Employee” screen.
- An institution must be able to create a position for the official on the “Create Position” screen before registering them.

2.1.6 Login

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

2.1.7 Changing Password

- All user types must be able to receive a change password verification code to their email by entering their email on the “Forgot Password” screen.
- All user types must be able to change their password on that screen after correctly entering the verification code and a new password.

2.1.8 Reporting a Problem

- A citizen must be able to go to the “Post Report” screen by clicking the “Post Report” button at the bottom of the screen (bottom tab).

- A citizen must be able to fill a form with the following fields: The problem's picture (*), description (*), category (*), responsible institution (*), and location information (*).
- After uploading or taking the picture with a camera, a citizen must be able to see the category that the application recommends and choose to select that category or choose a category freely which he/she thinks is appropriate for the report.
- A citizen must be able to upload the report by clicking the “Post” button at the bottom of the form.

2.1.9 Citizen Profile Navigation

- A citizen must be able to go to their profile by clicking the “Profile” button at the right bottom of the screen on the tab.
- A citizen must be able to see their profile picture, username, reliability score, a short bio, and two sections as Resolved Problems and Unresolved Problems, including their respective reports.
- A citizen must be able to see their reports that have not been resolved yet by clicking the "Unresolved" button that appears on their profile.
- A citizen must be able to see their reports that have been resolved by clicking the "Resolved" button that appears on their profile.
- A citizen must be able to edit the profile picture by clicking on his/her profile picture, and his/her bio by clicking the “pen” icon.

2.1.10 Institution Profile Navigation

- An institution must be able to go to their profile by clicking the “Profile” button at the right bottom of the screen.
- An institution must be able to see their username, reliability score, a short bio, two sections as Resolved Problems and Unresolved Problems, including their reports, and a section as Employees which contains the Officials' accounts.
- An institution must be able to see their reports that have not been resolved yet by clicking the "Unresolved" button that appears on their profile.
- An institution must be able to see their reports that have been resolved by clicking the "Resolved" button that appears on their profile.

- An institution must be able to see their officials from the Employees section, and perform registering and removing officials.

2.1.11 Notifications

- A citizen or an institution must be able to see their notifications by clicking the “Notifications” button (with the bell icon) on the top left of the screen. The notifications include commenting, upvoting, solution approving, and mark as solved actions.
- A citizen or an institution must be able to navigate to the related report post that the notification is associated with by clicking on the “See Report” button.

2.1.12 Report

- Any user must be able to click “See Report” on notification pop up and navigate to the “Report” screen, where there is a single report with all of its details (description, picture, category, responsible institution, location, upvotes, and comments).
- A citizen must be able to upvote the report.
- A citizen must be able to comment on the report.
- A user must be able to click on the “Comments” button under the report and see the comments of the citizens that commented on the report.
- A user must be able to see the solution that the institution uploaded on the “Report” screen if the report is solved.

2.1.13 Assigning an Official

- An institution must be able to see the “Assign Official” button on the reports that are not assigned an official yet and are not Volunteering reports.
- An institution must be able to assign an official to a report using the pop up screen upon clicking the “Assign Official” button.

2.1.14 Posting a Solution

- An official of the institution must be able to see reports that he/she is assigned to and not yet solved on the “Assigned Reports Feed” screen.

- An official of the institution must be able to see reports that he/she is assigned to and solved but not yet reviewed by the owner of the report on the “Pending Reports Feed” screen.
- An official of the institution must be able to post a solution to a report by clicking on the “Post Solution” button on the “Assigned Reports Feed” screen and filling out a solution form with the following sections to inform the user: an explanation, a picture showing the problem is resolved.
- A report owner citizen must receive the "Problem Resolved" notification, and they must be able to navigate to the report page and click on the “See Solution” button to see the details of the solution.
- A report owner citizen must be able to approve or reject a solution by clicking the “Approve” or “Reject” button on the solution screen.
- The report must be visible at the “Resolved Problems” of both the user’s account and institution account if the user approves the solution.
- When the approval process is completed, both the scores of the user, official and the institution must be incremented by a certain amount.

2.1.15 Map

- A citizen/institution must be able to click on the “Map” button on the bottom left side of the screen to navigate to the Map.
- A citizen/institution must be presented with a world map with markers from where the reports are posted.
- A citizen/institution must be able to navigate on the map by swiping the screen.
- A citizen/institution must be able to zoom out and zoom in on the map.
- A citizen/institution must be able to click on a marker on the map to see the report from that location.
- A citizen must be able to see their current location on the map.
- A citizen must be able to navigate to their current location on the map by clicking a button that is on the top right side of the screen.
- Institutions must be able to see reports that are tagged with themselves.

2.1.16 Feed

- A citizen must be able to see a feed that will consist of reports that other users posted after clicking the “Feed” button at the bottom of the screen.
- A citizen or an official must be able to filter reports by keyword and category from the “Filter Reports” button at the top of the feed.
- Any user must be able to upvote and comment on the reports on the feed.

2.1.17 Search

- A citizen must be able to search for other citizens and institutions by their name on the “Search” screen.
- Should the citizen want to search for a citizen, the citizen must navigate to “Search Users”.
- Should the citizen want to search for an institution, the citizen must navigate to “Search Institution”.
- After the search operation is done, a citizen must be presented with all results that contain the entered keywords under the appropriate screen.
- A user must be able to navigate the search results by clicking on the citizen result to go to his/her profile and institution result to go to its profile.

2.1.18 Volunteering

- A citizen must be able to post a report with the category “Volunteering”.
- If a citizen or institution wants to participate in volunteering, they must indicate so in the Comments.
- If the report owner citizen is content with the Volunteering post and thinks the problem is solved, the report owner citizen can resolve the report.

2.1.19 Logging Out

- All types of users must be able to log out by clicking the “Logout” button on the left side of the top of the page from any screen and should be navigated to the Login screen again.
- A logout success message must be displayed on the screen after successful logging out.

2.2 Nonfunctional Requirements

2.2.1 Usability

- Providing citizens with two different report tracking systems, a map and a feed give citizens the chance to choose the feature they want to use more, thus increasing the application's usability.
- Reports have an upvoting and commenting feature to let citizens voice details of the report and make a really important city problem known with high upvotes.
- In the map feature, the problems will be displayed on a map, and citizens will be able to access the reports in the city by going to the desired location on the map, which increases the usability.
- The feature of uploading photograph to the report in the application gives the citizens the opportunity to express the problems better and enables the institutions and other citizens 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 is a category recommendation with the help of computer vision so that citizens can create reports conveniently.
- 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 on the system. To avoid system failures, a MongoDB database should be used to obtain scalability.
- For a more scalable environment, Google Cloud App Engine should be utilized to serve the application.

2.2.3 Security

- The citizen/institution should receive a confirmation mail while registering to the system to prevent creating fake accounts by using other people's email 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 being created.
- When users forget their passwords, they will be able to create a new password using the verification code sent to their email so that they don't use their accounts.

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.

2.3 Pseudo Requirements

2.3.1 Implementation

- The application must run on Android and IOS mobile platforms.
- The application must be stateless (i.e. authorization/authentication should be done with JWT tokens).
- Git will be used as a version control system, and GitHub will be used for managing Git repositories and collaborative work for project management.
- GitHub Actions should be utilized for the Google Cloud App Engine deployment (CI).
- 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, and location)).
- Google Geolocation API will be used for location services and displaying the map.
- Python programming language, OpenCV library, and FastAPI framework should be utilized to serve the machine learning service.
- To host the database and machine learning service, a Google Cloud Compute Engine VM instance should be utilized.
- Google Cloud Storage and Google Cloud Storage API should be utilized to upload/download report/solution images.
- Lombok Java library increases maintainability without the need to write getter and setter methods.
- YOLOv4 Darknet algorithm, various datasets, and OpenCV library will be used for object detection.

3. Final Architecture and Design Details

In this section, the architecture of repоТown is discussed with the final architecture diagram.

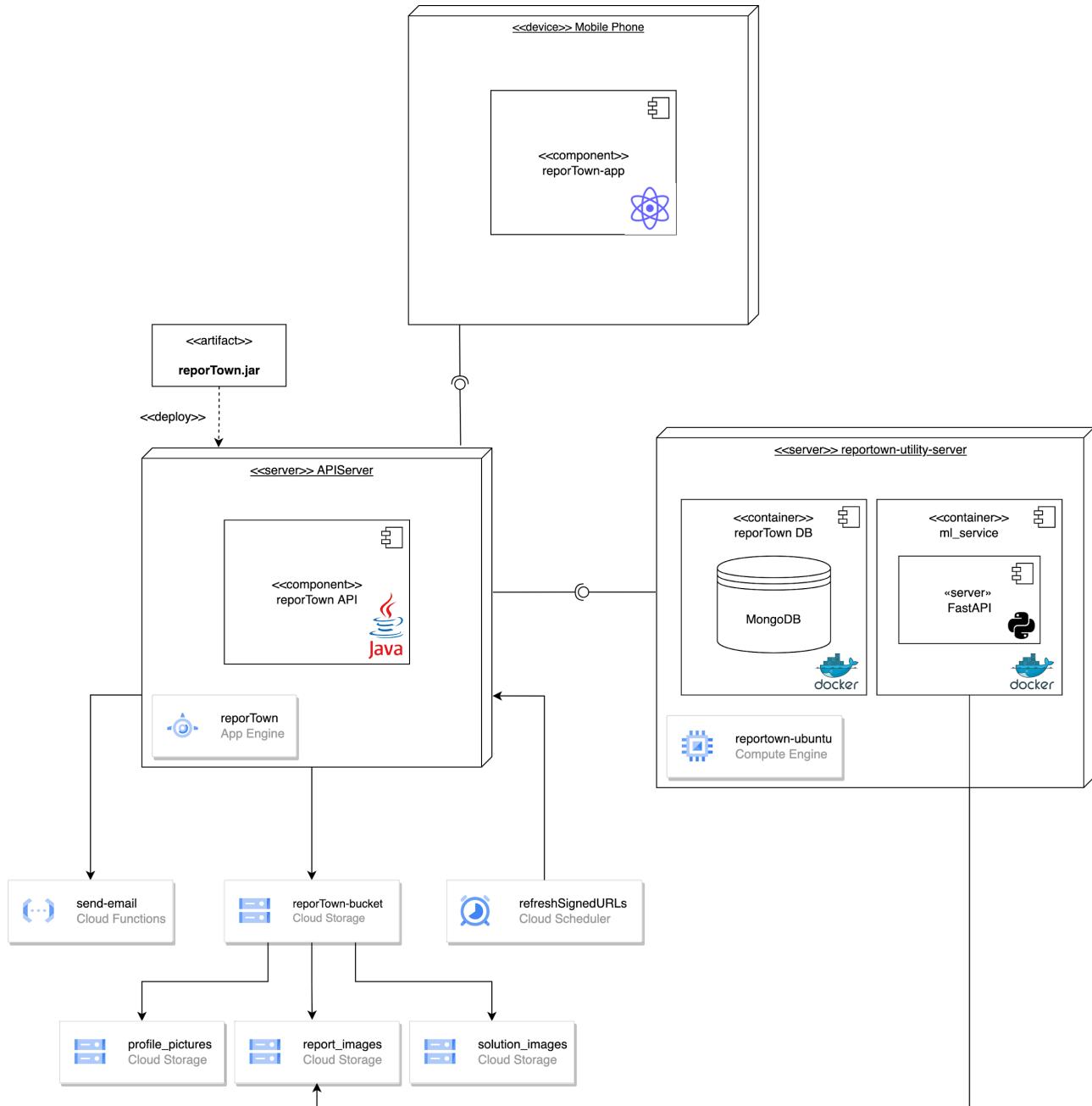


Figure 1. Final Architecture

3.1 General

In Figure 1, the final architecture of reporTown can be observed. The front-end application is a React-Native application that can run on both Android and iOS. The application communicates with the API server which is a Java Spring Boot application that is deployed to Google Cloud App Engine. A Google Cloud Compute Engine Ubuntu VM instance is utilized for certain services. Cloud Functions, Cloud Storage, and Cloud Scheduler are some services that the application utilizes for certain functionality.

3.2 API Server

The API Server is a Java Spring Boot application with the Maven package manager. The business logic is implemented in this application.

3.3 Continuous Integration

The deployment process is automated through a continuous integration (CI) pipeline (main.yml) on GitHub Actions. The pipeline allocates a worker process of GitHub Actions, checkouts the source code, builds and packages (in a jar file) the application with Maven, and finally deploys this jar file to Google Cloud App Engine.

3.4 Utility Server

reporTown has a utility server that is hosted on Google Cloud Compute Engine. This server is an Ubuntu VM instance with 10GB memory and 4GB RAM. The server contains running Docker containers for the MongoDB database connection and machine learning service. The back-end application communicates with this server for CRUD operations. Therefore, application data is stored on this server. The machine learning service, which is implemented in Python's FastAPI, is also containerized with Docker. This service predicts possible categories for a report in order to recommend to the user in the report posting process.

3.5 Other Services

reporTown utilizes Google's Cloud Functions, Cloud Storage, and Cloud Scheduler. A Cloud Function coded in Javascript sends emails to the users in the registration and password change phases. This function is triggered from the front-end

application through HTTP. Cloud Storage is used to store the report images, solution images, and profile pictures. Both the back-end application and machine learning service access the bucket through the Cloud Storage client library. Finally, a cron job associated with the back-end application periodically refreshes the signed URLs for the images that are stored in Cloud Storage.

4. Development/Implementation Details

In this section, the information regarding the implementation and development details are given with diagrams. The used technologies are also explained in appropriate headings.

4.1 Back-end Application

The implementation of the back-end application is done with Spring Boot, utilizing REST architecture.



Figure 2. Module Diagram

The cloud package is responsible for the Google Cloud configurations for the buckets that the application will utilize for storing images.

The controller package contains the controller that takes the request from the presentation layer and calls certain services that contain the application logic.

The request package contains the objectified requests with certain attributes that the application receives from the end-user. These request objects come in handy since they do not require JSON string parsing on the server-side.

The response package contains the objectified responses with certain attributes that the application sends back to the end-user. Again, these responses can be directly sent to the presentation layer without parsing to JSON.

The security package deals with the authorization and authentication of the application. The application uses Spring Boot Security, which is a highly reliable and easy-to-implement Spring Boot framework.

The service package contains the relevant services and application logic that the complex computations and data queries are done. These services are called as the controllers receive requests from the presentation layer.

The repository package serves as an interface to MongoDB, in which CRUD operations and queries are done.

The model package contains the business-specific objects (such as user types, objects related to reports, etc.) which are instantiated in the application logic for certain operations.

4.2 Front-end Application

On the front-end side of the project React-Native framework has been used so that the application will be compatible with both iOS and android operating systems. Everything that can be visible by the users have been separated into components, screens, assets, context, and gui components. The screens package has been divided into citizen screens, official screens, and institution screens. Each screen package is responsible for their own user types.

5. Testing Details

5.1 Functional Testing

In order to test the acceptance of functionalities in reporTown, each functional requirement is tested. Test results ensure that reporTown provides the promised functionalities to its users for the sake of purpose. Also, this testing allows us to

make UI design according to promised functional requirements in a short period of time.

5.2 Non-functional Testing

Reportown has to meet non-functional requirements listed in section 2 in order to provide better service to their users. Usability, scalability, security, reliability and performance are important non-functionalities for reporTown. In order to test usability, the UI part is designed according to feedback coming from people outside from the team. In order to provide better scalability, we deployed reporTown to Google Cloud and added CI/CD pipelines. Then, tests ensure that our Google Cloud Compute Engine did its job in the right way. Security is provided by using Spring Boot Security functions and protecting user's data in the database. In order to test security, leaking user's data from the database is tried and tests show that reporTown provides security. In order to test performance, we create multiple users and authenticate them at the same time to ensure whether the reporTown continues working properly.

5.3 Integration Testing

Reportown uses different services to provide promised functionalities to the users. These services include Google Cloud service, MongoDB service and ML service. Since these services are independent from each other, we need to test their integration to reporTown in order to ensure that different services can run without disturbing the reporTown users.

6. Maintenance Plan and Details

The model used for category recommendation is open to development more. We are planning to continue to train our model with the data coming from reporTown app. As the number of reports achieves a number that makes sense to train our model again, we will use this new data to improve our model. In this way, category recommendation of post report functionality will perform better.

We added CI/CD pipeline to this project so that any deployment is automatically checked whether any error exists in the project. This helps us in our maintenance plan to avoid errors before merging new functionalities.

We are planning to optimize the reporTown application for different web browsers so that user experience will be improved. In addition, we will need to improve our architecture in case the number of users increases rapidly. In order to maintain reporTown with a huge number of users, we will rearrange our module structure.

7. Other Project Elements

The following sections present how factors affect the project, work packages, ethical consideration, team collaboration details, acknowledged responsibilities, and new knowledge acquired throughout the project process.

7.1. Consideration of Various Factors in Engineering Design

Below are the factors that are taken into consideration in analysis which might have an effect on design and shape constraints/necessities for the project. Their degree of effect is given in Table 1.

Public Health

The project allows citizens to report city problems under ten significant categories, some of which pose risks for the public health, to encourage governmental institutions, private organizations and/or volunteers to eliminate the problem quickly. Consequently, protection of the public health is the main concern for reporTown. The most profound city problem that threatens public health and is one of the categories is improper garbage disposal on streets as it emits toxic gases and might create fire hazards [1]. The citizens can report improper waste management to prevent being exposed to harmful gases. Other reportable problems are for ill stray animals and dangerous road conditions. The citizens can report stray animals in bad health condition via reporTown to eliminate the risk of spread of zoonotic diseases like rabies. They can also report detrimental road conditions like flooded/eroded/icy to forestall traffic accidents so that other reporTown users would be cautious about the reported road in their region as institutions would be acting to repair it. The project must efficiently call for the collective aid of institutions and volunteers to early

eliminate public health threats mentioned above, so it must present a quick report generation with computer vision for problem categorization and create a linked community for increasing the visibility of a report with upvotes from other users. Additionally, the citizens themselves should accept the proof of solution, so they have control over satisfaction of protection of their health.

Public Safety

Similar to ensuring public health, some of the ten categories for city reports like traffic accident reports are chosen to promote public safety by notifying appropriate institutions for solutions. Besides, other citizens in the region should see these reports on their feed which decreases the risk of hazard exposure by the public.

It is of utmost importance that only verified institutions and their verified officials must be able to register to the system since fake institutions and official accounts would ruin the reputation of the imitated party as well as harm the purpose of the project which is providing communication to citizens with reliable problem solvers for city problems. For ensuring the safety of exchange between user groups, each institution's registration on the system and citizen's should be verified with email, and institutions must create accounts for their officials.

Moreover, reporTown's posting a report process requires special permissions such as accessing the camera and/or camera roll for photographic evidence, current location information, and authentication information from the citizens. The application should ask for permission for each feature from the citizens and should promise the safety of their data by stating the protection of their user data from third parties and encrypt the data.

Public Welfare

The project makes the aiding communication from governmental and private organizations and volunteers to citizens easier by providing an easy to use free native application for the general age group. Also, reporTown is expected to include a wide range of institutions to keep the array of help available and diverse. In the long run, the project should improve the cities at a better rate than when only traditional or limited media platforms are used. However, if the city's welfare is low,

the institutions and volunteers might not have the resources to solve city problems and citizens might not have access to a smartphone, so the download count of the application is affected negatively in that region and the application might not be preferred.

Global Factors

A world map is integrated into reporTown to support international users, which makes the application open for global use (any user type from any region will have its own region feed). reporTown is initially expected to support English (the most spoken language). A legal constraint due to globalization for the mobile project is the General Data Protection Regulation (GDPR) which is imposed by the European Union (EU) to any organization that collects user data from the European countries [2]. Consequently, reporTown should ask for permission explicitly and collect only necessary user data, encrypt them, and designate data protection responsibilities for the project team.

Cultural Factors

Cultural factors do not have a considerable impact on reporTown as ten city problem categories are independent from cultural backgrounds. For instance, region A and region B can both have improper garbage disposal problems. Possibly, interest in volunteer work reports might change depending on how supportive a culture is for helping people in need.

Social Factors

reporTown supports a volunteering system to ease the resource utilization of institutions and introduce a cooperative aiding platform between citizens of the same region. Thus, citizens can adopt social responsibility if they choose to. They can communicate for volunteer work via comments on posted reports. All user types have scores based on feedback and report metrics for social credibility.

Environmental Factors

The project's motivation is to sustainably ameliorate cities with quick response from authorities and collaboration within communities. Fixing the reported city problems

implies living in better environments for citizens from any social class with great accessibility. Following are environmental impacts of reporTown solutions:

- Proper waste management: Less greenhouse gas emission for the neighborhoods
- Proper road maintenance: Less frequent road reconstruction means less noise, dust and vibrations [3]
- Cooperative help for stray animals: Less disease spread means protection of local animal populations and citizens, more opportunities for adoption, decrease in overpopulation
- Reported traffic accidents: Preferring other routes with the acknowledgment of the accident decreases intensity of traffic on the lane of the accident and results in less traffic time and gas emission.

Economic Factors

The project functions better (better rate of problem solutions) as more users register from any type, so keeping the application free is essential. The more users the application attracts, more work is loaded to the database. Additional costs might be required to sustain the system in the future work. This might be solved by requiring an affordable periodic membership fee from institutions in that case.

Table 1: Factors that can affect analysis and design and their degree of effect.

	Effect level	Effect
Public health	10	reporTown considers report categories to eliminate city problems threatening public health.
Public safety	9	reporTown considers problem categories, verification of institutions, access permissions.
Public welfare	4	reporTown's active user rate in a region is affected by the region's welfare, and report solutions are equally provided to any social class citizen group.

Global factors	5	repоТown can be globally used on an interactive world map to support internationalization.
Cultural factors	2	repоТown's addressed city problems are global problems independent of cultural backgrounds, volunteering tendency might change due to cultural value for altruism.
Social factors	3	Volunteering and commenting system in repоТown introduces social responsibility to its users.
Environmental factors	10	repоТown provides access to solutions which have a wide range of good environmental impacts such as less gas emission, noise pollution, disease spread.
Economic factors	2	repоТown is a free to register and use native mobile application to encourage more user rate, costs might increase with time.

7.2.Ethics and Professional Responsibilities

ReportTown is an application that has an impact on the social, global, economic and environmental aspects, and this imposes many ethical and professional responsibilities on its developers and itself.

The most prominent aspect is the environmental aspect, because the goal of reportTown is to make cities more livable places and involve their residents in this process. Thanks to reportTown, it is possible to achieve a more beautiful, sustainable and livable environment with the help of people.

ReportTown is not just an application designed for local use, on the contrary, it can be used anywhere in the world; It is open to the use of every person, regardless of race, country. With the worldwide use of the application, hundreds of different cities in the world will become more livable, and the application will contribute to the increase in the satisfaction of the city residents and even the people who come to the city as tourists.

ReportTown, which has effects on the economic aspect, will also contribute positively to the economies of the cities as it will help the development of the cities. Another aspect taken into consideration while developing reportTown is the social aspect.

The application has a social media feature. Users' personal data will be used within the application and therefore this information is stored. Keeping this data safe is an important and huge responsibility. Also, along with its volunteering feature, reportTown aims to increase social cohesion, which is an important effect in the social aspect.

While developing an application with such effects, developers should meticulously fulfill the responsibilities brought by the project and be aware of the contribution value of their work.

7.3 Judgments and Impacts to Various Contexts

Judgement	reportTown is a free to download and use native application.	
Impact Context	Impact Level (out of 10)	Impact Description
Global	10	Everyone can access the application and become a user as long as they have a smartphone (IOS or Android).
Economic	10	Everyone can use the application without paying an additional cost. They do not have to pay for phone services to call organizations. They need an internet connection at most to contact authorities.
Environmental	10	Since everyone can use it from their cities, the environmental harm of the city problems can be solved across the globe.
Societal	10	Since everyone can use it, they can form a community with the citizens that are victimized by the same city problems. Also, due to the potential of great reach of volunteering service, it strengthens the awareness and cooperation within cities.

Judgement	Language of the application upon first release is English.	
Impact Context	Impact Level (out of 10)	Impact Description
Global	8	English is the most spoken language in the world, so the majority of global users can benefit from it. However, people that are non-English speakers might have to use external translation to
Economic	0	-
Environmental	0	-
Societal	6	Only basic English speakers are able to constitute the users for this application, but the potential user rate is still high due to the amount of English speakers in the world. Users are united under one language for communication which might ease the channel of communication in feed.

Judgement	Citizens can create volunteering reports.	
Impact Context	Impact Level (out of 10)	Impact Description
Global	8	It encourages users to locally form a community to fix a common problem, yet it also encourages the upvotes of other users that are non-local which leads to support from people for a really significant city problem.
Economic	10	Due to the volunteering category, citizens might be able to raise a public fund to solve some city problems without having to use institutions' budgets. This leads to institutions to use their budget effectively for

		greater in scale issues instead.
Environmental	6	Some environmental hazards like unmanaged garbage problem might be solved with volunteering (ie. garbage collection of locals) quickly to eliminate health threats.
Societal	10	The app encourages citizen cooperation and also with institutions, too, which empowers societal relations and awareness of common problems. It also emphasizes that together problems can be solved.

Judgement	The solutions must be approved by the report owner citizen to assure validity of the solution.	
Impact Context	Impact Level (out of 10)	Impact Description
Global	0	-
Economic	0	-
Environmental	6	It ensures environmental hazards are solved and fully eliminated.
Societal	10	It establishes the reliability of officials and institutions by ensuring the validity of solutions via this confirmation function.

Judgement	Computer vision technology aids in categorization of reports.	
Impact Context	Impact Level (out of 10)	Impact Description
Global	10	Regardless of the content of the uploaded picture (culturally etc.) it can identify a category for the report.
Economic	1	In terms of time cost, it

		fastens the process of posting a report.
Environmental	5	With correct recommendation, it ensures that the environmental city problem is marked correctly.
Societal	0	-

Judgement	Verification of registration is implemented through email service and officials are only created by institutions.	
Impact Context	Impact Level (out of 10)	Impact Description
Global	0	-
Economic	0	-
Environmental	0	-
Societal	10	It ensures safety within the application by preventing creation of suspicious institutions and officials. Thus, the reliability of officials and institutions are established. Citizen accounts are also validated this way (checking that they are the owner of the email).

Judgement	10 categories are defined for city problems.	
Impact Context	Impact Level (out of 10)	Impact Description
Global	10	The city problems defined by the categories are global city problems - they are not specific to a region. Thus, problems are global.
Economic	3	Categorization helps institutions and officials to identify the problem quickly (time cost).
Environmental	5	Environmental problems are categorized for efficient problem solving.
Societal	0	-

7.4 Teamwork Details

7.4.1 Contributing and functioning effectively on the team

To be able to have information about what each team member should do, the reporTown developer team met every week to determine their task. What must be done during the following sprint was determined and tasks were shared among the team members. The Jira tool was used for project management in which tasks were created, their deadlines were determined, and shared among the developers. Every detail of the tasks was written in Jira so that team members were aware of what other team members were doing, and since there were 4 status of a task which are 'To Do, In Progress, In Review, and Done', the implementation details of those tasks could be observed.

TO DO 1 ISSUE	IN PROGRESS 7 ISSUES	IN REVIEW 1 ISSUE	DONE
Write 4.5. and 4.6. Sections of the analysis report REP-12	Write 4.3. and 4.4. Sections of the analysis report REP-11 Create Mock-Ups REP-8 Create Object and Class Model REP-5 Create Sequence Diagrams REP-7 Create Functional and Non-functional Requirements REP-9 Write 4.1. and 4.2. Sections of the analysis report REP-10 Add Admin's functionalities to Use-Case diagram and create descriptions for the use cases	Write description of the diagrams you have created REP-13	

Figure 3. Jira board the reporTown team uses

7.4.2 Helping creating a collaborative and inclusive environment

Although everyone in the group has a certain job description and responsibility, it is also aimed that each group member also helps other teammates in their work. In this

way, it was possible to transfer experience between peers, each team member gained experience in areas that they have no previous experience, the completion time of the work was reduced, and each team member had detailed information about different parts of the project instead of only knowing about their own part. All of this increased team members' ownership of the project, made the process much more efficient, and created a more productive working environment by increasing peer-to-peer interaction, friendship and intimacy.

7.4.3 Taking lead role and sharing leadership on the team

There are a lot of tasks to consider in the workflow of the project, including the architecture of software, design decisions, requirements analysis, project management and many more which require different skill sets from the team. For this reason, and to manage our time and our team collaboration effectively, work packages were created and their leaderships were shared amongst the team, so that all team members were responsible for an equal load of work as a leader according to their interests, skills, and past experience. This way, each team member had a chance to work on their leadership skills. Each work package also had other members to work on one package in collaboration. Leaders had the responsibility to construct an outline of how the task will be undertaken (organization of work for the task).

Table 2: Work packages table for reporTown

WP	Work Package Title	Leader	Involved
WP1	Project Specification Report	Oğuz K. İmamoğlu	All team members
WP2	Analysis Report	Cemre Biltekin	All team members
WP3	High-level Design Report	Mustafa Yaşar	All team members
WP4	Low-level Design Report	Arda Akça Büyük	All team members
WP5	Final Report	Elif Özer	All team members
WP6	Presentations & Demos	Mustafa Yaşar	All team members
WP7	Project Website	Elif Özer	Oğuz K. İmamoğlu
WP8	Computer Vision	Arda Akça Büyük	Mustafa Yaşar
WP9	UI & UX Design	Cemre Biltekin	Oğuz K. İmamoğlu, Mustafa Yaşar

			Yaşar
WP10	Project Management	Cemre Biltekin	Arda Akça Büyük, Elif Özer
WP11	Machine Learning Models (Classification)	Oğuz K. İmamoglu	Cemre Biltekin, Arda Akça Büyük
WP12	Frontend Development	Mustafa Yaşar	Oğuz K. İmamoğlu, Cemre Biltekin
WP13	Backend Development	Elif Özer	Oğuz K. İmamoğlu, Arda Akça Büyük
WP14	Testing	Arda Akça Büyük	Elif Özer, Mustafa Yaşar, Cemre Biltekin
WP15	Deployment	Oğuz K. İmamoglu	Elif Özer, Mustafa Yaşar

Table 3: Project specifications wp table

WP1: Project Specifications Report
Start Date: Oct. 1, 2021 End Date: Oct. 11, 2021
Leader: Oğuz Kaan İmamoğlu Members Involved: All team members
Objectives: To give a title and a brief description of the proposed project, determine requirements and constraints of the project, evaluate the project in terms of different aspects such as sustainability.
Tasks: Task 1.1: Writing introduction Task 1.2: Writing requirements Task 1.3: Reviewing the report
Deliverables: Specifications Report

Table 4: Analysis report wp table

WP2: Analysis Report
Start Date: Oct. 13, 2021 End Date: Nov. 14, 2021
Leader: Cemre Biltekin Members Involved: All team members
Objectives: Analysing the problem the project addresses and the system to be developed in an object-oriented way regarding its detailed requirements, application model, constraints, risks, project plan, learning strategies; and inscribe

the system analysis on the analysis report as a deliverable.

Tasks:

- Task 1.1:** Work division for analysis report writing
- Task 1.2:** Preparation of the diagrams
- Task 1.3:** Writing the analysis report according to responsibility of each member
- Task 1.4:** Integration of written parts into final form of the analysis report
- Task 1.5:** Analysis report review and delivery

Deliverables: Analysis report

Table 5: High-level design report wp table

WP3: High-Level Design Report
Start Date: Nov. 16, 2021 End Date: Dec. 24, 2021
Leader: Mustafa Yaşar Members Involved: All team members
Objectives: Creating a high-level design report in which the analysis model is transported into a system design model including design goals of the project, creating smaller subgroups to share among the project members, and considerations about public health, safety, global, cultural, social, environmental, and economic factors in engineering design process.
Tasks: Task 1.1: Researching about the considerations about the project, and inspecting those considerations in terms of the project. Task 1.2: Creating the high-level design report of the project. Task 1.3: Deciding about the communication of the smaller subgroups among each other.
Deliverables: High-level design report

Table 6: Low-level design report wp table

WP4: Low-Level Design Report
Start Date: Dec. 25, 2021 End Date: Jan. 30, 2022
Leader: Arda Akça Büyük Members Involved: All team members
Objectives: Refinement of the high-level design to low-level.
Tasks: Task 1.1: Explanation of the extent and validity of the design principles that were used to carry out the project. Task 1.2: Explanation of engineering standards used during design.

Deliverables: Low-Level Design Report
--

Table 7: Final report wp table

WP5: Final Report
Start Date: Jan. 31, 2022 End Date: May 6, 2022
Leader: Elif Özer Members Involved: All team members
Objectives: Preparing a report stating final architecture, design and status of the project. Complete information about the system is provided and details about the test and maintenance plan will be given in detail with the other project elements such as ethics and professional responsibilities, judgements and impacts to various contexts and teamwork information.
Tasks: Task 1.1: Work division for final report writing. Task 1.2: Writing the details of requirements, final architecture and design, implementation and testing and maintenance. Task 1.3: Writing other project elements by explaining engineering/project standards used during design. Task 1.4: Final report review and delivery.
Deliverables: Final Report

Table 8: Presentations and demos wp table

WP6: Presentations & Demos
Start Date: Dec. 20, 2021 End Date: May 6, 2022
Leader: Mustafa Yaşar Members Involved: All members
Objectives: Preparing the presentation and presenting the demo
Tasks: Task 1.1: Prepare the presentation to explain the project in detail to the audience Task 1.2: Conduct the demo
Deliverables: Presentation

Table 9: Project website wp table

WP7: Project Website
Start Date: Oct. 13, 2021 End Date: May 6, 2022

Leader: Elif Özer Members Involved: Oğuz Kaan İmamoğlu
Objectives: Preparing web page for the reporTown describing the project. Each report will be available on this project website.
Tasks: Task 1.1: Preparing website Task 1.2: Make Project Specification available on the website. Task 1.3: Make Analysis Report available on the website. Task 1.4: Make High Level Design Report available on the website. Task 1.5: Make Low Level Design Report available on the website. Task 1.6: Make Final Report available on the website.
Deliverables: https://ardaakcabuyuk.github.io/reporTown/

Table 10: Computer vision wp table

WP8: Computer Vision
Start Date: Jan. 17, 2022 End Date: Mar. 23, 2022
Leader: Arda Akça Büyük Members Involved: Mustafa Yaşar
Objectives: Implementing a computer vision algorithm for object identification and detection with utilization of YOLO algorithm and COCO dataset with OpenCV for city problem categorization on images in reporting process by a citizen
Tasks: Task 1.1: Training of YOLO object detector on COCO dataset inclusive of all 8 categories Task 1.2: Evaluation of the YOLO object detector for city problem categorization with test sets Task 1.3: Integration of the computer vision model into the system Task 1.4: Testing of the computer vision model in the system
Deliverables: Computer vision model for city problem categorization

Table 11: UI/UX Design wp table

WP9: UI/UX Design
Start Date: Oct. 17, 2021 End Date: May 6, 2021
Leader: Cemre Biltekin Members Involved: Oğuz K. İmamoğlu, Mustafa Yaşar
Objectives: Designing the UI with compliance to enhanced UX. Ease the job of the front-end people.

Tasks:
Task 1.1: Defining navigational paths
Task 1.2: Creating mock-ups
Task 1.3: Communicating with front-end people during implementation
Deliverables: Mock-ups

Table 12: Project management wp table

WP10: Project Management
Start Date: Oct. 1, 2021 End Date: Apr. 26, 2022
Leader: Cemre Biltekin Members Involved: Arda Akça Büyük, Elif Özer
Objectives: Managing project management strategies, organizing efficient team meetings, setting up team environments for tracking team collaboration.
Tasks: Task 1.1: Choosing, setting up, and managing team collaboration tools which are GitHub, Jira, Google Docs. Task 1.2: Organizing sprints Task 1.3: Organizing team (weekly) meetings and supervisor meetings Task 1.4: Providing educational resources
Deliverables: Collaboration environments' pages

Table 13: Machine learning models wp table

WP11: Machine Learning Models (Recommendations and Classifications)
Start Date: Dec. 1, 2021 End Date: Mar. 27, 2022
Leader: Oğuz Kaan İmamoğlu Members Involved: Cemre Biltekin, Arda Akça Büyük
Objectives: Establishing machine learning models to be used for features such as problem classification, trend problems, obtaining the necessary datasets and training the models.
Tasks: Task 1.1: Getting or generating datasets Task 1.2: Building the models Task 1.3: Training the models Task 1.4: Testing the models Task 1.5: Adding necessary features to the application
Deliverables: Trained machine learning models

Table 14: Front-end development wp table

WP12: Front-end Development
Start Date: Nov. 15, 2021 End Date: Feb. 28, 2022
Leader: Mustafa Yaşar Members Involved: Cemre Biltekin, Oğuz K. İmamoğlu
Objectives: Implementing the user interface design of the application that aims to achieve maximum user friendliness and usability by using React Native
Tasks: Task 1.1: Starting implementation by creating registration, sign-in, sign-up, and profile screens Task 1.2: Implementing feed screen and integrating the front-end with the back-end Task 1.3: Implementing posting report, institution-volunteering recommendation, solutions, and institution profile screens Task 1.4: Implementing employee-related screens such as creating employees, adding employee screens Task 1.5: Implementing Map screens and integrating the machine learning algorithms to the front-end
Deliverables: All screens that can be reached while using the application

Table 15: Backend development wp table

WP13: Backend Development
Start Date: Nov. 23, 2021 End Date: Mar. 8, 2022
Leader: Elif Özer Members Involved: Oğuz K. İmamoğlu, Arda Akça Büyükk
Objectives: Developing server logic and handling functionality of the application with the database connection.
Tasks: Task 1.1: Providing database connection Task 1.2: Implementing reusable code parts Task 1.3: Implementing logic of the application Task 1.4: Implementing APIs for get and post processes
Deliverables: APIs and Database connection

Table 16: Testing wp table

WP14: Testing
Start Date: Mar. 24, 2022 End Date: Apr. 20, 2022

Leader: Arda Akça Büyük Members Involved: Elif Özer, Mustafa Yaşar, Cemre Biltekin
Objectives: Automated/Manual Unit & Integration Testing of the application and reporting the bugs to developers.
Tasks: Task 1.1: Code Unit & Integration Tests Task 1.2: Assuring Continuous Integration Task 1.3: Reporting the bugs to developers
Deliverables: Unit & Integration Tests, Automated testing environment

Table 17: Deployment wp table

WP15: Deployment
Start Date: Apr. 1, 2022 End Date: May. 6, 2022
Leader: Oğuz Kaan İmamoğlu Members Involved: Elif Özer, Mustafa Yaşar
Objectives: Making the application usable and accessible by building it, deploying new features, making it available in application stores
Tasks: Task 1.1: Building the application for Android Task 1.2: Building the application for IOS Task 1.3: Publishing the app in stores Task 1.4: Updating app and deploying new features if necessary
Deliverables: AAB file and IOS application

7.4.4 Meeting objectives

Referring to the objectives set on Analysis phase, all deliverables are completed and delivered appropriately which are included in WP1-2-3-4-5-6-7. The computer vision part is implemented as promised and does categorization as expected, but the categories were previously 8, now it is altered and it is 10: Road Problem, Road Maintenance, Electricity, Stray Animals, Missing, Volunteering, Transportation, Waste, Garbage, Accident.

The volunteering mechanism is changed: citizens do not have to press a button to imply that they are joining a volunteering event; instead they can comment and join the volunteering act and the report owner can mark the problem as solved. No

officials or institutions are needed to be assigned. UI & UX Design, Project Management, Frontend Development and Backend Development are completed as promised. However, Admin user is removed completely from the application itself and its requirements (functionalities) are also discarded (like the report post as spam functionality). The trending page is also discarded. The map interface design is changed to include clickable markers for created reports instead of implementing a heatmap - this design change was due to heatmap's non-user friendly clickable actions: it did not enable us to look at reports from regions individually, and did not enable citizens to look where they previously posted during posting the report action.

In terms of testing, integration and functional testing are completed as promised. Functional testing includes testing via Postman and real time testing through Expo service.

For deployment, our expectation was to deploy the application to application stores: Google Play and AppStore. This expectation is delayed to future work due to obtaining the developer license process.

7.5 New Knowledge Acquired and Applied

Significant amount of technologies that are used during the development process was completely new to the developers of the reporTown. For instance, Cemre Bilekin and Oğuz Kaan İmamoğlu had limited experience in React Native which is used to develop the front-end of the project, however, Mustafa Yaşar had no experience in that technology. Therefore, the framework should have been learned from scratch. Also, Elif Özer and Arda Akça Büyük were familiar with cloud technologies but this project deepened their knowledge and experience in Google Cloud as they did hands-on work. Computer vision functionality of the application required investigation of tutorials and learning of datasets.

While developing a big project such as reporTown, the usage of the technology that is newly learned must be used very carefully so as not to encounter problems whose reason is unclear. Thus, while applying a solution to a problem, the solution was

discussed among the team and possible problems that can appear because of inexperience were avoided.

While acquiring such knowledge and skills, online resources such as documentation, Udemy, Youtube, and Stackoverflow have been used. Those resources include the experienced developers' lectures, projects, and applications, therefore, they contain practical and significant information for the developers.

8. Conclusion and Future Work

In conclusion, all screens which provide the necessary functionalities of the application have been implemented. These screens are in three categories which are citizen screens, institution screens, and official screens. These 3 types of users can properly use the application according to their user type. Back-end which provides every information to the frontend has been fully implemented. Machine learning models for predicting the category of an image uploaded to the report have been connected to the front-end.

In the future, the machine learning models are expected to be extended so that there will be more recommendations for the users so that the users will be using the application more easily. Additionally, spam detection machine learning models will be trained so that the descriptions of the reports will be automatically filtered for spam and ensure safety for users. Language support is expected to be released in future additions. The application is expected to be deployed to AppStore after purchasing the developer license for IOS - however, through Expo, IOS device usage is already tested and fully functional to use. Categories might be extended to include further city problems which are recognized by the machine learning model.

9. Glossary

Reports: The content that the citizens posted to the application which includes the problem they encountered.

Citizens: People who are using the application by posting reports and requesting solutions to the problems they encountered.

Officials: Employees who are working for an institution. They can solve problems by analyzing reports and trying to solve the problems by taking necessary steps.

Institutions: Institutions like the municipality, governorship, or non-governmental organizations that are using the application in which. Officials work as employees in institutions.

MongoDB: An open source NoSQL database for large data.

AWS: Cloud computing platform for database storage.

S3: AWS based object storage service.

JSON: Human readable format for structure of the data/file.

10. Project Repositories

The first iteration of the project can be found in a joint repository linked below (done on first :

<https://github.com/ardaakcabuyuk/reporTown>

For the second iteration, the frontend repository and the backend repositories are separated. The frontend repository can be viewed on:

<https://github.com/mustafayasar28/reporTown-front-end>

The backend repository can be viewed on:

https://github.com/ardaakcabuyuk/reporTown/tree/backend_dev

11. References

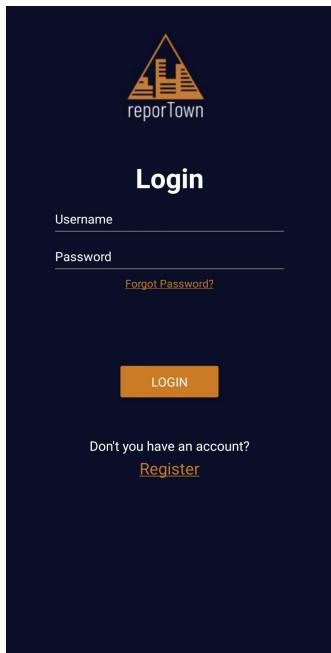
- [1] L. Osborne, “Garbage governance: Poor waste management causes environmental crises: DW: 27.08.2015,” *DW.COM*, 27-Aug-2015. [Online]. Available: <https://www.dw.com/en/garbage-governance-poor-waste-management-causes-environmental-crises/a-18677430>. [Accessed: 15-Nov-2021].
- [2] B. Wolford, “What is GDPR, the EU's new Data Protection Law?,” *GDPR.eu*, 13-Feb-2019. [Online]. Available: <https://gdpr.eu/what-is-gdpr/>. [Accessed: 15-Nov-2021].
- [3] “4. environmental issues related to road management,” *ROADEX Network*, 23-Apr-2020. [Online]. Available: <https://www.roadex.org/e-learning/lessons/environmental-considerations-for-low-volume-roads/environmental-issues-related-to-road-management/#:~:text=The%20three%20most%20damaging%20effects,roads%20and%20unbound%20aggregate%20layers>. [Accessed: 15-Nov-2021].

Appendix - User Manual

A. Installation

reporTown is a mobile application in closed beta. In order to use the application, “Expo” must be installed on the phone. Additionally, the source code of the application should be downloaded from the moodle. Once the source code is downloaded, change the current directory to the project’s directory (reportown-frontend) and run the following commands: “npm install”, “npm start”. When the installation and initialization are completed, the browser will be opened automatically and the user will be presented an expo page. Change the Connection to Tunnel from the left panel and wait for a second. When the QR code changes, open the Expo application from your phone and click “Scan QR Code”. When you scan the QR code, the application will be started.

B. Login, Register, & Authentication



User can enter their credentials (username and passwords) in order to log in.

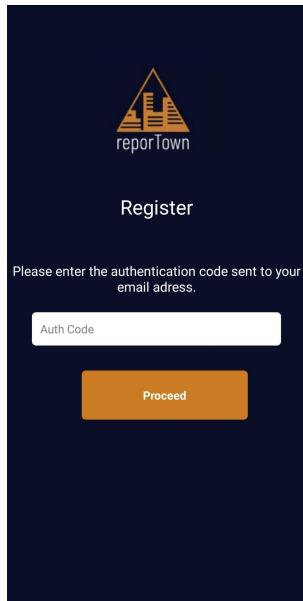


One can register to reporTown by clicking the register button if one does not have an account already.

Then the potential users must choose registration type, as a citizen or as an institution.

A comparison of two registration forms. On the left is the "As Citizen" form, showing fields for Name, Surname, Username, Email, Password, and Password (Again). It includes a note "Please fill all fields" and a checkbox "I understand and accept terms of usage." On the right is the "As Institution" form, showing fields for Institution Name, Select country (dropdown), Select city (dropdown), Username, Email, Password, and Password (Again). It also includes a checkbox "I understand and accept terms of usage." and a large orange "Register" button at the bottom.

Users must fill in the fields such as name, surname, user name, institution name (for institutions), password and password repetition provided to them in the registration form. The two passwords must match.

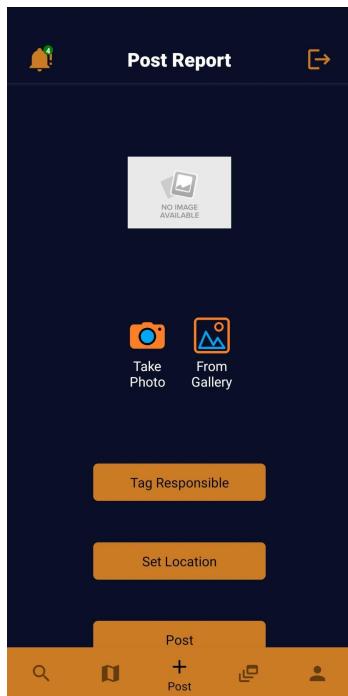


Then users who press the progress button come to the email verification screen. After entering the verification code sent to their e-mail addresses in this field, their email addresses are verified. The application redirects to the login screen.

On the screens, the logout button on the upper tab allows users to exit the application.

C. Citizens

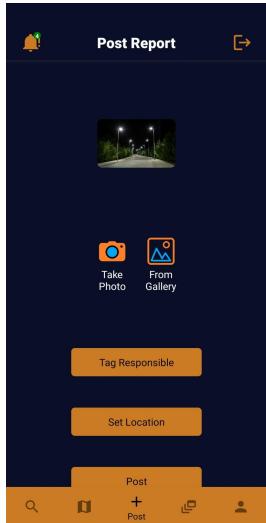
Citizens have 5 main screens. These are feed, post report, search, profile, and map screens.



In the post report screen, citizens can post reports by fulfilling some tasks.



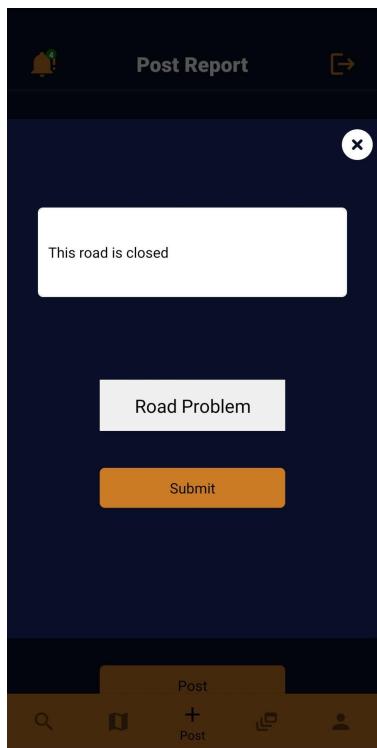
They should tag a related institution for the report by using a miniscreen. They can touch the logo near the institution's name.



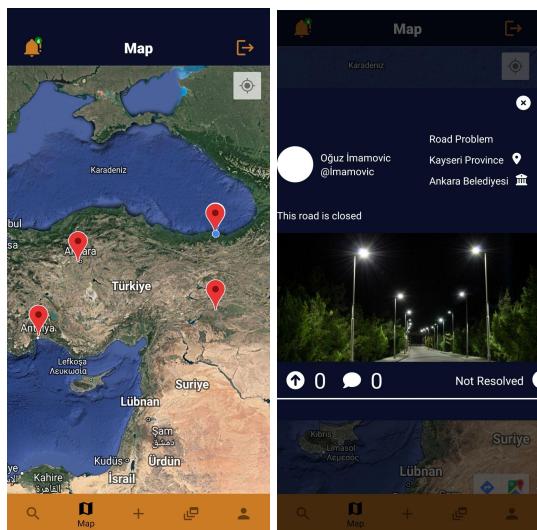
They can either choose an image for the report from their gallery or they can take a new picture. The picture will be displayed at the top of the screen.



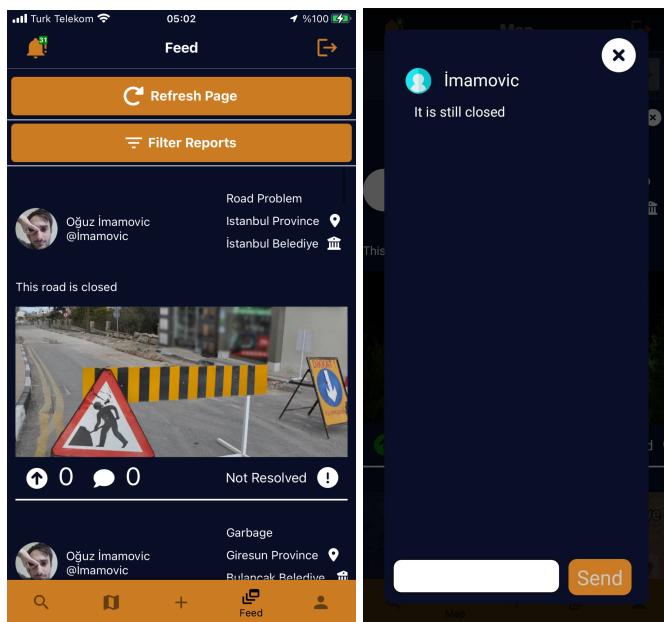
Users can choose a location for the report by using a map mini-screen. If they do not choose a location, their current location will be used.



After users click the submit button, a mini screen will be open. In this mini screen, users can set the description and the category for the report. If reporTown computer vision service has a recommendation, the system states it. After they click submit button, the report will be submitted.



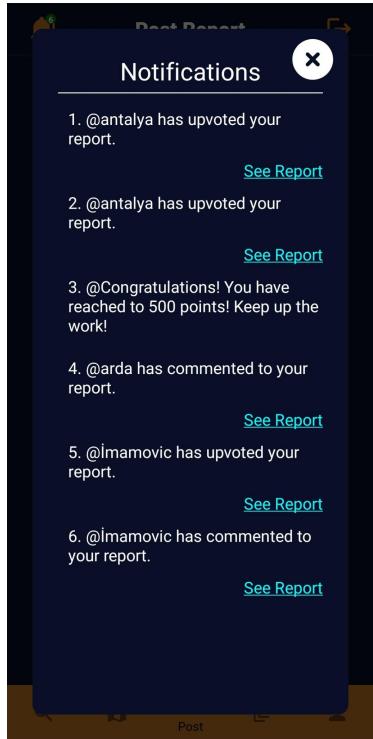
Citizens see the locations of the reports posted on the map screen as markers. By clicking on these markers, they can see the reports in more detail. Through this mini-screen, they can interact with the reports, like them, comment on them, and see the images larger.



On the feed screen, users can see the reports posted by other users, they can navigate through the reports by scrolling down on the screen. They can interact with the reports, like them, comment on them, and see the images larger.

On the profile screen, citizens can see the reports they have created under the title of solved or unsolved, they can interact with the reports, and if a solution has been sent for a report, they can see it and accept or reject it. Also on this screen, they can add a bio, edit their bio or choose a profile photo for themselves and upload it.

Citizens can search for other citizens or institutions on the search screen and visit their profiles by clicking on them.



Thanks to the notification button on the top tab, citizens are informed when they receive likes, comments, or solutions to their reports.

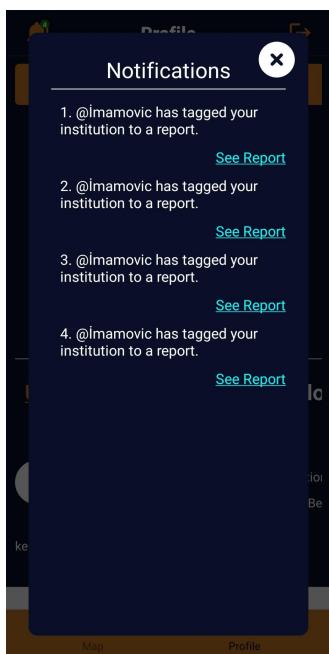
D. Institutions

Institutions have two main screens. Map and profile.

Institutions can see the reports assigned to them on the profile screen as resolved and unresolved reports. On this screen, they can add officials to work within the institution. On this screen, they can also like the reports or assign an official to the report as an assignee. They can also add or edit bios on their profile.



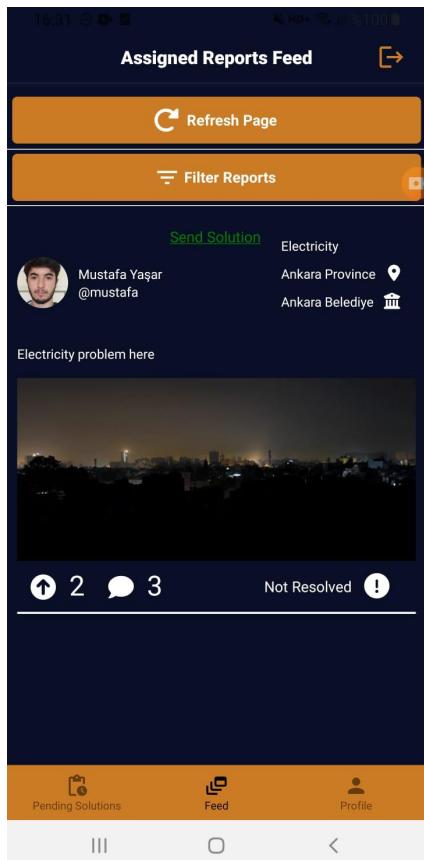
On the Map screen, they can view the reports they are tagged with on the map and click on the markers to see these reports in detail, assign them as assignee or like them.



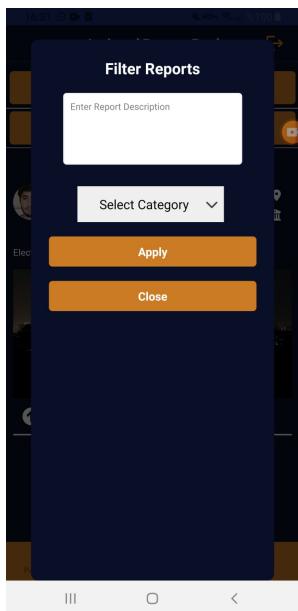
Thanks to the notification button on the top tab, institutions can be informed about the posts they are tagged in.

E. Officials

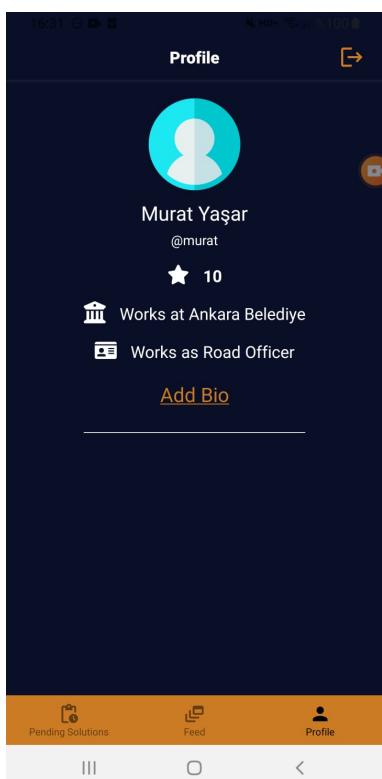
Official has 4 main screens. Assigned Reports, Profile, Pending Solutions and Solve Report.



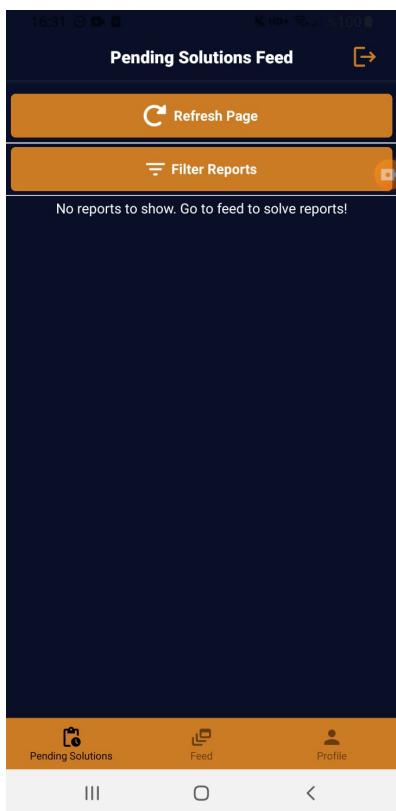
Officials can see reports assigned to them on this Assigned Reports feed screen. They can refresh the page or filter reports by pressing the appropriate buttons.



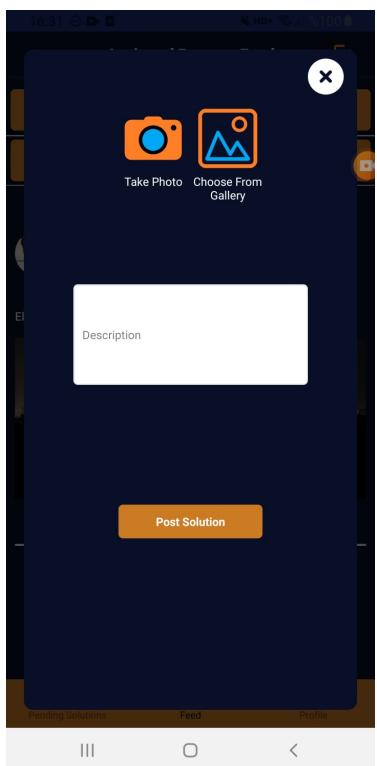
On this screen, the official can filter reports by Category and Description. Then the official clicks the Apply button and see the filtered reports.



On this screen, the official can see his/her profile and related information. The official can add a biography by clicking the "Add Bio" button.



On this screen, the official can see reports that he/she solved but the user did not review (accept or reject) yet.



On this pop up, officials can upload an image for the solution to be posted or take a new image by pressing the icons. The description input field also have to be filled before posting the solution. After posting the solution, a success alert is displayed.