Travelogram - Application Design Specification

ENPM809W – Special Topics in Engineering; Introduction to Secure Coding for Software Engineering

Name: Sindhuja Sane

Email ID: sane@umd.edu

Directory ID: sane

UID: 119384268

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# Section 1 - Project Description

## 1.1 Project

Travelgram - change name

## 1.2 Description

The Travel Blogging Application is a user-friendly web-based platform tailored to the needs of travel enthusiasts, content creators, and avid readers. It streamlines the process of creating, managing, and sharing travel-related content. Content creators can craft detailed travel blog posts, with narratives, images, and expense information to offer a comprehensive view of their journeys. Through interactive commenting, users engage in meaningful discussions, share insights, and build an active travel community.

**Contents**

[Section 1 - Project Description](#_yr1n1w74g294)

[1.1 Project](#_mpniepv3sji5)

[1.2 Description](#_kxtdscgtj3ad)

[1.3 Revision History](#_x1jfzxbgcdpj)

[Section 2 - Overview](#_v0z2li3lbhgp)

[2.1 Purpose](#_wkcniuopwsk)

[2.2 Scope](#_40qz41btxs84)

[2.3 Requirements](#_b5joi8kutkvr)

[2.3.1 Traceability Matrix](#_khayjwa8jo2y)

[Section 3 - System Architectur](#_yeyfpufiww8s)e

[Section 4 - Data Dictionary](#_drq5ibko1acc)

[Section 5 - Application Software](#_mc3s4e33quku) Components

[5.1 Application Software](#_hzjgbuitbc87) Components

[5.2 Application Software UI components](#_mb7tg0hm62qt)

[Section 6 –](#_wk232hcifrl1) Application Data components

Section 7 - Test cases

[Section 8 – References](#_zigwkt1lv1o6)

[Section 9 – Glossary](#_k7zsnsc4e2v7)

# Section 2 - Overview

## 2.1 Purpose

This document aims to provide a detailed overview of the Travel Blogging Application. It intends to explain the goals, functionalities, and features of the application, define the scope of the project, and define the roles and responsibilities of various user categories. Additionally, this document serves as a guide for the development, implementation, and maintenance of the Travel Blogging Application. It is intended for stakeholders, developers, and all individuals involved in the project's lifecycle.

Brief description of the focus of this module of the overall project and its intended audience.

## 2.2 Scope

Describe the scope of the module to be produced

The Travel Blogging Application is designed to simplify the creation, management, and sharing of travel-related content through a web-based platform. It caters to the needs of both content creators and avid readers interested in travel experiences.

The project's scope includes several key components, including the development of a web-based application that allows users to create and manage detailed travel blog posts, complete with text, images, and expense information. Additionally, it enables users to engage with one another by leaving comments on travel blog posts, encouraging discussions and interactions. The application also enables users to edit and manage their own comments, enhancing their control over their contributions. Moreover, search functionality is implemented to help users discover travel blog posts based on keywords. The platform encourages social engagement by allowing users to express their appreciation for travel blog posts. User registration and profile management features are provided to offer a personalized experience, and **administrators have the tools they need to effectively moderate the application.**

The primary goal of the Travel Blogging Application is to provide an accessible and engaging platform for travel enthusiasts to share their journeys and gain insights into various tourist destinations. With features such as post creation, interactive commenting, and searching capabilities, the Travel Blogging Application aims to enhance user engagement and overall satisfaction.

## 2.3 Requirements

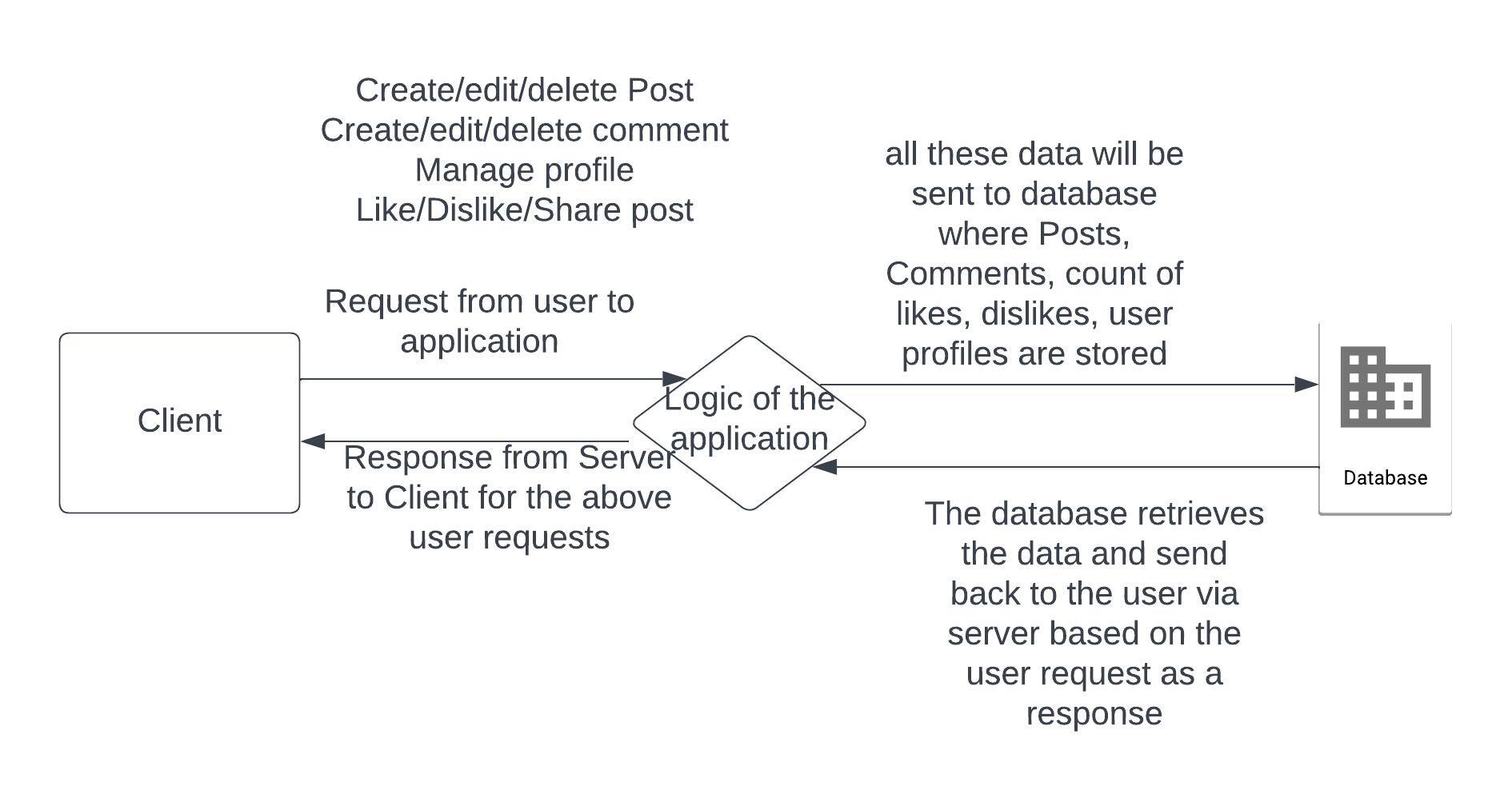
Your mileage may vary -- we typically break down the requirements to provide a ballpark estimate.

### 2.3.1 Traceability Matrix

Cross reference this document with your requirements document and link where you satisfy each requirement

|  |  |
| --- | --- |
| **SRS Requirement** | **SDD Module** |
| Req 1 Search Post | 5.2 |
| Req 2 Add Post | 6 |
| Req 3 Edit Post | 5.2 |
| Req 4 Delete Post | 5.2 |
| Req 5 Share Post | 6 |
| Req 6 Like Post | 5.2 |
| Req 7 Dislike Post | 5.2 |
| Req 8 Add Comment | 6 |
| Req 9 Edit Comment | 5.2 |
| Req 10 Delete Comment | 5.2 |

# Section 3 - System Architecture



The system architecture of my travel blogging application is based on a client-server model. On the client side, users access the application through web browsers, interacting with a responsive and interactive user interface. On the server side, ASP.NET Core MVC handles incoming HTTP requests, coordinating actions between the Model, View, and Controller components. The Model component, connected to a database, manages data retrieval, storage, and business logic, while Controllers process user requests and orchestrate responses. This architecture delivers a robust and responsive platform that enhances user engagement and content creation in the realm of travel blogging.

The main entities in my application are Login page, User, Post, Comment, Like, Dislike, Share.

Below are the attributes of each entity:

**User Entity Attributes**

Username

Email

Password (hashed and salted)

**Travel Post Entity Attributes:**

Title Content

Images or media attachments

Likes count Dislikes count

Shares count

**Comment Entity Attributes**:

Content (text)

User Name of the commenter

**Like Entity** **Attributes:**

Count

**Dislike Entity Attributes:**

Count

**Share Entity**:

Link of the Post

# Relationship between these entities

**User - Travel Post Relationship:**

One-to-Many Relationship

A user can create many travel posts.

A travel post is created by one user.

**User - Comment Relationship:**

One-to-Many Relationship

A user can create many comments.

A comment is created by one user.

**Travel Post - Comment Relationship:**

One-to-Many Relationship.

A travel post can have many comments.

A comment belongs to one travel post.

**User - Like Relationship:**

Many-to-Many Relationship

A user can like many travel posts.

A travel post can be liked by many users.

**Travel Post User - Dislike Relationship**:

Many-to-Many Relationship.

A user can dislike many travel posts.

A travel post can be disliked by many users.

**Travel Post User - Share Relationship:**

Many-to-Many Relationship.

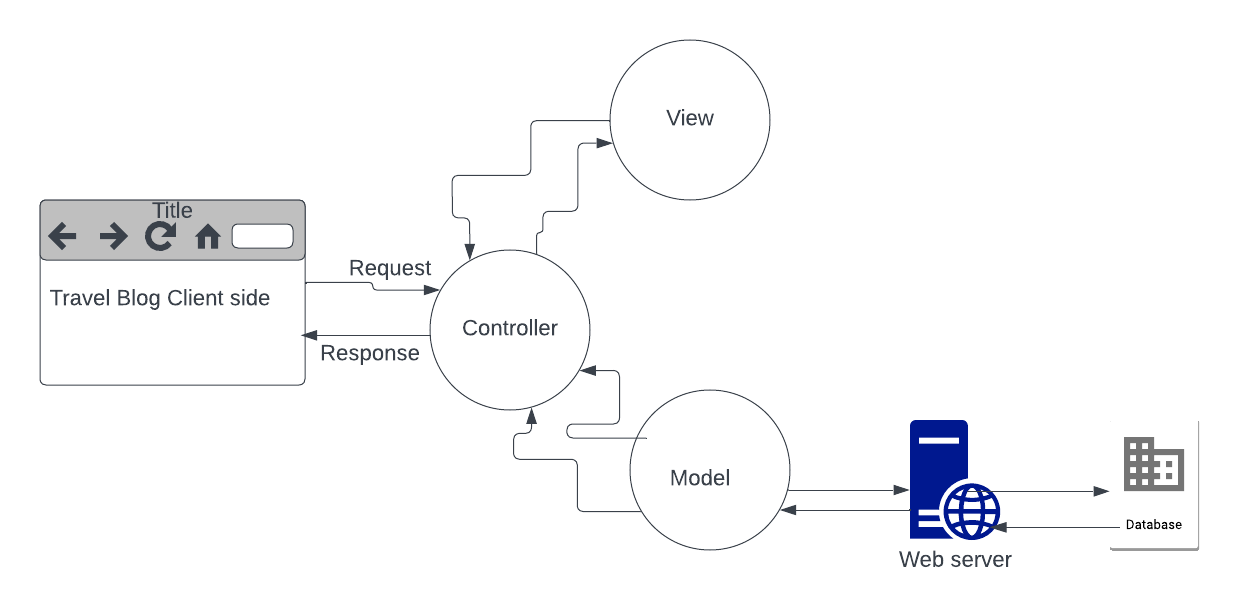
A user can share many travel posts.

A travel post can be shared by many users.

|  |
| --- |
| **Table** |

|  |  |
| --- | --- |
| **Field** | **Type** |
| FIRST NAME | VARCHAR(50) |
| LAST NAME | VARCHAR(50) |
| EMAIL ID | VARCHAR(50) |
| TITLE | VARCHAR(50) |
| PHOTO | VARBINARY(50) |
| TEXT | VARCHAR(50) |
| COMMENT | VARCHAR(50) |

# Section 5 - Application Software Components(*models*)



For the development of this blogging web application, I will be using C# programming language, ASP.NET MVC and SQL Database.

In my travel blogging project, ASP.NET MVC (Model-View-Controller) serves as the architectural framework. The Model component manages data-related tasks such as interacting with the database for user profiles, travel posts, and comments, while the View component handles the presentation layer, rendering dynamic views for users. The Controller acts as the intermediary, processing incoming requests, managing user interactions, and coordinating actions like post creation, comment submission, and user authentication. This framework ensures that my application's data, presentation, and logic are efficiently separated and allows for the seamless implementation of the specified use cases and test cases, resulting in a user friendly travel blogging platform.

**Model:**

Data Handling: The Model component will be responsible for handling all data related to the application. This includes user profiles, posts, comments, likes, dislikes, and shares.

Business Logic: The Model will contain the application's business logic, which includes determining how posts are created, how interactions are recorded, and how data is retrieved and displayed.

**View:**

User Interface: The View component will handle all aspects of the user interface and presentation. It will define how data is displayed to users, including post layouts, comment sections, and user profiles.

**Controller:**

Request Handling: The Controller component will handle incoming HTTP requests from users and determine how to respond to them. For example, when a user wants to create a post, the Controller will handle the request to create a new post.

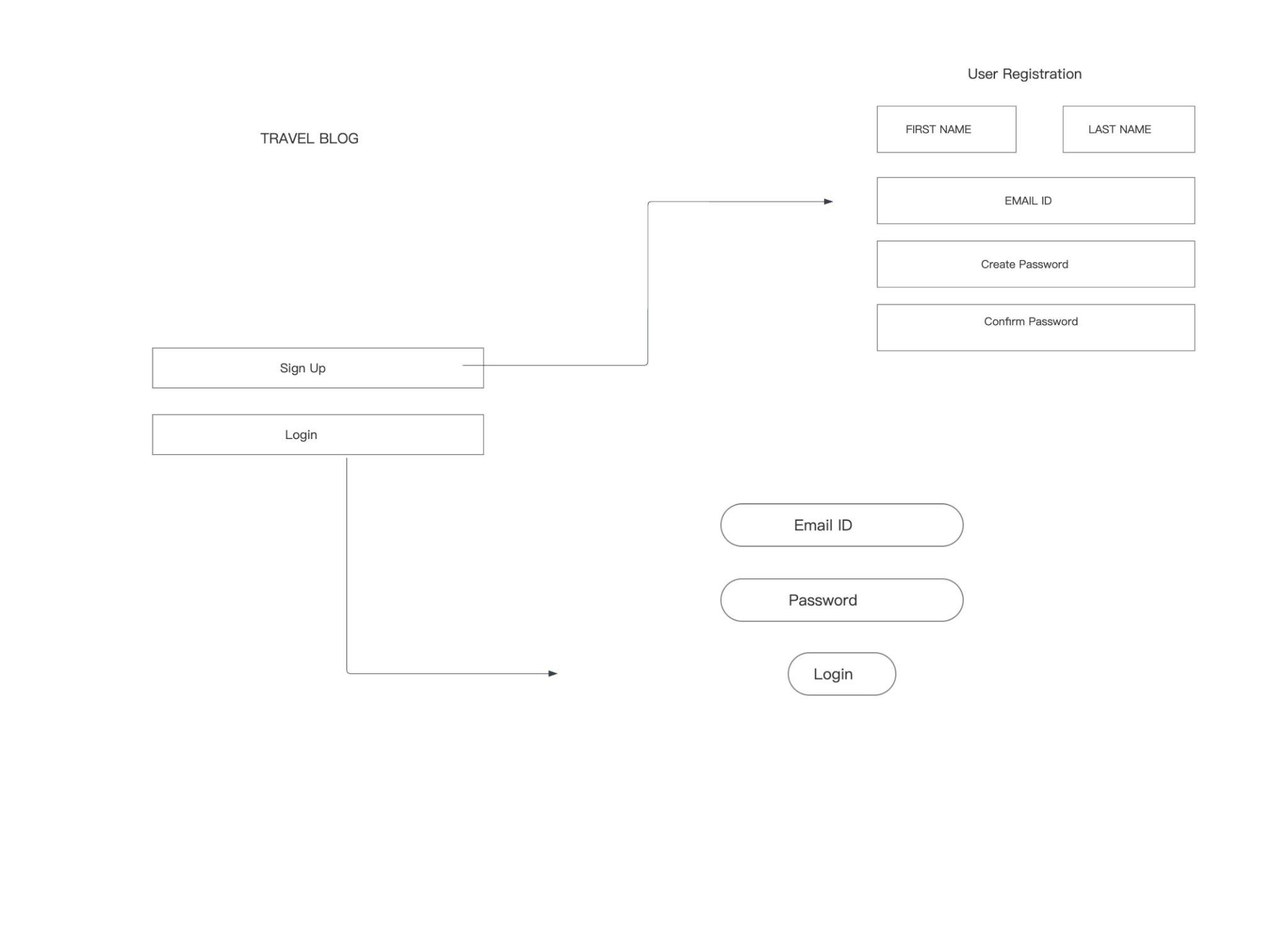
Model Interaction: It will interact with the Model to retrieve data when rendering views and to update data when processing user actions. For instance, when a user likes a post, the Controller will instruct the Model to record the like.

View Rendering: The Controller will decide which View to render based on the user's request and the data retrieved from the Model. It will pass the necessary data from the Model to the View for rendering.

Routing: It will manage URL routing, ensuring that the correct Controller actions are called based on the URL requested by the user.

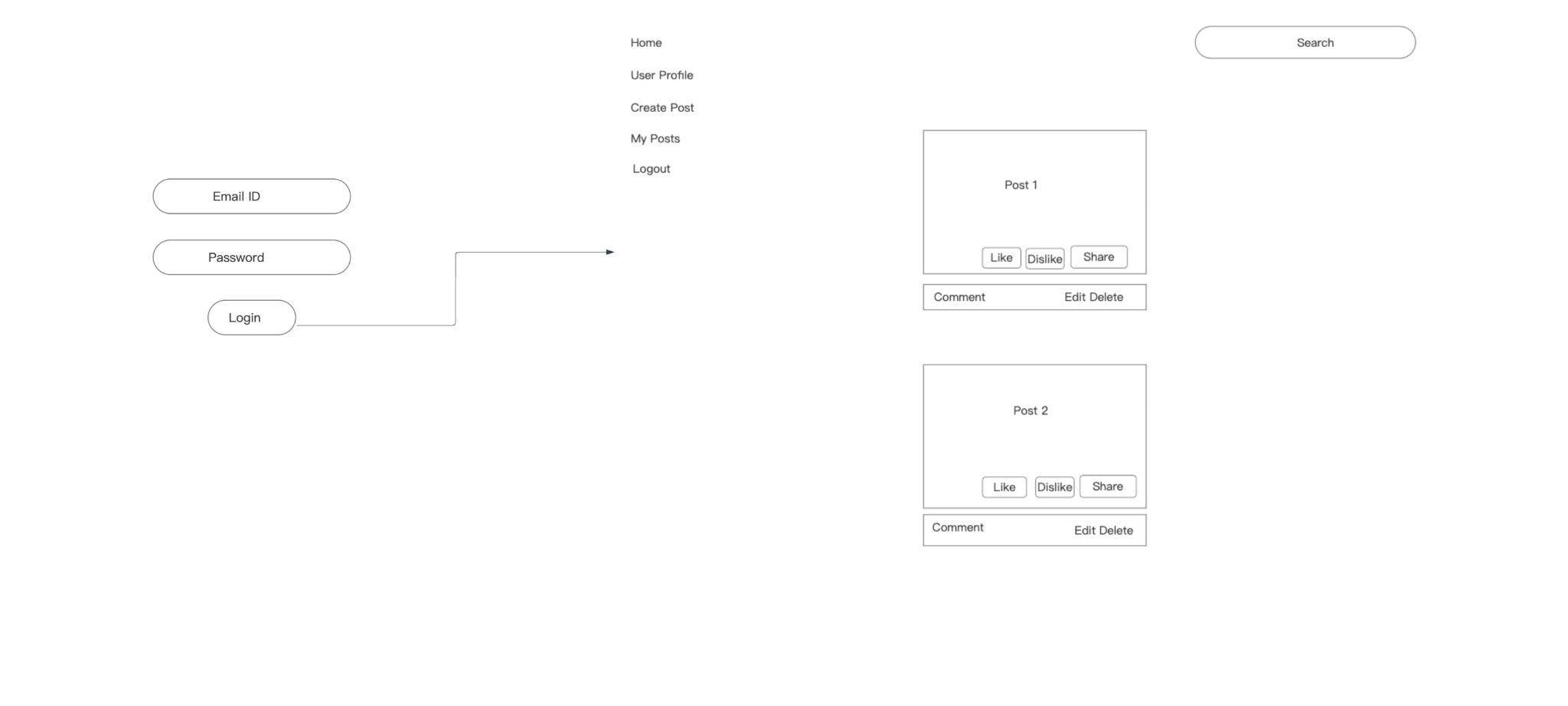
**5.2 Application Software UI Components.**

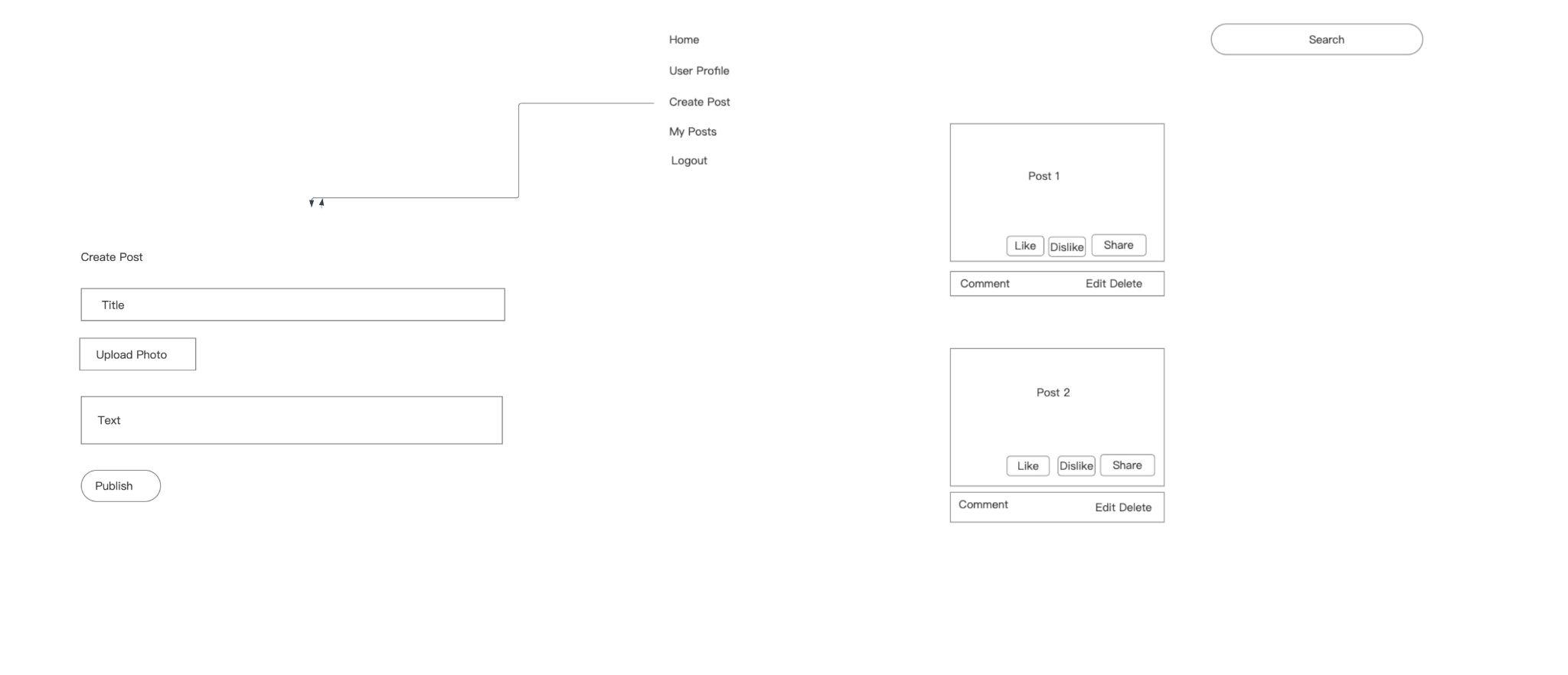
When the user visits the web application, he can sign up if he is a new user or can login if he is the existing user. If the user clicks on the sign up button he will be redirected to user registration page and when he clicks on the login button he can enter his credentials to login.

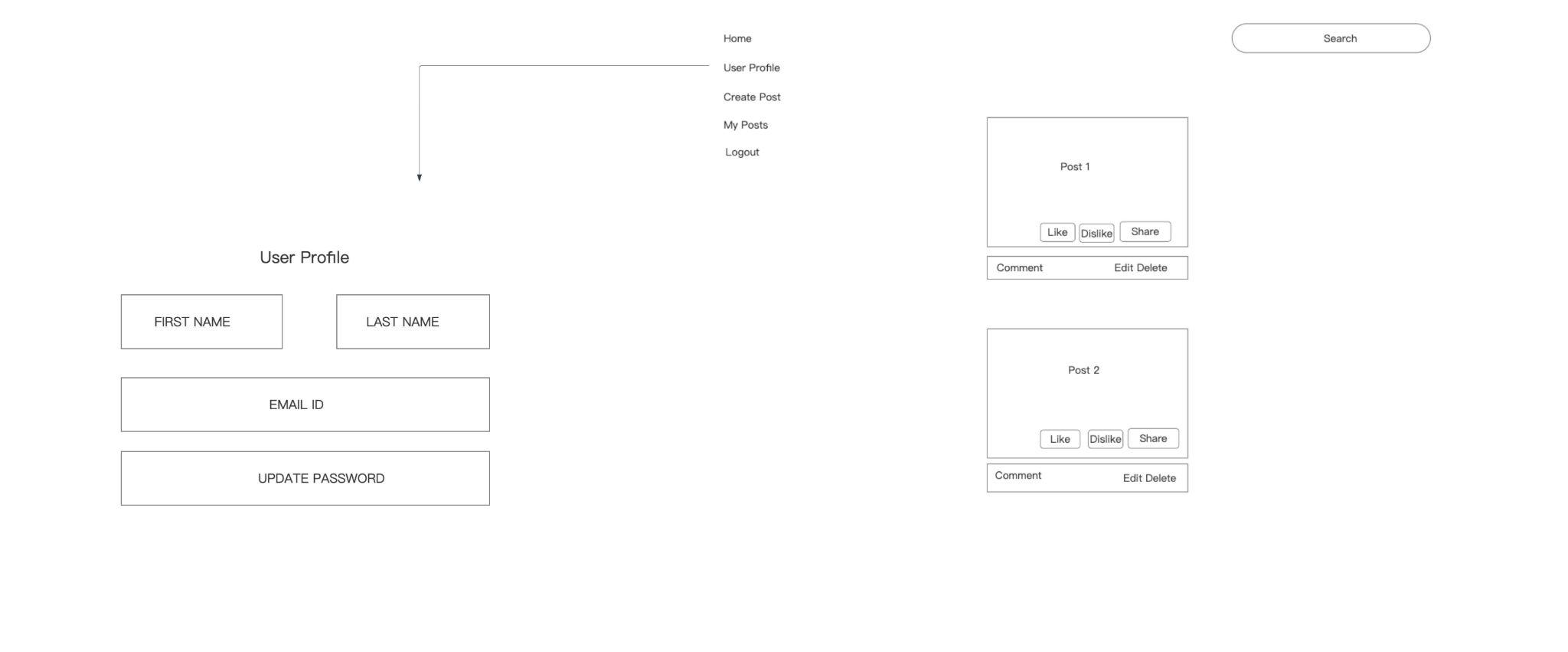
****

When the user login to the application he will be directed to the Home page where he can search for the Posts based on the titles given to the Posts, he has the option to Create Post and also visit his previous posts by clicking on the My Posts option on the home page where he can edit or delete the Post. Moreover, he can edit his profile details by clicking on User Profile where he can view his name and email Id and update his password. The following diagrams explain the flow when a user clicks on options available on the Home page.

On the Home page he can view Posts of other users too, where he can comment, like, dislike and share the Post. He also has the option to edit and delete the comment.

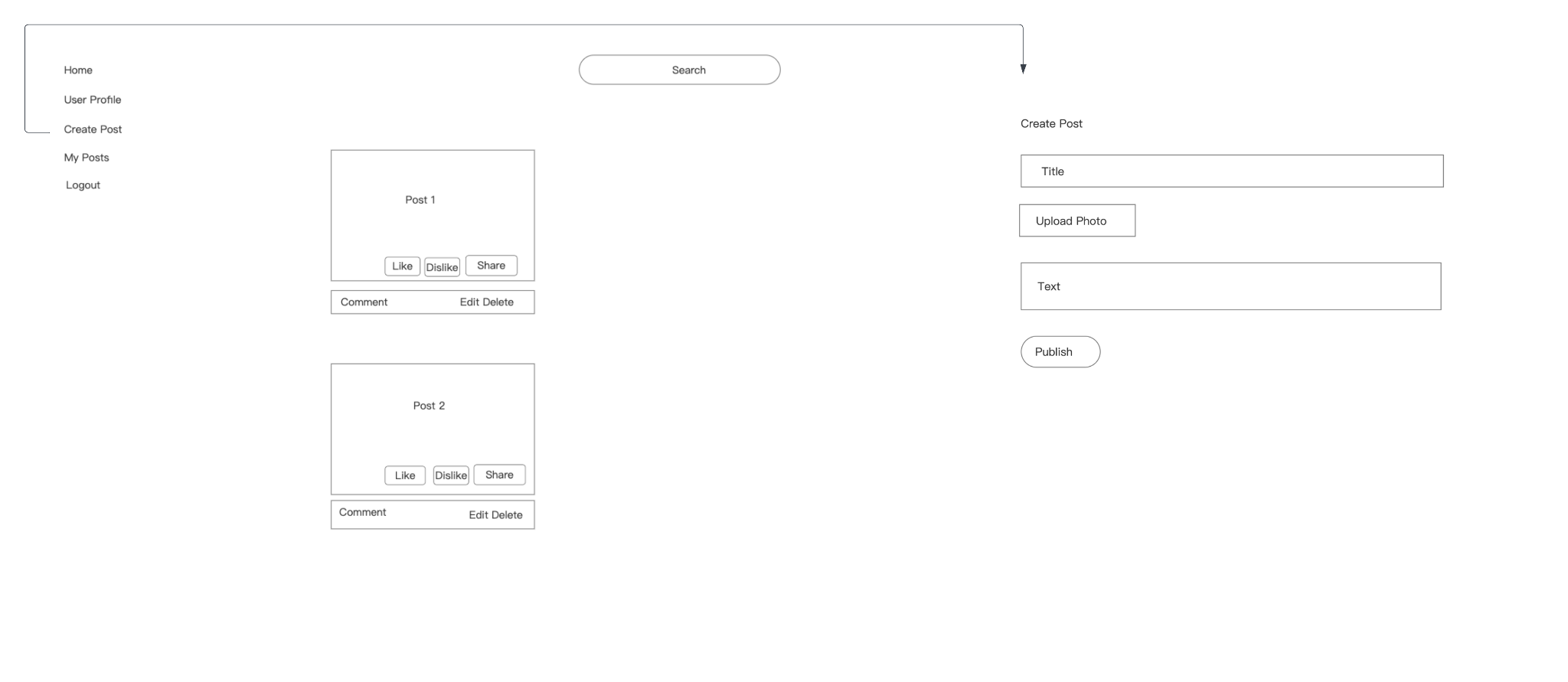








When a user clicks on Create Post, the Create Post page will open, where User can upload images, write text related to the post and then publish. Once the Post is published the Post is visible when searched by other users based on the keywords of the title.



Finally, when the user clicks on log out on the Home page the user logs out of the application.

From the above description, this application has UI components such as User registration and login, User Profile, Post Creation, editing and deleting, Post Comment, edit comment and delete comment, like and dislike the post, option to share the post amd also search functionality component to search for the Post based on the keyword.

# Section 6 –Application Data components

**1. User Data Component:**

This collection stores information about registered users. Username, Email, Hashed Password. User Name serves as the primary key, ensuring uniqueness.

**2. Travel Post Data Component:**

This collection stores data related to travel blog posts.Title, Content (text or HTML), Images.

**3. Comment Data Component:**

This collection stores comments made on travel blog posts. Fields include Comment Content (text), User Name who commented on the Post, serves as the primary key for comments.

**4. Like Data Component:**

This collection keeps track of likes on travel blog posts.

**5. Dislike Data Component:**

This collection records dislikes on travel blog posts.

**6. Share Data Component:**

This shares of travel blog posts.

This data-level architecture helps in efficient storage and retrieval of data for the travel blogging application while accommodating relationships and interactions between users, posts, comments, likes, dislikes, and shares.

# 7. Test Cases

**Test Case 1: Search Post**

* Verify that users can search for posts by keywords.
* Posts matching the search query are displayed. Verify that the displayed posts contain the search query in their titles or content.

**Test Case 2: Add Post**

* Verify that users can create a new travel experience post.
* Click the "Create Post" button then fill in the post details, including title, content and upload image.
* Click the “Publish” button.
* The expected result is a new post is created and added to the user's My Posts section or the main feed and the post's details match the information provided during creation.

**Test Case 3: Edit Post**

* Verify that users can edit their existing posts under My Posts.
* Precondition is that at least the user should have one post uploaded by him.
* Navigate to one of the user's own posts.
* Click the Edit button.
* Modify the post's title, content, or change the image
* Click the Update button.
* The expected result is the post is updated with the edited information and the post's details match the changes made during editing.

**Test Case 4: Delete Post**

* Verify that users can delete their own posts.
* Preconditionis that the user has at least one post.
* Navigate to one of the user's own posts.
* Click the "Delete" button.
* The expected result is that the post is deleted and no longer visible in the user's profile or the main feed.

**Test Case 5: Share Post**

* Verify that users can share a post on social media platforms.
* Precondition is that the user is viewing a post.
* Click the "Share" button on the post.
* Select a social media platform (e.g., Facebook, Twitter).
* Share the post on the selected platform.
* The expected result is that the post is shared on the chosen social media platform.

**Test Case 6: Like Post**

* Verify that users can like a post.
* Precondition is that the user is viewing a post.
* Click the "Like" button on the post.
* The expected result is that the post's like count increases by one.

**Test Case 7: Dislike Post**

* Verify that users can dislike a post.
* The precondition is the user is viewing a post.
* Click the "Dislike" button on the post.
* The expected result is that the post's dislike count increases by one.

**Test Case 8: Add Comment**

* Verify that users can add comments to a post.
* The precondition viewing a post.
* Enter a comment in the comment input field.
* Click the "Submit" button.
* The expected result is thatThe comment is added to the post's comment section and the comment's content matches the text entered by the user.

**Test Case 9: Edit Comment**

* Verify that users can edit their own comments on a post.
* The precondition is that the user has commented on a post, and the comment is visible.
* Locate the user's own comment on the post.
* Click the "Edit" button.
* Modify the comment's content.
* Click the "Update" button.
* The expected result is that the comment is updated with the edited content and the comment's content matches the changes made during editing.

**Test Case 10: Delete Comment**

* Verify that users can delete their own comments on a post.
* The precondition is that has commented on a post, and the comment is visible.
* Locate the user's own comment on the post.
* Click the "Delete" button.
* The expected result is that the comment is deleted and no longer visible in the post's comment section.

# Section 8 – References

<https://raw.githubusercontent.com/dotnet-architecture/eBooks/main/current/architecting-modern-web-apps-azure/Architecting-Modern-Web-Applications-with-ASP.NET-Core-and-Azure.pdf>

<https://jelvix.com/blog/software-design-document>

# Section 9 – Glossary

Glossary of terms / acronyms

|  |  |
| --- | --- |
| Term | Definition |
| Travel Blogging Application | A web-based platform designed to simplify the creation, administration, and sharing of travel-related content among content creators and readers interested in travel experiences. |
| Content Creator | User who generates and manages travel blog posts, including text, images. |
| Commenter | User who engages with travel blog posts by providing comments, opinions, and participating in discussions. |
| Visitor | User interested in travel experiences who reads travel blog posts and comments related to the post on the platform. |
| Search Functionality | Feature that enables users to find travel blog posts based on keywords. |
| Profile Management | Feature that enables users to edit and manage their profiles. |