Blogosphere

Software Requirements Specification

14.04.2023

Samet CAN 150120528
Kadir Berk YAĞAR 150120016
Ela Nur ALP 150119673
Hakan UZ 150119001
Ali KOŞAR 150119003
Berkkan RENÇBER 150119011
Murat ÖZCAN 150119008

Prepared for
CSE3044 Software Engineering Term Project

Table of Contents

1. Introduction

In this Software Requirements Specification (SRS), the website "Blogosphere" is introduced. This document contains a lot of information and materials from the purpose of the website to its technical details. It is detailed in such a way that a software engineer can design its application by examining it.

1.1 Purpose

The purpose of this Software Requirements Specification document is to provide information about the project, to determine the requirements and to explain in detail without misunderstandings. This document is aimed primarily at the developers and scalers of the project, and then at other interested parties.

1.2 Scope

First of all, Blogosphere is an information website with blog and social media features about the ecosystem and environment that prioritizes accurate information. Blogosphere receives information and comments from many users, however, these people must have certain permissions.Users with authority in Blogosphere can create and share content related to the environment, members can comment on these contents and admins can regulate users' authorizations. In this way, Blogosphere will provide accurate information about the environment.

1.3 Definitions, Acronyms, and Abbreviations

Blogosphere: Subject of this report. A website aimed to gather like minded people
Visitor: Those who did not log in to the website.
User: Those who did log in to the website.
Editor: Those who create content.
Content Creator: Editor.
Admin: Those who have the general authority of the website.
Database: A software that stores data.
SQL: Structured Query Language
MSSQL: Microsoft SQL Server
UML: Unified Modeling Language

1.4 References

- IEEE Computer Society SRS Book (830-1998)
- https://www.geeksforgeeks.org/functional-vs-non-functional-requirements
- https://www.lucidchart.com/pages/data-flow-diagram
- https://www.stickyminds.com/article/state-transition-diagrams

1.5 Overview

Three primary subdivisions and several subdivisions make up the remaining SRS. The "General Description" section, which is the first primary section, contains an explanation of the broad variables that influence the product and its specifications. Second, "Specific Requirements" contains the project's specifications, which are used to direct the development, testing, and design of software. The last section, "UML Classes," contains software and usage case illustrations for our product.

2. General Description

Blogosphere is a platform where you can read and get information about the articles and current contents on topics such as environment, climate and pollution. The platform offers you to be more conscious about the environment and the contents behind the scenes. You can also browse trending topics if you want.

2.1 Product Perspective

Although there are other websites and social media platforms that discuss environmental issues, Blogosphere is distinguished from other projects by giving importance to the accuracy of the content that users publish. To ensure this, users will be classified as administrators, content producers, members, and visitors. With this authorization, there will be restrictions in the production and comment section. Visitors will not be permitted to leave comments or share their opinions on the subject matter. Thus, the blogosphere will ensure the reliability of the information with authorization.

2.2 Product Functions

Blogosphere will allow its users to register to its system to let them give impressions about topics. They will do that by both leaving emotes and comments for the topic and they may even discuss it with other members of the community. Editors that are chosen among them will be able to produce new content that is trusted. They will have access to rich-text creation tools so that they can express their opinions and the content freely. Members will be notified for the categories that they are interested in and for the updated discussions they participated in.

2.3 User Characteristics

- The required level of education of the user - None
- The required experience of the user - Basic Internet Knowledge
- The required technical expertise of the user - None

2.4 General Constraints

- Blogosophere receives data(email and password) from users in accordance with legal permissions.
- The hardware support provided by the hosting server where this project will be installed will create a limit for the website.
- Blogosophere is designed to serve many users at the same time.
- The system has authorization and offers certain features according to authorizations.
- The blogosphere is written using .Net and is limited by the requirements of that language.
- Blogosphere always runs on an online hosting.
- Even if the system has received the password from the user, it encrypts it and saves it in the database.

2.5 Assumptions and Dependencies

Assumptions:

- The user is using one of the internet browsers that supports the latest web technology.
- The user must have basic internet usage skills.

Dependencies:

- The user must have a device that can connect to the internet. (smartphone, tablet, computer)
- The user must have an internet connection to use the website.

3. Specific Requirements

The SRS's biggest and most significant section is this one. The D-requirements are provided in this part and are used to direct the development, testing, and design of the project's software.

3.1 External Interface Requirements

3.1.1 User Interfaces

- Front-End Software: Angular
- Back-End Software: .Net, MS SQL
- The Blogosphere web-based application has some interfaces to keep its relations with its users intact. These interfaces will enable users to be on a secure platform and offer a wide variety of usage options.

3.1.1.1 Login Interface

This interface allows logged out users to log into their Blogosphere accounts. If the user has not logged out of their account, it also allows users to log in directly to their accounts with the help of cookies. It is important for users to log in to leave comments and likes. If the user is a content producer, he cannot share anything without logging into his account.

3.1.1.2 Register Interface

The registration interface is an interface created for new users to create a Blogosphere account. Here, users' accounts are created by obtaining information such as username, password. If there are already created accounts, necessary referrals are made.

3.1.1.3 In-App Interface

3.1.1.3.1 Homepage

On the homepage of Blogosphere, there is a button where the user can go to his profile, a scroll where the shared news is listed, a category tab to read the contents in a specific category, the titles of trending contents and a search bar where he can search for contents.

3.1.1.3.2 Profile

Profile is the part where the user's information is displayed. Here, the user will be able to see account information and make changes. You can also access his comments and likes from here. In addition, if he is a content producer, he can see the contents he publishes and make edits on them.

3.1.1.3.3 Post Content

This tier is a feature that will only be available to content creators. Here the user can post new contents.

3.1.1.3.4 Category

The category exists to enable the user to reach the specific contents they are interested in. From here, climate, pollution, etc. access the contents.

3.1.1.3.5 Trending Content

It is a structure created to attract the attention of other users to the contents that has recently attracted great interest by users.

3.1.1.3.6 Search Content

It is the structure that will exist so that the user can find contents more specifically other than categories. It aims to present the contents about the search made to the user.

3.1.2 Hardware Interfaces

Since Blogosphere is a web-based platform, the application must run via the Internet. Application requires any specific hardware interfaces used to access the internet. These are:

- Desktop computers
- Laptops
- Tablets
- Smartphones

3.1.3 Software Interfaces

All users will interact with the frontend part of the app via an internet browser. The frontend will send requests to the API Service and will perform depending on the response. The API Service will evaluate the request and depending on the operation that is requested, it either will send a query request to the database to handle that specific data or generate a basic response.

- Development Environments: Visual Studio, Visual Studio Code
- Frontend Technologies: Angular
- Backend Technologies: .Net
- Database Technologies: MSSQL

3.1.4 Communications Interfaces

The Client uses web connection from mobile or computer device to use Blogosphere.

3.2 Functional Requirements

S.No	Features	Functionality	Description
3.2.1	Authorization	The users of the site are authorized by the administrators.	Admins can view all users on their own admin page. On this page, s/he can give the desired authority to the user s/he chooses.

S.No	Features	Functionality	Description
3.2.2	Login	Members, editors and admins login to access their functions.	Members, editors and admins can access their own pages using their login information. On this page, they see their own information, transactions and options that are restricted according to their authority.
3.2.3	Loading the Home Page	This page is loaded the first time the site address is accessed.	The home page is loaded according to the user's knowledge and authorization, according to the first time or regular access to the site.
3.2.4	Add Comment	Comments can be added to the content.	Authorized users can add comments under the content. These comments are visible to anyone who viewing the content.
3.2.5	Loading the User Page	The user's page is loaded.	The user information is fetched from the database and the functional user page is loaded.
3.2.6	Registration	Visitors register to use additional functions.	Visitors can only see the contents. To add comment and like exist contents register to the website.
3.2.7	Add Content	Editors can add new content.	There are two types of user that member and editor. Editors can publish new content on the website. The published content is saved the database.
3.2.8	Like	Members can like a comment.	Members give likes to the content that interests them and that they like. Likes count are stored in database.

3.3 Non-Functional Requirements

Non-functional requirements are crucial for the program's performance, reliability, and continuous operation.

3.3.1 Performance

- The system should respond to the user's requests as soon as possible.
- It is important to bring information from the database in the smallest possible timeframes.
- Performance should be web-optimized across platforms such as mobile, desktop, and tablet.

3.3.2 Reliability

Blogosphere is a sharing platform designed to provide a reliable and consistent user experience for all users. To ensure the reliability of the platform, measures such as regular backups, high availability and security measures will be taken to protect user data and content from unauthorized access.

3.3.3 Availability

Blogosphere should be accessible almost every time during the day so that every single person around the globe will not experience any downtime. We are expected to write resilient code that covers most of the exceptions.

3.3.4 Security

- Only the admin will be able to access most of the data in the database.
- User passwords will be stored with cryptology methods.
- API will be used to provide more secure data flow between server and client.

3.3.5 Maintainability

First of all, Blogosphere has a very understandable file structure and code flow, as it is written with OOP, the Design Pattern, and the up-to-date frameworks .Net and Angular, which are very compatible with each other. In addition, with the class structure, error recording and detection codes and code blocks can be tested together or separately and can be easily changed. In the project, the server and the user side are clearly separated and provide security with a partially open database Microsoft SQL Server to the user.

3.3.6 Portability

As a website, Blogosphere can be accessed on any device that has an internet browser (Windows, Linux, IOS, Android etc).

3.4 Inverse Requirements

- Users cannot use the comment and like features without registering or logging in. They can only read what is shared on the website.
- Even if users log in, they cannot share new contents if they are not authorized. They can only use the like and comment features.

3.5 Design Constraints

There are some restrictions on this Blogosphere web-based project. These can be listed as follows:

- Time constraint: Blogosphere has to end within a time determined by our instructors.
- Budget constraint: The Blogosphere project does not have any budget.
- Reliability constraint: Since the blogosphere is a content source on issues such as climate, forests, pollution and the environment, the reliability of these contents is a criterion. We believe that this situation will be resolved by identifying the content producers.
- Blogosphere should work web-based on devices such as mobile, desktop and tablet.
- Front-End Software: Angular
- Back-End Software: .Net, Mssql

3.6 Logical Database Requirements

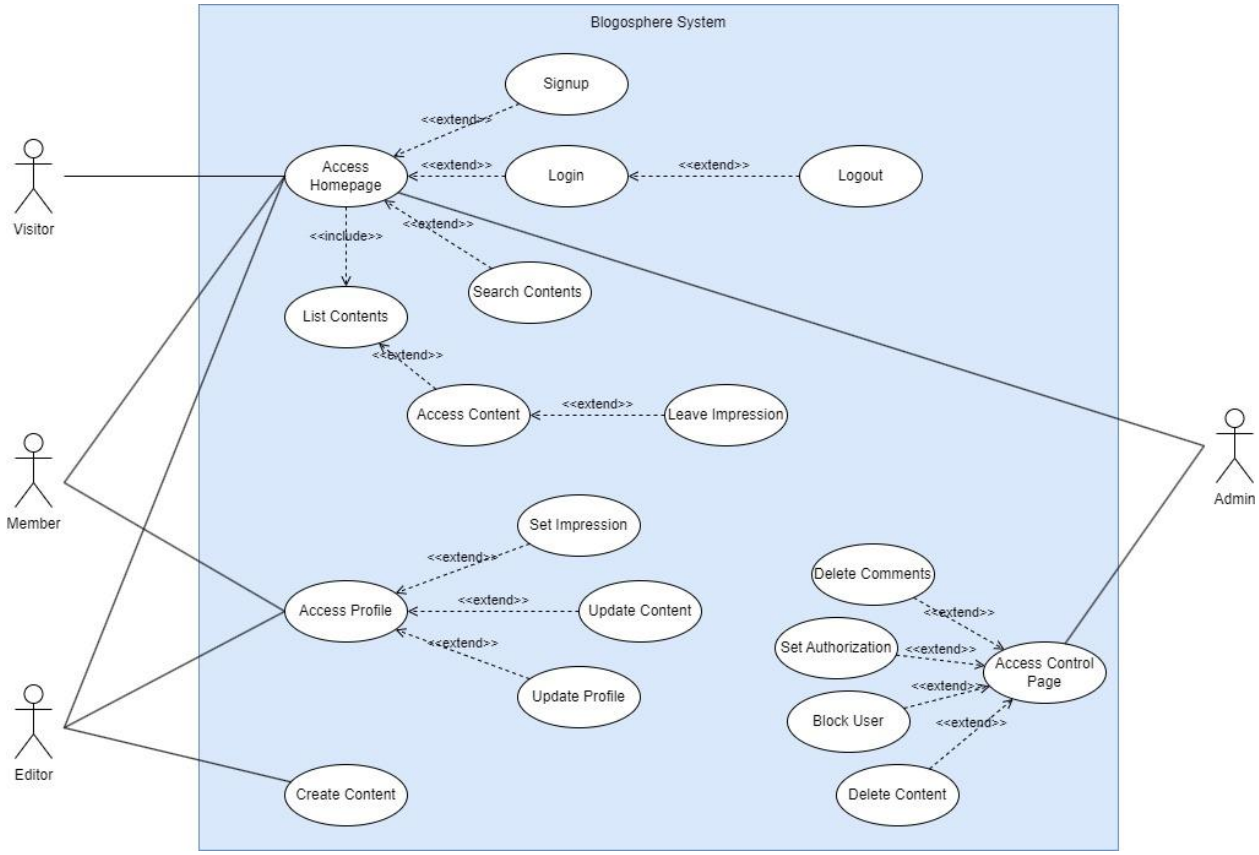
Blogosphere will utilize a relational database management system (RDBMS) to store and manage its data. Microsoft SQL Server (MSSQL) will be used as our database management system due to its reliability, scalability, and performance.

The database will be used to store information about users, articles, comments, and other related entities. There are some logical requirements for the database:

- Supporting standard data formats, such as text, numbers and dates,
- Ability to store large amounts of data,
- Supporting a wide range of data integrity,
- Retaining data for as long as it is needed by the application.

4. UML Diagrams

4.1 Use Cases



4.1.1 Access Homepage

Actors: Visitor, Member, Editor, Admin
Description: All users who will log in to our site can view the homepage. Users who are members can view the homepage, as well as visitors without the condition of being a member.
Includes: List Contents
Extends: Search Contents, Login, Signup

4.1.2 List Contents

Actors: Visitor, Member, Editor, Admin
Description: When users log in to the homepage, the contents will be displayed.
Includes: Access Content
Extends: None

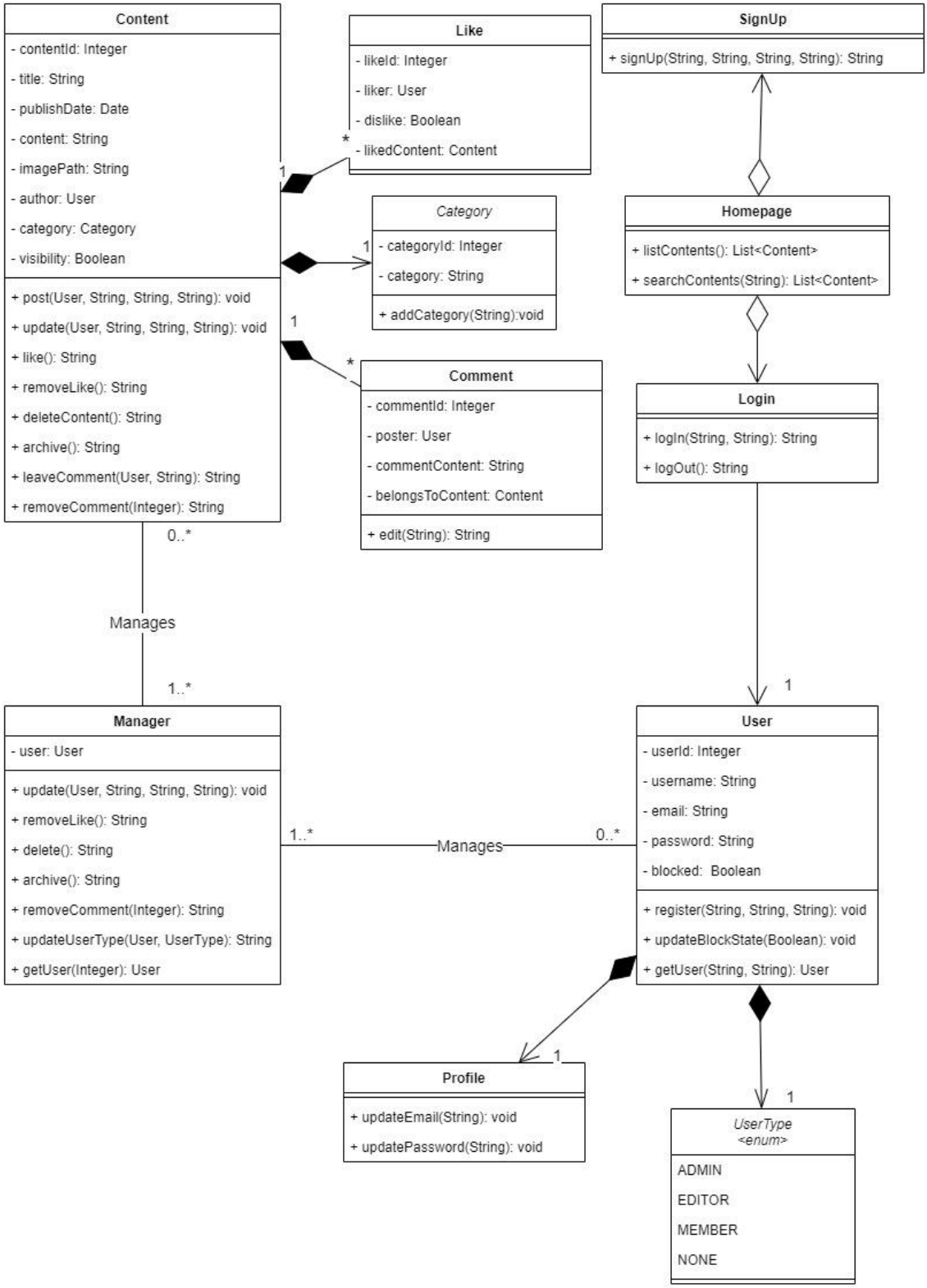
4.1.3 Access Content

Actors: Member, Editor, Admin
Description: Actors will be able to view the details of the content they have listed.
Includes: None
Extends: Leave Impression

4.1.4 Leave Impression
Actors: Member, Editor, Admin
Description: Actors will be able to like and comment on the content they view.
Includes: None
Extends: None
4.1.5 Search Contents
Actors: Visitor, Member, Editor, Admin
Description: When actors access the homepage, they will be able to search for the content they want on this page, thanks to the search button.
Includes: None
Extends: None
4.1.6 Signup
Actors: Visitor
Description: Visitors can become a member via the sign up button in order to use the specific functionalities such as comment and like.
Includes:None
Extends: None
4.1.7 Login
Actors: Member, Editor, Admin
Description: Actors will be able to login to the site when they view the homepage.
Includes: None
Extends: Logout
4.1.8 Logout
Actors: Member, Editor, Admin
Description: Users who have logged in to the site will be logged out of the site with the log out button.
Includes: None
Extends: None
4.1.9 Access Profile
Actors: Member, Editor, Admin
Description: Users who are members of the website will have their own profile pages. And they will be able to access their own profiles via the My Profile button.
Includes: None
Extends: Set Impression, Update Content, Update Profile
4.1.10 Set Impression
Actors: Member, Editor, Admin
Description: Users will be able to edit the likes and comments they have left on the posts from their own profile pages.
Includes: None
Extends: None
4.1.11 Update Content
Actors: Editor, Admin
Description: Users will be able to edit the content they have previously shared from their profile pages.
Includes: None
Extends: None

4.1.12 Update Profile
Actors: Member, Editor, Admin
Description: Actors will be able to edit their profiles as they wish.
Includes: None
Extends: None
4.1.13 Create Content
Actors: Editor
Description: Actors will be able to share the content they want thanks to the create content button.
Includes: None
Extends: None
4.1.14 Access Control Page
Actors: Admin
Description: Only admin will be able to access the access control page to ensure the security of the site
Includes: None
Extends: Delete Comments, Set Authorization, Block User, Delete Content
4.1.15 Delete Comments
Actors: Admin
Description: The admin has the right to control the content shared by the users and if they wish, they can delete any comments made on this content via the access control page.
Includes: None
Extends: None
4.1.16 Set Authorization
Actors: Admin
Description: Admin will be able to control the users' ability to share content. So, the process of authorizing users is carried out by the admin through the access control page.
Includes: None
Extends: None
4.1.17 Block User
Actors: Admin
Description: Admin will be able to block the users they want via the access control page.
Includes: None
Extends: None
4.1.18 Delete Content
Actors: Admin
Description: Admin will be able to delete the content shared by users through the access control page.
Includes: None
Extends: None

4.2 Classes / Objects



4.2.1 <Content>

4.2.1.1 Attributes

- - contentId: Integer
- - title: String
- - publishDate: Date
- - content: String
- - imagePath: String
- - author: String
- - category: Category
- - visibility: Boolean

4.2.1.2 Functions

- + post(User, String, String, String): void
- + update(User, String, String, String): void
- + like(): String
- + removeLike(): String
- + deleteContent(): String
- + archive(): String
- + leaveComment(User, String): String
- + removeComment(Integer): String

4.2.2 <Like>

4.2.2.1 Attributes

- - likeId: Integer
- - liker: User
- - dislike: Boolean
- - likedContent: Content

4.2.3 <Category>

4.2.3.1 Attributes

- - categoryId: Integer
- - category: String

4.2.3.2 Functions

- + addCategory(String): void

4.2.4 <Comment>

4.2.4.1 Attributes

- - commentId: Integer
- - poster: User
- - commentContent: String
- - belongsToContent: Content

4.2.4.2 Functions

- + edit(String): void

4.2.5 <Manager>

4.2.5.1 Attributes

- - user: User

4.2.5.2 Functions

- + update(User, String, String, String): void
- + removeLike(): String
- + delete(): String
- + archive(): String
- + removeComment(Integer): String
- + updateUserType(User, UserType): String
- + getUser(Integer): User

4.2.6 <SignUp>

4.2.6.1 Attributes

- No Attribute

4.2.6.2 Functions

- + signUp(String, String, String, String): String

4.2.7 <HomePage>

4.2.7.1 Attributes

- No Attribute

4.2.7.2 Functions

- + listContents(): List<Content>
- + searchContents(String): List<Content>

4.2.8 <Login>

4.2.8.1 Attributes

- No Attribute

4.2.8.2 Functions

- + logIn(String,String): String
- + logOut(): String

4.2.9 <User>

4.2.9.1 Attributes

- - userId: Integer
- - username: String
- - email: String
- - password: String
- - blocked: Boolean

4.2.9.2 Functions

- + register(String,String,String): void
- + updateBlockState(Boolean): void
- + getUser(String,String): User

4.2.10 <Profile>

4.2.10.1 Attributes

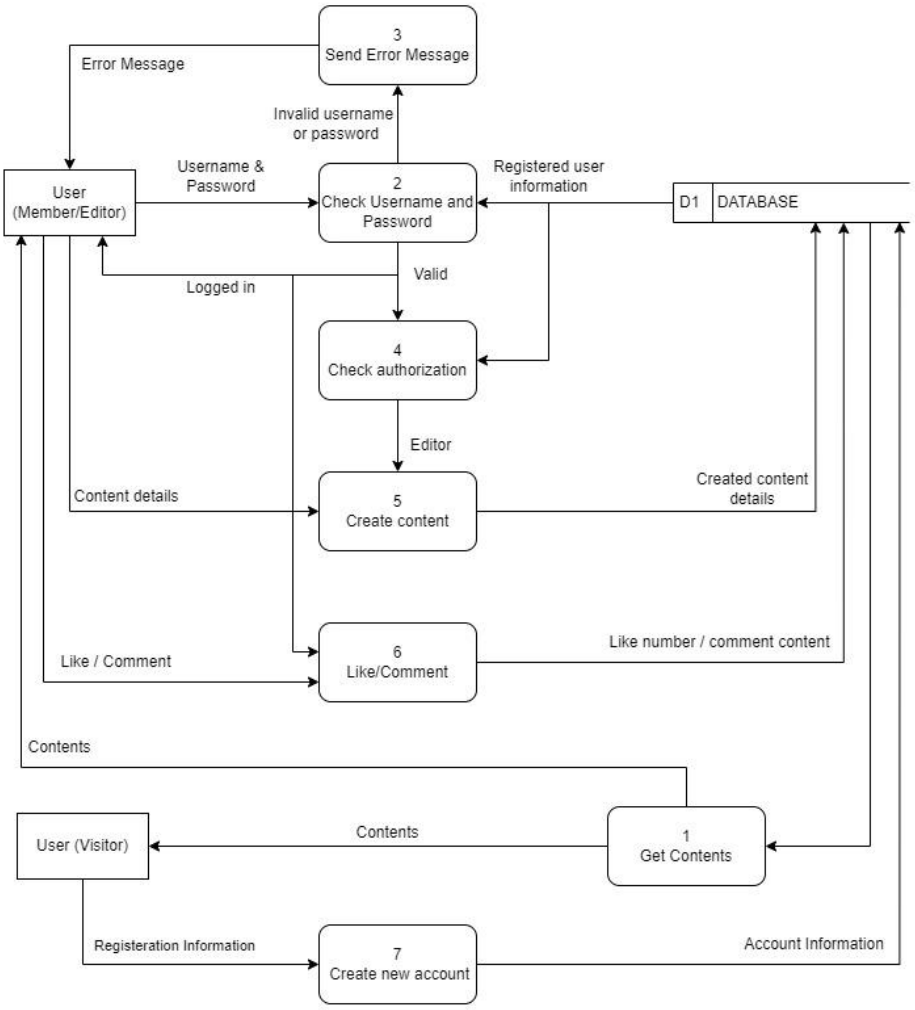
- No Attribute

4.2.10.2 Functions

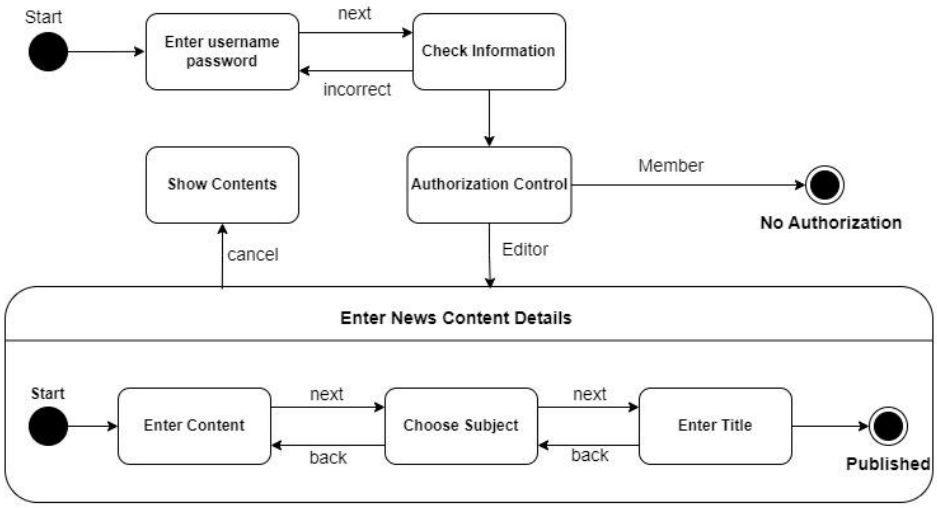
- + updateEmail(String): void
- + updatePassword(String): void

4.3 Sequence Diagrams

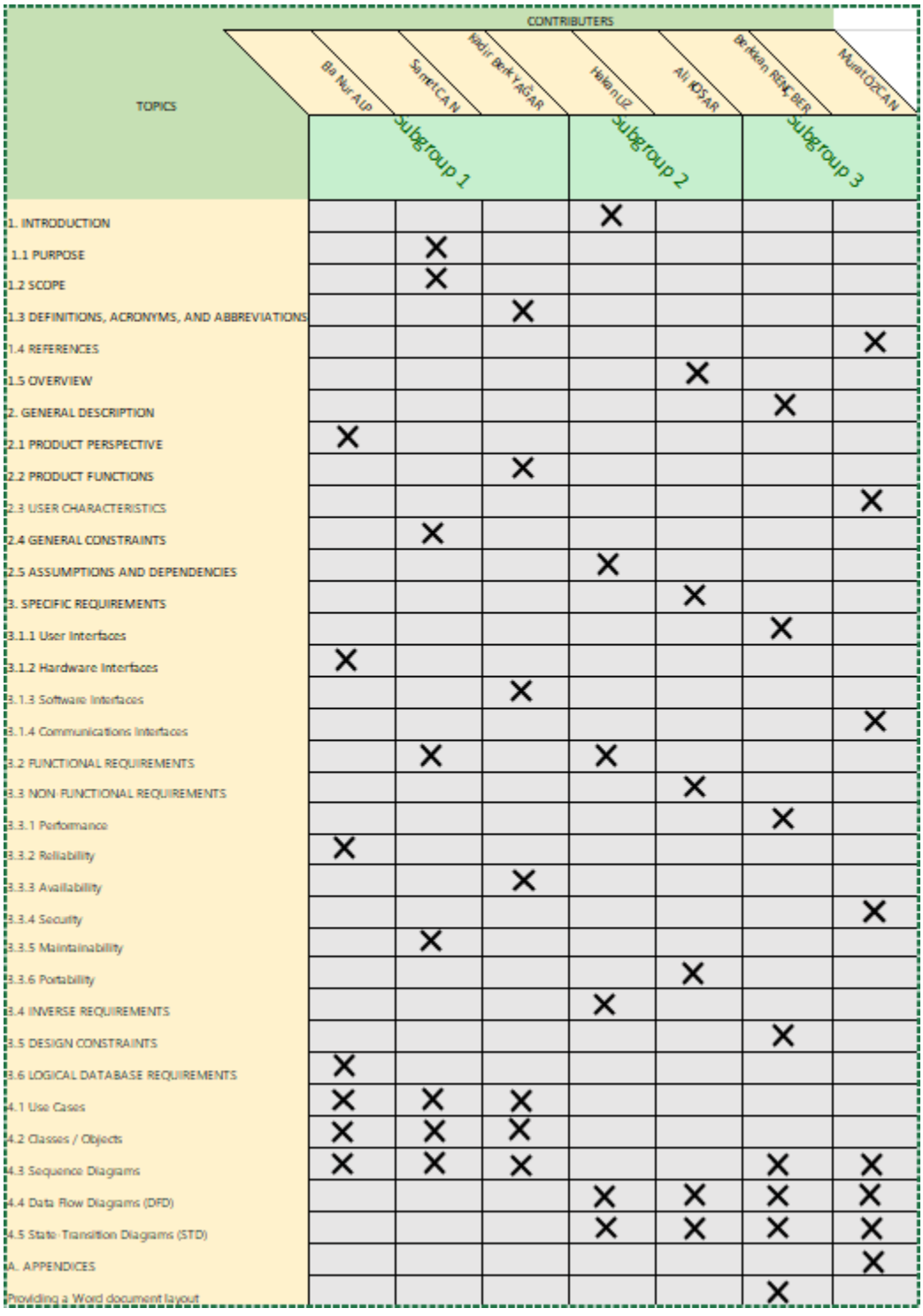
4.4 Data Flow Diagrams (DFD)



4.5 State-Transition Diagrams (STD)



5. Work Sharing Table



TOPICS	Bia Nur ALP	Subgroup 1	Subgroup 1	Subgroup 2	Subgroup 2	Subgroup 3	Subgroup 3
		Sümeyye CAN	Yadir Berk YAĞAR	Helen UZ	Ali KOSAR	Berkcan REŞBER	Murat OZCAN
1. INTRODUCTION				X			
1.1 PURPOSE		X					
1.2 SCOPE		X					
1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS			X				
1.4 REFERENCES							X
1.5 OVERVIEW					X		
2. GENERAL DESCRIPTION						X	
2.1 PRODUCT PERSPECTIVE	X						
2.2 PRODUCT FUNCTIONS			X				
2.3 USER CHARACTERISTICS							X
2.4 GENERAL CONSTRAINTS		X					
2.5 ASSUMPTIONS AND DEPENDENCIES				X			
3. SPECIFIC REQUIREMENTS					X		
3.1.1 User Interfaces						X	
3.1.2 Hardware Interfaces	X						
3.1.3 Software Interfaces			X				
3.1.4 Communications Interfaces							X
3.2 FUNCTIONAL REQUIREMENTS		X		X			
3.3 NON FUNCTIONAL REQUIREMENTS					X		
3.3.1 Performance						X	
3.3.2 Reliability	X						
3.3.3 Availability			X				
3.3.4 Security							X
3.3.5 Maintainability		X					
3.3.6 Portability					X		
3.4 INVERSE REQUIREMENTS				X			
3.5 DESIGN CONSTRAINTS						X	
3.6 LOGICAL DATABASE REQUIREMENTS	X						
4.1 Use Cases	X	X	X				
4.2 Classes / Objects	X	X	X				
4.3 Sequence Diagrams	X	X	X			X	X
4.4 Data Flow Diagrams (DFD)				X	X	X	X
4.5 State Transition Diagrams (STD)				X	X	X	X
A. APPENDICES							X
Providing a Word document layout						X	