Akdeniz University

Department of Computer Engineering

Software Engineering Project

Project short-name: project title

Software Requirements Specification

Project Group Member Name, Project Group Member Name …

**Team Leader:** Team Leader’s Name

**Product Owner:** Product Owner’s Name

**Instructor:** Prof. Ümit Deniz ULUŞAR

<Date>

This report is submitted to the Department of Computer Engineering of Akdeniz University of the Software Engineering course CSE332.

|  |  |
| --- | --- |
| **Abbreviations** | |
| IP | Internet Protocol |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Contents

[1 Introduction 1](#_Toc1659209)

[1.1 Purpose 1](#_Toc1659210)

[1.2 Product Scope 1](#_Toc1659211)

[1.3 References 1](#_Toc1659212)

[2 Overall Description 1](#_Toc1659213)

[2.1 Product Perspective 1](#_Toc1659214)

[2.2 Product Functions 1](#_Toc1659215)

[2.3 User Classes and Characteristics 1](#_Toc1659216)

[2.4 Operating Environment 1](#_Toc1659217)

[2.5 Design and Implementation Constraints 1](#_Toc1659218)

[2.6 User Documentation 2](#_Toc1659219)

[2.7 Assumptions and Dependencies 2](#_Toc1659220)

[3 External Interface Requirements 2](#_Toc1659221)

[3.1 User Interfaces 2](#_Toc1659222)

[3.2 Hardware Interfaces 2](#_Toc1659223)

[3.3 Software Interfaces 2](#_Toc1659224)

[3.4 Communications Interfaces 2](#_Toc1659225)

[4 System Features 2](#_Toc1659226)

[4.1 System Feature 1 2](#_Toc1659227)

[4.1.1 Description and Priority 2](#_Toc1659228)

[4.1.2 Stimulus/Response Sequences 2](#_Toc1659229)

[4.1.3 Functional Requirements 3](#_Toc1659230)

[4.2 System Feature 2 (and so on) 3](#_Toc1659231)

[5 Use Cases 3](#_Toc1659232)

[5.1 Creating an new account 3](#_Toc1659233)

[5.2 Deleting an account 3](#_Toc1659234)

[6 Nonfunctional System Requirements 4](#_Toc1659235)

[6.1 Performance Requirements 4](#_Toc1659236)

[6.2 Safety Requirements 4](#_Toc1659237)

[6.3 Security Requirements 4](#_Toc1659238)

[6.4 Software Quality Attributes 4](#_Toc1659239)

[6.5 Business Rules 4](#_Toc1659240)

[7 Other Requirements 4](#_Toc1659241)

[8 Proposed System 5](#_Toc1659242)

[8.1 Overview 5](#_Toc1659243)

[8.2 Functional Requirements 5](#_Toc1659244)

[8.3 Non-functional Requirements 5](#_Toc1659245)

[8.4 Pseudo Requirements 6](#_Toc1659246)

[8.5 System Models 6](#_Toc1659247)

[8.5.1 Scenarios 6](#_Toc1659248)

[8.5.1.1. Patient and Doctor Member of the System 6](#_Toc1659249)

[8.5.2 Use-Case Model 6](#_Toc1659250)

[8.5.3 Object and Class Model 6](#_Toc1659251)

[8.5.4 Dynamic Models 7](#_Toc1659252)

[8.5.5 User Interface 7](#_Toc1659253)

[8.5.6 Test 7](#_Toc1659254)

[9 References 7](#_Toc1659255)

[10 Appendix A: Glossary 7](#_Toc1659256)

# Introduction

## Purpose

The purpose of this document is to explain the requirements of our application. You’ll find the answer for system structure, application constraints, user interface, and Interaction between entities. This document is to be demonstrated for the customer’s approval or as a reference for further development.

## Product Scope

The [NameOfApp] protects people from cancel culture. The toxicity of cancel culture destroyed thousands of people's lives, they had to leave their job, school, and their city but these are may have been the luckiest ones, cause lots of people couldn’t take it, attempted suicide and many of them died. The new is of Cancel Culture continue to destroy Celebrities because of something that they said 10 years ago. Kevin Hurt – the most successful stand-up comedian judged by some twits which he published almost 10 ago. Last Year, Kevin was hosting Oscars and that was the beginning of everything. The haters just dig down 10 years and find some joke about “Gay People” and they made pressure on the council of Oscars to get hosting from Kevin. And they did. They got the change of hosting from Kevin. Could you imagine that? Today we said something and 10 years later someone find them, hey, you said this so you can’t do this now. I wish I were told you a fantastic story but this happened in the last few years. And It continues to happen. Toxic people abusing freedom of speech. We are going to let people say things anonymously to protect them. People will be able to socialize as usual but when they want to say something and if they aren’t comfortable with they will be able to publish their thoughts anonymously.

## References

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Overall Description

## Product Perspective

 Our application mainly made of 2 Parts:

1- Cross-platform Frontend UI

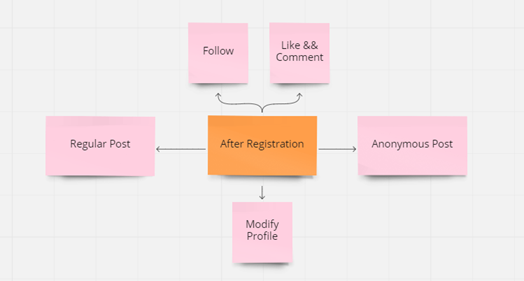
2- .Net Core backend

UI going to be written in Flutter so The application will be available both on IOS and Android. We are going to use Flutter responsive utility to run Application also in Web Browsers. Frontend Communicate with backend on .Net Core Web APIs. Data will be stored in PostgreSQL but in the future, this may change. If the Application needs Real-time data to follow, we are also planning to use Web sockets. Communication with the Database will be established in the backend and the frontend won’t be able to communicate database directly no matter what. In our application Data Flow will be from Database to .Net Core then from there to Front-end or vice Versa.

## Product Functions

The frontend application will be written in Flutter and for All platforms, it will preserve its functions. Some small changes may occur while we are developing Applications like using a webcam instead of a phone camera, disabling taking a picture, etc.

* Users can share a post
  + Image
  + Text
* Users can share posts anonymously
* Users can follow other users
* Users can like and Comment on posts
* Users can modify their profile



## User Types and Characteristics

Our application [Twitter] can use many types of users over the age of 18. Real or legal persons will be able to use the application. All users who have any technology that can use the application will be able to subscribe to it. Our app will work in the same way on the mobile and the web side.

All users can use the same features and functions.

* Be able to share (Anonymous or Own ID)
* Search for existing users
* To be able to follow other users
* To be able to send direct messages between users

## Operating Environment

Relational Database: PostgreSQL

Operating system: Any device used with iOS or Android and an operating system running a web browser

Development Platform: Visual Studio Code, Visual Studio, Android Studio

## Design and Implementation Constraints

* Flutter technology will be used while creating the website and mobile application. Flutter is an application development kit that has been very popular recently, especially in mobile application development. We specifically chose Flutter because of its web support. As in the mobile application, after the application is written from a single source code with Dart programming language, Flutter with using "build" can be used to create HTML, CSS, JS, etc. it creates web files for us. However, as this is the first time we will be using this technology, it will cause us to spend more time learning Flutter.
* Tweets that users want to share are restricted to a maximum of 300 characters. Such a method was followed because it was noticed that the tweets were concise and more creative tweets were posted with limited opportunities.
* We noticed that users sometimes have problems expressing their opinions and want to tweet more freely while hiding their identities. For this reason, the user was allowed to tweet anonymously by hiding his identity with the restriction of only one tweet per day.
* Since no one in the team has a history of working in the back-end the part before, we do not have much information in this part, we will need to improve ourselves quickly.
* Since we do not have an IOS device, we cannot test our application on IOS devices.
* The user must be at least 18 years old to use our services.
* Users are not allowed to register with the same email address or phone number.
* It is important that users enter their credentials correctly when creating an account. Otherwise, if users provide incorrect information (for example, not complying with age restrictions), this reduces the reliability of the application.

## User Documentation

In the "Help Center" section of the menu, the application contains useful texts for users in many sub-titles about the product.

Our subtitles and their explanations:

Manage your account - This section assists the user in informing about private accounts, adding a phone number to the account, and updating the e-mail address.

The procedure for using XXXX(app name) - This section assists the user in registering to the application, how to create a post within the application.

Security and Safety - In this section, our user is provided with assistance on how your personal data is protected, privacy policies, sensitive content (specific age restriction or violent posts), and harassment cases.

Rules and policies - In this section, we assist our users by explaining why we need cookies, what our notifications do, and how we identify sensitive media.

## Assumptions and Dependencies

Assumptions

* Data inaccessibility to the user if the Database's Web Server stops responding.
* The user's phone and operating system are not suitable for our application.

Dependency

* Flutter libraries required within the application
* Requires using Android Jellybean v16 and higher. On the iOS side, the necessity of iOS 8 and later versions.

# External Interface Requirements

## User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

After first installation of mobile application , A welcome page appears on screen. On Welcome page users see register or login button, according to their choice they will be redirect to desired page to perform their actions. In web application we will present both login and welcome page at the same time, but to register we will redirect the users to register page.

From here the user interface mainly focuses on mobile application. Web version of application will be extended from mobile application so everything in mobile also be presented in web UI.

After users logged in they are redirected to their home page which is consist of posts of people who they chose to follow. Left hidden menu will contain all navigation button for user to move around. Navigation bar will contains buttons for: users profile, settings and other features of application.

Whenever users make a long top scroll a search bar will appear. Search bar firstly finds people but may in the future version it may include posts.

## Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

## Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

* .Net Core 3.1x: Backend written .Net Core 3.1x , all others libraries have to be compatible with this version of .Net Core.
* Entity Framework 3.1x: Application written with code first approach , in order to this entity framework will be used to map our object to database.
* Entity Framework Tools and Design: Needed for managing Entity framework.
* Npgsql Entity Framework Provider 3.1x: It needed for connect database and backend
* PostgreSQL: is main data storage for application. Data Accessing and management parts will be handled with .Net Core, With .Net Core APIs we will send and get data from users. Frontend never communicate with database directly. Data Flow will be from Database to backend from there to frontend.

* SignalR 3.1x: May application need real-time access in the future .
* IIS , Azure or AWS : Application Backend may launch on ISS directly but we also thinking use cloud functionality in our application.
* SMTP Server : It’s need to send mail to users in order to verify their emails.

## Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

# System Features / Requirements

ID: FR1

TITLE: Download mobile application

DESC: A user should be able to download the mobile application through either an application store or similar services on the mobile phone. The application should be free to download.

RAT: In order for a user to download the mobile application.

DEP: a device that support the app

ID: FR3

TITLE: User registration

DESC: Any user that can access application should be able to register through the mobile application. The user must provide user-name, password and phone number. The user can

RAT: In order for a user to register on the mobile application.

DEP: FR1

ID: FR4

TITLE: User log-in

DESC: Given that a user has registered, then the user should be able to log in to the application. The log-in information will be stored on the phone and cookies and in the future the user should be log in automatically.

RAT: In order for a user to register thapplication.

DEP: FR1, FR3

ID: FR5

TITLE: Retrieve password

DESC: Given that a user has registered, and he doesn’t remember his/her password with, user should be able get an a code for the change his password

RAT: In order for a user to retrieve his/her password. DEP: FR1

ID: FR6

TITLE: Search

DESC: Given that a user is logged in to the mobile application, user should be able to search other users, according to several search options.

RAT: In order for a user to search for other users.

DEP: FR4

ID: FR7

TITLE: Customize Profile

DESC: On the application, a user should have a profile page. On the profile page a user can edit his/her information, which includes the password, e-mail address and phone number. The application may contain dark and light mode.

RAT: In order for a user to have a profile page on the application.

DEP: FR1

ID: FR9 Feature: Disable an account

ID: FR10 Feature: Publish Post

ID: FR11 Feature: Flow Other Users

ID: FR12 Feature: Like and Comment posts

ID: FR13 Feature: share anonymous posts

ID: FR14 Feature: Private chat

# Use Cases

## Creating an new account

The purpose of this use case is to describe the procedure of creating an account in the system

**Pre-conditions:**

Phone Number

**Post-conditions:**

• An account is created for the user

**Basic Flow:**

1. The user is in the homepage

2. The user clicks on “Sing UP” link and is taken to the account creation page

3. The user enters all their information and clicks “Sing IN” button

4. If the phone already exits, an error message appears alerting the user and asking the user to choose another phone number. If the username does not exist, a confirmation appears letting the user know that the account has been created

## Disable an account

The purpose of this use case is to describe the procedure of creating an account in the system

**Pre-conditions:**

Accessing an Account and the phone associated with account

**Post-conditions:**

• An account is created for the user

**Basic Flow:**

1. The user is in the homepage

2. The user clicks on “Sing UP” link and is taken to the account creation page

3. The user enters all their information and clicks “Sing IN” button

4. If the phone already exits, an error message appears alerting the user and asking the user to choose another phone number. If the username does not exist, a confirmation appears letting the user know that the account has been created

# Nonfunctional System Requirements

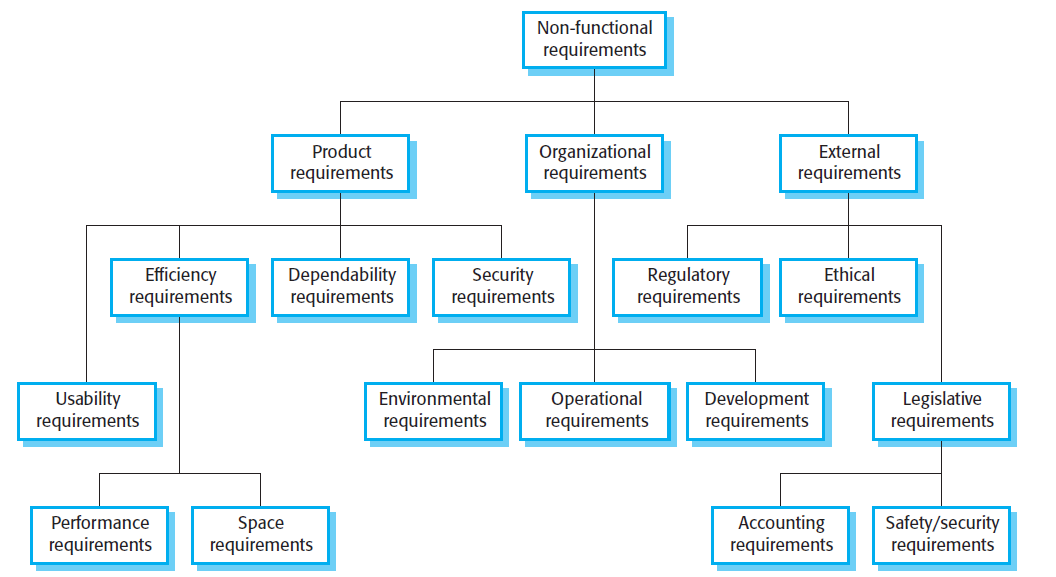


Figure 1 An example figure

## Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

ID: NF1

TAG: Response Time < ID: FR6 TITLE: Mobile application - Search >

GIST: The fastness of the search

SCALE: The response time of a search

METER: Measurements obtained from 1000 searches during testing.

MUST: No more than 2 seconds 100% of the time.

WISH: No more than 1 second 100% of the time.

ID: NF2

TAG: Response Time for < ID: FR4 TITLE: User log-in - Mobile application >

GIST: The fastness of the search

SCALE: The response time of a search

METER: Measurements obtained from 1000 searches during testing.

MUST: No more than 2 seconds 100% of the time.

WISH: No more than 1 second 100% of the time.

## Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>

## Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

## Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

## Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

# Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

# References

For the references you can use programs like Zotero (https://www.zotero.org/)

Object-Oriented Software Engineering, Using UML, Patterns, and Java, 2nd Edition, by Bernd Bruegge and Allen H. Dutoit, Prentice-Hall, 2004, ISBN: 0-13-047110-0.

# Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>