BLOGOSPHERE

Software Design Specification

12/05/2023

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

This document contains all general and detailed information about the Blogosphere web application. The architecture of the project is explained in detail. All components in the project are specified and explained in detail. In addition to the detailed descriptions of the components in the project, their behavior in the application is also shown using diagrams.

1.1. Purpose

Overall goals and software objectives are described.
The overall goals and software objectives of the Blogosphere web application are to:

- Create a platform that is easy to use so that users may read and share content about environmental issues.
- Allow users to produce, comment on, and share content, among other features and capabilities that will enhance user engagement.
- Implement security measures to protect user data and prevent incorrect data sharing.
- Establish a content strategy to ensure that the site is regularly updated with new and relevant content.

By achieving these goals and objectives, the Blogosphere web application aims to provide a platform or a user-friendly web application for users to share their ideas and opinions on environmental issues such as climate change, deforestation, and pollution. The project aims to increase environmental awareness, to provide up-to-date information and at the same time to create an accurate and reliable platform.

1.2. Statement of scope

The Blogosphere web application is a sharing platform designed to promote environmental awareness and provide up-to-date information on environmental issues. The project will allow users to create, read, and share content related to topics such as climate change, deforestation, and pollution. The major inputs for the system will include user-generated content in the form of blog posts and comments. The processing functionality of the system will include content moderation, user authentication and authorization, content creation and editing.

1.2.1 Essential Requirements

- User authentication and authorization: Users will be able to create an account , log in, and manage their profile information. Only authorized users will be able to create, edit or delete content
- Content creation and editing: Authorized users will be able to create, edit and delete their own blog posts and comments
- Search functionality: Users will be able to search for content by keywords, tags, or other criteria
- Content categorization: Allow users to categorize their content by topic, type, or other criteria for easier search and discovery

1.2.2 Desirable Requirements

- User interface customization: Allow users to customize their dashboard and profile pages with different themes and layouts.
- Related content suggestions: Suggest related blog posts to users based on their browsing history and interests

1.2.3 Future Requirements

- Mobile application: Develop a mobile application for the Blogosphere platform to enable users to access the site on-the-go.
- User-generated polls: Allow users to create and participate in polls related to environmental issues to encourage engagement and feedback.

1.3. Software context

Environmental issues such as climate change, deforestation, and pollution are becoming important concerns for individuals and organizations around the world. There is a growing need for a platform that allows people to share their ideas and knowledge about these issues, as well as stay informed about the latest developments in the field. This need is intended to be met by the Blogosphere web application, which gives users a convenient, approachable, and entertaining platform to share and access environmental-related content. This project has the potential to help create a more sustainable and environmentally aware future by fostering discussions, raising awareness, and inspiring action.

1.4. Major constraints

Our product is a web application and since we don't have investment we could only test and debug on Chromium environments. We must host our database servers on Windows environments for the fact that we will use Microsoft SQL Server. The development environment is generally built around the Windows ecosystem such as Visual Studio and Visual Studio Code.

1.5. Definitions

MVC: MVC is one of the architectural patterns used in software. It consists of the initials of the words Model View Controller. This design consists of 3 layers. And these layers work without affecting each other. This makes it easier to manage and control large projects

.NET: .NET is an open source platform for building desktop, web and mobile applications that can run on any operating system. .NET includes many libraries and languages to develop high-performance and modern applications. It is supported and maintained by an active developer community.

Angular: Angular is an open source JavaScript framework written in TypeScript. Angular is a framework supported by Google. Its primary purpose is to develop single-page applications. Angular is also a very widely used framework because JavaScript is so widely used.

MSSQL: It is the information management and storage product owned by the Windows company. It provides data processing, querying, reporting operations. It meets the database needs very commonly in websites, desktop applications. It is also a frequently used tool in the field of data mining.

1.6. References

https://www.tutorialspoint.com/uml/uml_interaction_diagram.htm
https://angular.io/guide/what-is-angular
https://www.wrike.com/product-management-guide/faq/what-is-a-product-design-requirement-document/
https://www.geeksforgeeks.org/mvc-design-pattern/

2. Design Consideration

This section describes many of the issues which need to be addressed or resolved before attempting to devise a complete design solution.

2.1. Design Assumptions and Dependencies

2.1.1 Related software or hardware

Software:

- Blogosphere was designed using Visual Studio or Visual Studio Code on Windows 10/11 and MacOS.
- Blogosphere works web-based on devices such as mobile ,desktop and tablet.

Hardware:

- In order for the Blogosphere web application to be used effectively, browsers such as Google Chrome, Safari, Firefox, Opera, Microsoft Edge can be used on devices such as computers, phones and tablets, where visitors, members, editors and admins can access the internet safely.

2.1.2 Operating systems

- Window XP/7/10/11 , MacOS, Linux, IOS, Android, HarmonyOS

2.1.3 End-user characteristics

- Users have certain features according to their authorization. Every authorized person should know their rights. Apart from this, the language used on the website is understandable, but the news shared may contain a technical language.

2.1.4 Possible and/or probable changes in functionality

- The functionality will not change, but the information provided by the editors in the database part may cause a change in the interface.
- Communication between members can be achieved through comments.
- Members can read news from categorized topics and this feature can be offered to members.
- It can be ensured that members can search for a news they want within the website.
- Admin can block users who do not conform to the application's mission and vision.
- The editor can share news and informative articles.

2.2. General Constraints

- Blogosphere needs a database to store users' information, contents, comments and impressions.
- Attention should be paid to database design and management, effective data storage, fast querying, and database security.
- Blogosphere should work fine in different browsers.
- Browser compatibility requires proper use of HTML and CSS languages.
- Blogosphere must take security measures to protect data.
- Must save the user's password by encrypting it
- Performance improvements should be made using caching and database optimization techniques.
- The Blogosphere must comply with legal and ethical constraints.
- It should have a clear and easy to use interface.
- The design, development and presentation processes must be completed within the time frame set for the Blogosphere.

2.3. System Environment

- Windows 10/11 and MacOS
- Intel CPU
- .NET
- Angular
- Visual Studio
- VSCode
- MSSQL

2.4. Development Methods

After the development languages were decided, task distributions were made. Similar to the Agile method, we held small meetings frequently and made the latest situation assessment and planning. This also ensured that everyone was informed about the latest status of the project.

We tried to keep the dependencies as minimal as possible in our code. Thanks to Dependency Injection one of the Design Patterns, we separated the concerns of constructing and using objects. In this way, we ensured loosely coupled programs and it became easy to change the code later.

3. Architectural and component-level design

The intended architecture is heavily dependent on MVC (Model-View-Controller) and MVC-like patterns. A website, for the generalization, has three layers that are used to be built and executed on. These are client-side interface, server-side logic and server-side data management in our context. This is a common pattern used on most projects and is relied upon for the simplicity and maintainability the pattern offers.

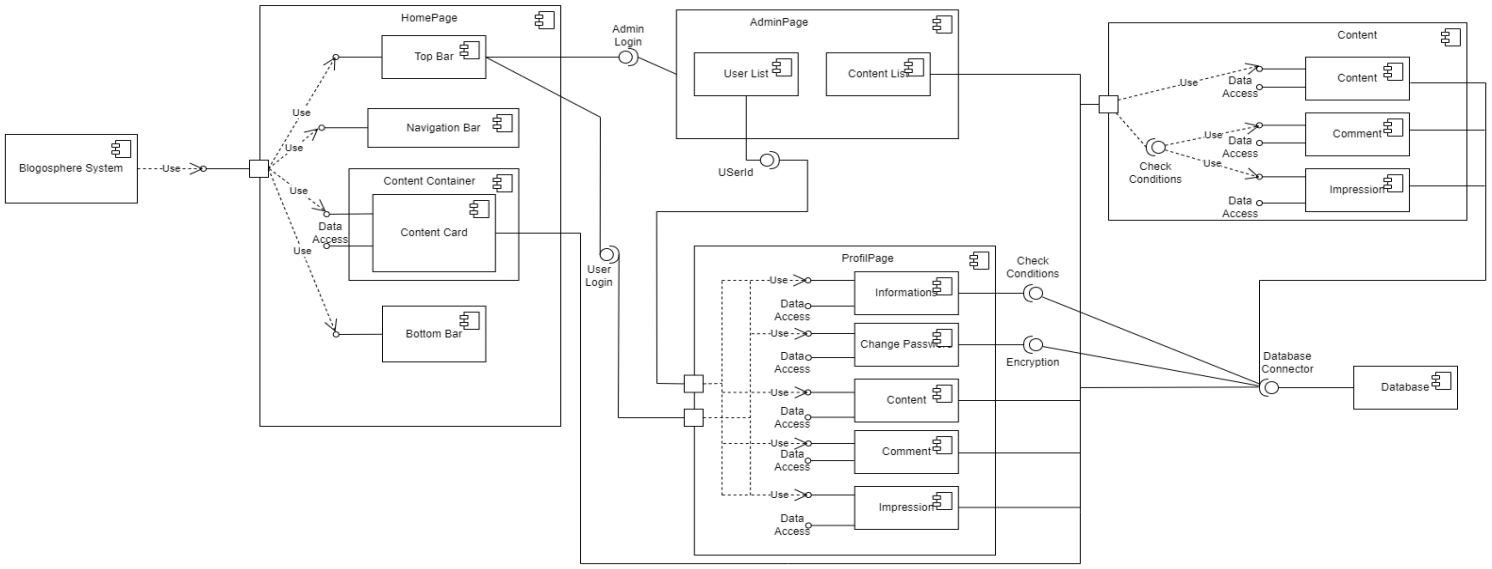
Users will be interacting with the client-side interface to get their job done. Everything that is understandable by the moderate user/human will be presented through this interface. The server-side logic is executed under the hood by the server so that the operations made through the client-side interface have an effect on the data and, through that, the inner workings of the system.

3.1. System Structure

Basically, the Blogosphere web application consists of five main components: Login, Homepage, Content, AdminPage, ProfilePage. When the user enters the application as a visitor, it opens to the homepage. When the visitor accesses the homepage, they can view the contents and view the contents of the contents. The visitor can enter an input to access the desired news content from the search bar at the top of the homepage. If the visitor wants to access the content in a certain category, the visitor can choose one of the existing categories.

If the visitor wants to register, the visitor is directed to the login page. Registration is done on this page. After the visitor registers, he becomes a user for the web application. The difference of the user from the visitor is that the user can comment and like after entering the details of the content. If the user wants to become an editor, he has to pass admin approval. After the approval, unlike the user, they can create content, add categories, and edit their own content. Finally, the admin can view all shared content and all users and can edit content and comments.

3.1.1. Architecture diagram



3.2.1 Description for Component Login

A detailed description of each software component contained within the architecture is presented. Section 3.2 is repeated for each of n components.

The Login component is responsible for authenticating users and providing access to the blog website. It enables registered users to securely log in to the system by entering their username and password. Once the user's credentials are verified, they are granted access to the site and its features based on their assigned role (i.e., user or admin).

Once a user logs in, they will have access to the platform's content, including blog posts related to environmental issues such as climate change, deforestation, and pollution. Authorized users will also be able to post content to the platform and comment on other users' posts.

For users who have not yet registered, the Login component provides a button for the Signup component where they can create an account. The Signup component requires the user to provide their email address, username, and password (twice for verification). Once the user has successfully signed up, they can log in to the system using their chosen username and password.

The login component also checks the accuracy of the information. If the user is already a member of the system and has entered the wrong user name or password, a warning screen is displayed to the user and after this warning screen is closed, the login information is requested from the user again, whenever the correct information is entered, the user is directed to the system's homepage.

Visitors who have not registered to the system can still access the site, but they will not have permission to perform any actions such as commenting, liking, or sharing posts. Instead, they can only view the content posted by registered users.This is necessary to prevent spam and ensure that the platform remains a reliable source of information and discussion on environmental issues.

Overall, the login component plays a crucial role in ensuring the security and integrity of the Blogosphere platform, as well as allowing for the reliable flow of information and discussion between users and visitors.

3.2.1.1 Processing narrative (PSPEC) for component Login

A processing narrative for component n is presented. It should describe the responsibilities of the component.

The Login component is responsible for authenticating users who have previously registered to the system. When a user attempts to log in, the component first verifies that both the username and password fields have been filled out. If either field is left blank, the user is prompted to fill out both fields. Once both fields have been filled out, the component checks if the entered username exists in the system's user database. If the username is not found, the component displays an error message indicating that the username is not registered. If the username is found, the component checks if the entered password matches the password associated with that username. If the password is incorrect, the component displays an error message indicating that the password is invalid. If the password is correct, the component grants the user access to the system by setting a session token for the user, which can be used to identify the user and their authorization level for subsequent requests. If the user is an admin, they will be redirected to the admin dashboard, while regular users will be redirected to the main page of the blog website.

3.2.1.2 Component Login interface description.

A detailed description of the input and output interfaces for the component is presented.

The Login and Signup component provides interfaces for users to log in to the system and create new accounts. The component accepts user inputs and provides appropriate outputs based on the user's actions.

Input interfaces:

- Username: A field where the user enters their username when logging in or signing up.
- Password: A field where the user enters their password when logging in or signing up.
- Email: A field where the user enters their email address when signing up.
- Password for confirmation: A field where the user enters their password again when signing up for verification.

Output interfaces:

- HomePage(Component interaction): When a user successfully logs in, the system will redirect them to the Homepage component, which displays the latest posts and allows users to interact with them.
- Error messages: If a user enters an incorrect username or password during login, an error message will be displayed, and the user will be prompted to re-enter their credentials. Similarly, if a user tries to sign up with an email address that is already registered, or if their password confirmation does not match their initial password entry, an error message will be displayed.

- When users log in successfully, the login component interacts with the HomePage component to display the latest posts and allow the user to interact with them. Likewise, after successful registration, the component will redirect the user to the Login interface to log in with the new credentials.

3.2.1.3 Component Login processing detail

When a user enters their login credentials (username and password) and clicks the login button, the system checks if the credentials are correct by searching for a user with a matching username and password in the database. If the credentials are correct, the system redirects the user to the homepage. If the credentials are incorrect, the system displays an error message to the user. For new users who want to sign up for an account, they provide their desired username, password, and email address. The system checks if the username is unique, and if it is, the system creates a new user account with the provided information and stores it in the database. The system then redirects the user to the homepage. If the username is not unique, the system displays an error message to the user and prompts them to choose a different username. Overall, the Login/Signup processing component of the system is responsible for authenticating users and managing user account creation.

3.2.1.3.1 Design Class hierarchy for component Login

The Login component has a simple class hierarchy consisting of two classes: Login and User. The Login class has two functions: login and logout.The "login" function takes two string parameters representing the user's username and password, and returns a string indicating whether the login was successful or not. The logOut function logs the user out of the system.The User class, on the other hand, has several functions and attributes related to user registration and management. These functions include register, updateBlockState, and getUser, which allow the user to create an account, block or unblock a user, and retrieve user information, respectively.

The Login class has a relation with the User class, as it is responsible for handling the user authentication process. Additionally, there is an aggregation relationship between the User class and the Profile class, which is responsible for updating user information such as email and password.

Furthermore, the SignUp class is also part of the component, that is responsible for allowing users to sign up for a new account. There is an aggregation relationship between the SignUp class and the HomePage class, as the sign-up functionality is part of the home page. Finally, there is an aggregation relationship between the User class and the UserType enumeration, which defines the different types of users in the system: ADMIN, EDITOR, MEMBER, and NONE.

3.2.1.3.2 Restrictions/limitations for component Login

There are no specific restrictions or limitations for accessing the Login component. Everyone can access the Blogosphere website including visitors without registration. However, the users are required to provide valid login credentials (username and password) in order to access the system. The Login component will verify the provided credentials and allow access to the system if they are valid. If the credentials are invalid, the user will not be able to access the system.

3.2.1.3.3 Performance issues for component Login

The performance issue of the Login component is the response time of the system when a large number of users attempt to log in simultaneously. This may cause delays in the login process and negatively impact the user experience. In case of this large number of users problem, we need to improve our database.

3.2.1.3.4 Design constraints for component Login

The Blogosphere web application must be designed to work on a variety of devices and operating systems, including mobile phones, tablets, and desktop computers. The login component must be designed to work with different web browsers, such as Google Chrome, Firefox, and Safari. The design of the login component should prioritize user experience and ease of use, with clear instructions and error messages provided to the user if necessary.

3.2.1.3.5 Processing detail for each operation of component Login

Login Operation:

1. User enters their username and password.
2. System checks if the username and password match the data stored in the user database.
3. If the entered data is correct, the system grants access to the user and redirects them to the homepage.
4. If the entered data is incorrect, the system displays an error message and prompts the user to enter their credentials again.

Signup Operation:

1. User enters their email, username, password, and confirms the password.
2. System checks if the email is in the correct format and if the username is unique.
3. If the data entered is valid, the system stores the user data in the user database and redirects the user to the homepage.
4. If the data entered is invalid, the system displays an error message and prompts the user to enter their information again.

3.2.2 Description for Component Home Page

Home page component is the main part of the Blogosphere web application. Home page the component contains mainly content shared by editors, and this component has 6 sub-components. One of these components is the content card, which contains every content shared by the editor. In this component, the user accesses the image, title, author, and date of the content and if the user clicks on this component, they can switch to the page of the details of the content. Another component is the content container where the content cards are listed. The component containing the navigation properties, on the other hand, includes functions that sort the content cards in the content container according to any rule and list the contents related to the category that the user will choose. The main container contains these three components in a specific layout. The component at the top of the page

includes the site's features such as logo, name, and the search bar where the user can enter input. The last component is located at the bottom of the page and contains some information about the site.

3.2.2.1 Processing narrative (PSPEC) for component Home Page

Home page the main task of the component is to show users the latest shared content. The content card, one of the components, gives the user preliminary information about that content. There are previews about the content such as images and titles. The responsibility of the content container component, which contains content cards, is to list the contents according to the most recently shared and to provide the user with access to multiple contents in the appropriate view. In this component, the user provides the opportunity to access more than one news with the scroll function. The responsibility of the component containing the navigation functions is to perform this sorting function if the user wants to sort the contents according to any rule, for example by the most recent date, by the most likes, etc. Also, this component offers different categories to the user. Whichever of these the user chooses, the content related to that category is listed. The main container component is responsible for displaying these three components to the user in a specific layout. The responsibility of the top bar component at the top of the page is to find the content containing the input that the user will enter in the search bar. The user can also go to the user profile or log out of their account from the buttons on the right of the component. The responsibility of the component at the bottom of the page is to show the user some information about the Blogosphere web application.

3.2.2.2 Component Home Page interface description.

The home page component receives text as input from the user in the search bar in the top bar component. It displays the content in which that text is included as output to the user. The user selects the category as an input. The component receives a click on the content as an optional input from the user. It outputs the user to the detail page of the content they clicked. The component displays content by category as the selected output. The user chooses a sorting method as input. The component sorts the contents according to the sorting method chosen by the user as output. Finally, an optional input is received from the user as a request to access the user profile. The component outputs a redirect to the user's profile page.

3.2.2.3 Component Home Page processing detail

The home page component contains the Model-View-Controller design pattern. Our models will be used mostly for the content card section. Operations and processes are carried out with the Controller part. The contents published by the editors with the controller will be pulled from the database and listed on the user's home page. In addition, processes such as login, logout, search, category, trend and communication are provided. In this way, CPU intensive usage is avoided by not using a single class structure.

3.2.2.3.1 Design Class hierarchy for component home page

Home page component consists of a total of 6 sub-components. These are topBar, bottomBar, mainContainer, navigationBar, contentContainer, contentCard respectively. The top bar contains the website's logo, search button and login or logout button as appropriate. With the Search button, the user can search for content. The bottom bar is positioned to provide information about the designer of the website. Main container was created to design the interface of the website and it contains navigationBar, contentContainer and contentCard. Navigation bar allows the user to access trending topics, select categories and contact the website administrator. It contains a content card in a content container and helps organize its design. Content card contains the picture of the content, its title and a single sentence summary if written, and the content is listed thanks to these cards.

3.2.2.3.2 Restrictions/limitations for component home page

In the home page component, users will only be able to choose from predetermined categories on the website.

3.2.2.3.3 Performance issues for component home page

In the home page component, the excess of the content pulled from the database can cause difficulties on the website. With this problem, the website may run slower and cause the user to move away from the website. In addition, it can cause difficulties while pulling the desired content from the database for the content sought in the search button.

3.2.2.3.4 Design constraints for component home page

Blogosphere application does not contain any restrictions for users with internet access and hardware stability. It is sufficient for the device to be used to have one of the Google Chrome, Firefox, Safari, Opera or Microsoft Edge browsers.

3.2.2.3.5 Processing detail for each operation of component home page

In the home page component, at first, the user displays the uncategorized content in the content container section. If he does not want to go to the detail page about any content here, he can observe the titles, pictures and a one-sentence summary of the news listed, depending on the editor. In addition, a category option has been provided for the user to view the contents in certain categories and it has been enabled to choose as he wishes. If you want to view a special content again, you can search with the search button. The top bar also has the ability for the user to exit. Again, another option is presented to the user for communication.

3.2.3 Description for Component Profile Page

The Profile Page component is the place where user information is found and the impressions made and the contents written are displayed as a list. This component includes 5 subcomponents. These components are newPass, infos, my posts, comments and impressions. The profile page component consists of these 5 components, which we will explain later. It contains these components and allows switching between them. This component can be accessed by the admin for any user or the user can access for their own information.

3.2.3.1 Processing narrative (PSPEC) for component Profile Page

The main purpose of this component is to arrange the functions of the sub-components correctly. It should allow the user to switch between components. Here is the information about the subcomponents:

- **Informations Tab:**

  This tab is the tab selected by default in the page. The task of this tab is to inform users about their username, email and authorization. Here, two types of operations can be performed after the display operation. If the user who reaches the page is himself, he can change the mail and username information. If the admin reaches the page, he can block the user or change his authority.

  The user changes the existing e-mail address or user name in the textbox and presses the update button. If the new user selected is suitable for use, the database is updated and the SUCCESS information is printed on the screen. Otherwise, failure information is printed on the screen.

  Admin selects authorization for the user or presses the block user button. The transaction takes place in the database and successful information is written on the screen.

- **NewPassword Tab**

  The purpose of this component is to update the user's password in the database by retrieving the user's old password, new password, and repetition of the new password.

  It takes old password, new password and repetition of new password as input when the update button is pressed. It gives a message on the screen as password update successful or unsuccessful.

  The process starts with the user entering the old password, the new password and the repetition of the new password into the relevant textbox and pressing the update button. Then, firstly, the user is logged from the session to check the correctness of the old password. If the old password is not correct, an error message will be displayed and the process will be terminated, and if correct, the next step will be taken. In the next step, it is checked whether the new password and its repetition match, if they do not match, an error message is displayed and the process is terminated, and if they match, the next step is taken. The new password is checked for compliance with the password criteria, the new password must meet the minimum length criteria, if not, an error message is displayed. Finally, the new password is updated in the database and the successful information is displayed on the screen and the process is completed. In addition, in the last step, database disconnection etc. cases, an error message is displayed on the screen.

- **Contens Tab**

  This tab is only visible to editors and admins who enter the editor profile.

  The sole purpose of this tab is to list user-created content. The list contains title and date information about the content and when you click on it, the content page opens with the content id.

- **Comments Tab**

  The sole purpose of this tab is to list contents which are user commented. The list contains title and date information about the content and when you click on it, the content page opens with the content id.

- **Impressions Tab**

  The sole purpose of this tab is to list contents which are user left impression. The list contains title and date information about the content and when you click on it, the content page opens with the content id.

3.2.3.2 Component profile page interface description.

There may be changes in the presentation layer depending on who is logged in. The page loads with the Informations tab selected. Tabs at the top of the component are respectively Informations, NewPassord, Contents, Comments and Impressions. If the input is not the editor, the contents tab is not included. If the login is admin, the NewPassword tab is not included. The contents of the tabs are as follows:

- **Informations Tab**

  If the user is logged in, this tab gives the username and mail information in the changeable textboxes, in the form of an unchangeable text. In addition, it gives an update button.

  If the login is admin, username and mail information are given in unchangeable text format. There is one dropdown for authorization and the admin can change the user's authorization from here. In addition, there is a block button.

- **NewPassword Tab**

  In the presentation layer, there are 3 textboxes where the user writes the old password, new password and new password and one update button for input. Successfully updated, incorrect old password, mismatch passwords, password not long enough or failed to save popups will be shown as output.

  There will be an updatePassword method on the application layer. It will provide the specified controls for the entered entries, if all queries are successful, the new password will be written to the relevant place in the database and successful information will be returned.

- **Contents Tab-Comments Tab-Impressions Tab**

  The interface of the this three component consists of the list which is consist of headers and date information about relevant contents. When clicked, it functions to pull the list of posts from the database and display it. The interface uses a listing method to allow users to simply find and navigate through their posts.

3.2.3.3 Component profile page processing detail

Information update process, password renewal process, authorization editing process, listing process and blocking are the main operations of this component. Details will be given in section 5 under this title.

3.2.3.3.1 Design Class hierarchy for component profile page

Profil page component consists of 5 different sub-components, these are 5 different tags in the presentation layer. In the application layer, the relevant methods of these 5 components are located in the profile page component class.

3.2.3.3.2 Restrictions/limitations for component profile page

- In the presentation layer, only the admin block button is provided.
- Only admin can use change authorization dropdown in presentation layer.
- The newPassword tag is not presented to the admin login in the presentation layer
- Contents tag is only presented to the user with the editor type and the admin accessing the profile of the editor type user.
- Editor sees only the list which are own.
- Under the comment tag, only the list of comments left is listed.
- Under the impressions tag, only the list of impressions left is listed.
- In the data layer, only the relevant tables and rows of the session object are accessed.

3.2.3.3.3 Performance issues for component profile page

While creating the lists, it is necessary to determine and pull the relevant rows from the database. Here, slowdown is inevitable as the number of transactions will increase as the number of relationships and rows increases. In addition, since the component will behave differently for admin and user logins, the page must be loaded separately for two different logins.

3.2.3.3.4 Design constraints for component profile page

Design restrictions are determined by the framework and software languages we will use. There are no extra restrictions. Frameworks and languages are specified in this document.

3.2.3.5 Processing detail for each operation of component profile page

First of all, the component is loaded with the infomations tag by default, and the transition between these tags is the priority process. This process allows to access sub-components in the user interface by switching between tags. Subcomponent operations are as follows:

- **Information Tab**

  This component, on the other hand, is the admin who logs in, changes user authorization and makes blocking operations. As soon as the admin presses the block button, the user's activity is set to false in the database and a message is returned to the screen. Admin selects any authorization from dropbox and user type is changed to selected type in user's database.

  The logged in user, on the other hand, changes the username or email from the textboxes on the screen. If the username entered has not been taken before, the error message is displayed on the screen. Otherwise, the user's information is updated from the database and a successful message is displayed on the screen.

- **NewPassword Tab**

  First, get the User's old password, new password and new password again and check the old password is correct, if it is incorrect, Invalid password message will be given and the process will be terminated. The new password and its repetition are compared. If they do not match, a Password Mismatch message is issued and the process is terminated. Then a length comparison is made for the new password. If the transaction does not comply with the conditions, a Short Password message is given and the transaction is terminated. Finally, the password is updated in the Database and the message SUCCESSFUL is displayed.

- **Contents Tab**

  The Contnents component will not be presented in the presentation layer if the editor is not logged in or the admin is not logged into the editor profile. The only operation of this component is to list the contents written by the user and direct them to the content page with its id when clicked.

- **Comments Tab**

  The only operation of this component is to list the contents that the user has commented on, and when clicked on it, it redirects to the content page with its id.

- **Impressions Tab**

  The only operation of this component is to list the contents left by the user's impression and when clicked on it, it will redirect to the content page with its id.

3.2.4 Description for Component Content

Content component is the component where the details of the news accessed from the main page are accessed. It allows the entire clicked news to be read and the news to be commented and liked. It consists of 4 components. The first of these, "inside component", is the one in the center and consists of 3 subcomponents. Content component contains the details of the news along with the photo of the news. Like component allows the member to like or dislike the news. Finally, the comment component allows the member to leave comments on the relevant news and read the comments left.

3.2.4.1 Processing narrative (PSPEC) for component content

The main task of the Component is to show the entire news to the user. The content container, which is one of the components, contains the title of the news, its image and the entire article. As you scroll down, it accesses the rest of the news. It also shows an edit-delete button for the content, depending on who the user is. For example, it provides the opportunity to delete any news if the user is an admin, and to edit his own news if the user is an editor. The like container allows the user to leave a like or dislike for the news. He can access the like he left later from his own profile. Likewise, if the user is a reader, that is, if he is not a member, the like button cannot be accessed. Finally, the Comment Container contains comments left by members. If the user is a member, it also allows them to leave comments. He can also access the comment he left later on his own profile.

3.2.4.2 Component content interface description.

Content Component can receive input from the user as a like or dislike. It notifies the user as a view along with an increase in the number of likes as output. Another input from the user is a comment. If the user is a member of the site, he receives the text he will leave for the news he reads as input. It publishes the comment with a successful message as output.

3.2.4.3 Component content processing detail

A detailed algorithmic description for each component is presented.

3.2.4.3.1 Design Class hierarchy for component content

Content component consists of 4 components. Inside the Content Component, it consists of Inside, news, like and comment components. There are edit and delete buttons in News. Admin or editor can edit or delete news content from here. Inside there are like and dislike buttons. The user can react about the news here. Finally, there is an empty text box in the comment where the user can comment. The user can write what he thinks about the news or read what is written here.

3.2.4.3.2 Restrictions/limitations for component content

Content Component includes some features according to user type. So some users have some limitations in terms of features and interface.

While the edit and delete buttons in the News component are only visible to the admin or editor, they are not visible to the member or reader. Likewise, the comment location in the Comment component is visible to every member while not visible to the readers.

3.2.4.3.3 Performance issues for component content

Retrieving the images, content, comments and likes of the related news from the database may also cause some slowness. Factors such as too many comments on the news, too many interactions can negatively affect the loading speed of the news.

3.2.4.3.4 Design constraints for component content

Since Blogosphere is a platform independent site, it can be used on any device without any problems.

3.2.4.3.5 Processing detail for each operation of component content

Content Component is the place you come to after clicking the news on the main page. At first it includes the title, image and content of the news. As the user scrolls down, they read the rest of the news. If the user likes the news or does not like it, he can react with the like-dislike buttons. In addition to commenting below at the end of the page, he also reads some comments if they have been made. If the user is an admin or an editor, they can edit or delete the news. For this, click on the buttons that appear next to the news. It will then be directed to the fix screen.

Finally, when the user goes back to leave the Content page, he is transferred to the page he was directed to.

3.2.5 Description for Component Admin Page

The Admin Page is a page that is only accessible by a user with admin permission. Admin uses this page to operate on User permissions and contents. It has a Side Menu to choose for the intended operation. This Side Menu has "List Users" and "List Contents" buttons so that they will be listed to be operated on.

3.2.5.1 Processing narrative (PSPEC) for component admin page

Through the usage of this page: The Admin can operate on User privileges, by banning them or promoting them as authors/content creators. Contents might be updated or deleted, their comments might be deleted as well.

3.2.5.2 Component admin page interface description.

The Admin Page will have a Sıde menu on the left with buttons named "List Users" and "List Contents". "List Users" button will list Users with their usernames, emails, types, etc. on the right side. "List Contents" button will list Contents with their name, publish date, author, comment count, impression count, etc.

3.2.5.3 Component admin page processing detail

The Admin Page does not have much usability on its own. Its generally used to search for the concrete objects so that they can be operated on, and the Admin Page is specialized on that. The admin has operation privileges on both content and user profile pages to update depending on needs.

3.2.5.3.1 Design Class hierarchy for component admin page

The Admin Page has three components which are specialized for it: Control Container, Side Menu, Panel Contents. Besides this there are another 2 components that are common on all pages such as the top menu bar and the footer.

Control Container holds the other components the Admin Page relies on. It has Side Menu and Panel Contents side by side.

Side Menu has buttons that indicate which data will be operated by the admin. At this point, "List Users" and "List Contents" buttons can be seen by the Admin and those buttons, when clicked on, will trigger the listing operation on the Panel Contents component with related columns.

3.2.5.3.2 Restrictions/limitations for component admin page

The Admin Page does not have a direct option to make updates on contents or users. Instead, it provides a link to the content or user so that they can be operated on.

3.2.5.3.3 Performance issues for component admin page

Since the Admin Page is generally used for listing contents and users, the query that will be executed on the sql database may encounter performance issues due to the database having so much data stored.

3.2.5.3.4 Design constraints for component admin page

The website is expected to be run on desktop web browsers as the Admin for the most cases so this component is designed for wide screens and relies heavily on mouse usage.

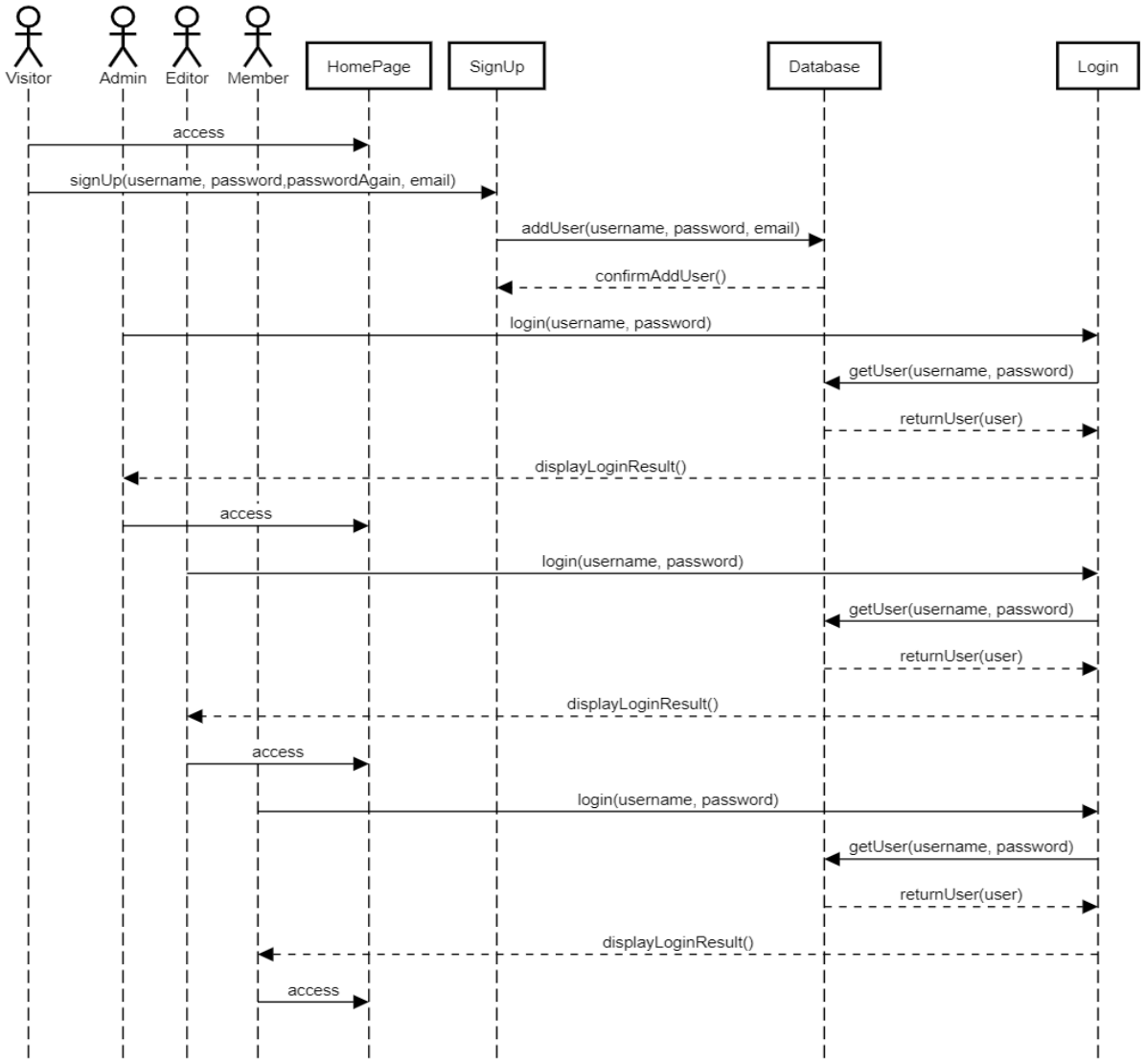
3.2.5.3.5 Processing detail for each operation of component admin page

This page includes a Side Menu at the left side and a Content Panel at its right. The Side Menu has buttons that will list its corresponding data on the right side when they are clicked on. In our case, there are two buttons that can list data: “List Users” and “List Contents”. These buttons will list their corresponding data with the related attributes as columns for it. The Admin can click on the elements of the list to go to their pages so that they can be operated on as intended.

3.3.1 Dynamic Behavior for Component Login Page

Dynamic behavior of component Login interacts with HomePage, SignUp, Database and Login objects/classes. In the behavior of the Login component the actor Visitor accesses the HomePage and can see the contents without logging in or signing up. Visitor can sign up by providing their username, password, password confirmation, and email address through the SignUp component.The SignUp component sends the provided information to the Database to add a new user account.If the addUser() operation is successful, the Database confirms the addition of the new user to the SignUp component.Admin, Editor, and Member can log in by providing their username and password through the Login component.The Login component sends the provided information to the Database to retrieve the user account.If the getUser() operation is successful, the Database returns the user account to the Login component.The Login component then displays a login result, either a success message or an error message if the login failed, to the corresponding user. If the login is successful, the user can access the HomePage and see the contents.

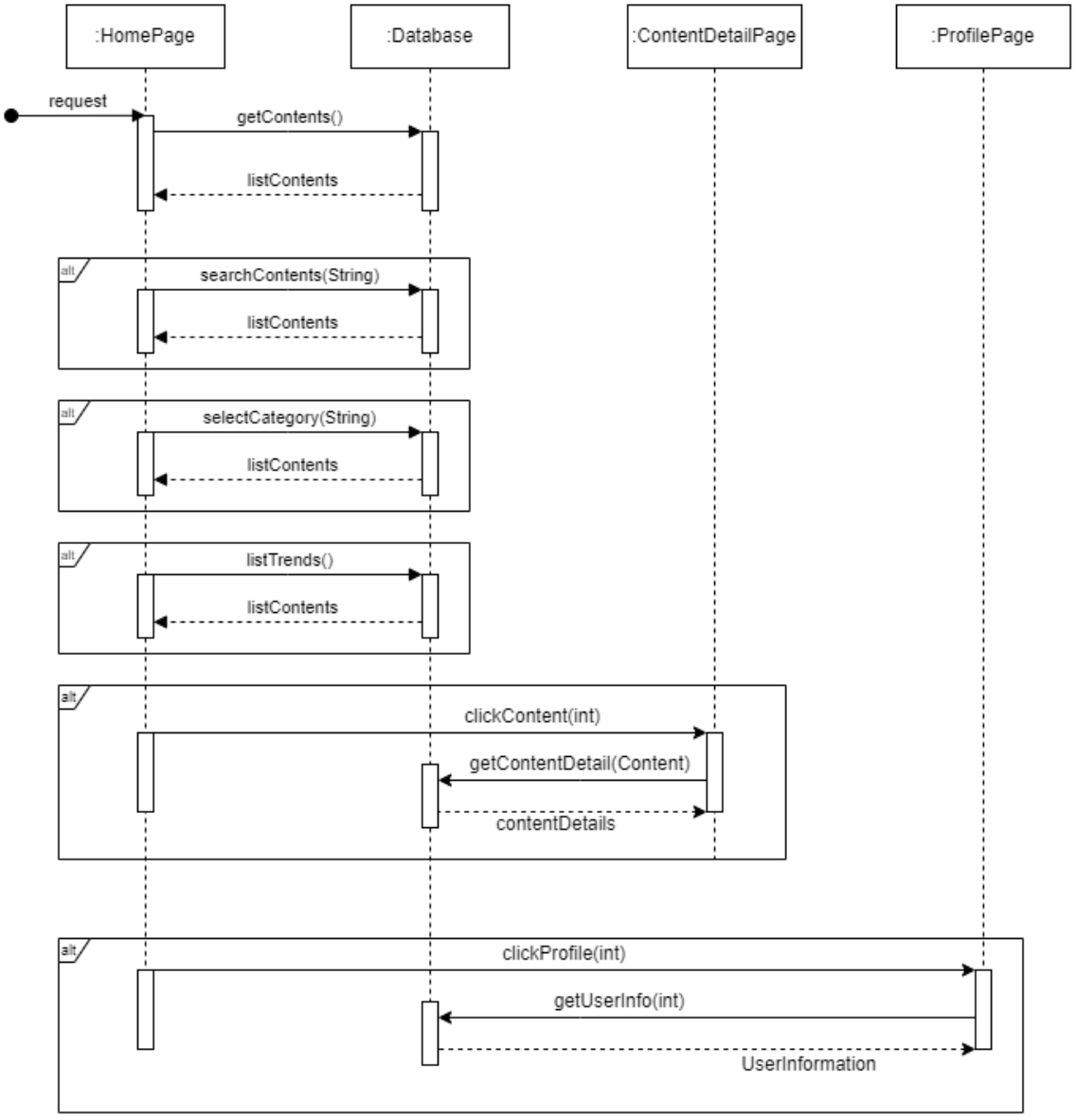
3.3.1.1 Interaction Diagrams



3.3.2 Dynamic Behavior for Component Home Page

The options that the user can interact with in the Homepage component are, respectively, searching for a content, listing content in a category, viewing trending content, being directed to the detail page when he clicks on a content, and viewing his profile if he is not a visitor. Apart from this, the contents in the database are listed for the user when the first request is made to the website.

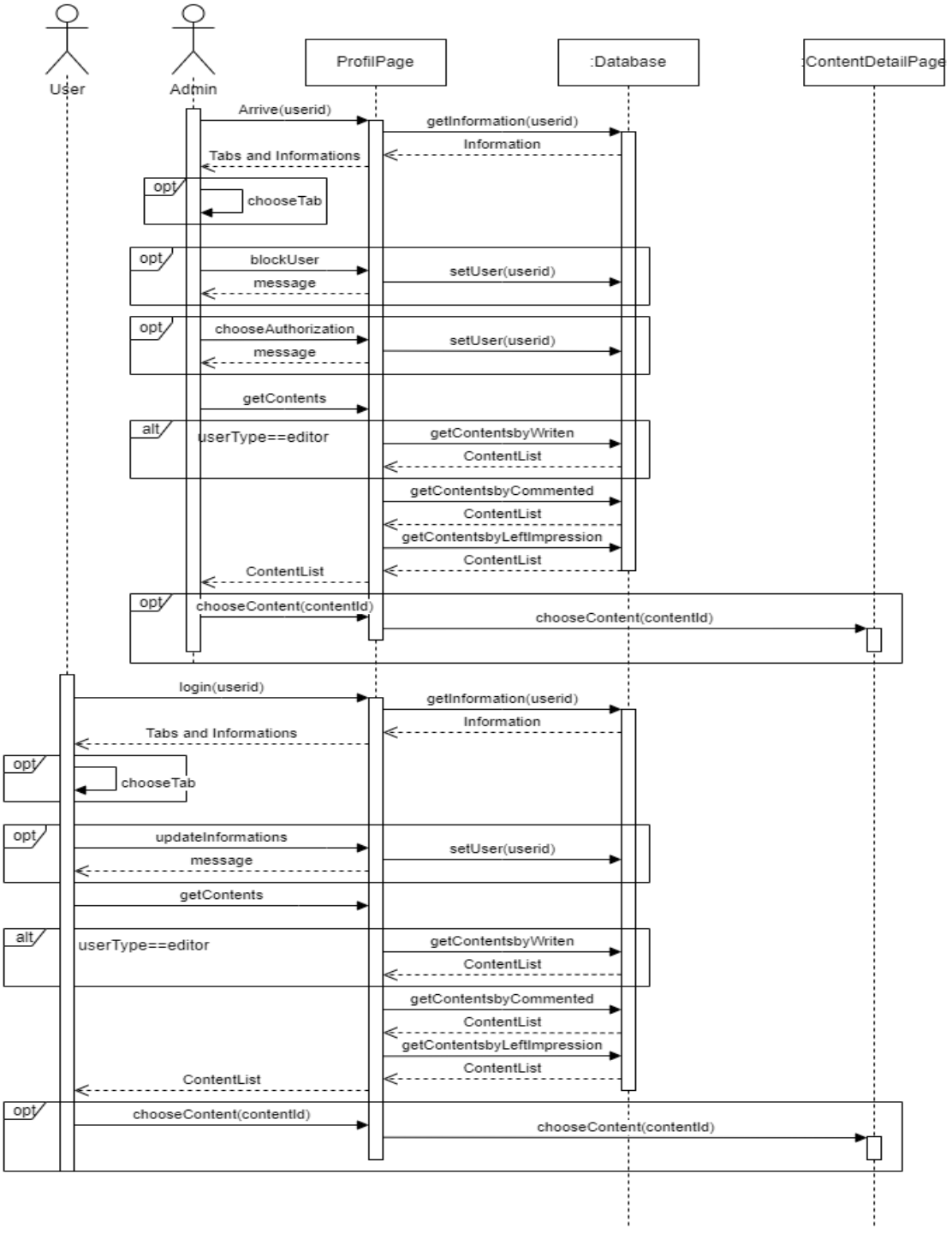
3.3.2.1 Interaction Diagrams



3.3.3 Dynamic Behavior for Component Profile Page

User and Admin have the options to change the information about the user from the informations tab in Profile Page. The user can use the NewPassword tab to change the password. Admin and User can fetch a content list with Contents, Comments and Impressions tabs. Clicking on the desired Content from the incoming list opens this content on the content page.

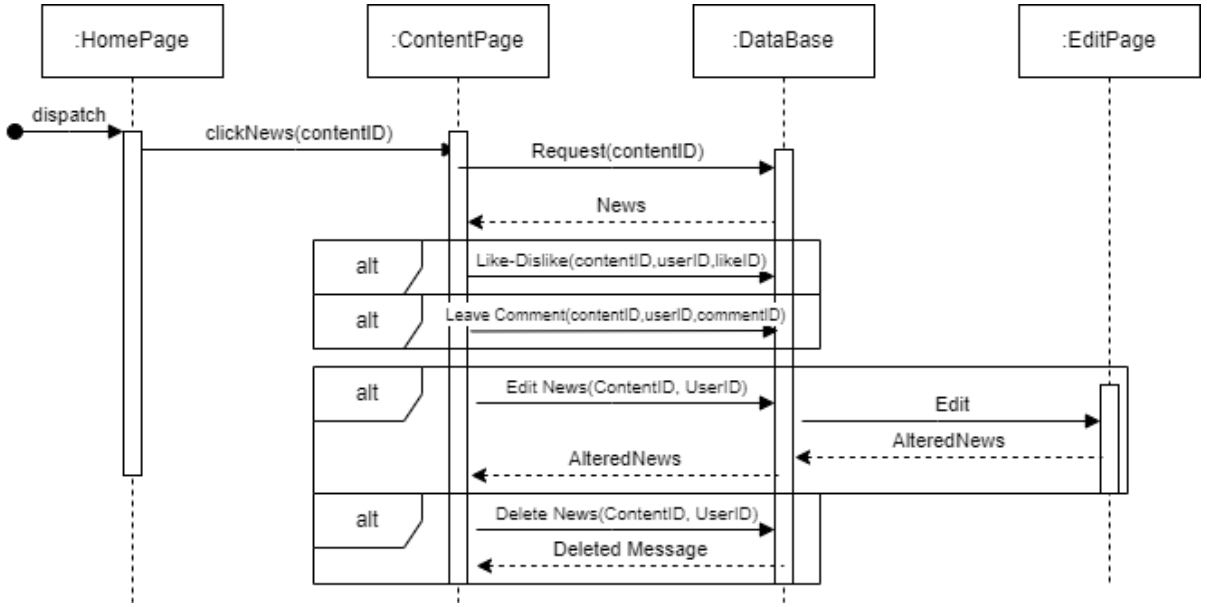
3.3.3.1 Interaction Diagrams



3.3.4 Dynamic Behavior for Component Content Page

The user reaches the content page with the redirect from the home page. Optionally, the user can leave a like-dislike. Likewise, he/she can write or read comments. If he is an editor or admin, he can edit the news content. For this, a redirect is made to the Edit Page. After saving changes, it redirects to the content page. In addition, the admin or editor can delete the news directly.

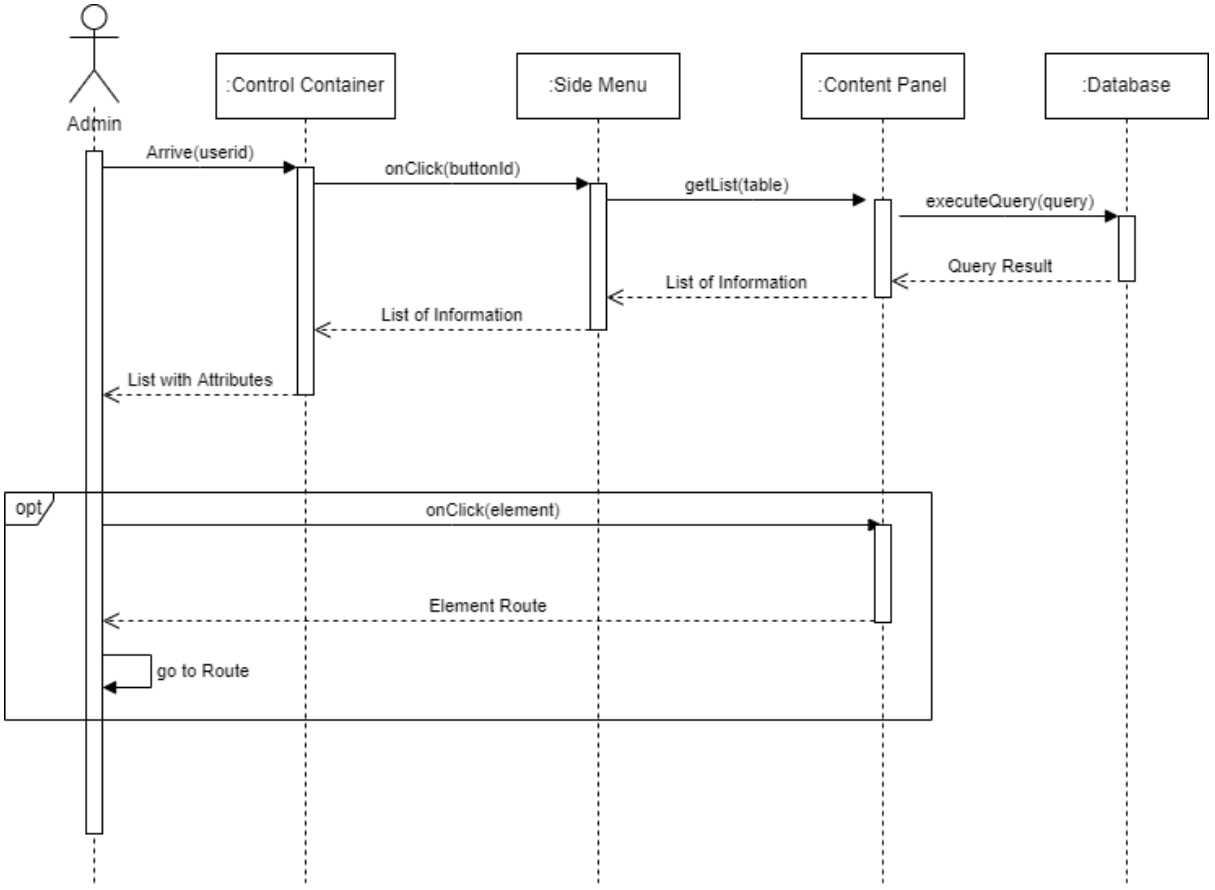
3.3.4.1 Interaction Diagrams



3.3.5 Dynamic Behavior for Component Admin Page

The Admin Page is used for listing contents and users so that they can be accessed and be operated on. Admin should first list which data to be listed and then should choose an element to route the page.

3.3.5.1 Interaction Diagrams



4. Restrictions, limitations, and constraints

4.1. Time Constraints

The project has a strict deadline, which limits the time available for design, development, testing, and deployment. To meet the deadline, the system will adopt agile development methodologies to ensure that each phase of the development cycle is completed on time.

4.2. Compatibility with Different Browsers and Devices

The system must be designed to work seamlessly across different browsers and devices. This includes desktop and mobile devices, as well as different operating systems and web browsers. The system must be compatible with major web browsers such as Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge.

4.3. Data Privacy and Security

Data privacy and security is a major concern for this project. As a result, the system must comply with all relevant data protection laws and regulations. The system must ensure that user data is stored securely and is not accessible by unauthorized users.

4.4. Performance

Performance is another major concern for this project. The system must be designed to handle a large number of users concurrently. It must also be able to handle a high volume of traffic without slowing down or crashing. To achieve this, the system must be optimized for speed and efficiency.

4.5. Usability and Accessibility

The system must be designed to be easy to use and accessible for all users, including those with disabilities. The user interface must be intuitive and user-friendly, with clear navigation and labeling.

5. Conclusion

In this document, we explained how we developed the Software Design Specification of our project called Blogosphere. We have explained the architecture of the project in detail in order to save the Design Information and communicate that information to the design's stakeholders. We wrote Design Consideration and General Constraints.

Then we covered Architectural and Component Level Design. We have explained in detail the 5 components (Login, Home Page, Profile Page, Content, Admin) that make up our project. We have covered the Processing narrative as well as the process detail of each component.

While creating the components, we also considered the back-end as well as the design. In other words, the components we created are also used in the back-end of our project, connected to the database.

Finally, we discussed the dynamic behavior of each Component and drew their interaction diagrams.