



Bilkent University
Department of Computer Engineering

Senior Design Project

reporTown

High-Level Design Report

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

Supervisor: Dr. Ayşegül Dünder

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

Innovation Expert: Mehmet Çakır

Dec. 24, 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	3
1.1 Purpose of the System	3
1.2 Design Goals	4
1.2.1 Top Design Goals	4
1.2.2 Further Design Goals	5
1.3 Definitions, Acronyms, and Abbreviations	6
1.4 Overview	6
2. Current Software Architecture	7
2.1 Similar Applications	7
2.1.1 Şikayetvar	8
2.1.1 Instagram - Twitter	8
2.1.2 Tedaş	8
2.1.3 Municipality Applications	8
3. Proposed Software Architecture	9
3.1 Overview	9
3.2 Subsystem Decomposition	10
3.3 Hardware/Software Mapping	11
3.4 Persistent Data Management	11
3.5 Access Control and Security	12
3.6 Global Software Control	13
3.7 Boundary Conditions	13
3.7.1 Initialization	13
3.7.2 Termination	14
3.7.3 Failure	14
4. Subsystem Services	15
4.1 User Interface Layer	15
4.2 Business Logic Layer	15
4.2.1 Controller	15
4.2.1.1 User	15
4.2.1.2 Report	15
4.2.1.3 Auth	15
4.2.2 Service	15
4.2.2.1 User	16
4.2.2.2 Admin	16
4.2.2.3 Report	16
4.2.2.4 Notification	16
4.2.2.5 ML	16
4.2.2.6 Auth	16
4.3 Data Layer	16
4.3.1 Model	17
4.3.2 Repository	17

5. Consideration of Various Factors in Engineering Design	17
6. Teamwork Details	19
6.1 Contributing and Functioning Effectively on the Team	19
6.2 Helping Creating a Collaborative and Inclusive Environment	19
6.3 Taking Lead Role and Sharing Leadership on the Team	19
7. Glossary	20
8. References	21

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 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 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.

1.1 Purpose of the System

reporTown is a free to download and use mobile application for Android and iOS smart mobile devices that aims to bring citizens and appropriate institutions (governmental and private organizations) together to quickly solve city problems by creating reports and responding with solutions. reporTown is supported with machine

learning and computer vision technologies for efficient reach of the created reports to the institutions, aiding the citizen users in categorizing their problems and recommending them to the related organization. Additionally, reporTown seeks to encourage volunteer work for a collaborative town.

1.2 Design Goals

The following design goals are being considered in the development of reporTown project.

1.2.1 Top Design Goals

Usability

The target user of reporTown is the general public since everyone should feel included in aiding their cities; so the application should have a user-friendly UI/UX design with clear and coherent self-explanatory page components like button tags, and navigation between pages. In this way, users with any demographic background can effectively, intuitively and easily use the functionalities the application offers. This is important in maintaining and growing a community in the application because usability encourages users to keep using it, and for reporTown, the more users register and use actively, the better the city conditions get around the world. It should be compatible with most of the smartphone models, too.

Scalability

The application should be given enough resources (i.e. database maintenance, server issues, responsive machine learning models) to support the growing number of users since the users are expected to be all around the world from many cities. Thus, the application should not crash should it be overwhelmed with the amount of users and their requests, created reports, comments, and city entries (amount of data held in general).

Reliability

It is crucial that the created reports are visible to the appropriate organization, and that the reports are not marked inappropriately (i.e. when the report has not been solved yet but somehow is marked as solved) to cause reliability of the application to

drop to urge users to leave the application because their reports are not answered or faulty. Additionally, it is important for computer vision and machine learning models to do mostly accurate recommendations and classifications to prevent flooding for wrong organizations and wrong spam detection which in turn creates an extra delay in solving the reported problem.

1.2.2 Further Design Goals

Extensibility

The software architecture should be designed and developed to allow for future feature extensions on the current application system should new needs arise and new city problems, volunteering functionalities, communication extensions are needed to be included. The application should be well implemented for different mobile operating systems (OS). Design patterns should be used to allow easy integration of new features.

Performance

Although citizens can upload images to their reports from their gallery, meaning it is not mandatory to snap a photograph right on the scene, many citizens might want to report city problems that need urgent care and attention like traffic accidents on the go. For this reason, it is significant that reporTown offers adequate performance for instant report creation by citizens and responding to a report from the institution's scope. Moreover, the machine learning and computer vision parts should not block the flow of report creation by taking too long to respond.

Security

The application asks for permission for using location services and accessing the photo gallery of the user to generate the report for the specified location. The recordings, images, location should not be shared with a third party service, nor should it be distributed. reporTown should consider the privacy of the users and use encryption algorithms to protect user data from data breaches.

1.3 Definitions, Acronyms, and Abbreviations

User Interface (UI) Design: Process designers use to build interfaces in software, focusing on looks or style [1].

User Experience (UX) Design: It includes all aspects of the end-user's interaction with the company, its services, and its products [2].

Machine Learning: It is a branch of artificial intelligence; a computer science technology that uses certain algorithms to imitate learning processes like humans.

Computer Vision: Like machine learning, it is a branch of artificial intelligence that allows computers to derive information from digital visual input.

Application Programming Interface (API): A set of protocols for building an application software.

CRUD (Create, Read, Update, Delete) Operations: The four foundational functions of persistent storage.

1.4 Overview

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.

2. Current Software Architecture

Currently, there are applications that are utilized so that people can report problems about their cities, technological devices, etc in the market. However, they are either focused solely on reporting problems, or having social media features. The main focus of reporTown is to bring those two features together by maintaining solving problem focus as the most significant aim. While being able to communicate with authorities, they will additionally be able to respond to other people's reports.

2.1 Similar Applications

Information about similar applications in the market will be given in this section, at the end of the section features that make reporTown different from these applications will be explained.

2.1.1 Şikayetvar

The main purpose of Şikayetvar application is to report problems about products bought from the Internet, cargo operations, and services [3]. Users can create reports which are published only if admins of the application approve them. To solve the problem, authorities of the companies directly contact the user. Users can respond to the reports created by other users.

2.1.1 Instagram - Twitter

Even though Instagram and Twitter do not focus on solving problems, these applications can be used for that purpose. Their main purpose is to provide social networking among users, however, observing problems by authorities and solving problems are possible.

2.1.2 Tedaş

Tedaş has a reporting application for detecting malfunctioning street lamps. However, this application serves only for street lamps. Therefore, the broad range of this application is narrow.

2.1.3 Municipality Applications

Many municipalities own their own local applications to observe their cities and detect problems. However, these applications focus solely on detecting problems, therefore, do not contain social media features.

In many applications, because there are many problems, significant problems may be unnoticed or ignored. The reporTown's aim is to solve any report posted if it is possible to be solved by reminding authorities those problems repeatedly until they are solved. By assigning scores to users and authorities, the application aims to provide a more efficient environment as users do not want to lose scores which may indicate their actions or even themselves are not trustworthy.

Additionally, there is no application in which the volunteers who are not authorities but citizens who want to help and authorities work collaboratively. The reporTown

introduces a new user type which is “volunteer” that can help resolving reports by solving problems posted by other citizens.

Moreover, the reporTown has a categorization of reports feature which enables reports to be filtered, sorted, and displayed to the authorities, volunteers, and citizens more efficiently. By using machine learning and computer vision algorithms, reports are categorized and interlocutors are recommended.

3. Proposed Software Architecture

3.1 Overview

The subsystem is shown in the decomposition, layered, and uses module styles. There are three main layers which are User Interface, Business Logic, and Data. The User Interface layer interacts with the Controller module in the Business Logic layer. The modules in the Controller module use the modules in the Service module. The Service module uses the Repository module in the Data Layer for database queries.

3.2 Subsystem Decomposition

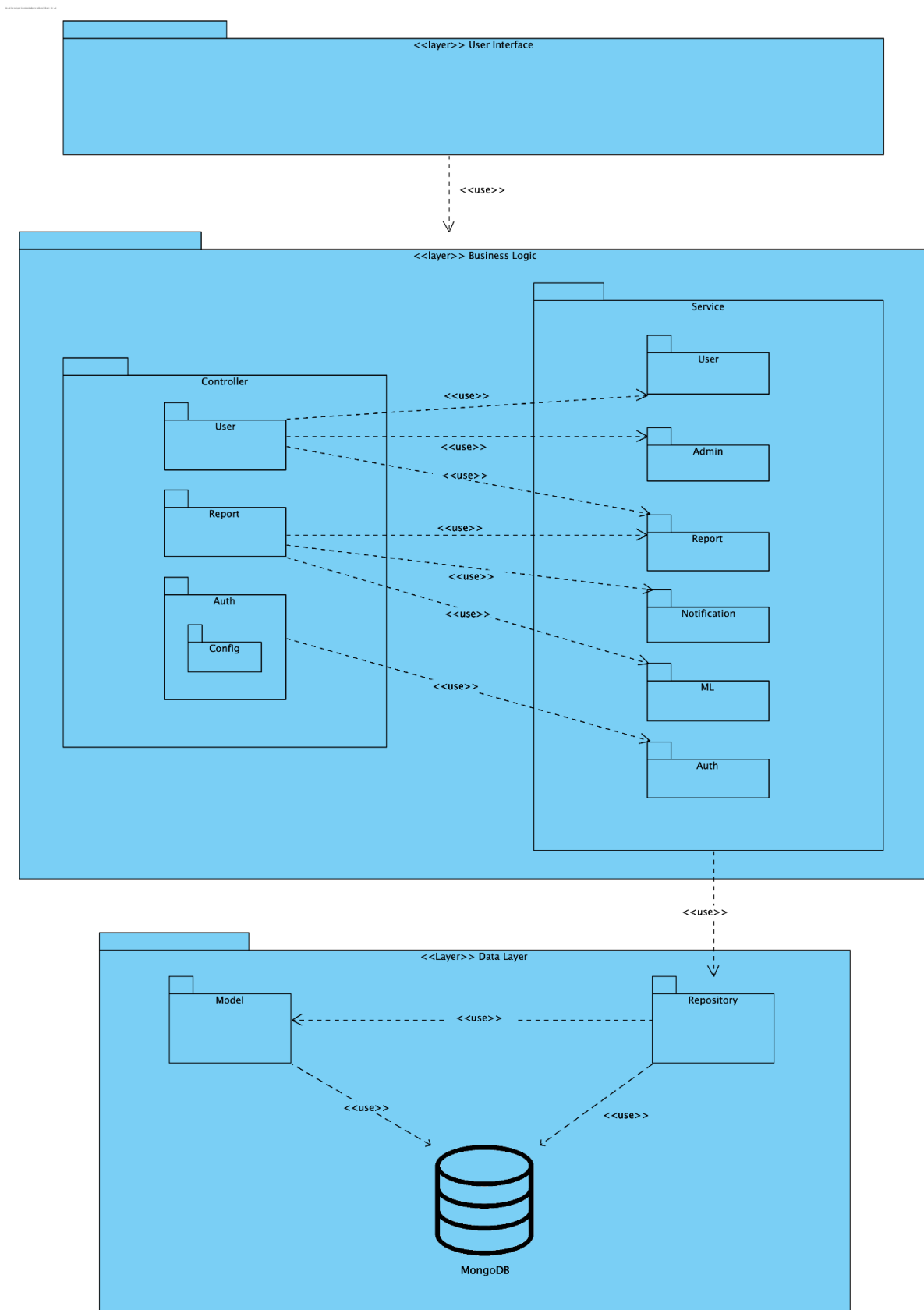


Figure 1. The diagram for the subsystem decomposition of reportTown

3.3 Hardware/Software Mapping

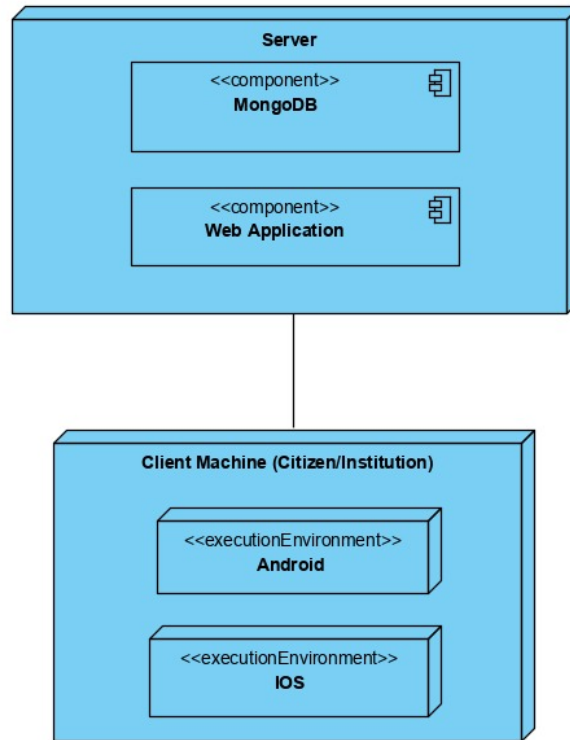


Figure 2. The diagram that shows the hardware/software mapping for reportTown

reportTown basically consists of two hardware items, they are client side and server side. Client side is the mobile device of the user, this user can be an institution or a citizen. This mobile device can be a mobile device with Android or iOS operating system. reportTown works on both as a React Native app. The server part covers the parts of the application running on the internet. This part consists of MongoDB where data such as user information, photos, report details are stored, and the web application running as the backend of the application. This web application undertakes very important tasks such as running database queries and providing API connections.

3.4 Persistent Data Management

We use the MongoDB database for the data management of the application. Users will register to the system with their username, user type, email and password and

this information is held on the database. Users must login to the application in order to use the application. Since user credentials (usernames, user types and passwords) are held in MongoDB database, the login process requires database connection. In addition, reports created by citizens and solutions are also held in MongoDB so that reportTown manages the reports and solutions on the server side.

Since citizens and officials are able to upload images to the system, data management for the images is also required. In order to store images, AWS's S3 service will be used. Uploaded images will be transferred to S3 buckets and the access to images will be restricted on AWS. In this way, images will be visible only on reportTown.

Settings of the application and some log files (in case of errors) are kept on the users' device. Settings of the application will be in JSON format and log files will be in .log format.

3.5 Access Control and Security

reportTown is a system where users have individual accounts. Therefore, in order to access the application and use its functions, it is necessary to first register with the application and then verify the account with email verification. Afterwards, users access the application by logging in. To log in, users must enter their username and password. These credentials are kept in a hashed form in the database to ensure security. Users stay logged in until they log out, and every time they open the application, they are directed to the main screen instead of the login screen. Again, the email verification phase ensures security and verifies that the e-mail addresses used by the users really belong to them.

In order to verify the institutional accounts, the institution that will open the account is requested to upload an official document. After this document is evaluated manually as soon as possible, a return is provided to the institution regarding whether the institution's account has been approved or not. If the account is verified, the account can be used.

reportTown respects users' private information and tries not to collect private information as much as possible. While collecting only name and surname information from citizens, the country and city of affiliation are requested from institutions in addition to the name of the institution.

reportTown requests the user's current location information so that the location of a report can be obtained easily and accurately. Users can allow this, or manually enter location information if they wish. Users must also approve camera access to use reportTown. Camera access is used to upload the photo of the report when creating a report. Users can also approve the gallery access request and choose a photo from their gallery.

3.6 Global Software Control

Our product has an event driven control system. Since our application will be used by a large number of people at the same time, users need fast and uninterrupted access to the servers. For this reason, event handlers should be used. The event handler will be used as follows: users' requests trigger the relevant part of the program asynchronously. Thus, users do not have to wait for others' requests to be done and they perform their requests quickly. When users send requests, this request will be sent to the relevant subsystem and completed there. The Server package in the Business Logic Layer will receive requests from the Controller package of the same layer. This request will be handled with the work of Server package and Data Layer.

3.7 Boundary Conditions

reportTown has three boundary conditions; start-up, shut-down and error behaviors which are described in detail below:

3.7.1 Initialization

Any type of user should have reportTown installed on their mobile device from Google Play if their device supports Android or from App Store if their device supports iOS for free to start using the application. The user must have an active internet connection. After the installation, a new user should sign up by filling in appropriate

information according to the user type or the existing user should log in to continue using the application. After the authentication process, the user lands on the main page and reporTown suggests the citizen user marks cities to see their report feed and asks for permission to use location and accessing photo gallery services for report creation. The user follows these steps if it is a first time user or the application data was erased and the application was redownloaded. From here, the user can make use of the features reporTown offers.

3.7.2 Termination

The user can exit from the application by closing its window like any other mobile application. This terminates running background processes for the application. This does not cause automatic logging out for the user; to log out, the user has to specifically click on the log out button before closing. The preferences and initialization data for the user is cached for starting the application again with the user's credentials if the user is still logged in at termination.

3.7.3 Failure

If there is no internet connection, the map in reporTown cannot be loaded and the user cannot see their location. Network dependent features such as posting a report, signing in and signing up cannot be used at all. Feeds and pages are displayed with their last modified (fetched) state until internet connection is restored. If the internet connection is interrupted, the new areas of the map (not yet loaded) cannot be loaded, the posting of a report or responding to a report can be interrupted in the middle, and any changes made are not stored (changes need network connection to apply). If the application crashes due to a heavy number of users or a failure in requesting data, and if the citizen user is in the process of creating a report, making a comment, joining a volunteer work, it will be lost. Similarly, if the officials are in the middle of responding to a report, the response will be lost when the application crashes. Database management errors might cause a report or a solution to be lost permanently. Machine learning result timeout errors might cause the application to not progress to a further stage and require restarting by the user.

4. Subsystem Services

4.1 User Interface Layer

This layer will contain the views that will be the client-side of the application.

4.2 Business Logic Layer

This layer will contain the subsystem services that will be integral to drive the business logic of the application.

4.2.1 Controller

The Controller package will be responsible for calling certain services according to client-side requests.

4.2.1.1 User

The User controller package will contain the controller classes that will call the User, Admin, and Report services. This package will handle user-related requests.

4.2.1.2 Report

The Report controller package will contain the controller classes that will call the Report, Notification, and ML services. This package will handle the report flow of the application.

4.2.1.3 Auth

The Auth controller package, which contains the Config package that does the authentication-related configurations, will contain classes that will call the Auth service. This package will handle authentication, authorization, and session flows.

4.2.2 Service

The Service Package will be responsible for the logical flow of the client-side requests.

4.2.2.1 User

The User service will contain the classes that will take part in the user-related logical flows such as management of the user documents in the database.

4.2.2.2 Admin

The Admin service will contain the classes that will take part in the admin-related logical flows such as banning a user from the platform, authorizing a constitution, etc.

4.2.2.3 Report

The Report service will contain the classes that will take part in the report-related logical flows such as posting a report, posting a solution, upvoting, commenting, etc.

4.2.2.4 Notification

The Notification service will contain the classes that will take part in the notification-related logical flows such as sending notifications, receiving notifications, etc.

4.2.2.5 ML

The ML service will contain the classes that will take part in the machine-learning-related logical flows such as category classification, spam detection, etc.

4.2.2.6 Auth

The Auth service will contain the classes that will take part in the authentication-related logical flows such as registering, logging in, session management, etc.

4.3 Data Layer

This layer is responsible for communicating with the database through queries and modeling.

4.3.1 Model

The Model package will contain the classes that model the data in the database to Java objects that can be utilized.

4.3.2 Repository

The Repository package will contain the classes that will be called by the services and make the queries for CRUD (create, read, update, delete) operations.

5. Consideration of Various Factors in Engineering Design

In order to implement a better product, various factors are significant in engineering design. We consider technological, economic, cultural, environmental factors as well as public health, safety and wealth.

Technological Factors

Since technology is continuously developing, we keep track of developments and try to use an innovative perspective by following new technologies. Also, we should investigate whether there are products similar to our product and follow a path where we differentiate from other products in implementation.

Public Health

Since our product aims to report the problems in the city, we also considered public health. If important problems in the city are not resolved quickly, they may cause accidents that pose a health threat. For this reason, the application also aims to solve such problems quickly by moving them to the top of the feed with users' upvotes.

Public Safety

Since this application aims to report the problems in a city, we first aimed to develop a secure application. For this, we set a design goal as described in section 1.2.2. In order for the application to be secure so that we protect the confidential data of users. For this, we set a design goal as described in section 1.2.2. In order for the application to be secure, we attach importance to authentication and keeping user

information safe. We also aim to do spam detection via NLP to prevent spam reports.

Social Factors

Since citizens in the city will use this application, we consider human factors as well as technical problems. For this purpose, we have an approach that this product should be user friendly. In order to make this application user friendly, we provide 2 options to the users as map usage and feed usage as mentioned in the introduction.

Public Welfare

Since public welfare is affected by the social and economic conditions of the country, we have taken into account factors such as server usage prices and the need for users to connect to the application over the internet while developing our product.

Economic Factors

We need to use cloud service for user data in the implementation phase of our product and after the deploy phase. For this reason, the Cloud service usage fee determined by the exchange rate was taken into account in the analysis and design part of the project.

Cultural Factors

Cultural factors do not have an important impact in the engineering design of our product.

Global Factors

Global factors do not have an important impact in the engineering design of our product.

Environmental Factors

Environmental factors do not have an important impact in the engineering design of our product.

6. Teamwork Details

This section introduces the details about how the team works on the project collaboratively while every team member contributes to the project as much as they are able to do.

6.1 Contributing and Functioning Effectively on the Team

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

6.2 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 will also help other teammates in their work. In this way, it is possible to transfer experience between peers, each team member gains experience in areas that they have no previous experience, the completion time of the work is reduced, and each team member has detailed information about different parts of the project instead of only knowing about their own part. All of this increases team members' ownership of the project, makes the process much more efficient, and creates a more productive working environment by increasing peer-to-peer interaction, friendship and intimacy.

6.3 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 are created and their leaderships are shared amongst the team, so that all team members are responsible for an equal load of work as a leader according to their interests, skills, and past experience. This way, each team member has a chance to work on their leadership skills. Each work package also has other members to work on one package in collaboration. Leaders have the responsibility to construct an outline of how the task will be undertaken (organization of work for the task).

7. Glossary

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

8. References

- [1] “What is user interface design?,” *The Interaction Design Foundation*. [Online]. Available: <https://www.interaction-design.org/literature/topics/ui-design>. [Accessed: 24-Dec-2021].
- [2] World Leaders in Research-Based User Experience, “The definition of user experience (UX),” *Nielsen Norman Group*. [Online]. Available: <https://www.nngroup.com/articles/definition-user-experience/>. [Accessed: 24-Dec-2021].
- [3] “Müşteri memnuniyetinin adresi - şikayetvar,” *Ana Sayfa*. [Online]. Available: <https://www.sikayetvar.com/>. [Accessed: 24-Dec-2021].