

ExploreBooking: A Web Application for Travel

Submitted by: Team-Eight

Submitted to: Jake Nesovic

ICET Department,

School of Engineering Technology and Applied Science

Progress Campus, Block A

Centennial College

Discipline: Software Engineering

Due Date: April 21, 2023

Declaration of Sole Authorship

We, Team-Eight, confirm that this work submitted for assessment is our own and is expressed in our own words. Any uses made within it of the works of any other author, in any form (ideas, equations, figures, texts, tables, programs), are properly acknowledged at the point of use. A list of the references used is included.

Signed:

- Arvin Almario
- Chung Yin Tsang
- Kristi Goxhaj
- Gustavo G M L R F Almeida
- David Gutierrez Jimenez
- Adityasinh Makwana
- Egor Shabalkin
- Karanjot Singh
- Saima Rafik Patel

Date: February 18, 2023

Abstract

With an increasing number of traveling tourists from around the world to Albania, more and more people are looking for travel recommendations to Albania. Current websites or applications are offering limited travel ideas for travelers and detailed information for many places in Albania is hard to find. Therefore, the team decides to create an application catering to tourists wanting to explore unfamiliar spots that are only known to locals. The solution the team proposed is a web application called ExploreBooking. The application allows registered users to log in to the system and create posts to share their recommended travel spots in any region in Albania. Users can view posts, filter posts by region, and make comments under each post.

Table of Contents

| | |
|---|-----|
| Declaration of Sole Authorship | 2 |
| Abstract | 3 |
| List of Figures | 6 |
| List of Tables | 8 |
| 1.0 INTRODUCTION | 9 |
| 2.0 METHODOLOGY AND RESULTS | 11 |
| 2.1 Literature Review | 11 |
| 2.2 Proposed Solution | 11 |
| 2.3 User Role Modelling | 15 |
| 2.3.1 Brainstorm and Group | 15 |
| 2.3.2 Consolidated User Roles | 17 |
| 2.3.3 Description of User Roles and Persona | 18 |
| 2.3.4 Additional Documentation | 23 |
| 2.4 Release 1.0 | 24 |
| 2.4.1 User Stories | 24 |
| 2.4.2 Additional Documentation | 54 |
| 2.4.3 Release Plan 1.0 | 55 |
| 2.4.4 Iteration Plan (Release 1.0) | 63 |
| 2.4.4.1 Iteration 1 Planning | 63 |
| 2.4.4.2 Iteration 2 Planning | 71 |
| 2.4.5 Additional Documentation | 76 |
| 2.4.7 Acceptance Tests for Release 1.0 | 77 |
| 3.0 CONCLUSIONS | 86 |
| 4.0 RECOMMENDATIONS | 88 |
| CREDITS, LICENSE, AND REFERENCES | 89 |
| Credits | 89 |
| License | 89 |
| References | 89 |
| APPENDIX A: DESIGN DOCUMENTS | 90 |
| APPENDIX B: TEST PLAN | 104 |
| 1.0 Introduction | 104 |
| 1.0.1 Goals | 104 |

| | |
|--|-----|
| 1.0.2 Assumptions | 104 |
| 1.0.3 Risks and Assets | 105 |
| 2.0 Scope | 106 |
| 2.0.1 Features to Be Tested | 106 |
| 2.0.2 Features Not to Be Tested | 108 |
| 3.0 Testing Procedures | 108 |
| 3.0.1 Test Objectives | 115 |
| 3.0.2 Types of Testing | 115 |
| 3.0.2.1 Unit Testing | 115 |
| 3.0.2.2 Integration Testing | 116 |
| 3.0.2.3 Acceptance Testing | 116 |
| 3.0.2.4 Stress Testing | 123 |
| 3.0.2.5 Performance Testing | 123 |
| 3.0.3 Testing Tools | 123 |
| 4.0 Schedule and Deliverables | 124 |
| APPENDIX C: END-USER & ADMINISTRATOR MANUALS | 126 |
| APPENDIX D: PROGRESS MONITORING | 150 |
| APPENDIX E: PROJECT PROPOSAL | 153 |

List of Figures

- Figure 1: A software network architecture diagram.13
Figure 2: Organizing the user role cards on a table.16
Figure 3: The consolidated role cards.17
Figure 4: Low-fidelity Tree for Traveler.25
Figure 5: Low-fidelity Tree for Post Writer.25
Figure 6: Low-fidelity Tree for Administrator.26
Figure 7: User story card 'Traveler - View Post Overviews Story'41
Figure 8: User story card 'Traveler - Filter Posts Story'42
Figure 9: User story card 'Traveler - Sign Up Story'42
Figure 10: User story card 'Traveler - Search Post Story'43
Figure 11: User story card 'Traveler - View Profile Story'43
Figure 12: User story card 'Traveler - Edit Profile Story'44
Figure 13: User story card 'Traveler - View Post Detail Story'44
Figure 14: User story card 'Traveler - Add Comments Story'45
Figure 15: User story card 'Traveler - View Comments Story'45
Figure 16: User story card 'Post Writer - View Post Overview Story'46
Figure 17: User story card 'Post Writer - Filter Posts Story'46
Figure 18: User story card 'Post Writer - Sign Up Story'47
Figure 19: User story card 'Post Writer - View Profile Story'47
Figure 20: User story card 'Post Writer - Edit Profile Story'48
Figure 21: User story card 'Post Writer - Add Post Story'48
Figure 22: User story card 'Post Writer - Edit Post Story'49
Figure 23: User story card 'Post Writer - Delete Post Story'49
Figure 24: User story card 'Administrator - View Post Overviews Story'50
Figure 25: User story card 'Administrator - Filter Posts Story'50
Figure 26: User story card 'Administrator - Search Post Story'51
Figure 27: User story card 'Administrator - View Post Detail Story'51
Figure 28: User story card 'Administrator – Delete Post Story'52
Figure 29: User story card 'Authentication Constraint'52
Figure 30: User story card 'Authentication Constraint' (Continued)53
Figure 31: Paper Prototype 'A user can view post overviews and filter posts'.90
Figure 32: Paper Prototype 'A user can view and add comments'.91
Figure 33: Paper Prototype 'The system should do authentication'.92
Figure 34: Paper Prototype 'A user can sign up'.93
Figure 35: Paper Prototype 'A user can add a post'.94
Figure 36: UML Class Diagram.96
Figure 37: ERD Diagram.97
Figure 38: CRC Cards.98
Figure 39: Use Case Diagram.99
Figure 40: Login Activity Diagram.100
Figure 41: Add Post Activity Diagram.101
Figure 42: Login Sequence Diagram.102

- Figure 43: Create a Post Sequence Diagram.103
Figure 44: Download Visual Studio Code.126
Figure 45: Download the ExploringBooking app ZIP file.127
Figure 46: Open Folder in Visual Studio Code.127
Figure 47: Change the Terminal in Visual Studio Code.128
Figure 48: Running npm install.129
Figure 49: Result of Running npm install.129
Figure 50: Running node server.js.130
Figure 51: Navigate to localhost:3000.130
Figure 52: Login/Signup button on Home page.131
Figure 53: Sign Up hyperlink in Login page.131
Figure 54: Signup Form.132
Figure 55: MongoDB Compass Collection.133
Figure 56: MongoDB Compass - User folder.133
Figure 57: An example of data in MongoDB Compass.134
Figure 58: Update the data in MongoDB Compass.135
Figure 59: Login Form.136
Figure 60: "Admin" shown in the Home page.137
Figure 61: Posts in the Home page.138
Figure 62: Delete button in the Post Details of a post.138
Figure 63: Confirmation after deletion.139
Figure 64: Iteration Burndown Chart.151

List of Tables

- Table 1: Initial List of User Stories 30
Table 2: The must-have stories for Release – Initial version. 39
Table 3: The should-have stories for Release – Initial version. 41
Table 4: Combined User Stories For Release Plan 58
Table 5: Must-have Stories for Release Plan. 59
Table 6: Should-have Stories for Release Plan. 60
Table 7: Could-have Stories for Release Plan. 61
Table 8: The Finished Release Plan. 61
Table 9: Tasks for ‘A user can view post overviews including post titles and post images’. 67
Table 10: Tasks for ‘A user can filter posts by region’. 67
Table 11: Tasks for ‘A user can search post with keywords’. 68
Table 12: Tasks for ‘A user can add a post’. 68
Table 13: Tasks for ‘A user can edit a post’. 69
Table 14: Tasks for ‘A user can view the post details’. 69
Table 15: Tasks for ‘A user can delete a post’. 70
Table 16: Tasks for ‘A user can add comments under a post’. 72
Table 17: Tasks for ‘A user can view comments from other travelers’. 73
Table 18: Tasks for ‘A user can sign up by email’. 73
Table 19: Tasks for ‘The system should do authentication to only allow authorized users to add, edit, and delete posts’. 74
Table 20: Tasks for ‘A user can view profile information’. 74
Table 21: Tasks for ‘A user can edit profile information’. 75
Table 22: Stories, acceptance tests, and contributors for Release 1.0 (Green=Passed; Red=Failed). 77
Table 23: Features within Testing Scope. 106
Table 24: Testing scenarios in scope. 116
Table 25: Schedule for Testing activities. 124
Table 26: Deliverables. 125
Table 27: Progress and changes for a release with two iterations. 150
Table 28: Total Estimated and Actual Hours for Each Developer. 152

1.0 INTRODUCTION

Recently, a Virtuoso study reveals that the number of travelers booking trips for 2023 on average is far more than that in 2019. According to the article on Euractiv and Global Voices, 2022 was a bumper year of tourism for Albania, and there is a raising trend in tourism. However, there are difficult to gain travel detailed recommendations for Albania due to the lack of applications focusing on a travel guide to Albania. The current solutions provide limited travel recommendations for Albania and lack of comments from travelers and local people. To facilitate the development of tourism in Albania, the government and travel-related companies may want a solution to promote different attractions in Albania. It brings a vast opportunity to connect adventurous tourists with the potential but not-widely-known touristic attractions recommended by locals.

ExploreBooking includes several functions such as the sign-up and login functions. Users can create an account and log in to the system. The main feature of the web application is the posting function for travel recommendations. Logged-in users can click the ‘add a new post’ button to create a new post. They can input the content of the post which includes the title, the text content, and an image. The web application also has the view post summary, filtering post, search post, view post details, and comment function. The users can view the overviews of all posts on the homepage. They can filter the posts by selecting a specific region in Albania. They can search posts by keyword. When they click on

a post, they can view the details of the post and all comments on the post. They can also add a comment under the post no matter whether they are logged in or not. If they are logged in, the username of the user will be shown beside the comment.

This project has several deliverables when it is completed. Deliverables include a web application zip file and access to the database (using MongoDB) which is connected to the application. Deliverables exclude test data and servers.

2.0 METHODOLOGY AND RESULTS

2.1 Literature Review

There are web applications related to travel recommendations such as Rough Guides, which is a travel guide website. In Rough Guides, there is a column for Albania. It has articles stating some travel advice for Albania including where to go, when to go, and what activities to join. It also provides some basic facts about Albania, such as the culture and the crime rate. However, as the web application, Rough Guides, is an application providing travel guides for many different countries, it does not focus on introducing Albania. It only introduces a few regions in Albania with a few recommendations. There are only one to two short articles, which are mostly written by one author, sharing travel information. There are some Albanian-centered travel guide pages such as tourisminalbania.com and anitahendrieka.com, but they only have information on common attractions and there is a lack of local people's involvement in the recommendations made. Our product will include every region in the country and allow Albanians to make posts and share their travel ideas and recommendations for each region.

2.2 Proposed Solution

The solution proposed here is a web application sharing travel recommendations from locals in Albania. The application will offer a platform for locals to share and promote their recommended places, and for travelers to gain up-to-date and

detailed travel information as well as real comments from other travelers. The project will benefit travelers that they can get detailed travel recommendations easily. It will also benefit the travel industry of Albania as the project will promote potential attractions in Albania. The project will be successful if at least 1% of tourism to Albania obtained travel information from it.

One of the strengths of this system is that the web application provides a platform for locals in Albania, who may not have a large influence on the Internet, to share their recommended places easily only by signing up for an account. Another strength is that the web application allows users to view travel information for all areas in Albania. The application also has the strength of being easy to access. As it is a web application, users can easily access the application through mobile devices or computers via the Internet, and no need to download or install anything except a browser. However, due to the very limited time for the project, the weakness of our solution is that the booking of flights and hotels cannot be done within the same websites.

Architecture Diagram

The architecture diagram of our application is shown below.

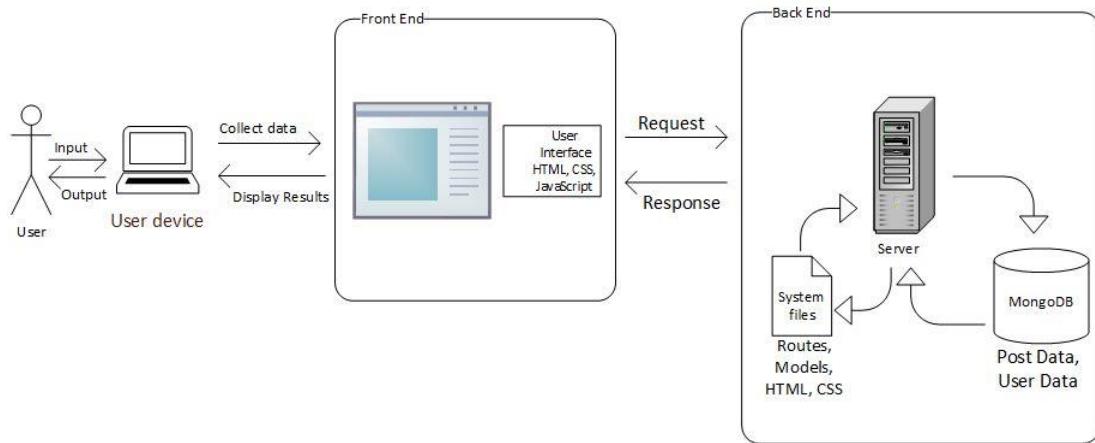


Figure 1: A software network architecture diagram.

The software network architecture diagram has various components that interact with each other in following ways:

1. User: The end-user of the application, who interacts with the user interface via user device.
2. User device: The device used by user to interact with the application, which takes the user input and displays the output.
3. User Interface: The Front End of the application, which is displayed on the user device, it is responsible to process the collected data and send request to the Backend and receives the Response and display the generated response, it is made up of HTML, CSS, JavaScript.
4. Server: The backend of the application, Server takes the request from the

user interface (front end) and interacts with system files and database and provides front end with a response.

5. System files: System files, consist of models, routes, HTML, CSS and JavaScript, which helps the server to generate a desired response.

6. MongoDB: The database for the web application which is used to store, update, and retrieve the post and user data.

The Data flow the software network architecture diagram (figure 1) is as follow:

1. With the help of user device, a user interacts with the user interface by searching, adding posts, adding comments, etc....
2. The user interface collects the data and send the requests made by user to the backend server and displays the received response.
3. The backend server takes receives the request from the user interface and interacts with system files and Database to process the desired request.
4. The backend server sends the generate response to the user interface, which is displayed to the user.

2.3 User Role Modelling

2.3.1 Brainstorm and Group

From our brainstorming session, there came out with 10 initial user roles as listed below:

- Traveler (a person who travels to Albania)
- First-time Traveler (A person who has no previous travel experience)
- Constant Traveler (A person who travels constantly)
- Travel Agency (A company that provides travel services)
- External Content Creator (A person who creates travel content on social media such as YouTube)
- Post Writer (A person who writes posts of recommendations on the web application)
- Local Attraction Owner (A person who owns a local attraction in Albania)
- Local in Albania (A person living in Albania)
- Administrator (A person who manages the system of the application)
- Content Moderator (A person who monitors the contents of posts on the web application)

We organized them into three groups due to similarity as shown in the figure below. The first group contains five roles, which are *Traveler*, *First-time Traveler*, *Constant Traveler*, *Travel Agency*, *External Content Creator*. They are grouped together because they are the one who gains information from the application. The second group contains three roles, which are *Post Writer*, *Local Attraction Owner*, *Local in Albania*. They are in the same group since they are the one who

provides recommendations and creates posts. The last group contains two roles, which are *Administrator*, and *Content Moderator*. The two roles are grouped together as they are the one who manages and monitors the application.

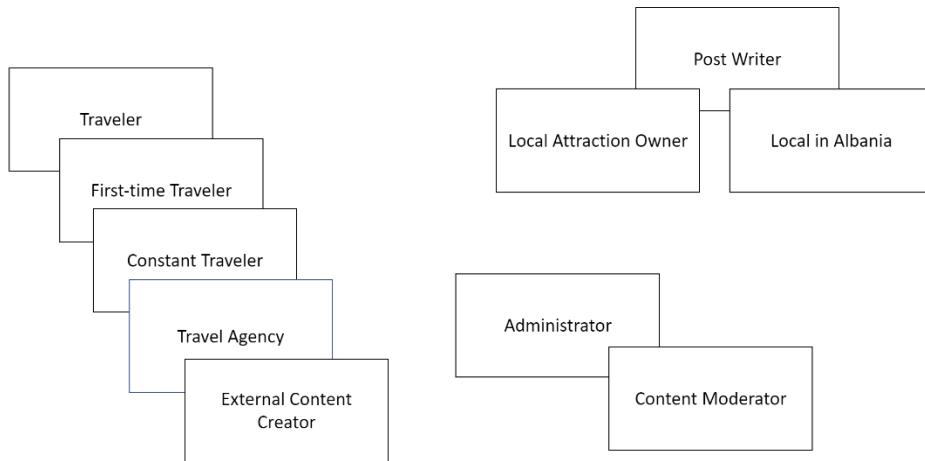


Figure 2: Organizing the user role cards on a table.

2.3.2 Consolidated User Roles

We consolidated the user roles into three main user roles, which are *Traveler*, *Administrator*, and *Post Writer*. *Traveler* is the main role because *First-time Traveler*, *Constant Traveler*, and *External Content Creator* are different types of *Traveler*, and *Travel Agency* will use the application in a similar way to obtain travel recommendations. *Administrator* is the main role as *Content Moderator* is a special type of administrator. Both *Local in Albania* and *Local Attraction Owner* will create posts on the application, and therefore, they are a type of *Post Writer*.

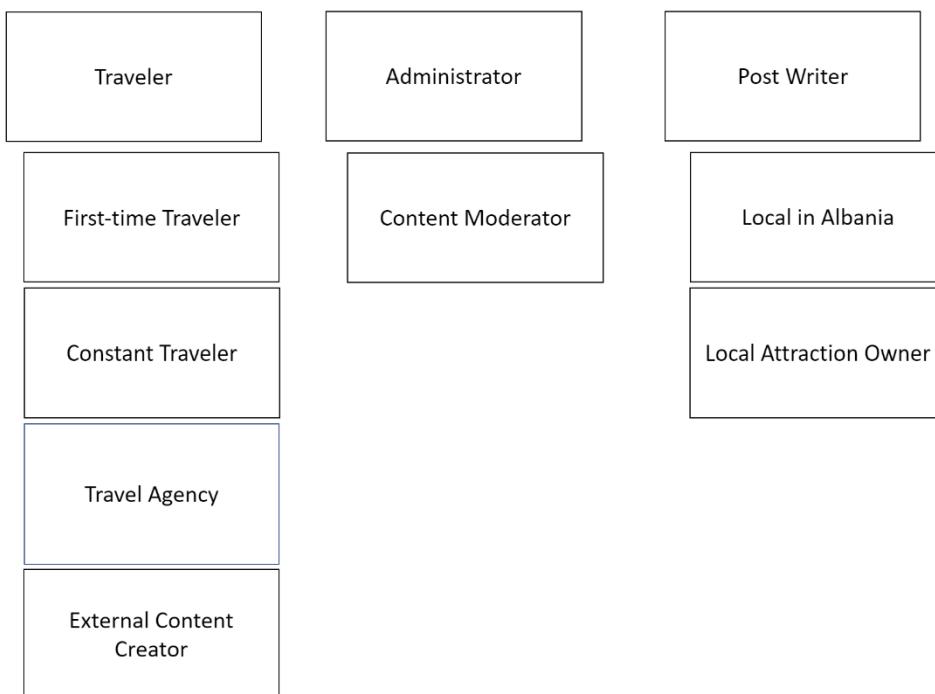


Figure 3: The consolidated role cards.

2.3.3 Description of User Roles and Persona

Traveler

- The user has a high frequency of using the software.
- The user has a low level of expertise with the domain.
- The user has a low general level of proficiency with computers and software.
- The user has a low level of proficiency with the software being developed.
- The user's general goal for using the software is to obtain travel information and recommended places in Albania.

First Time Traveler

- The user has a moderate frequency of using the software.
- The user has a low level of expertise with the domain.
- The user has a low general level of proficiency with computers and software.
- The user has a low level of proficiency with the software being developed.
- The user's general goal for using the software is to obtain travel information and diverse or alternative tourist recommendations to visit Albania.

Constant Traveler

- The user has a high frequency of using the software.

- The user has low level of expertise with domain.
- The user has low level of proficiency with computers and software.
- The user has moderate level of proficiency with the software being developed.
- The user's general goal for using the software is to get travel information, and recommendations about places to visit in Albania.

Travel Agency

- The user has a high frequency of using the software.
- The user has a low level of expertise with the domain.
- The user has a low general level of proficiency with computers and software.
- The user has a low level of proficiency with the software being developed.
- The user's general goal for using the software is to search for local recommendations from Albania for them to use in their travel Itinerary on their potential clients.

External content creator

- The user has moderate frequency of using the software.
- The user has low level of expertise with domain.
- The user has moderate level of proficiency with computers and software.
- The user has low level of proficiency with the software being developed.

- The user's general goal for using the software is to get travel recommendation from locals in Albania, in order to create personal digital content.

Administrator

- The user has a high frequency of using the software.
- The user has a high level of expertise with the domain.
- The user has a high general level of proficiency with computers and software.
- The user has a high level of proficiency with the software being developed.
- The user's general goal for using the software is to maintain the application so that other users can navigate and use the web application smoothly.

Content moderator

- The user has a high frequency of using the software.
- The user has a high level of expertise with the domain.
- The user has a high general level of proficiency with computers and software.
- The user has a high level of proficiency with the software being developed.

- The user's general goal for using the software is to maintain and enforce the principles of civility, relevancy, and locality on the tourist attraction recommendations; and the ones of civility and relevancy in every other posting activity.

Post Writer

- This user has a high frequency of using the software.
- This user has a low level of expertise with the domain.
- The user has a low general level of proficiency with computers and software.
- The user has a low level of proficiency with the software being developed.
- The user's general goal for using the software is to post the most attractive destinations and tour guides in Albania.

Local in Albania

- This user has a low frequency of using the software.
- This user has a low level of expertise with the domain.
- This user has a low general level of proficiency with computers and software.
- This user has a low level of proficiency with the software being developed.
- This user's general goal for using the software is to recommend places in Albania for the travelers.

Local Attraction Owner

- This user has a low frequency of using the software.
- This user has a low level of expertise with the domain.
- This user has a low general level of proficiency with computers and software.
- This user has a low level of proficiency with the software being developed.
- This user's general goal for using the software is to recommend the places they own in Albania for the travelers.

2.3.4 Additional Documentation

The video from our workshop is included here.

URL to the video on YouTube: https://youtu.be/MR_drb37PKw

2.4 Release 1.0

2.4.1 User Stories

At this stage, we decided to combine user roles because of similarity. In the above sections, nine user roles are defined initially. *Traveler*, *First Time Travel*, *Constant Traveler*, and *Travel Agency* are combined since they have the same general goal, which is to obtain information about attractions in Albania. *Post Writer*, *Local in Albania*, and *Local Attraction Owner* are combined since they have the same general goal, which is to recommend local attractions. *Administrator* and *Content Moderator* are combined since they both have a high frequency of using the software, and a high level of expertise with the domain, proficiency with computers and software, proficiency with the software being developed, and they have the same general goal, which is to maintain the software. Three main user roles defined in the User Modeling section, *Traveler*, *Post Writer*, and *Administrator*, are used for creating user stories.

We have created three low-fidelity trees for the three main user roles, *Traveler*, *Post Writer*, and *Administrator*. The three low-fidelity trees are shown below. For all roles, there are Home Page containing post overviews and a post filter. There are Profile and Sign Up for *Traveler* and *Post Writer*. There are Search Post, Search Results, and Post Details for *Traveler* and *Administrator*. Moreover, there are Comments for *Traveler*, Add Post, Edit Post, and Delete Post for *Post Writer*, and Delete Post for *Administrator*.

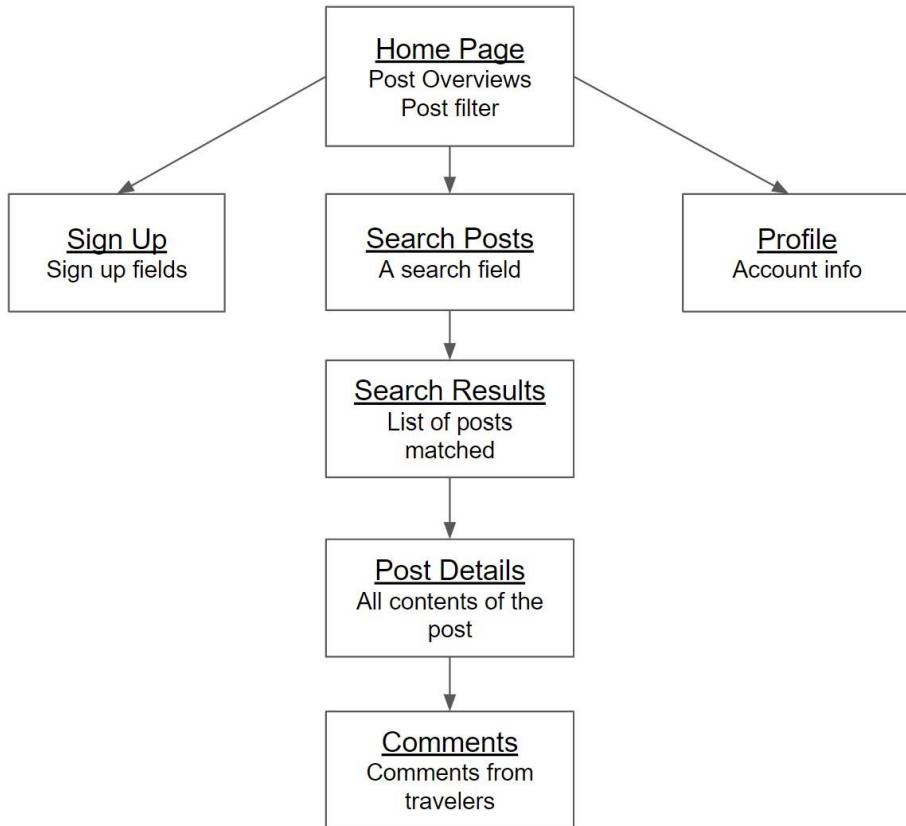


Figure 4: Low-fidelity Tree for Traveler.

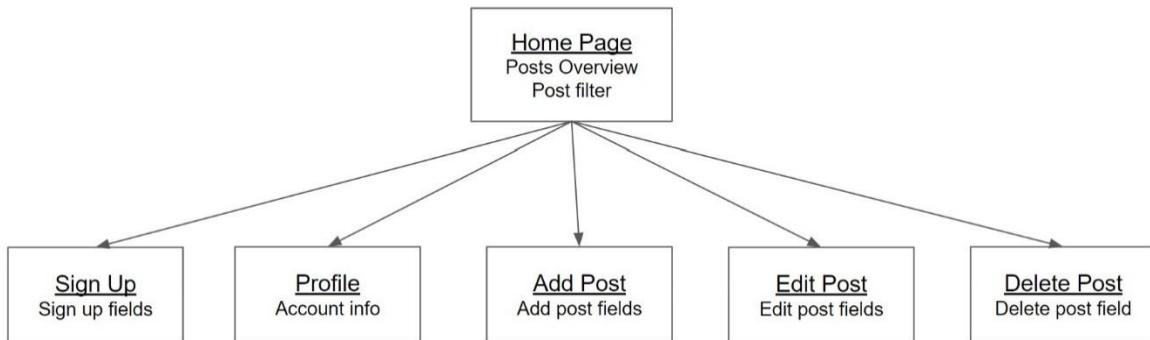


Figure 5: Low-fidelity Tree for Post Writer.

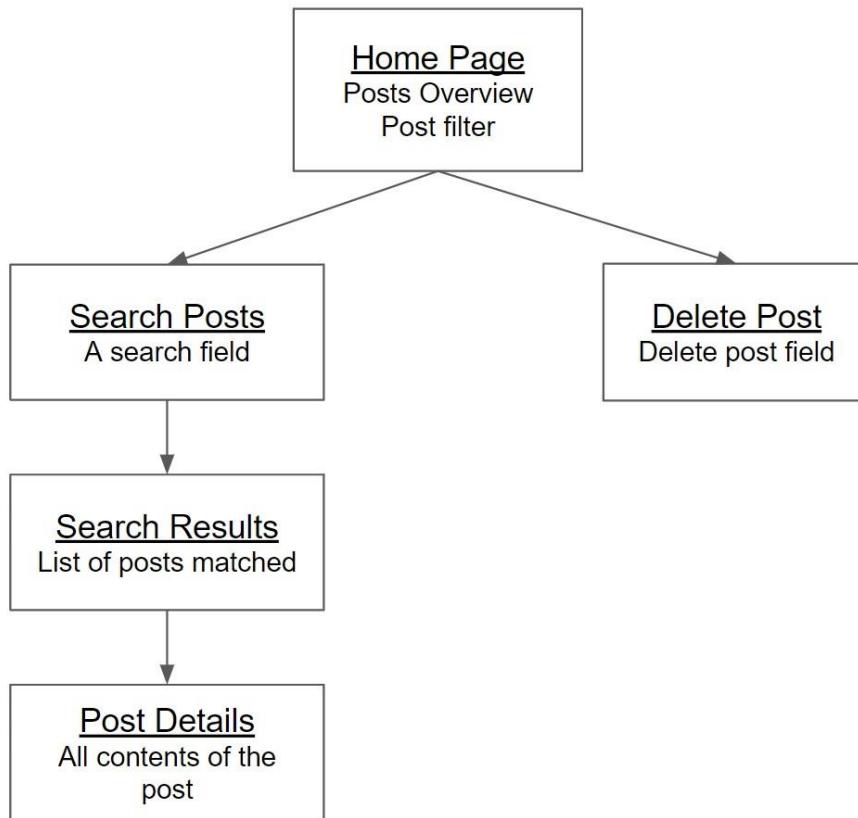


Figure 6: Low-fidelity Tree for Administrator.

Stories for Traveler

We decided to start with traveler. Based on the low-fidelity diagram, from actions under Home Page, we have come up with two stories.

- As a traveler, I want to view post overviews including post title and post image, so that I can know what the posts are briefly about.
- As a traveler, I want to filter posts by regions, so that I can find posts in the region.

For next, *Traveler* can do actions for sign up, search posts or profile. In the low-

fidelity diagram, the user reaches search results after search posts. We have defined one search post story for search posts and search results. For the profile actions, the user can view and edit the profile information. Therefore, we have the following stories.

- As a Traveler, I want to sign up with email, so that I can get an account easily.
- As a Traveler, I want to search posts with keywords, so that I can find posts easily.
- As a Traveler, I want to view my profile information including user type, username, first name, last name and email address, so that I can know which account it is.
- As a Traveler, I want to edit my profile information including password, first name, last name and email address, so that I can update my information.

After searching a post, *Traveler* can do actions related to post details followed by comments. We have defined one story for post details and two for comments.

- As a Traveler, I want to view the post details, so that I can get recommendation information.
- As a Traveler, I want to add comments under a post, so that I can provide feedback.
- As a Traveler, I want to view comments from other travelers, so that I can

see the feedback to help me make travel decisions.

Stories for Post Writer

Next, we discussed user stories for *Post Writer*. Similar to *Traveler*, *Post Writer* can do actions under the Home Page, and do actions for sign up and profile after that. Hence, we have defined the following stories.

- As a Post Writer, I want to view post overviews including post title and post image, so that I can know the existing posts on the page.
- As a Post Writer, I want to filter posts by regions, so that I can find posts in the region.
- As a Post Writer, I want to sign up with email, so that I can get an account easily.
- As a Post Writer, I want to view my profile information including user type, username, first name, last name and email address, so that I can know which account it is.
- As a Post Writer, I want to edit my profile information including password, first name, last name and email address, so that I can update my information.

Furthermore, *Post Writer* can add, edit and delete a post. We have come with three stories corresponding to each action.

- As a Post Writer, I want to add a post, so that I can make

recommendations.

- As a Post Writer, I want to edit a post, so that I can change what is written in the recommendation.
- As a Post Writer, I want to delete a post, so that I can remove the post from the web page.

Stories for Administrator

After that, we discussed the stories for *Administrator*. Similar to *Traveler*, *Administrator* can reach the Home Page, and then search posts and view post details. More than that, *Administrator* can also delete any post on the website. We have defined the following stories.

- As an Administrator, I want to view post overviews including a post title and a post image, so that I can know what the posts are briefly about.
- As an Administrator, I want to filter posts by regions, so that I can find posts in the region.
- As an Administrator, I want to search posts with keywords, so that I can find posts easily.
- As an Administrator, I want to view the details of the post, so that I can evaluate the content.
- As an Administrator I want to delete a recommendation post so I can manage what will be on the page.

Constraint

After discussing stories for all user roles, we still want a constraint to do authentication to only allow specific users to do certain actions. We want the system to prevent *Traveler* from performing actions including add post, edit post and delete post, but allow *Post Writer* to add a post, edit and delete its own post, and allow *Administrator* to delete any posts. Therefore, we have come up with the below constraint.

- The system should do authentication to only allow authorized users to add, edit, and delete post.

List of Stories

A list of user stories initially defined is shown below.

Table 1: Initial List of User Stories

| Story |
|--|
| • As a Traveler, I want to view post overviews including post title and post image, so that I can know what the posts are briefly about. |
| • As a Traveler, I want to filter posts by regions, so that I can find posts in the region. |
| • As a Traveler, I want to sign up with email, so that I can get an account easily. |
| • As a Traveler, I want to search posts with keywords, so that I can find posts easily. |
| • As a Traveler, I want to view my profile information including user type, username, first name, last name and email address, so that I can know which account it is. |
| • As a Traveler, I want to edit my profile information including password, first name, last name and email address, so that I can update my information. |
| • As a Traveler, I want to view the post details, so that I can get recommendation information. |
| • As a Traveler, I want to add comments under a post, so that I can provide feedback. |

| |
|---|
| <ul style="list-style-type: none"> As a Traveler, I want to view comments from other travelers, so that I can see the feedback to help me make travel decisions. |
| <ul style="list-style-type: none"> As a Post Writer, I want to view post overviews including post title and post image, so that I can know the existing posts on the page. |
| <ul style="list-style-type: none"> As a Post Writer, I want to filter posts by regions, so that I can find posts in the region. |
| <ul style="list-style-type: none"> As a Post Writer, I want to sign up with email, so that I can get an account easily. |
| <ul style="list-style-type: none"> As a Post Writer, I want to view my profile information including user type, username, first name, last name and email address, so that I can know which account it is. |
| <ul style="list-style-type: none"> As a Post Writer, I want to edit my profile information including password, first name, last name and email address, so that I can update my information. |
| <ul style="list-style-type: none"> As a Post Writer, I want to add a post, so that I can make recommendations. |
| <ul style="list-style-type: none"> As a Post Writer, I want to edit a post, so that I can change what is written in the recommendation. |
| <ul style="list-style-type: none"> As a Post Writer, I want to delete a post, so that I can remove the post from the web page. |
| <ul style="list-style-type: none"> As an Administrator, I want to view post overviews including a post title and a post image, so that I can know what the posts are briefly about. |
| <ul style="list-style-type: none"> As an Administrator, I want to filter posts by regions, so that I can find posts in the region. |
| <ul style="list-style-type: none"> As an Administrator, I want to search posts with keywords, so that I can find posts easily. |
| <ul style="list-style-type: none"> As an Administrator, I want to view the details of the post, so that I can evaluate the content. |
| <ul style="list-style-type: none"> As an administrator I want to delete a recommendation post so I can manage what will be on the page. |
| <ul style="list-style-type: none"> The system should do authentication to only allow authorized users to add, edit, and delete posts. |

Estimation

To estimate the stories, we have an online meeting. We used an online tool to do the estimation. The tool allows us to select a card from a set of numbers secretly and then reveal the selections at the same time. We estimated the workloads using story points, 3, 5, 8, or 13 points. The definition of story points is that 3

points represent 5 hours, 5 points represent almost 8 hours of work as a team, 8 points represent 13 hours, and 13 points represent 24 hours.

[View Post Overviews Stories](#)

The first group of stories is the view post overviews stories listed below:

- As a Traveler, I want to view post overviews including post title and post image, so that I can know what the posts are briefly about.
- As a Post Writer, I want to view post overviews including post title and post image, so that I can know the existing posts on the page.
- As an Administrator, I want to view post overviews including a post title and a post image, so that I can know what the posts are briefly about.

We started the estimation. Each of us selected a card with a number. The result of the first round of estimation was:

- David: 3
- Karanjot: 3
- Gustova: 5
- Saima: 5
- Chung Yin: 8
- Adityasinh: 8
- Egor: 13
- Kristi: 13
- Arvin: 13

David explained that the task will be easy as it is to display something only. Arvin considered the three view post overviews stories as a heavy workload because we will need to connect to the database and retrieve data of all posts to show the title and image of each post. We have another round of estimation. The result was:

- David: 13
- Karanjot: 13
- Gustova: 13
- Saima: 13
- Chung Yin: 13
- Adityasinh: 13
- Egor: 13
- Kristi: 13
- Arvin: 13

Our team agreed that it is a heavy task. Hence, the three stories are estimated as 13 points.

Filter and Search Stories

Next, we grouped stories related to filter and search posts together since the tasks are similar, which are to find specific data in the database and display them. The following stories were estimated together.

- As a Traveler, I want to filter posts by regions, so that I can find posts in

the region.

- As a Post Writer, I want to filter posts by regions, so that I can find posts in the region.
- As an Administrator, I want to filter posts by regions, so that I can find posts in the region.
- As a Traveler, I want to search posts with keywords, so that I can find posts easily.
- As an Administrator, I want to search posts with keywords, so that I can find posts easily.

We did the estimation using the online tool. The result was:

- David: 13
- Karanjot: 5
- Gustova: 8
- Saima: 8
- Chung Yin: 13
- Adityasinh: 13
- Egor: 8
- Kristi: 8
- Arvin: 8

Karanjot shared his thought that it is a simple task as we only need to add one line of code to limit the displayed data. Chung Yin explained that the task is heavy because we need to deal with the database and spend time to do

discussion and testing. We did the estimation again. The result was:

- David: 13
- Karanjot: 13
- Gustova: 13
- Saima: 13
- Chung Yin: 13
- Adityasinh: 13
- Egor: 13
- Kristi: 13
- Arvin: 13

As a result, the five stories were estimated as 13 points.

[View Profile, Post and Comments](#)

The stories related to display data from the database are grouped together.

- As a Traveler, I want to view my profile information including user type, username, first name, last name and email address, so that I can know which account it is.
- As a Traveler, I want to view the post details, so that I can get recommendation information.
- As a Post Writer, I want to view my profile information including user type, username, first name, last name and email address, so that I can know which account it is.

- As an Administrator, I want to view the details of the post, so that I can evaluate the content.
- As a Traveler, I want to view comments from other travelers, so that I can see the feedback to help me make travel decisions.

Our estimation was:

- David: 5
- Karanjot: 8
- Gustova: 5
- Saima: 8
- Chung Yin: 8
- Adityasinh: 8
- Egor: 8
- Kristi: 8
- Arvin: 5

Kristi explained that to only display data is an easy task but dealing with the database would take some time. Daivd, Gustova and Arvin agreed. As a result, we estimated these five stories as 8 points.

Add and Edit Stories

The stories related to adding data to the database or editing data in the database are considered together. They are listed below.

- As a Traveler, I want to add comments under a post, so that I can provide

feedback.

- As a Traveler, I want to edit my profile information including password, first name, last name and email address, so that I can update my information.
- As a Post Writer, I want to edit my profile information including password, first name, last name and email address, so that I can update my information.
- As a Traveler, I want to sign up with email, so that I can get an account easily.
- As a Post Writer, I want to sign up with email, so that I can get an account easily.
- As a Post Writer, I want to delete a post, so that I can remove the post from the web page.
- As an administrator I want to delete a recommendation post so I can manage what will be on the page.

The result of our estimation was:

- David: 8
- Karanjot: 8
- Gustova: 5
- Saima: 8
- Chung Yin: 8
- Adityasinh: 8
- Egor: 5

- Kristi: 3
- Arvin: 8

David shared his thought that although the tasks look like straight forward, we need to handle the database similar to the previous estimated tasks. Kristi agreed that 3 points is too low. Finally, we all agreed to estimate these stories as 8 points.

Add and Edit Post

We grouped the add post and edit post story together as the task are similar that we need to add the data of a post into the database. The two stories were estimated separately from the previous group because despite text data, we also need to handle images.

- As a Post Writer, I want to add a post, so that I can make recommendations.
- As a Post Writer, I want to edit a post, so that I can change what is written in the recommendation.

We agreed that this group of stories should have a higher story point than the previous group. Hence, we estimated them as 13 points.

Constraint

Finally, we estimated the constraint ‘The system should do authentication to only allow authorized users to add, edit, and delete posts’. The result was:

- David: 8
- Karanjot: 5
- Gustova: 5
- Saima: 8
- Chung Yin: 8
- Adityasinh: 8
- Egor: 8
- Kristi: 8
- Arvin: 8

Gustova expressed his ideas that the task seems easy because we had experience of adding authentication for the website, but he understood that we may need more time to work together. Karanjot had the same opinion. Easily, we had a consent that the constraint should be estimated as 8 points.

Must-have/ Should-have Stories

Two stories, ‘Traveler – Edit Profile Story’ and ‘Post Writer – Edit Profile Story’, are defined as should-have stories, and the remaining stories are defined as must-have stories. The stories with estimation are shown below.

Table 2: The must-have stories for Release – Initial version.

| Story | Estimate (Points) |
|-------|----------------------|
|-------|----------------------|

| | |
|---|----|
| • As a Traveler, I want to view post overviews including post title and post image, so that I can know what the posts are briefly about. | 13 |
| • As a Traveler, I want to filter posts by regions, so that I can find posts in the region. | 13 |
| • As a Traveler, I want to sign up with email, so that I can get an account easily. | 8 |
| • As a Traveler, I want to search posts with keywords, so that I can find posts easily. | 13 |
| • As a Traveler, I want to view my profile information including user type, username, first name, last name and email address, so that I can know which account it is. | 8 |
| • As a Traveler, I want to view the post details, so that I can get recommendation information. | 8 |
| • As a Traveler, I want to add comments under a post, so that I can provide feedback. | 8 |
| • As a Traveler, I want to view comments from other travelers, so that I can see the feedback to help me make travel decisions. | 8 |
| • As a Post Writer, I want to view post overviews including post title and post image, so that I can know the existing posts on the page. | 13 |
| • As a Post Writer, I want to filter posts by regions, so that I can find posts in the region. | 13 |
| • As a Post Writer, I want to sign up with email, so that I can get an account easily. | 8 |
| • As a Post Writer, I want to view my profile information including user type, username, first name, last name and email address, so that I can know which account it is. | 8 |
| • As a Post Writer, I want to add a post, so that I can make recommendations. | 13 |
| • As a Post Writer, I want to edit a post, so that I can change what is written in the recommendation. | 13 |
| • As a Post Writer, I want to delete a post, so that I can remove the post from the web page. | 8 |
| • As an Administrator, I want to view post overviews including a post title and a post image, so that I can know what the posts are briefly about. | 13 |
| • As an Administrator, I want to filter posts by regions, so that I can find posts in the region. | 13 |
| • As an Administrator, I want to search posts with keywords, so that I can find posts easily. | 13 |
| • As an Administrator, I want to view the details of the post, so that I can evaluate the content. | 8 |
| • As an administrator I want to delete a recommendation post | 8 |

| | |
|--|---|
| so I can manage what will be on the page. | |
| • The system should do authentication to only allow authorized users to add, edit, and delete posts. | 8 |

Table 3: The should-have stories for Release – Initial version.

| Story | Estimate (Points) |
|---|-------------------|
| • As a Traveler, I want to edit my profile information including password, first name, last name and email address, so that I can update my information. | 8 |
| • As a Post Writer, I want to edit my profile information including password, first name, last name and email address, so that I can update my information. | 8 |

User Story Cards

We have defined stories for the three users. A user story card is created for each story. The user story cards are shown below.

User Stories for *Traveler*:

Traveler – View Post Overviews Story

User Story Card

| | |
|---|--|
| (Front) <u>Story:</u> As a traveler, I want to view post overviews including post title and post image, so that I can know what the posts are briefly about. <u>Estimate:</u> 24 hrs. | (Back) <u>Test 1:</u> Test with loading homepage. <u>Expected outcome:</u> The system should display all posts, each one with a title and an image |
|---|--|

Figure 7: User story card 'Traveler - View Post Overviews Story'

Traveler – Filter Posts Story

User Story Card

| (Front) | (Back) |
|--|---|
| <p><u>Story:</u> As a traveler, I want to filter posts displayed by regions, so that I can find recommendations only for a specific regions.</p> <p><u>Estimate:</u> 24 hrs.</p> | <p><u>Test 1:</u> Test with selecting one region.</p> <p><u>Expected outcome:</u> The system should only display posts for the selected region.</p> <p><u>Test 2:</u> Test with selecting no region.</p> <p><u>Expected outcome:</u> The system should display all posts for all regions.</p> |

Figure 8: User story card 'Traveler - Filter Posts Story'

Traveler – Sign Up Story

User Story Card

| (Front) | (Back) |
|--|--|
| <p><u>Story:</u> As a traveler, I want to sign up with email, so that I can get an account easily</p> <p><u>Note:</u> Sign up form should appear</p> <p><u>Estimate:</u> 13 hrs.</p> | <p><u>Test 1:</u> Test with filling out sign up form with all validate information</p> <p><u>Expected outcome:</u> the system should accept the form and profile is successfully created</p> <p><u>Test 2:</u> Test by filling out the form with existing username</p> <p><u>Expected outcome:</u> the system should reject the form and display error message</p> |

Figure 9: User story card 'Traveler - Sign Up Story'

Traveler – Search Post Story

User Story Card

| | |
|--|---|
| (Front) | (Back) |
| <u>Story:</u> As a traveler, I want to search posts with keywords, so that I can find posts easily. | <u>Test 1:</u> Test with any keywords existing in posts. <u>Expected Outcome:</u> The system should display post containing the keyword. |
| <u>Estimate:</u> 24 hrs. | <u>Test 2:</u> Test with keywords not existing in any posts. <u>Expected Outcome:</u> The system displays no posts. |
| | <u>Test 3:</u> Test with no keyword. <u>Expected Outcome:</u> The system display all posts. |

Figure 10: User story card 'Traveler - Search Post Story'

Traveler – View Profile Story

User Story Card

| | |
|---|---|
| (Front) | (Back) |
| <u>Story:</u> As a traveler, I want to view my profile information including user type, username, first name, last name and email address, so that I can know which account it is. | <u>Test 1:</u> Test clicking view profile. <u>Expected outcome:</u> the system should display the profile information of the user. |
| <u>Estimate:</u> 13 hrs. | |

Figure 11: User story card 'Traveler - View Profile Story'

| Traveler – Edit Profile Story | |
|---|--|
| User Story Card | |
| <p>(Front)</p> <p><u>Story:</u></p> <p>As a traveler, I want to edit my profile information including password, first name, last name and email address, so that I can update my information.</p> <p><u>Note for UI (if any):</u></p> <p>The user interface should display a form with fields for password, first name, last name, and email address</p> <p><u>Estimate:</u></p> <p>13 hrs.</p> | <p>(Back)</p> <p><u>Test 1:</u></p> <p>Test with a new valid password.</p> <p><u>Expected Outcome:</u></p> <p>The system should accept the changes and store the password.</p> <p><u>Test 2:</u></p> <p>Test with a new interest.</p> <p><u>Expected Outcome:</u></p> <p>The system should make the changes and display the new interest.</p> <p><u>Test 3:</u></p> <p>Test with a new username.</p> <p><u>Expected Outcome:</u></p> <p>The system should accept the changes and change the username.</p> <p><u>Test 4:</u></p> <p>Test with a new description.</p> <p><u>Expected Outcome:</u></p> <p>The system should make the changes and display the new description.</p> |

Figure 12: User story card 'Traveler - Edit Profile Story'

| Traveler – View Post Detail Story | |
|---|---|
| User Story Card | |
| <p>(Front)</p> <p><u>Story:</u></p> <p>As a traveler, I want to view the post details, so that I can get recommendation information.</p> <p><u>Estimate:</u></p> <p>13 hrs.</p> | <p>(Back)</p> <p><u>Test 1:</u></p> <p>Test with accessing a post.</p> <p><u>Estimated Outcome:</u></p> <p>The system should display all details of the post.</p> |

Figure 13: User story card 'Traveler - View Post Detail Story'

Traveler – Add Comments Story

User Story Card

| (Front) | (Back) |
|--|---|
| <p>Story: As a traveler, I want to add comments under a post, so that I can provide feedback.</p> | <p>Test 1: Test with blank comment. Expected outcome: the system should prompt user to fill in the comment field.</p> |
| <p>Estimate: 13 hrs.</p> | <p>Test 2: Test with any non-blank comment. Expected outcome: the system should display the comment.</p> |
| | <p>Test 3: Test with adding a comment. Expected outcome: the system should add the comment to the database.</p> |

Figure 14: User story card 'Traveler - Add Comments Story'

Traveler – View Comments Story

User Story Card

| (Front) | (Back) |
|---|--|
| <p>Story: As a traveler, I want to view comments from other travelers, so that I can see the feedbacks to help me make travel decisions.</p> | <p>Test 1: Test with any post detail page containing comments. Expected outcome: the system should display all the comments of the post.</p> |
| <p>Note (if any):</p> | <p>Test 2: Test with any post detail page that does not have comment. Expected outcome: the system should display no comment.</p> |
| <p>Estimate: 13 hrs.</p> | |

Figure 15: User story card 'Traveler - View Comments Story'

User Stories for *Post Writer*:

Post Writer – View Post Overviews Story

User Story Card

| (Front) | (Back) |
|--|---|
| <p><u>Story:</u> As a post writer, I want to view post overviews including post title and post image, so that I can know what the posts are briefly about.</p> | <p><u>Test 1:</u> Test with loading homepage.</p> |
| <p><u>Estimate:</u> 24 hrs.</p> | <p><u>Expected outcome:</u> The system should display all posts, each one with a title and an image</p> |

Figure 16: User story card 'Post Writer - View Post Overview Story'

Post Writer – Filter Posts Story

User Story Card

| (Front) | (Back) |
|---|---|
| <p><u>Story:</u> As a post writer, I want to filter posts displayed by regions, so that I can find recommendations only for a specific regions.</p> | <p><u>Test 1:</u> Test with selecting one region.</p> |
| <p><u>Estimate:</u> 24 hrs.</p> | <p><u>Expected outcome:</u> The system should only show posts for the selected region</p> <p><u>Test 2:</u> Test with selecting no region.</p> <p><u>Expected outcome:</u> The system should display all posts for all regions.</p> |

Figure 17: User story card 'Post Writer - Filter Posts Story'

Post Writer – Sign Up Story

User Story Card

| | |
|--|---|
| (Front) | (Back) |
| <p><u>Story:</u> As a post writer, I want to sign up with email, so that I can get an account easily</p> | <p><u>Test 1:</u> Test with filling out sign up form with all validate information</p> |
| <p><u>Note for UI (if any):</u> Sign up form should appear</p> | <p><u>Expected outcome:</u> the system should accept the form and profile is successfully created</p> |
| <p><u>Estimate:</u> 13 hrs.</p> | <p><u>Test 2:</u> Test by filling out the form with existing username</p> <p><u>Expected outcome:</u> the system should reject the form and display error message</p> |

Figure 18: User story card 'Post Writer - Sign Up Story'

Post Writer – View Profile Story

User Story Card

| | |
|--|--|
| (Front) | (Back) |
| <p><u>Story:</u> As a Post Writer, I want to view my profile information including user type, username, first name, last name and email address, so that I can know which account it is.</p> | <p><u>Test 1:</u> Test clicking view profile.</p> |
| <p><u>Estimate:</u> 13 hrs.</p> | <p><u>Expected outcome:</u> the system should display the profile information of the user.</p> |

Figure 19: User story card 'Post Writer - View Profile Story'

Post Writer– Edit Profile Story

User Story Card

| | |
|---|---|
| (Front) <u>Story:</u> As a Post Writer, I want to edit my profile information including password, first name, last name and email address, so that I can update my information. | (Back) <u>Test 1:</u> Test with a new valid password. <u>Expected Outcome:</u> The system should accept the changes and store the password. <u>Test 2:</u> Test with a new interest. <u>Expected Outcome:</u> The system should make the changes and display the new interest. <u>Test 3:</u> Test with a new username. <u>Expected Outcome:</u> The system should accept the changes and change the username. <u>Test 4:</u> Test with a new description. <u>Expected Outcome:</u> The system should make the changes and display the new description. |
| <u>Note for UI (if any):</u> The user interface should display a form with fields for password, first name, last name, and email address | |

Figure 20: User story card 'Post Writer - Edit Profile Story'

Post Writer – Add Post Story

User Story Card

| | |
|---|---|
| (Front) <u>Story:</u> As a Post Writer, I want to add a post, so that I can make recommendations. | (Back) <u>Test 1:</u> Test with filling all information. <u>Expected outcome:</u> The system should display the new added post and its details. |
| <u>Estimate:</u> 24 hrs. | |

Figure 21: User story card 'Post Writer - Add Post Story'

Post Writer – Edit Post Story

User Story Card

| (Front) | (Back) |
|---|--|
| <p><u>Story:</u></p> <p>As a Post Writer, I want to edit a post, so that I can change what is written in the recommendation</p> | <p><u>Test 1:</u> Test with changing the post content. <u>Expected outcome:</u> The system should display the edited post with the updated changes.</p> |
| <p><u>Estimate:</u> 24 hrs.</p> | <p><u>Test 2:</u> Test with no changes <u>Expected outcome:</u> The system should display the post as it was.</p> |
| | <p><u>Test 3:</u> Test with changing the post title. <u>Expected outcome:</u> The system should display the edited post with the updated changes.</p> <p><u>Test 4:</u> Test with changing the image. <u>Expected outcome:</u> The system should display the edited post with the updated changes.</p> |

Figure 22: User story card 'Post Writer - Edit Post Story'

Post Writer – Delete Post Story

User Story Card

| (Front) | (Back) |
|--|---|
| <p><u>Story:</u></p> <p>As a Post Writer, I want to delete a post, so that I can remove the post from the web page</p> | <p><u>Test 1:</u> Test with deleting an existent post.</p> |
| <p><u>Estimate:</u> 13 hrs.</p> | <p><u>Expected outcome:</u> The system should make the post not visible on the page and display a message saying the post was successfully deleted.</p> |

Figure 23: User story card 'Post Writer - Delete Post Story'

User Stories for *Administrator*:

Administrator – View Post Overviews Story

User Story Card

| (Front) | (Back) |
|---|---|
| <p><u>Story:</u> As an administrator, I want to view post overviews including post title and post image, so that I can know what the posts are briefly about.</p> <p><u>Estimate:</u> 24 hrs.</p> | <p><u>Test 1:</u> Test with loading homepage.</p> <p><u>Expected outcome:</u> The system should display all posts, each one with a title and an image</p> |

Figure 24: User story card 'Administrator - View Post Overviews Story'

Administrator – Filter Posts Story

User Story Card

| (Front) | (Back) |
|--|---|
| <p><u>Story:</u> As an administrator, I want to filter posts displayed by regions, so that I can find recommendations only for a specific regions.</p> <p><u>Estimate:</u> 24 hrs.</p> | <p><u>Test 1:</u> Test with selecting one region.</p> <p><u>Expected outcome:</u> The system should only display posts for the selected region.</p> <p><u>Test 2:</u> Test with selecting no region.</p> <p><u>Expected outcome:</u> The system should display all posts for all regions.</p> |

Figure 25: User story card 'Administrator - Filter Posts Story'

Administrator – Search Post Story

User Story Card

| | |
|--|---|
| (Front) | (Back) |
| <u>Story:</u> As an Administrator, I want to search posts with keywords, so that I can find posts easily. | <u>Test 1:</u> Test with any keywords existing in posts. <u>Expected Outcome:</u> The system should display post containing the keyword. |
| <u>Estimate:</u> 24 hrs. | <u>Test 2:</u> Test with keywords not existing in any posts. <u>Expected Outcome:</u> The system display post search error message. |
| | <u>Test 3:</u> Test with no keyword. <u>Expected Outcome:</u> The system display all posts. |

Figure 26: User story card 'Administrator - Search Post Story'

Administrator – View Post Detail Story

User Story Card

| | |
|---|--|
| (Front) | (Back) |
| <u>Story:</u> As an Administrator, I want to view the details of the post, so that I can evaluate the content. | <u>Test 1:</u> Test with accessing a post. <u>Estimated Outcome:</u> The system should display all details of the post. |
| <u>Estimate:</u> 13 hrs. | |

Figure 27: User story card 'Administrator - View Post Detail Story'

Administrator – Delete Post Story

User Story Card

| | |
|--|---|
| (Front) | (Back) |
| <u>Story:</u> As an administrator, I want to delete a recommendation post so I can manage what will be on the page. | <u>Test 1:</u> Test with deleting an existent post. |
| <u>Note:</u> Administrator can delete all posts from all users. | <u>Expected outcome:</u> The system should make the post not visible on the page and display a message saying the post was successfully deleted. |
| <u>Estimate:</u> 13 hrs. | |

Figure 28: User story card 'Administrator – Delete Post Story'

Authentication Constraint

User Story Card

| | |
|---|---|
| (Front) | (Back) |
| <u>Story:</u> The system should do authentication to only allow authorized users to add, edit, and delete post. | Test with not logged in to add a posts. Expected outcome: the system should prompt the user to the sign in page. |
| <u>Note:</u> Administrator users can delete all posts from all users, but cannot add or edit posts. Post Writer users can add and edit posts but can only delete their own posts. Traveler users cannot add, edit, or delete any post. | Test with not logged in to edit a posts. Expected outcome: the system should prompt the user to the sign in page. |
| <u>Estimate:</u> 13 hrs. | Test with not logged in to delete a post. Expected outcome: the system should prompt the user to the sign in page. |
| | Test with logged in as a normal user to add posts. Expected outcome: the system should allow the user to access the add post page. |

Figure 29: User story card 'Authentication Constraint'

Authentication Constraint (Continued)

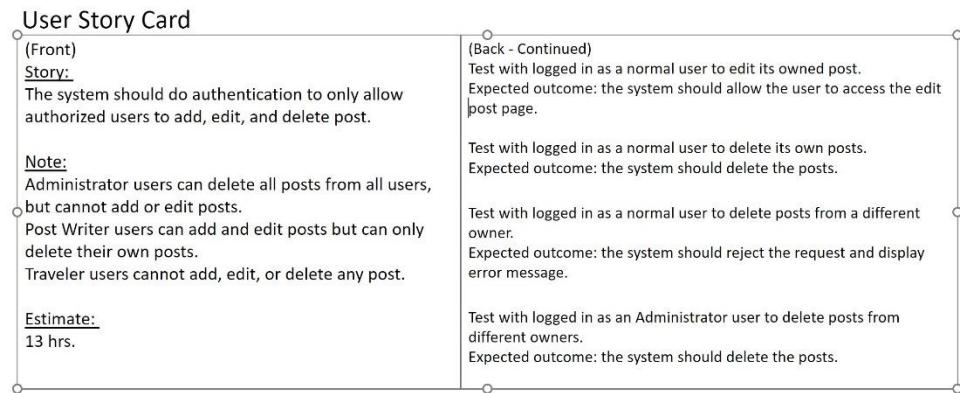


Figure 30: User story card 'Authentication Constraint' (Continued)

2.4.2 Additional Documentation

The video from our workshop is included here.

URL to the video on YouTube: <https://youtu.be/rzNj-Ti3XnA>

2.4.3 Release Plan 1.0

Roadmap

The main areas of focus for Iteration 1 as follows:

- View Post titles and images for user to view. (Core feature 1)
- Post filtering by region. (Core feature 2)
- Search post using keywords. (Core feature 3)
- View post details. (Core feature 4)
- Add, edit, and delete post. (Core feature 5)

The main areas of focus for Iteration 2 as follows:

- View and add comments from a post.
- Prevent not authorized user to access add, edit, and delete functions.
- View and edit profile information of users.

Software Release:

- Bug fixing.

Iteration Length and Release Date

Because the new features of the website need to be done in one month, our team decided to split that month into 2 parts for each iteration. The group should

finish iteration 1 in 2 weeks period and the rest 2 weeks will be delegated to iteration 2. Since the release date is April 16, the team considered 2 weeks for each iteration as the best time. This will allow the group to finish both iterations and review them before the deadline. The first iteration will focus on the highest-priority features. By focusing on the high-priority features in the first iteration, the team can ensure that the most critical parts of the project are completed on time and meet the expectations of stakeholders. Additionally, dividing the work into two iterations can also help the team to better manage their workload, identify potential issues early on, and adjust plans accordingly. When the first iteration is finished, the team will be able to analyze the velocity and adjust the work for the second iteration if needed. To conclude, the release date and iteration information are listed below.

Release Date: April 16, 2023

Iteration 1: March 13, 2023 – March 26, 2023 (2 weeks)

Iteration 2: March 30, 2023 – April 12, 2023 (2 weeks)

Combining Stories

For doing the release plan, we have combined some stories with high similarity.

- We've combined 'View Post Overview' user stories for Traveler, Post Writer and Administrator into one story.
- We've combined 'Filter Post by Region' user stories for Traveler, Post

Writer and Administrator into one story.

- We've combined 'Sign Up by Email' user stories for Traveler and Post Writer into one story.
- We've combined 'View My Profile Information' user stories for Traveler and Post Writer into one story.
- We've combined 'Edit Profile Information' user stories for Traveler and Post Writer into one story.
- We've combined 'Search Posts with Keyword' user stories for Traveler and Administrator into one story.
- We've combined 'View the Post Details' user stories for Traveler and Administrator into one story.
- We've combined 'Delete Post' user stories for Post Writer and Administrator into one story.

Table 4: Combined User Stories For Release Plan

| Stories for Release Plan | Estimate Story Points |
|--|-----------------------|
| A user can view post overviews including post titles and post images. | 13 |
| A user can filter posts by region. | 13 |
| A user can sign up by email. | 8 |
| A user can search posts with keywords. | 13 |
| A user can view profile information. | 8 |
| A user can edit profile information. | 8 |
| A user can view the post details. | 8 |
| A user can add comments under a post. | 8 |
| A user can view comments from other travelers. | 8 |
| A user can add a post. | 13 |
| A user can edit a post. | 13 |
| A user can delete a post. | 8 |
| The system should do authentication to only allow authorized users to add, edit, and delete posts. | 8 |

User Stories Prioritization

We have refined the priority of the user stories. By prioritizing the must-have, should-have and could-have stories, the team can ensure that the most critical features of the website are completed within the given timeline and budget.

Must-have stories:

The choice of must-have user stories that comprise the main functionalities of the system was based on several factors. The team identified the core objectives of the system and the needs of its users. This helped to determine the essential features and capabilities that the system must provide. The team also prioritized

the user stories based on their importance and complexity. This ensured that the highest-priority stories were included as must-have user stories. And lastly, the team considered the resources available, including time, budget, and personnel, to determine the feasibility of each user story. This helped to ensure that the must-have user stories could be implemented within the given constraints. Overall, the choice of must-have user stories was based on a thorough analysis of the system's objectives, user needs, priorities, and available resources. By focusing on these critical features and functionalities, the team can ensure that the system meets the needs of its users and delivers value to them.

Table 5: Must-have Stories for Release Plan.

| Must-have Stories | Estimate (Points) |
|---|-------------------|
| A user can view post overviews including post titles and post images. | 13 |
| A user can filter posts by region. | 13 |
| A user can search posts with keywords. | 13 |
| A user can add a post. | 13 |
| A user can edit a post. | 13 |

Should-have stories:

The should-have user stories that have been chosen are essential to enhance the system usability because they are critical features that can improve the user experience and engagement. For example, allowing users to add comments under a post and view comments from other travelers can encourage user interaction and collaboration. Similarly, allowing users to view post details and delete their own posts can improve their control and satisfaction with the

platform. In addition to that, providing an option for an administrator user to delete any post can ensure the integrity and quality of the content on the platform. These should-have user stories can make the platform more interactive, user-friendly, and secure, in the long run leading to higher user satisfaction and retention.

Table 6: Should-have Stories for Release Plan.

| Should-have Stories | Estimate (Points) |
|--|-------------------|
| A user can add comments under a post. | 8 |
| A user can view comments from other travelers. | 8 |
| A user can view the post details. | 8 |
| A user can delete a post. | 8 |

Could-have stories:

For the could-have stories, our team decided to add features that could enhance the user experience and these features include the ability for users to view and edit their profile information, such as their password, first name, last name, and email address. The team has also decided to postpone the implementation of two user stories: "A user can sign up by email" and "The system should do authentication to only allow authorized users to add, edit, and delete posts." While these features may not be critical for users to use the website, they can provide a personalized experience and make it easier for users to manage their account information. Furthermore, having these features can enhance the overall satisfaction of users, which can lead to increased engagement and loyalty to the website.

Table 7: Could-have Stories for Release Plan.

| Could-have Stories | Estimate (Points) |
|--|-------------------|
| A user can view profile information. | 8 |
| A user can edit profile information. | 8 |
| A user can sign up by email. | 8 |
| The system should do authentication to only allow authorized users to add, edit, and delete posts. | 8 |

Actual Release Plan

The release plan is decided as shown in below table. Where the team will work on Iteration 01 for 2 weeks, giving their best efforts to complete it before time, after completion team will evaluate the performance and resolve possible issues and then proceed towards Iteration 02 which is estimated to be completed in another 2 weeks.

Table 8: The Finished Release Plan.

| Iteration 1 | Iteration 2 |
|---|--|
| A user can view post overviews including post titles and post images. | A user can add comments under a post. |
| A user can filter posts by region. | A user can view comments from other travelers. |
| A user can search posts with keywords. | A user can view profile information. |
| A user can add a post. | A user can edit profile information. |
| A user can edit a post. | A user can sign up by email. |
| A user can view the post details. | The system should do authentication to only allow authorized users to add, edit, and delete posts. |

| | |
|---------------------------|--|
| A user can delete a post. | |
|---------------------------|--|

2.4.4 Iteration Plan (Release 1.0)

2.4.4.1 Iteration 1 Planning

During the first iteration of the project, the team plans to complete 7 user stories which have been broken down into 30 tasks, with a total estimated predicted velocity of 81 points (as per stated story points for each User Story). Iteration 1 will be completed within 2 weeks' period. The team consists of 8 developers and the task distributed among them is almost even.

The tasks have been estimated with some extra time in mind to account for unforeseen issues that may arise during the development process. Some tasks have been assigned extra points to allow for additional research or assistance with more complex issues. The team will also focus on ensuring that code quality and testing are prioritized throughout the development process.

During the planning meeting, the team members reviewed the user stories and discussed the tasks required to complete them. Each developer volunteered for the tasks that they felt best suited their skills and expertise. They also considered their workload from the iteration and considered any potential conflicts or dependencies. The team members worked collaboratively to ensure that each task had a clear owner, and that everyone understood the tasks they were responsible for completing. The developers showed great initiative in taking ownership of their tasks and were committed to ensuring that the project was completed on time and to a high standard.

Leveraging each member's specialization and expertise, the task ownership is consistent for every member. Arvin's tasks are related to database and controllers which are his experienced fields. Chung Yin and Kristi have taken tasks related to add and view post due to past experience of constructing similar functions. Egor is good at UI design, so he has taken several UI tasks. Saima, Adityasinh and Karanjot have taken tasks related to SQL query which is the area they are good at. Every member has confident to complete their tasks. From the assigned tasks we can distinguish responsibilities of each of the team members:

- David: Agile Customer
- Kristi: Full-stack developer

Responsible for developing the controller, which involves writing code that can manage user inputs and manipulate data. He will also be responsible for ensuring that the application follows best practices and adheres to design patterns.

- Egor: Frontend developer

Responsible for designing and implementing the user interface for various features of the application. He will also be responsible for ensuring that the UI is intuitive and easy to use, and that it meets design specifications.

- Chung Yin: Backend developer

Responsible for writing controllers and business logic to process and

manipulate data. She will also be responsible for ensuring API endpoint security and handling error messages.

- Karanjot: Full-stack developer

Responsible for designing and implementing the user interface for the application. She will also be responsible for ensuring that the UI is responsive and accessible, and that the application is properly connected to the backend.

- Adityasinh: Backend developer

Responsible for the model development, designing the database schema, and ensuring that the database is properly modeled to support the application's functionality. He will also be responsible for writing efficient code to interact with the database.

- Arvin: Backend developer

Responsible for working with the database and ensuring efficient and optimized data retrieval. He will also be responsible for debugging and troubleshooting issues with the database.

- Saima: Full-stack developer
- Gustavo: Front-end developer

After further discussion, we have done some modifications to the content of the tasks to make them more specific. We have also changed some of the assignments due to no willing of participation from some members. The

modifications are shown below.

- The tasks related to writing the SQL query were removed, but tasks related to coding the router were added. It is because after the discussion, we decided that the structure of our web application will include models, controllers, routers, and views. It is not necessary to have tasks for writing SQL query, while we need a task to construct the router.
- Some tasks related to design and code UI were separated to two tasks, design task and code task to make the task smaller.
- As we were not able to contact some members of our team after the start of the iteration, some of their tasks were distributed to other members if the task is not done nor in progress for over 1 week. Members who completed their originally assigned tasks earlier and had confident to complete the newly assigned tasks took those tasks.

Velocity

The iteration velocity is calculated by the number of the tasks. In Iteration 1, we have completed 7 user stories which contain 33 tasks, all accounting for 81 story points. Therefore, the velocity of Iteration 1 is 81 points.

Following are the tables that describe the breakdown of each user story in tasks, one table per user story.

- A. A user can view post overviews including post titles and post images.

Table 9: Tasks for 'A user can view post overviews including post titles and post images'.

| Task | Who | Estimate | Actual (hour) |
|--|------------|----------|------------------|
| Design the UI for the recommendations view. | Saima | 2 | 3 |
| Implement the table for posts in the DB. | Arvin | 2 | 1 |
| Design and implement DB table for images. | Arvin | 2 | 1 |
| Code the router for view post overviews page. | Chung Yin | 2 | 1 |
| Code the model for post object. | Adityasinh | 2 | 2 |
| Code the controller for post visualization. | Chung Yin | 4 | 1 |
| Connect the application to the DB. | Gustavo | 1 | 1 |
| Build Initial File Structure with Required Dependencies. | Chung Yin | 2 | 1 |
| Code the UI for the recommendations view. | Chung Yin | 2 | 1 |

- B. A user can filter posts by region.

Table 10: Tasks for 'A user can filter posts by region'.

| Task | Who | Estimate (hour) | Actual (hour) |
|--|------|--------------------|------------------|
| Code the UI for the search screen (button, or dropdown). | Egor | 1 | 1 |

| | | | |
|--|-----------|---|---|
| | Chung Yin | | |
| Code the ejs file for showing filtered results. | Chung Yin | 1 | 2 |
| Code the router for filtering the posts by region. | Saima | 1 | 1 |
| Code the controller for the filtered search. | Chung Yin | 2 | 3 |

C. A user can search post with keywords.

Table 11: Tasks for 'A user can search post with keywords'.

| Task | Who | Estimate (hour) | Actual (hour) |
|--|-----------|--------------------|------------------|
| Design and code the UI for the keyword search. | Arvin | 1 | 1 |
| Code the ejs for showing the result. | Chung Yin | 1 | 2 |
| Code the router for getting the post filtered by keywords. | Saima | 1 | 1 |
| Code the controller for filtered by keyword search. | Chung Yin | 2 | 4 |

D. A user can add a post.

Table 12: Tasks for 'A user can add a post'.

| Task | Who | Estimate (hour) | Actual (hour) |
|--|------------|--------------------|------------------|
| Design and code the UI that holds the input screen. | Kristi | 1 | 1 |
| Design and code the UI that holds the feedback screen. | Kristi | 1 | 1 |
| Code the router for adding a post. | Adityasinh | 2 | 1 |

| | | | |
|-----------------------------------|-----------|---|---|
| Code the controller for addition. | Chung Yin | 2 | 1 |
|-----------------------------------|-----------|---|---|

E. A user can edit a post.

Table 13: Tasks for 'A user can edit a post'.

| Task | Who | Estimate (hour) | Actual (hour) |
|--|--------|--------------------|------------------|
| Design and code the UI that will have the input screen. | Egor | 1 | 1 |
| Design and code the UI that will have the feedback screen. | Egor | 1 | 0.5 |
| Code the router for editing a post. | Saima | 1 | 1 |
| Code the controller for the edit post. | Kristi | 2 | 1 |

F. A user can view the post details.

Table 14: Tasks for 'A user can view the post details'.

| Task | Who | Estimate (hour) | Actual (hour) |
|--|-----------|--------------------|------------------|
| Design and code the UI that shows the relevant details for the post. | Saima | 2 | 2 |
| Code the router for viewing post details. | Egor | 1 | 1 |
| Code the controller for post details. | Chung Yin | 2 | 1 |
| Code the UI that shows the relevant details for the post. | Chung Yin | 2 | 1 |

G. A user can delete a post.

Table 15: Tasks for 'A user can delete a post'.

| Task | Who | Estimate (hour) | Actual (hour) |
|--|------------|--------------------|------------------|
| Design and code the UI for deletion. | Egor | 1 | 1 |
| Design and code the UI for feedback on deletion. | Egor | 1 | 0.5 |
| Code the router for deleting a post. | Adityasinh | 1 | 1 |
| Code the controller for post deletion. | Kristi | 2 | 2 |

2.4.4.2 Iteration 2 Planning

During the second iteration of the project, the team plans to complete 6 user stories which have been broken down into 16 tasks, with a total estimated predicted velocity of 16 points (1 task for 1 point). Iteration 2 will be completed within 2 weeks' time period. The team consists of 8 developers. However, only 6 developers participated in the task distribution this time. The tasks distributed among them are almost even (around 4 – 6 estimated hours per person).

Having the experience from Iteration 1, we realized that it is better for each member to take tasks that are closely related. The tasks have been distributed according to each developer's experience and ability. Saima, Chung Yin and Arvin took tasks related to add comments, authentication and view profile since they have the experience of constructing those functions. Kristi has confident at building router and controllers, and therefore he took two controller tasks and one UI task that is closely related. Adityasinh volunteered for the user model task as he has the experience of building a model in Iteration 1, and he also took the UI and backend tasks related to user data. Egor took post model task for comments and UI task for edit profile because of his experience of building another project containing similar functions.

Some developers of our team have not participated in the iteration 2 planning, but they showed willingness to take up some tasks after that. Hence, we have changed some of the task assignments.

Velocity

The iteration velocity is calculated by the number of the tasks. In Iteration 1, we have completed 7 user stories which contain 33 tasks: all accounting for 81 story points. So, the velocity of Iteration 1 is 81 points. For this second Iteration we have a total of 48 story points to develop. Considering that we cannot exceed the actual velocity for the first iteration, we expect to be able to execute at a velocity of 48 points, covering the totality of the remaining User Stories.

Following are the tables that describe the breakdown of each user story in tasks, one table per user story.

H. A user can add comments under a post.

Table 16: Tasks for ‘A user can add comments under a post’.

| Task | Who | Estimate (hour) | Actual (hour) |
|--|----------|--------------------|------------------|
| Design and code the UI for adding comments under a post. | Saima | 1.5 | 3 |
| Update or create the post model to accommodate comments. | Karanjot | 1 | 4 |
| Design and code controller and route to add comments to | Saima | 1 | 3 |

| | | | |
|---------|--|--|--|
| a post. | | | |
|---------|--|--|--|

I. A user can view comments from other travelers.

Table 17: Tasks for ‘A user can view comments from other travelers’.

| Task | Who | Estimate (hour) | Actual (hour) |
|--|--------|--------------------|------------------|
| Design and code the UI for viewing comments under a post. | Kristi | 2 | 2 |
| Update or create the post model to query comments list. | Egor | 2 | 2 |
| Design and code controller and route to get and show the comments to a post. | Kristi | 1 | 1 |

J. A user can sign up by email.

Table 18: Tasks for ‘A user can sign up by email’.

| Task | Who | Estimate (hour) | Actual (hour) |
|---|------------|--------------------|------------------|
| Design and code the UI for signing up and get feedback about success/failure. | Adityasinh | 2 | 2 |
| Create the model to work and save users/authorization level (access profile). | Adityasinh | 2 | 1 |
| Design and code controller and route for sign-up. | Kristi | 1 | 1 |

K. The system should do authentication to only allow authorized users to add, edit, and delete posts.

Table 19: Tasks for 'The system should do authentication to only allow authorized users to add, edit, and delete posts'.

| Task | Who | Estimate (hour) | Actual (hour) |
|--|-----------|--------------------|------------------|
| Design and code the UI to login to the system, including feedback about success/failure. | Chung Yin | 1 | 1 |
| Update post controller to enforce authentication requirement for addition, edition and deletion. | Chung Yin | 1 | 2 |
| Design and code controller and router to handle GET and POST routines for authentication. | Chung Yin | 2 | 2 |

L. A user can view profile information.

Table 20: Tasks for 'A user can view profile information'.

| Task | Who | Estimate (hour) | Actual (hour) |
|--|-------|--------------------|------------------|
| Design and code the UI to view user profile. | Arvin | 2 | 2 |
| Design and code controller and router to handle the display of user profile. | Arvin | 2 | 3 |

M. A user can edit profile information.

Table 21: Tasks for ‘A user can edit profile information’.

| Task | Who | Estimate (hour) | Actual (hour) |
|---|------------|----------------------------------|--------------------------------|
| Design and code the UI to edit the user profile. | Egor | 2 | 2 |
| Design and code controller and router to handle the changes for the user profile. | Gustavo | 2 | 3 |

2.4.5 Additional Documentation

The video report of Iteration 1 Planning is included here.

The URL of the video on YouTube: <https://youtu.be/1Pyue96Kn7k>

2.4.7 Acceptance Tests for Release 1.0

Table 22: Stories, acceptance tests, and contributors for Release 1.0 (Green=Passed; Red=Failed).

| Full description of user story | Acceptance test(s) | Name(s) of contributing Developer(s) |
|--|--|---|
| A. A user can view post overviews including post titles and post images. | <p>Test with loading homepage</p> <p>Expected outcome: The system should display all posts, each one with a title and an image</p> | Arvin Almario, Chung Yin Tsang, Adityasinh Makwana Gustavo Almeida |
| B. A user can filter posts by region. | <p>Test with selecting one region</p> <p>Expected outcome: The system should only show posts for the selected region.</p> <p>Test with selecting no region</p> <p>Expected outcome: The system should display all posts for all regions.</p> | Chung Yin Tsang, Saima Rafik Patel |

| | | |
|---|--|---|
| C. A user can search posts with keywords. | <p>Test with any keywords existing in posts. Expected outcome: Test with any keywords existing in posts.</p> <p>Test with keywords not existing in any posts. Expected outcome: The system displays post search error message.</p> <p>Test with no keyword. Expected outcome: The system displays all posts.</p> | Arvin Almario, Chung Yin Tsang, Saima Rafik Patel |
| D. A user can add a post. | <p>Test with filling all information. Expected outcome: The system should display the new added post and its details.</p> | Kristi Goxhaj, Adiyasinh Makwana, Chung Yin Tsang |
| E. A user can edit a post. | <p>Test with changing the post content.</p> | Egor Shabalkin, Saima Rafik Patel, |

| | | |
|--------------------------------------|---|--|
| | <p>Expected outcome: The system should display the edited post with the updated changes</p> <p>Test with no changes</p> <p>Expected outcome: The system should display the post as it was.</p> <p>Test with changing the post title</p> <p>Expected outcome: The system should display the edited post with the updated changes</p> <p>Test with changing the image.</p> <p>Expected outcome: The system should display the edited post with the updated changes.</p> | Kristi Goxhaj |
| F. A user can view the post details. | <p>Test with accessing a post.</p> <p>Expected outcome: The system should display all details of selected post.</p> | Saima Rafik Patel, Egor Shabalkin, Chung Yin Tsang |

| | | |
|--|---|---|
| | | |
| G. A user can delete a post. | <p>Test with deleting an existent post.</p> <p>Expected outcome: The system should make the post not visible on the page and display a message saying the post was successfully deleted.</p> | Egor Shabalkin, Adityasinh Makwana, Kristi Goxhaj |
| H. A user can add comments under a post. | <p>Test with blank comment.</p> <p>Expected outcome: the system should prompt user to fill in the comment field.</p> <p>Test with any non-blank comment.</p> <p>Expected outcome: the system should display the comment.</p> <p>Test with adding a comment.</p> <p>Expected outcome: the system should add the comment to the</p> | Saima Rafik Patel, Karanjot Singh |

| | | |
|---|--|--------------------------------------|
| | database. | |
| I. A user can view comments from other travelers. | <p>Test with any post detail page containing comments. Expected outcome: the system should display all the comments of the post.</p> <p>Test with any post detail page that does not have comment. Expected outcome: the system should display no comment.</p> | Kristi Goxhaj, Egor Shabalkin |
| J. A user can sign up by email. | <p>Test with filling out sign up form with all validate information Expected outcome: the system should accept the form and profile is successfully created</p> <p>Test by filling out the form with existing username Expected outcome: the system should reject the form and</p> | Adityasinh Makwana, Kristi Goxhaj |

| | | |
|---|---|-----------------|
| | display error message | |
| K. The system should do authentication to only allow authorized users to add, edit, and delete posts. | <p>Test with not logged in to add a posts.</p> <p>Expected outcome: the system should prompt the user to the sign in page.</p> <p>Test with not logged in to edit a posts.</p> <p>Expected outcome: the system should prompt the user to the sign in page.</p> <p>Test with not logged in to delete a post.</p> <p>Expected outcome: the system should prompt the user to the sign in page.</p> <p>Test with logged in as a normal user to add posts.</p> | Chung Yin Tsang |

| | | |
|--|---|--|
| | <p>Expected outcome: the system should allow the user to access the add post page.</p> <p>Test with logged in as a normal user to edit its owned post.</p> <p>Expected outcome: the system should allow the user to access the edit post page.</p> <p>Test with logged in as a normal user to delete its own posts.</p> <p>Expected outcome: the system should delete the posts.</p> <p>Test with logged in as a normal user to delete posts from a different owner.</p> <p>Expected outcome: the system should reject the request and display error message.</p> | |
|--|---|--|

| | | |
|---|--|---|
| | <p>Test with logged in as an Administrator user to delete posts from different owners.</p> <p>Expected outcome: the system should delete the posts.</p> | |
| L. A user can view profile information. | <p>Test clicking view profile.</p> <p>Expected outcome: the system should display the profile information of the user.</p> | Arvin Almario |
| M. A user can edit profile information. | <p>Test with a new valid password.</p> <p>Expected Outcome: The system should accept the changes and store the password.</p> <p>Test with a new interest.</p> <p>Expected Outcome: The system should make the changes and display the new interest.</p> | Egor Shabalkin, Gustavo Gomes M L R F Almeida |

| | | |
|--|--|--|
| | <p>Test with a new username.</p> <p>Expected Outcome: The system should accept the changes and change the username.</p> | |
| | <p>Test with a new description.</p> <p>Expected Outcome: The system should make the changes and display the new description.</p> | |

URL to video demo:

https://centennialcollegeedu.sharepoint.com/:v/s/COMP231W23_T8/EZIytHKbMn9LnWx_Y2ilveUBCIO34mVkywWRaMXhtc1O8A?e=PtTv0x

3.0 CONCLUSIONS

After analyzing, it can be concluded that the explore booking website has been successful in providing a user-friendly platform that allows travelers to filter, search, add, edit, view, and delete posts. The website's success can be attributed to several factors, such as its instinctive user interface that makes it easy for travelers to find and access the information they need. The ability for users to add, edit, and view comments on travel-related content creates a community-like atmosphere on the website. Users can share their experiences, ask for advice, and provide feedback to other users, which enriches the content and provides a more comprehensive perspective.

Furthermore, the website ability to allow users to personalize their experience by editing and viewing their profile information is a notable feature. This feature authorizes users to customize their preferences and interests, creating a more customized and personalized experience. It also allows users to keep their information up to date, ensuring that the website has the most current and accurate user data.

The ability for users to filter posts by region and search for posts with specific keywords is particularly useful, as it allows travelers to quickly and easily find relevant posts that match their interests and preferences.

The success of the website also relies on the transparent and accountable nature of the platform, as users can review each other's posts. This system encourages quality content and ensures that users are held accountable for the information

they provide.

However, to ensure continued success, it is important for the website to regularly monitor and update its features and functionalities to address any potential issues and keep up with changing user needs and expectations. While the project had only two releases, the user stories and their acceptance criteria were implemented and successfully tested, as the results show.

4.0 RECOMMENDATIONS

Due to the inherent nature and constraints of this project, it constitutes just the basis to expand from and build an enterprise grade application, if the project is to be continued, additional development and testing processes are needed.

As stated on section 1.0.3 of the Test Plan, there are some risks derived from time and expertise constraints, so it would be advisable to design and conduct a thorough process to test for security, third party modules, and performance issues before considering launching the system in a production environment. It might as well be necessary to conduct a projected demand study and adjust the architecture to respond to the resulting design requirements.

CREDITS, LICENSE, AND REFERENCES

Credits

Author of the report template layout: Jake Nesovic

<jnesovic@my.centennialcollege.ca>

License

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front-Cover Texts, and with no Back-Cover Texts.

References

[1] Cohn, Mike. 2004. *User Stories Applied: For Agile Software Development*, Addison-Wesley Professional.

[2] Rough Guides. (n.d.). *Rough Guides | Travel Guide and Travel Information | Rough Guides*. <https://www.roughguides.com/>

[3] Tourism in Albania. (n.d.). *Tourism in Albania*. <https://tourisminalbania.com/>

[4] Anita Hendrieka. (n.d.). *Anita Hendrieka*. <https://www.anitahendrieka.com/>

[5] Software Testing Help. (n.d.). *BDD (Behavior Driven Development) Framework: A Complete Tutorial*. <https://www.softwaretestinghelp.com/bdd-framework/>

APPENDIX A: DESIGN DOCUMENTS

Paper Prototype

We have designed a paper prototype of the web application. The following paper prototype shows our ideas for design of the web application. As it is an initial version, the UI design could be changed at a later stage.

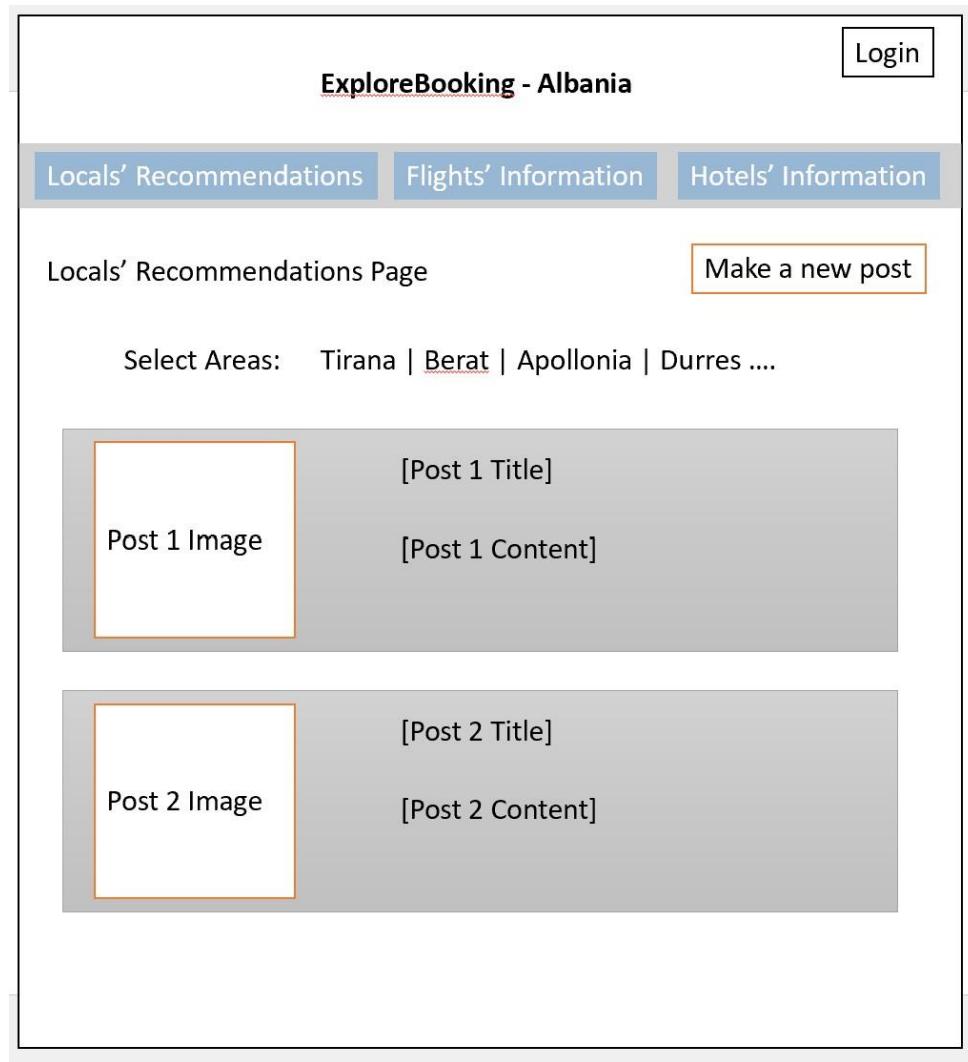


Figure 31: Paper Prototype 'A user can view post overviews and filter posts'

This would be our home page and from this page, we could view posts, recommendations, information about flights and hotels, and log in. It includes several user stories, such as:

- A user can view post overviews including post titles and post images.
- A user can filter the posts by region.

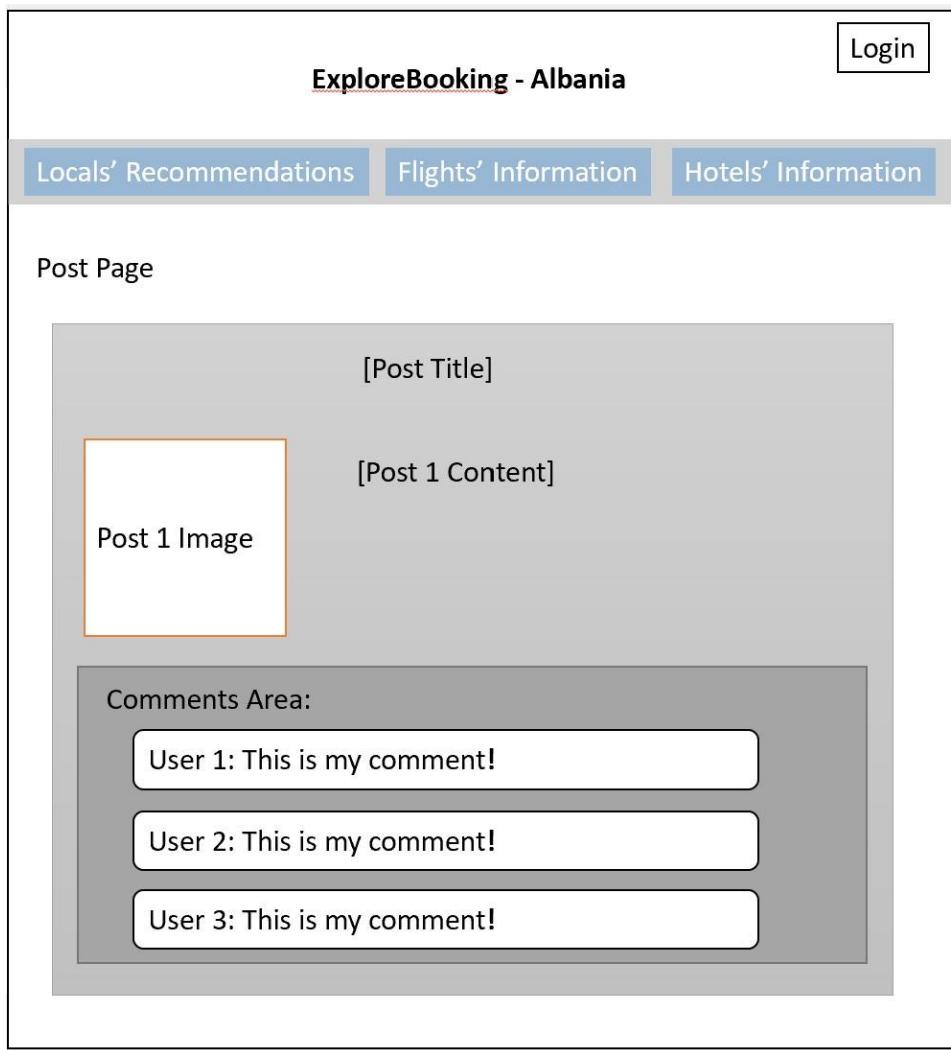


Figure 32: Paper Prototype 'A user can view and add comments'.

This would be the post page and from this page, the user could view posts in detail, recommendations, information about flights and hotels, log in, and user's comments about the post. It includes several user stories, such as:

- A user can add comments under a post.
- A user can view comments from other travelers.

The paper prototype is a rectangular card with a light gray background. At the top center, it says "ExploreBooking - Albania". In the top right corner, there is a small rectangular box labeled "Login". Below the title, there is a horizontal navigation bar with three tabs: "Locals' Recommendations", "Flights' Information", and "Hotels' Information". The "Flights' Information" tab is highlighted with a darker shade. Underneath the navigation bar, the text "Login Page" is centered. There are two input fields: one for "Username" and one for "Password", both enclosed in rectangular boxes. Below these fields is a central "Login" button. To the right of the "Login" button, the text "Click here to Sign Up" is written. At the bottom left, a note in parentheses reads "(Users need to login before making a post and any comments)".

Figure 33: Paper Prototype 'The system should do authentication'.

This would be our login page and from this page, the user could log in or sign up, and also view recommendations, and information about flights and hotels. It includes several user stories, such as:

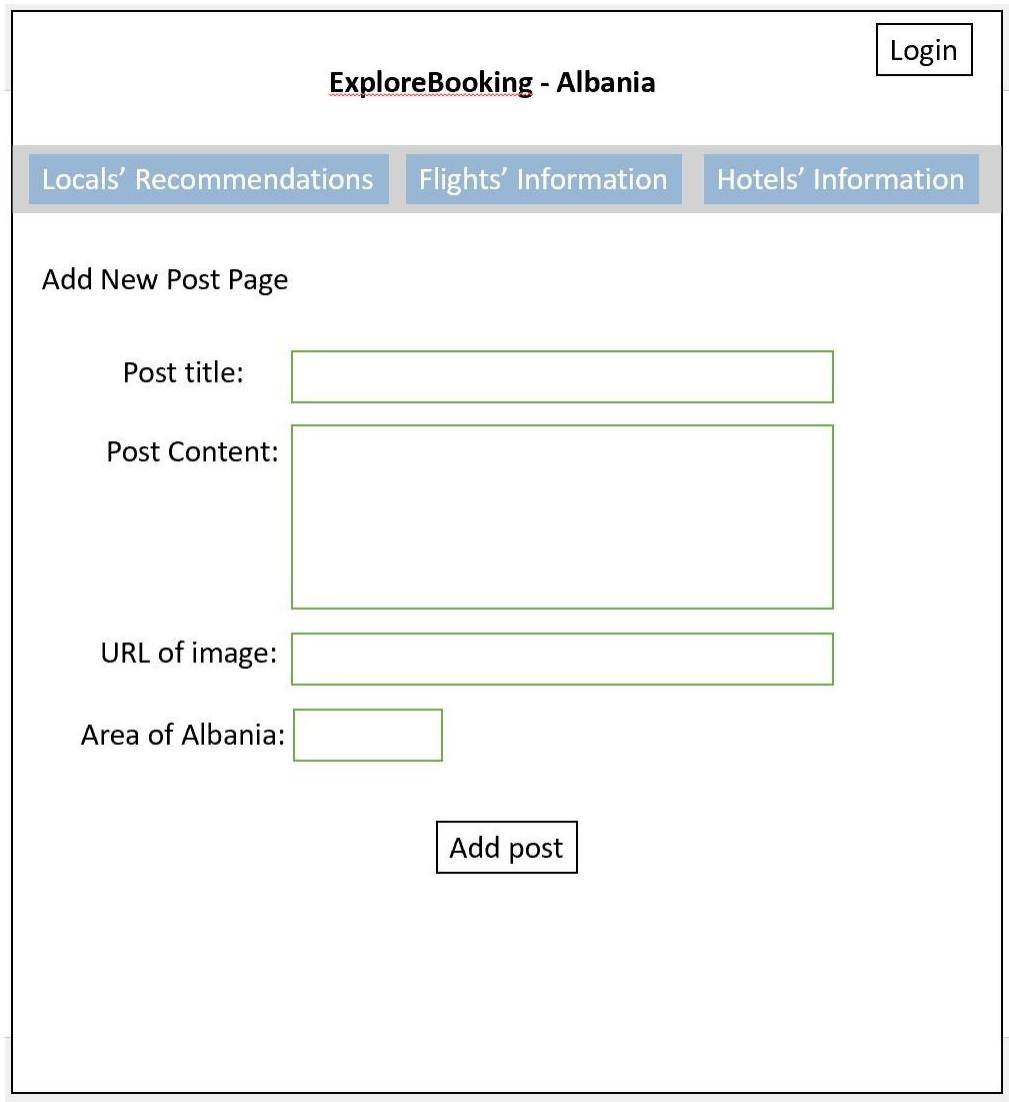
- The system should do authentication to only allow authorized users to add, edit, and delete posts.

The paper prototype is a rectangular card with a black border. At the top center, it says "ExploreBooking - Albania". In the top right corner, there is a small rectangular box labeled "Login". Below the title, there is a horizontal bar with three buttons: "Locals' Recommendations", "Flights' Information", and "Hotels' Information". The main content area is titled "Sign Up Page". It contains four text input fields with green borders, each preceded by a label: "Username:", "Password:", "First Name:", and "Last Name:". Below these fields is a "Sign Up" button with a black border. At the bottom left of the main area, there is a link "Click here to Login".

Figure 34: Paper Prototype 'A user can sign up'.

This would be our sign up page and from this page, the user could sign up or log in and view recommendations, and information about flights and hotels. It includes a user story:

- A user can sign up by providing an email, password, first name, and last name.



The paper prototype is a rectangular card divided into sections. At the top right is a 'Login' button. Below it is the 'ExploreBooking - Albania' logo. A horizontal navigation bar at the top has three tabs: 'Locals' Recommendations', 'Flights' Information', and 'Hotels' Information'. The main content area is titled 'Add New Post Page'. It contains four input fields with green borders: 'Post title:', 'Post Content:', 'URL of image:', and 'Area of Albania:'. Below these fields is a central 'Add post' button.

Figure 35: Paper Prototype 'A user can add a post'.

This would add a new post page and from this page, the user could add a new post by providing the post title, post content, URL of an image, area of Albania, view recommendations, and information about flights and hotels. It includes the user story:

- A user can add a post.

UML Class Diagram

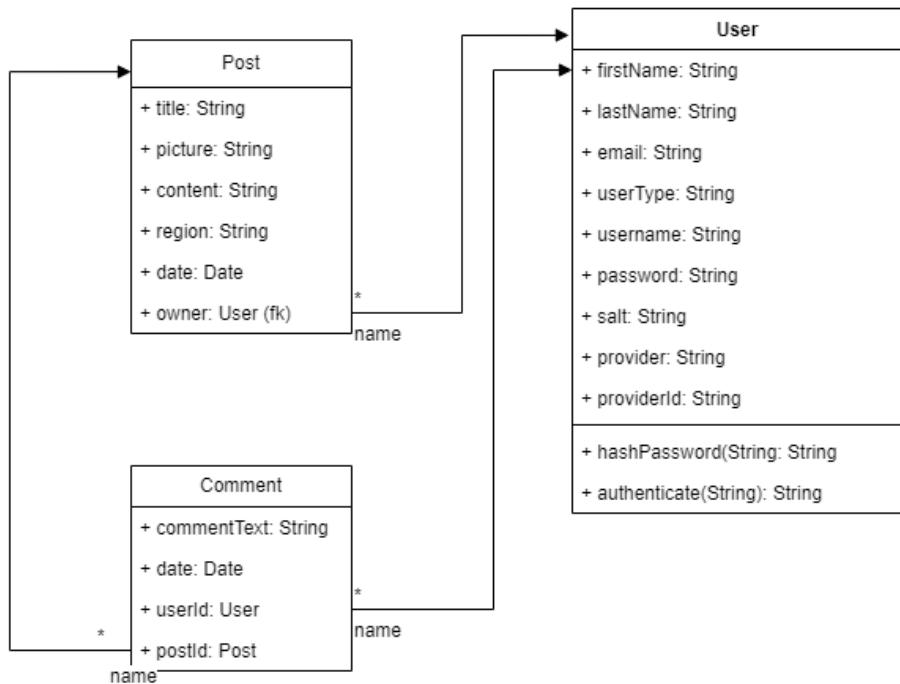


Figure 36: UML Class Diagram.

ERD Diagram

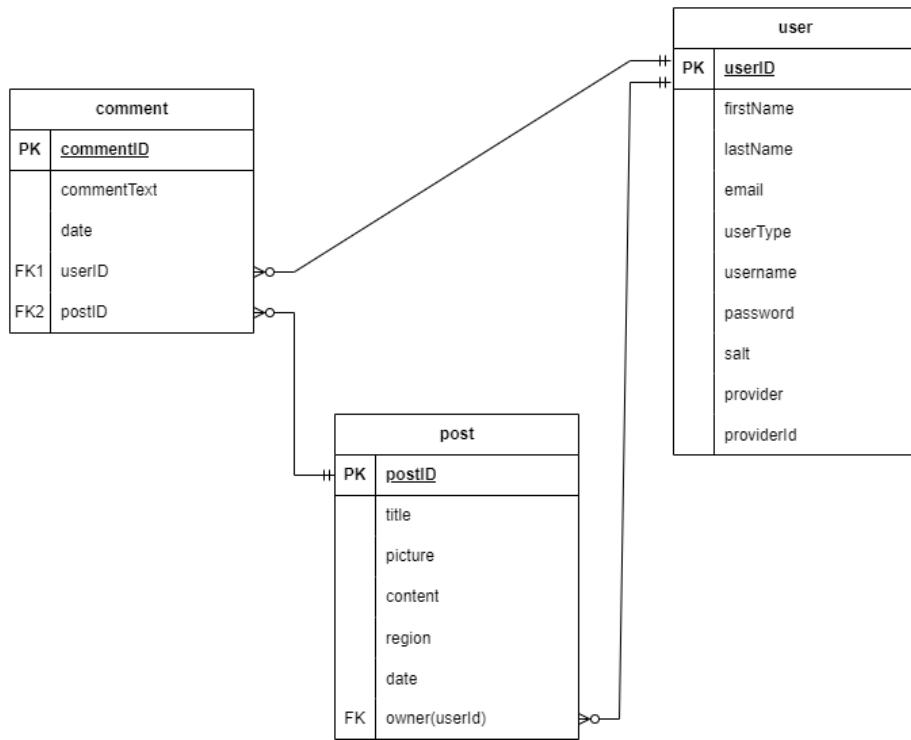


Figure 37: ERD Diagram.

CRD cards

| Post | | Comment | |
|-------------------|----------------|-------------------|----------------|
| Responsibilities: | Collaborators: | Responsibilities: | Collaborators: |
| View post | | View comment | |
| Add post | | Add comment | |
| Edit post | | | |
| Delete post | | | |

| User | |
|-------------------|----------------|
| Responsibilities: | Collaborators: |
| Login | |
| Sign up | |
| Logout | |
| | Post |

Figure 38: CRC Cards.

Use Case Diagram

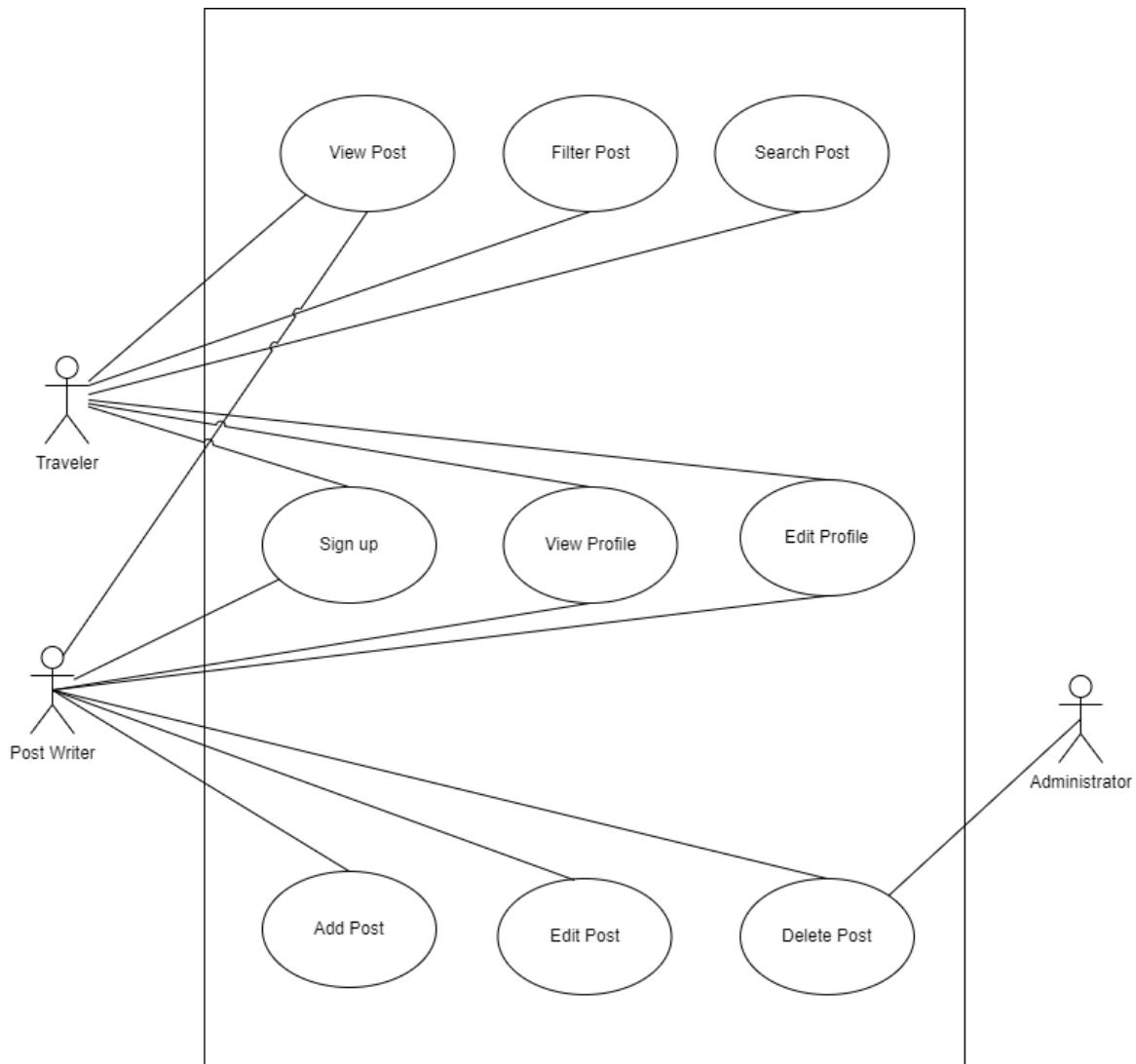


Figure 39: Use Case Diagram.

Activity Diagram

Login:

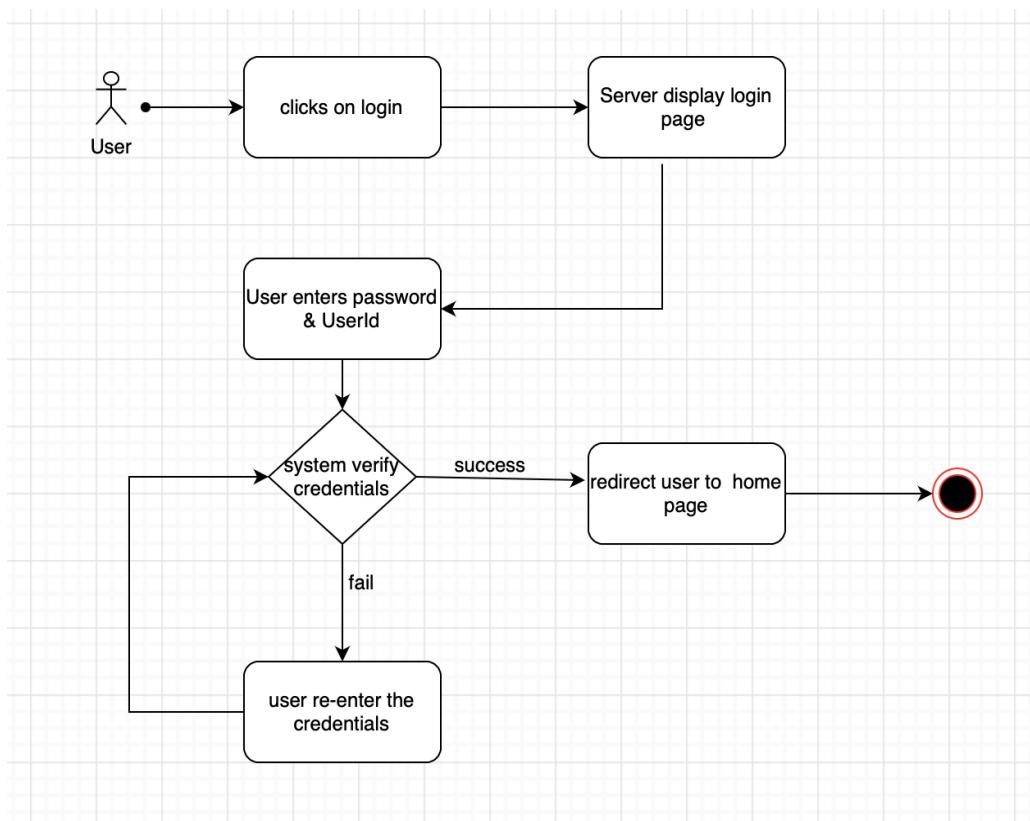


Figure 40: Login Activity Diagram.

Add Post:

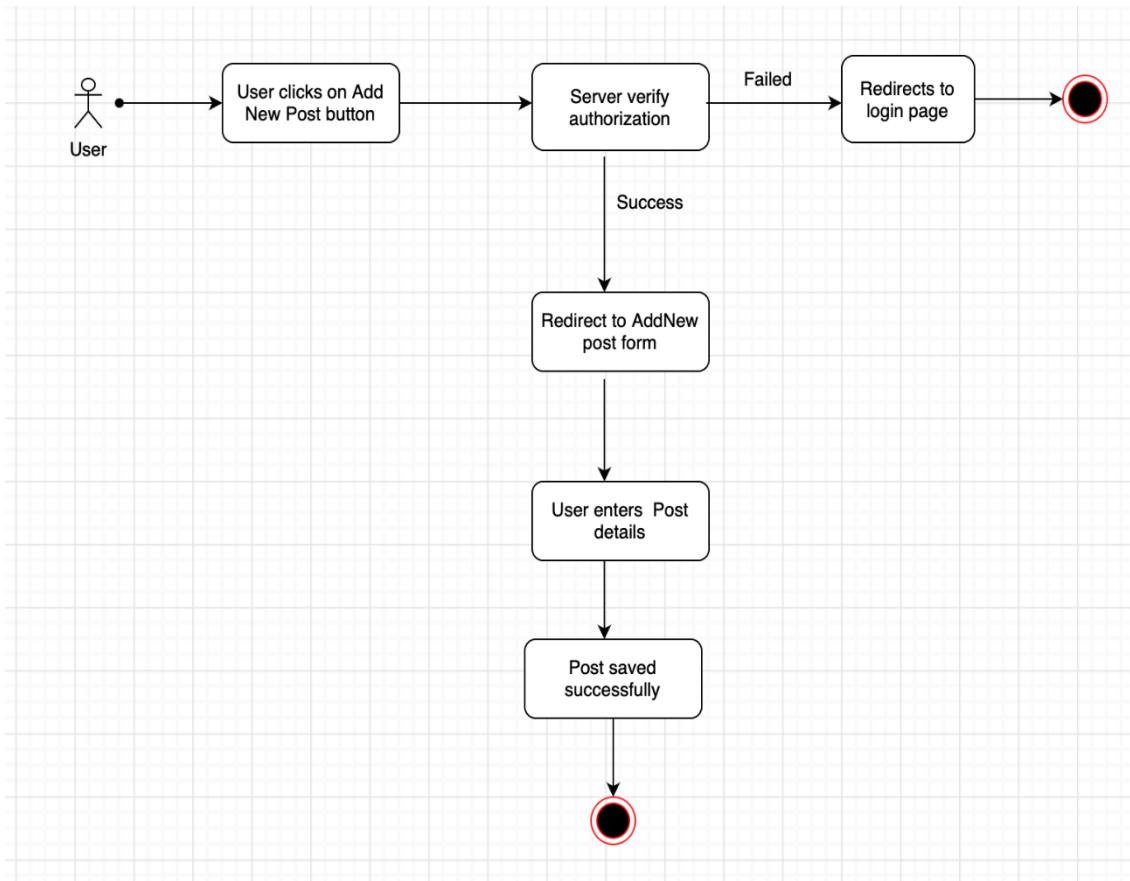


Figure 41: Add Post Activity Diagram.

Sequence Diagrams

Login:

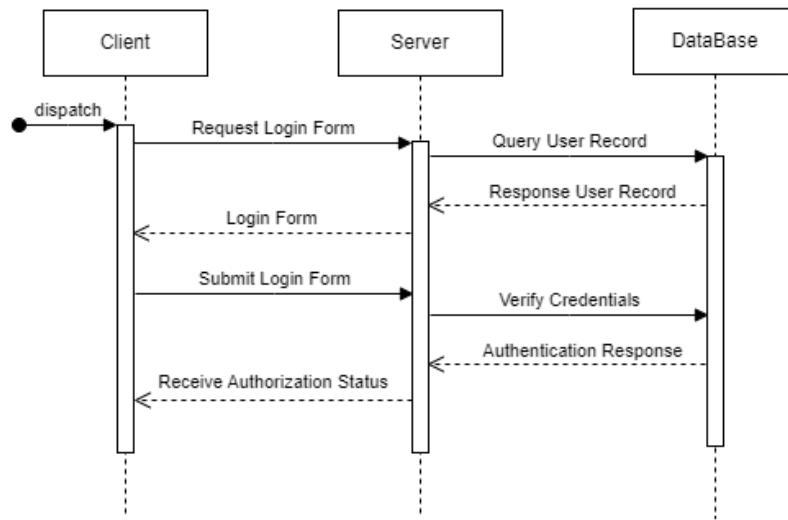


Figure 42: Login Sequence Diagram.

Add Post:

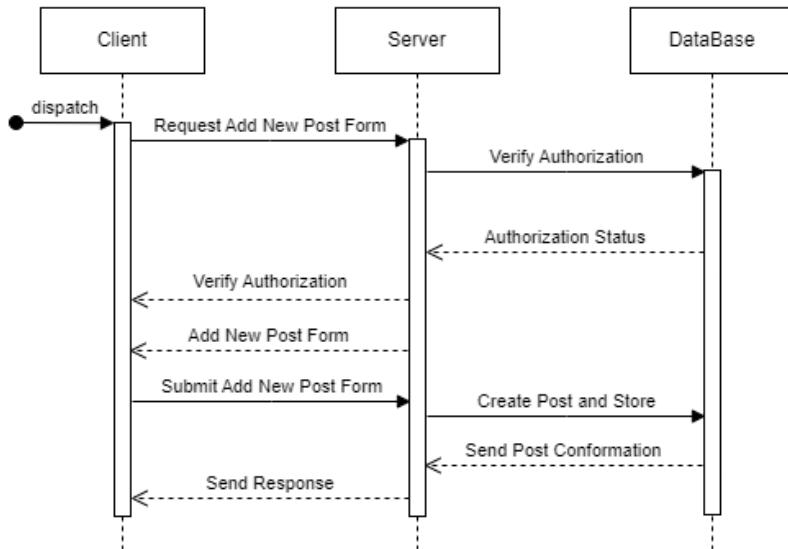


Figure 43: Add Post Sequence Diagram.

APPENDIX B: TEST PLAN

1.0 Introduction

1.0.1 Goals

This test plan is aiming to ensure that the application functions correctly by demonstrating the user stories related to its core features as well as the non-core features selected for implementation in the scope of this release. We will consider the tests passed when the objective of each user story is met within the expected results as per defined on each user story card (see section 2.4.1). As a complementary goal, the application needs to perform and operate in a user-friendly fashion, so there will be basic usability tests to avoid confusing user interfaces.

1.0.2 Assumptions

1. The application is developed and in position to be tested. It should run without errors.
2. All merging issues have been addressed and corrected.
3. The tests are conducted with full access to the source code and each responsible developer is available to clarify or explain how the application was coded.
4. The scope for the test plan includes every user story within Release 1.0.
5. Testing will be conducted using the code of a testing GitHub branch that

- will hold a true copy of the latest commission of main branch.
6. Testing will be conducted in a separate environment that will have the same dependencies and packages that are necessary for deployment.
 7. Testing will be conducted by the Costumer/SCRUM Master in close collaboration with the development team identify and address any issue that would impede the application from being deployed.
 8. Due to constraints in time and resources, security and performance test are not being conducted for Release 1.0.
 9. Unit testing will be conducted by development team before marking each task as ready to merge to the main branch.

1.0.3 Risks and Assets

1. Limited Testing Expertise. Derived from the student nature of the project, the testing team and plan may not cover all potential issues, particularly on security, performance, and integration of third parties' components (software dependencies), resulting in critical issues been missed or inadequately tested.
2. Limited Platform Availability for Testing. Given the circumstances and the fact that performance testing is out of the scope, large scale deployment is not being tested and this Release 1.0 could not be adequate to withstand numerous concurrent sessions or large numbers of users.
3. Limited Test Coverage. The current testing plan may not cover all

important test scenarios, resulting in several bugs going unidentified.

4. Unforeseen Changes in Third Parties' Components. Since this application is built upon a series of software components or platforms that are developed and maintained by entities outside of this development team, it is susceptible to changes in said components that may break its functionality or result in performance or security issues.

2.0 Scope

2.0.1 Features to Be Tested

For this Release 1.0, the test plan will include all the 13 User Stories selected for it. The stories are listed below, grouped in core and non-core features:

Table 23: Features within Testing Scope.

| Features |
|--|
| Core feature 1 - View Post titles and images for user to view. <ul style="list-style-type: none">• User story [A]: A user can view post overviews including post titles and post images. |
| Core feature 2 - Post filtering by region. <ul style="list-style-type: none">• User story [B]: A user can filter posts by region. |
| Core feature 3 - Search post using keywords. |

- | |
|--|
| <ul style="list-style-type: none"> • User story [C]: A user can search posts with keywords. |
|--|

| Core feature 4 - View post details. |

- | |
| --- |
| - User story [F]: A user can view the post details. |

| Core feature 5 - Add, edit, and delete post. |

- | |
| --- |
| - User story [D]: A user can add a post. - User story [E]: A user can edit a post. - User story [G]: A user can delete a post. |

| Non-core feature 1 - View and add comments from a post. |

- | |
| --- |
| - User story [H]: A user can add comments under a post. - User story [I]: A user can view comments from other travelers. |

| Non-core feature 2 - Prevent unauthorized user to access add, edit, and delete functions. |

- | |
| --- |
| - User story [J]: A user can sign up by email. - User story [K]: The system should do authentication to only allow authorized users to add, edit, and delete posts. |

| Non-core feature 3 - View and edit profile information of users. |

- | |
| --- |
| - User story [L]: A user can view my profile information including username, first name, description and interests. - User story [M]: A user can edit profile information |

including password, username, description and interests.

2.0.2 Features Not to Be Tested

- a. All features related to application performance are outside the scope of the testing plan because of time and expertise constraints.
- b. Features related to accessibility for users with some forms of disability are left out due to time constraints.
- c. Features pertaining to security domain are outside the scope due to limited expertise of the development and testing time, but also because of time constraints.

3.0 Testing Procedures

Given the skillset present in the development and testing teams, we are going to use a general Behaviour Driven Development (BDD) framework to have a solid structure for the test even though no automated testing tool is involved. The BDD test scenarios are expressed in Gherkin Syntax:

User Story: A user can view post overviews including post titles and post images.

Scenario: User views post overview

Given the user is on the home page

When the user views the post overviews

Then the post titles and images should be displayed

User Story: A user can filter posts by region.

Scenario: User filters posts by region

Given the user is on the home page

When the user selects a region filter

Then the posts displayed should be filtered by the selected region

User Story: A user can sign up by email.

Scenario: User signs up by email

Given the user is on the sign-up page

When the user enters a valid email and password

And clicks the sign-up button

Then the user should be registered and redirected to the home page

User Story: A user can search posts with keywords.

Scenario: User searches posts with keywords

Given the user is on the home page

When the user enters a search term in the search bar

And clicks the search button

Then the posts displayed should be filtered by the search term

User Story: A user can view profile information.

Scenario: User views profile information

Given the user is logged in

When the user clicks on their profile

Then their profile information should be displayed

User Story: A user can edit profile information.

Scenario: User edits profile information

Given the user is logged in

When the user clicks on their profile

And clicks on the edit button

And makes changes to their profile information

And clicks the save button

Then the changes should be saved and displayed on the profile page

User Story: A user can view the post details.

Scenario: User views post details

Given the user is on the home page

When the user clicks on a post title or image

Then the post details should be displayed

User Story: A user can add comments under a post.

Scenario: User adds comment under a post

Given the user is logged in

When the user views a post

And enters a comment in the comment section

And clicks the add comment button

Then the comment should be added and displayed under the post

User Story: A user can view comments from other travelers.

Scenario: User views comments from other travelers

Given the user is on a post page

When the user scrolls down to the comments section

Then the comments from other travelers should be displayed

User Story: A user can add a post.

Scenario: User adds a post

Given the user is logged in

When the user clicks on the add post button
And enters post details
And clicks the submit button
Then the post should be added and displayed on the home page

User Story: A user can edit a post.

Scenario: User edits a post
Given the user is logged in
When the user clicks on the edit post button for a post
And makes changes to the post details
And clicks the save button
Then the changes should be saved and displayed on the post page

User Story: A user can delete a post.

Scenario: User deletes a post
Given the user is logged in
When the user clicks on the delete post button for a post
And confirms the deletion
Then the post should be deleted and removed from the home page

User Story: The system should do authentication to only allow authorized users

to add, edit, and delete posts.

Scenario: User logs in successfully

Given the user is on the login page

When they enter valid login credentials

Then they should be redirected to the dashboard

Scenario: User fails to log in with invalid credentials

Given the user is on the login page

When they enter invalid login credentials

Then they should see an error message

Scenario: User logs out successfully

Given the user is logged in

When they click on the logout button

Then they should be redirected to the login page

Scenario: User tries to add a post without being authenticated

Given the user is not logged in

When they try to add a post

Then they should be redirected to the login page

Scenario: User tries to edit a post without being authenticated

Given the user is not logged in

When they try to edit a post

Then they should be redirected to the login page

Scenario: User tries to delete a post without being authenticated

Given the user is not logged in

When they try to delete a post

Then they should be redirected to the login page

Scenario: User adds a post successfully

Given the user is logged in

When they add a new post

Then the post should be added to the list of posts

Scenario: User edits a post successfully

Given the user is logged in

When they edit an existing post

Then the changes should be reflected on the post details page

Scenario: User deletes a post successfully

Given the user is logged in

When they delete an existing post

Then the post should be removed from the list of posts

Those scenarios will be considered and harmonized with the previously defined in User Story Cards, which will be the leading criteria to select and conduct testing.

3.0.1 Test Objectives

- a. Ensure that the application meets the functional requirements derived from selected User Stories.
- b. Ensure that the application provides an adequate User Experience, and it is sufficiently easy to use.
- c. Ensure that the application meets the client's expectations as they were expressed in workshops throughout the process.

3.0.2 Types of Testing

3.0.2.1 Unit Testing

In general, Unit Testing will be one of the main criteria for the development team to be able to request pull procedures to integrate their code to the main branch of the repository, so there is distributed validation embedded on the development process.

For unit test execution after the development stage ended, the following methods are selected:

- a. Show user profile.

- b. Render login.
- c. Process login.
- d. Display post details.
- e. Render post overview.

Since this stage will take place after all components are developed and integrated, we choose to follow a similar approach as End-to-End testing.

3.0.2.2 Integration Testing

Due to time constraints, Integration Testing will be merged with the Acceptance Testing procedures. As the goal for this type of testing is to ensure that all modules and subsystems, including third parties' software, interacts correctly from user interface to data storage.

3.0.2.3 Acceptance Testing

For Acceptance Testing, we will execute End-to-End Testing using BDD test scenarios outlined in section 3.0 Testing Procedures, which will be conducted with User Story Cards as a guideline of which of them will be recorded to document those tests. The following table includes all scenarios in scope:

Table 24: Testing scenarios in scope.

| Full description of | Acceptance test(s) |
|---------------------|--------------------|
| user story | |

| | |
|--|--|
| A. A user can view post overviews including post titles and post images. | <p>Test with loading homepage</p> <p>Expected outcome: The system should display all posts, each one with a title and an image</p> |
| B. A user can filter posts by region. | <p>Test with selecting one region</p> <p>Expected outcome: The system should only show posts for the selected region.</p> <p>Test with selecting no region</p> <p>Expected outcome: The system should display all posts for all regions.</p> |
| C. A user can search posts with keywords. | <p>Test with any keywords existing in posts.</p> <p>Expected outcome: Test with any keywords existing in posts.</p> <p>Test with keywords not existing in any posts.</p> <p>Expected outcome: The system display post search error message.</p> <p>Test with no keyword.</p> <p>Expected outcome: The system displays all posts.</p> |

| | |
|--------------------------------------|---|
| | |
| D. A user can add a post. | <p>Test with filling all information.</p> <p>Expected outcome: The system should display the new added post and its details.</p> |
| E. A user can edit a post. | <p>Test with changing the post content.</p> <p>Expected outcome: The system should display the edited post with the updated changes</p> <p>Test with no changes</p> <p>Expected outcome: The system should display the post as it was.</p> <p>Test with changing the post title</p> <p>Expected outcome: The system should display the edited post with the updated changes</p> <p>Test with changing the image.</p> <p>Expected outcome: The system should display the edited post with the updated changes.</p> |
| F. A user can view the post details. | <p>Test with accessing a post.</p> <p>Expected outcome: The system should display all</p> |

| | |
|---|---|
| | details of selected post. |
| G. A user can delete a post. | <p>Test with deleting an existent post.</p> <p>Expected outcome: The system should make the post not visible on the page and display a message saying the post was successfully deleted.</p> |
| H. A user can add comments under a post. | <p>Test with blank comment.</p> <p>Expected outcome: the system should prompt user to fill in the comment field.</p> <p>Test with any non-blank comment.</p> <p>Expected outcome: the system should display the comment.</p> <p>Test with adding a comment.</p> <p>Expected outcome: the system should add the comment to the database.</p> |
| I. A user can view comments from other travelers. | <p>Test with any post detail page containing comments.</p> <p>Expected outcome: the system should display all the comments of the post.</p> |

| | |
|---|--|
| | <p>Test with any post detail page that does not have comment.</p> <p>Expected outcome: the system should display no comment.</p> |
| J. A user can sign up by email. | <p>Test with filling out sign up form with all validate information</p> <p>Expected outcome: the system should accept the form and profile is successfully created</p> <p>Test by filling out the form with existing username</p> <p>Expected outcome: the system should reject the form and display error message</p> |
| K. The system should do authentication to only allow authorized users to add, edit, and delete posts. | <p>Test with not logged in to add a posts.</p> <p>Expected outcome: the system should prompt the user to the sign in page.</p> <p>Test with not logged in to edit a posts.</p> <p>Expected outcome: the system should prompt the user to the sign in page.</p> |

| | |
|--|---|
| | <p>Test with not logged in to delete a post.</p> <p>Expected outcome: the system should prompt the user to the sign in page.</p> |
| | <p>Test with logged in as a normal user to add posts.</p> <p>Expected outcome: the system should allow the user to access the add post page.</p> |
| | <p>Test with logged in as a normal user to edit its owned post.</p> <p>Expected outcome: the system should allow the user to access the edit post page.</p> |
| | <p>Test with logged in as a normal user to delete its own posts.</p> <p>Expected outcome: the system should delete the posts.</p> |
| | <p>Test with logged in as a normal user to delete posts from a different owner.</p> <p>Expected outcome: the system should reject the</p> |

| | |
|---|--|
| | <p>request and display error message.</p> <p>Test with logged in as an Administrator user to delete posts from different owners.</p> <p>Expected outcome: the system should delete the posts.</p> |
| L. A user can view profile information. | <p>Test clicking view profile.</p> <p>Expected outcome: the system should display the profile information of the user.</p> |
| M. A user can edit profile information. | <p>Test with a new valid password.</p> <p>Expected Outcome: The system should accept the changes and store the password.</p> <p>Test with a new interest.</p> <p>Expected Outcome: The system should make the changes and display the new interest.</p> <p>Test with a new username.</p> <p>Expected Outcome: The system should accept the</p> |

| | |
|--|--|
| | <p>changes and change the username.</p> <p>Test with a new description.</p> <p>Expected Outcome: The system should make the changes and display the new description.</p> |
|--|--|

3.0.2.4 Stress Testing

Due to time and expertise constraints, Stress Testing will be out of the scope for this Test Plan.

3.0.2.5 Performance Testing

Due to time and expertise constraints, Performance Testing will be out of the scope for this Test Plan.

3.0.3 Testing Tools

The Test Plan does not include automated testing due to lack of experienced testers who can design, configure and execute them. Instead, we are going to be using Behaviour Driven Development as the base framework to structure and organize tests. BDD methodology involves developers, testers and stakeholders, so it is deemed adequate given the team composition and roles associated. The key elements of BDD are listed below:

1. User Stories are transformed or decomposed in scenarios written in plain language using a structured format called Gherkin syntax, which format follows a Given-Then-When structure. This allows every team member to understand and learn about the logic behind each test.
2. Collaboration between developers, testers, and stakeholders is promoted, ensuring that there is a shared understanding of the requirements and acceptance criteria, giving developers a clear goal and a clear idea on how their code is going to be evaluated.
3. Business value and outcomes are emphasized to ensure the application delivers what the client requested, as opposed as just being technically correct only.
4. Behaviour and interaction with the user are the main focal point, so user perspective is always present at development and testing stages.

4.0 Schedule and Deliverables

Scheduled task are programmed according to the next table.

Table 25: Schedule for Testing activities.

| Task | Execution date |
|----------------------------------|----------------|
| Acceptance Tests for Iteration 1 | March 26, 2023 |
| Acceptance Tests for Iteration 2 | April 12, 2023 |

| | |
|------------|----------------|
| Unit Tests | April 12, 2023 |
|------------|----------------|

Deliverables for Test Plan are described in following table.

Table 26: Deliverables.

| Deliverable | Description |
|--------------------------------------|--|
| Test Log (included in section 2.4.7) | According to requirements from course material, the table will include the User Story to be tested, description of the scenarios being tested, the team members responsible for development and a color-coded result marker (green if passed, red if failed) |
| Demonstration | In line with course requirements, a video demonstration will be produced. All test scenarios and Unit Tests are being re-executed for documentation and showcase purposes. The core and non-core features, including all User Stories that are comprised in them, are going to be show in functionality, with table from section 2.4.7 as guide. |

APPENDIX C: END-USER & ADMINISTRATOR MANUALS

Instructions on Installation and Configuration

1. Download Visual Studio Code at <https://code.visualstudio.com/download>

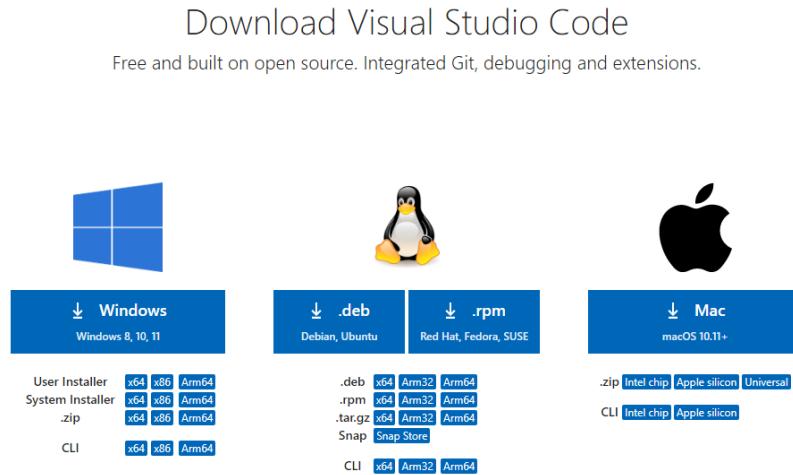


Figure 44: Download Visual Studio Code.

2. Follow the instructions on the screen to install Visual Studio Code on your computer.
3. Download the ExploreBooking app as ZIP at
<https://github.com/COMP231W23-T8/explorebooking>

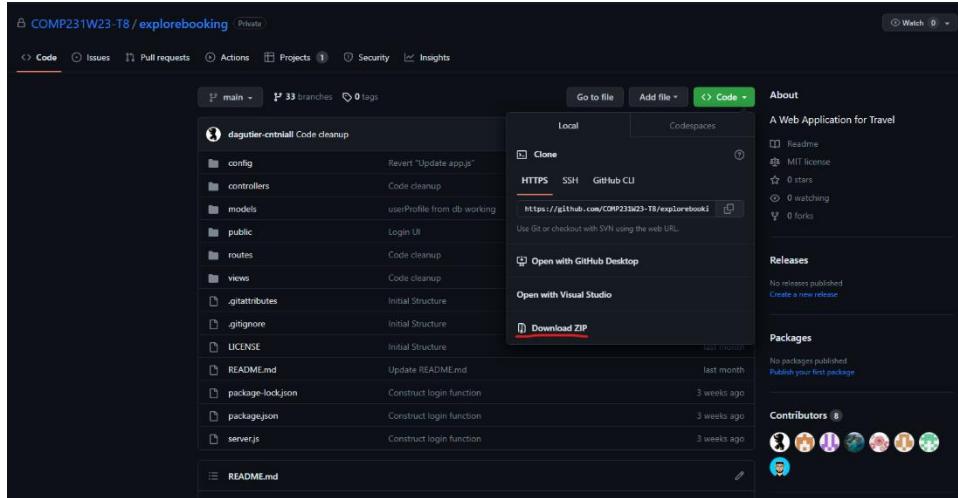


Figure 45: Download the ExploringBooking app ZIP file.

4. Extract the ZIP file.
5. Open Visual Studio Code.
6. Click on "Open Folder" in the welcome screen or "File" > "Open Folder" in the top menu to open the folder containing the EJS program that you downloaded and extracted.

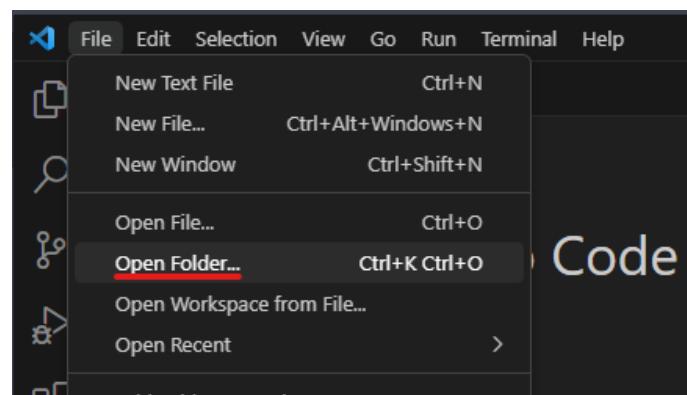


Figure 46: Open Folder in Visual Studio Code.

7. In Visual Studio Code, open the terminal by pressing **Ctrl+Shift+T**

(Windows/Linux) or Command+ (macOS), or by clicking on "Terminal" > "New Terminal" in the top menu.

8. Change the Terminal to a GitBash terminal.

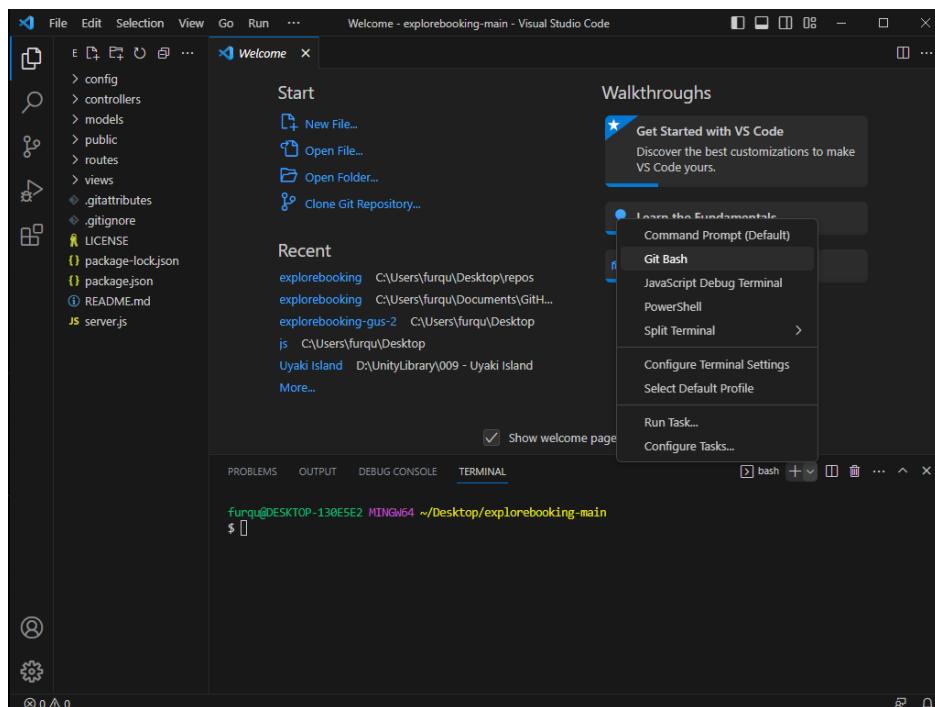
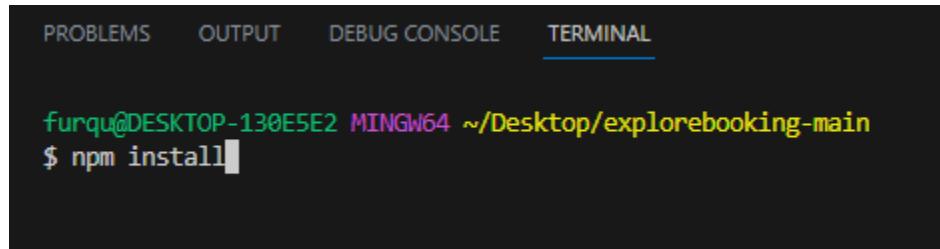


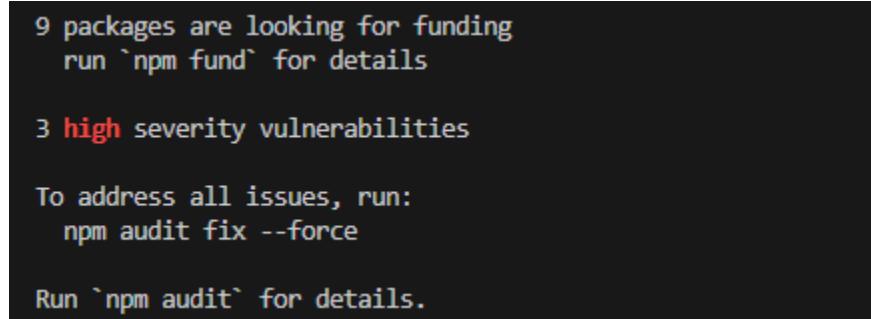
Figure 47: Change the Terminal in Visual Studio Code.

9. Install any required dependencies or packages for the program by running npm install in the terminal. This will download and install all the necessary packages listed in the package.json file.



A screenshot of a terminal window titled "TERMINAL". The window has a dark background and light-colored text. At the top, there are tabs labeled "PROBLEMS", "OUTPUT", "DEBUG CONSOLE", and "TERMINAL", with "TERMINAL" being underlined. Below the tabs, the terminal prompt is "furqu@DESKTOP-130E5E2 MINGW64 ~/Desktop/explorebooking-main". A command line is shown with the text "\$ npm install" followed by a cursor. The entire terminal window is enclosed in a black border.

Figure 48: Running npm install.



9 packages are looking for funding
run `npm fund` for details

3 **high** severity vulnerabilities

To address all issues, run:
npm audit fix --force

Run `npm audit` for details.

Figure 49: Result of Running npm install.

10. Run the program by typing node server.js in the terminal and pressing Enter.

```

furqu@DESKTOP-130E5E2 MINGW64 ~/Desktop/explorebooking-main
$ node server.js
(node:46788) [MONGOOSE] DeprecationWarning: Mongoose: the `strictQuery` option will be switched back to `false` by default in Mongoose 7. Use `mongoose.set('strictQuery', false);` if you want to prepare for this change
. Or use `mongoose.set('strictQuery', true);` to suppress this warning.
(Use `node --trace-deprecation ...` to show where the warning was created)
Express app running on http://localhost:3000
====> Connected to MongoDB.
[]

```

Figure 50: Running node server.js.

11. Open a web browser and navigate to <http://localhost:3000> to access the application.

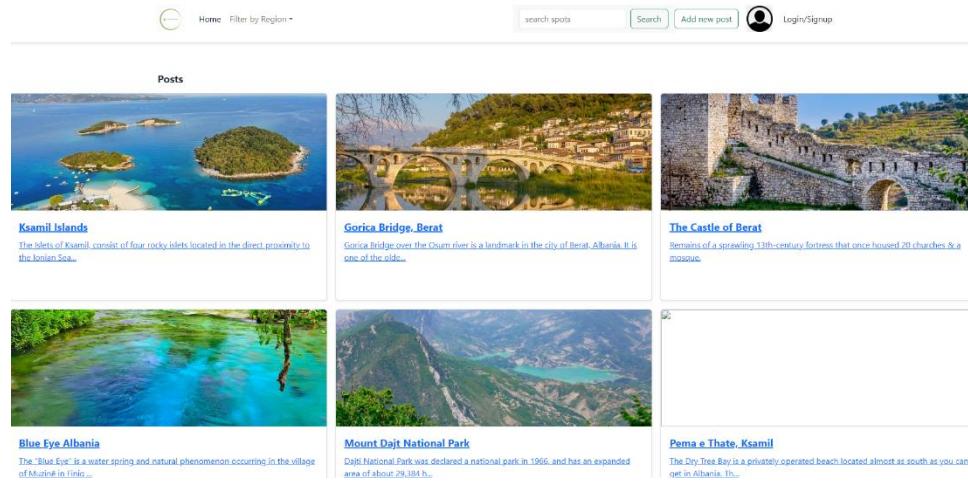


Figure 51: Navigate to localhost:3000.

Guide for Administrator

An Administrator can delete any post regardless of who owns it.

To create an account with Admin privileges:

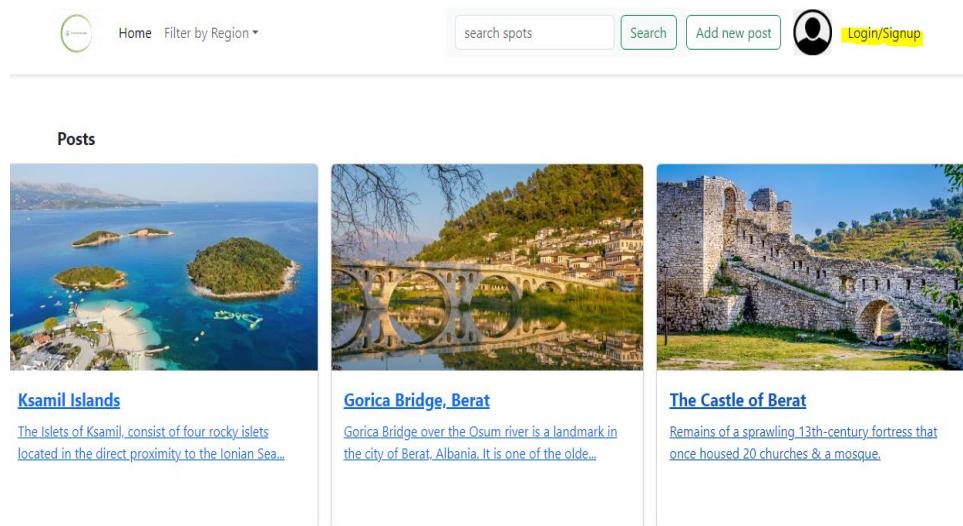


Figure 52: Login/Signup button on Home page.

- Click the Login/Signup button located in the upper right of the web application.

A screenshot of a login form titled 'Login Form'. The form has two input fields: 'Username:' and 'Password:', each with a corresponding input box. Below the password field is a large blue 'Login' button. At the bottom of the form, there is a link 'No Account? [Sign Up](#)'.

Figure 53: Sign Up hyperlink in Login page.

- Click the Sign-Up hyperlink.

Signup Form

Firstname:

Lastname:

Email:

Username:

Password:

signup

Figure 54: Signup Form.

- Fill-out the form and press the signup button.

comment

- Storage size: 20.48 kB
- Documents: 20
- Avg. document size: 106.00 B
- Indexes: 1
- Total index size: 36.86 kB

post

- Storage size: 36.86 kB
- Documents: 63
- Avg. document size: 500.00 B
- Indexes: 1
- Total index size: 36.86 kB

user

- Storage size: 20.48 kB
- Documents: 9
- Avg. document size: 334.00 B
- Indexes: 2
- Total index size: 73.73 kB

Figure 55: MongoDB Compass Collection.

- Once created, open the MongoDB Compass

databank.user

9 DOCUMENTS 2 INDEXES

| Documents | Aggregations | Schema | Explain Plan | Indexes | Validation |
|--|--------------|--------|--------------|---------|------------|
| Filter <input type="text"/> Type a query: { field: 'value' } <input type="button" value="Reset"/> <input type="button" value="Find"/> <input type="button" value="More Options"/> | | | | | |
| <input type="button" value="ADD DATA"/> <input type="button" value="EXPORT COLLECTION"/> 1 - 9 of 9 | | | | | |
| <pre>_id: ObjectId('641538c9b4462d11e3665c76') firstName: "John" lastName: "Smith" email: "john@smith.ca" username: "john" password: "PVSaJQ888Bg\$14creU/1fa+FucC6/39c8bPY/PmSOG+Ky8/5hSanEVTCzC89oFmn9khrkie..." gender: "Male" nationality: "Albanian" interest: "travelling" address: "Ontario, Canada" aboutMe: "I am a local traveller from Albania" provider: "local"</pre> | | | | | |

Figure 56: MongoDB Compass - User folder.

- Expand the databank and select the user folder.



A screenshot of the MongoDB Compass interface showing a single document in a collection. The document contains the following fields and values:

- _id: ObjectId('6440922ad9fd472c3f3530d9')
- firstName: "ArvinFname"
- lastName: "ArvinLname"
- email: "arvintestdata@contoso.com"
- userType: "normal"
- username: "acatest"
- provider: "local"
- created: 2023-04-20T01:15:22.270+00:00
- salt: "1353139775215472ba8918c35f86dd31"
- password: "z9ixh\$4U+bZcf5m3H65F732wLMqbjZXHq6lXsOC0dIgWzZ05i2NP9YTNRNDASIMVBfo2n..."
- v: 0

The interface includes a toolbar with edit, insert, delete, and refresh buttons at the top right.

Figure 57: An example of data in MongoDB Compass.

- Search for your created Account and press the edit (pencil) button.

The screenshot shows a MongoDB Compass interface with a document update dialog. The document fields and their types are listed:

| Field | Type |
|-----------|----------|
| _id | ObjectId |
| firstName | String |
| lastName | String |
| email | String |
| userType | String |
| username | String |
| provider | String |
| created | Date |
| salt | String |
| password | String |
| __v | Int32 |

A message at the bottom left says "Document modified." There are "CANCEL" and "UPDATE" buttons at the bottom right.

```
1 _id: ObjectId('6440922ad9fd472c3f3530d9')  
2 firstName: "ArvinFname"  
3 lastName: "ArvinLname"  
4 email: "arvintestdata@contoso.com"  
5 userType: "admin"  
6 username: "acatest"  
7 provider: "local"  
8 created: 2023-04-20T01:15:22.270+00:00  
9 salt: "1353139775215472ba8918c35f86dd31"  
10 password: "z9ixhS4U+bZcf5m3H65F732wlKMqbjZXHq6lXsOC0dIgWzZ05i2NP9YTNRNDAS"  
11 __v: 0
```

Figure 58: Update the data in MongoDB Compass.

- Change the userType value from normal to admin and press the update button.

Login Form

Username:

Password:

No Account? [Sign Up](#)

Figure 59: Login Form.

- Once updated, navigate back to the webpage and log-in using your credentials.

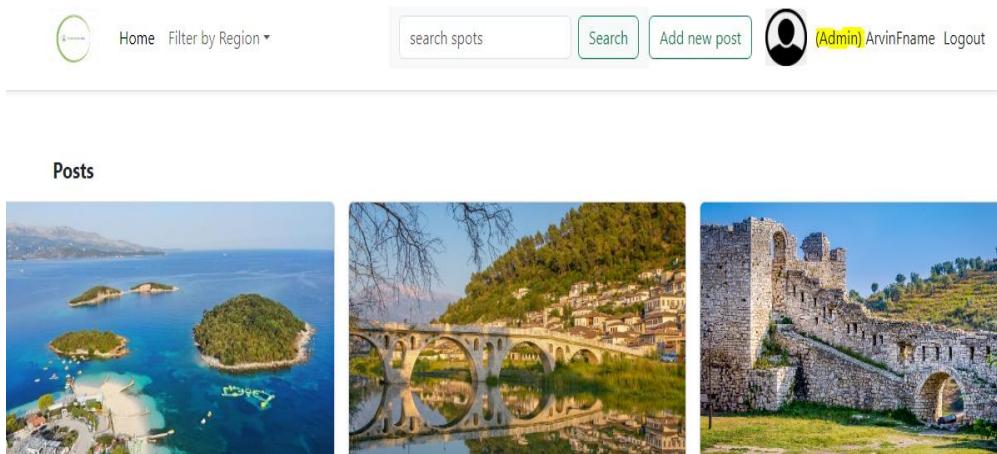


Figure 60: "Admin" shown in the Home page.

- Verify the account by checking if (Admin) word is beside the name of the user.

Delete any post using Admin privileges:

Pre-requisites:

- Admin account is logged in.
- Posts are existing in the web application.

To delete post:

- Select a post the user wants to edit by clicking on the picture.

The screenshot shows a user interface for a travel or tourism website. At the top, there is a navigation bar with a logo, 'Home', 'Filter by Region', a search bar ('search spots'), a 'Search' button, a 'Add new post' button, and a user profile icon ('(Admin) ArvinName Logout'). Below the navigation bar, the word 'Posts' is displayed. Three posts are listed in a grid:

- Ksamil Islands**: An aerial photograph of several small, rocky islands in the sea. Below the image, the text reads: 'The Islets of Ksamil consist of four rocky islets located in the direct proximity to the Ionian Sea...'.
- Gorica Bridge, Berat**: A photograph of a stone arch bridge reflected in a body of water. Below the image, the text reads: 'Gorica Bridge over the Osum river is a landmark in the city of Berat, Albania. It is one of the olde...'.
- The Castle of Berat**: A photograph of a large, ancient stone fortress built into a hillside. Below the image, the text reads: 'Remains of a sprawling 13th-century fortress that once housed 20 churches & a mosque.'

Figure 61: Posts in the Home page.

Rozafa Castle

Author: May Smith
Thu Mar 30 2023 21:05:44 GMT-0400 (Eastern Daylight Time)

[Edit Post](#) [Delete Post](#)

Figure 62: Delete button in the Post Details of a post.

- Once the user is in the post details, select the delete post.
- Once the button is triggered, it will navigate the user in the confirmation

page that the post is deleted.

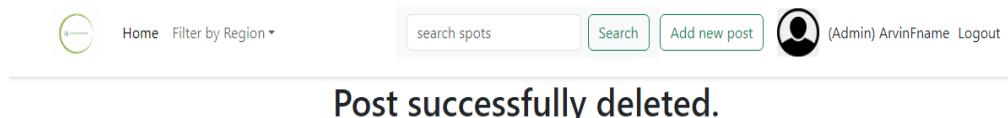


Figure 63: Confirmation after deletion.

Guide for Normal User

For Visitor:

Signup page:

- five required text inputs: firstname, lastname, email, username, password
- visitor will be logged in and redirected to homepage if signup process completed successfully.
- a link to the login page if visitor already has an account.

The screenshot shows a web application interface with a navigation bar at the top. The navigation bar includes a home icon, 'Home', 'Filter by Region ▾', a search bar ('search spots'), a 'Search' button, a 'Add new post' button, a user profile icon, and a 'Login/Signup' link. Below the navigation bar is a 'Signup Form' modal window. The form contains fields for Firstname (Kristi), Lastname (Goxhaj), Email (Kristigoxhaj@live.com), Username (herkuli), and Password (represented by a series of dots). At the bottom of the form is a blue 'signup' button.

Login page:

- two required text inputs: username, password
- visitor will be logged in and redirected to homepage if signup process completed successfully.
- a link to the login page if visitor already has an account.

search spots Add new post Login/Signup

Login Form

Username:

Password:

No Account? [Sign Up](#)

© Copyright 2023 Explore Booking

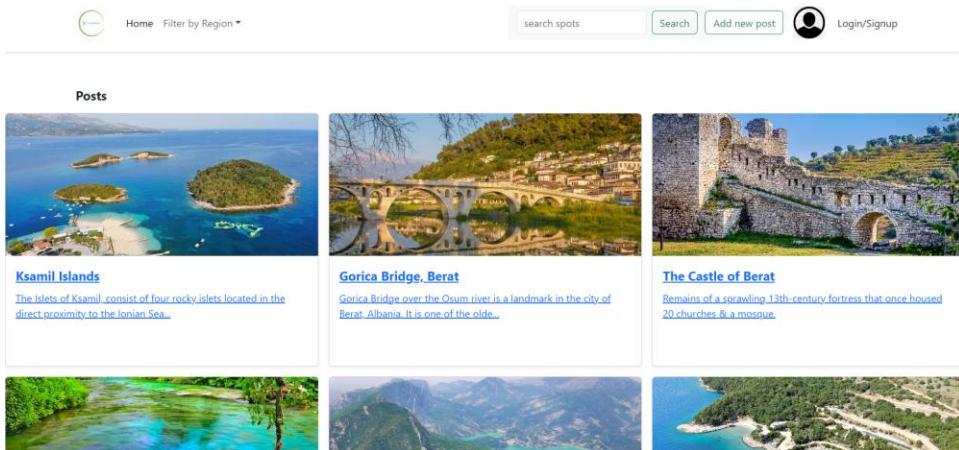
Navbar:

- dropdown list to filter Spots by region
- text input to search the wanted spot
- Add new post button, if clicked it will redirect to login page
- and a button to redirect for login/signup pages



Home page:

- Grid list of posts will be displayed for the visitor before signing in to the website.
- Footer to include the copyright to our brand.



Post Details page:

- to have the all details about the post that the visitor needs to know
 - visitor will be redirected to login page if clicked on edit or delete post buttons
 - visitor can write a comment and it will be submitted without an author name

Comments

Enter your comment

Submit comment

Comments

may2: a comment Wed Apr 05 2023 05:59:05 GMT-0400 (Eastern Daylight Saving Time)

may2: Wed Apr 05 2023 06:04:47 GMT-0400 (Eastern Daylight Saving Time)

may2: Wed Apr 05 2023 06:05:23 GMT-0400 (Eastern Daylight Saving Time)

: heello Wed Apr 05 2023 06:14:59 GMT-0400 (Eastern Daylight Saving Time)

: abc Thu Apr 06 2023 01:01:54 GMT-0400 (Eastern Daylight Saving Time)

: Beautiful City Thu Apr 06 2023 01:33:24 GMT-0400 (Eastern Daylight Saving Time)

: test Sat Apr 08 2023 00:20:23 GMT-0400 (Eastern Daylight Saving Time)

testasdasd: testasdasd Sat Apr 08 2023 10:46:10 GMT-0400 (Eastern Daylight Saving Time)

: Beautiful Place Sun Apr 16 2023 12:09:52 GMT-0400 (Eastern Daylight Saving Time)

: Nice Place Sun Apr 16 2023 12:17:36 GMT-0400 (Eastern Daylight Saving Time)

: Nice Image Sun Apr 16 2023 12:50:37 GMT-0400 (Eastern Daylight Saving Time)

Post Details

Ksamil Islands



Author: John Smith

Thu Mar 16 2023 20:00:00 GMT-0400 (Eastern Daylight Saving Time)

[Edit Post](#) [Delete Post](#)

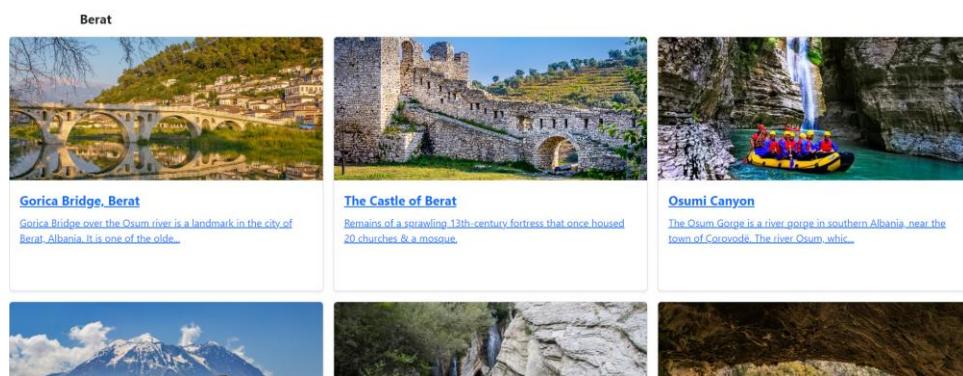
The Islets of Ksamil, consist of four rocky islets located in the direct proximity to the Ionian Sea in Southern Albania. The village of Ksamil, after whom the islets are named, is located to the east of the islets. Furthermore, the islands are situated within the boundaries of the Butrint National Park.

Filter by region results page:

- will search for each post that has the region value as the selected input.

Filter by Region ▾

- Berat
- Dibër
- Durrës
- Elbasan
- Fier
- Gjirokastër
- Korçë
- Kukës
- Lezhë
- Shkodër
- Tirana
- Vlorë



For User:

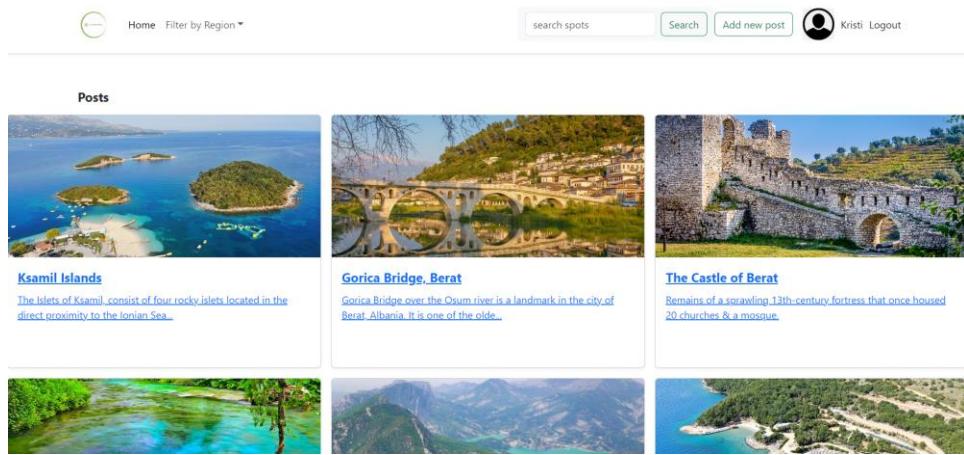
Navbar:

- All the features that the visitor have are included
- instead of the **login/signup** user will have **logout** button.
- “Profile” Icon to redirect the user to the profile page
- Add new post will redirect user to the “Add a new post page”



Home page:

- all the features that the visitor has.



Profile page:

- all the details about the user.
- edit profile button to redirect to edit profile page.

The screenshot shows a user profile page. On the left, there's a sidebar with a placeholder profile picture, the text "Hi there! I'm Kristi", and a section titled "About Me" with a "Edit profile" link. On the right, there are several text input fields: Name (Kristi Goxhaj), Gender, Joined Date (Fri Apr 21 2023 15:55:00 GMT-0400 (Eastern Daylight Saving Time)), Nationality, Interests, and Address. At the top, there are navigation links for Home, Filter by Region, search spots, Search, Add new post, and a user icon for Kristi with a Logout option. A copyright notice at the bottom reads "© Copyright 2023 Explore Booking".

Edit Profile page:

- five text inputs for the values that user needs to change:
username, about me, interests, password, confirm password.
- on success, user will be directed to the profile page with updated data shown.

The screenshot shows the "Edit Profile" form. It contains five text input fields: "Change Username" with the value "herkuli", "Change about me" with the value "I'd recommend anyone th", "Change interests" with the value "Traveling", "Password" with a redacted value, and "Confirm Password" with a redacted value. Below the inputs is an "Update" button. A copyright notice at the bottom reads "© Copyright 2023 Explore Booking".

Add a new post page:

- it has 4 required inputs:
- text inputs: title, picture URL, content

- dropdown select input: Region

after submission user will be redirected to homepage

Add a New Post

Title:

Region:

Picture URL:

Content:

| <u>Post</u> | <u>Details</u> | <u>page:</u> |
|--|----------------|--------------|
| <ul style="list-style-type: none">- user can view all the details about the post.- user can comment and his username will be shown with his comment- user can only edit or delete the post if he/she is the owner of it, other than that user will be redirected to Rejection error page. | | |

Post Details

Ksamil Islands



Author: John Smith

Thu Mar 16 2023 20:00:00 GMT-0400 (Eastern Daylight Saving Time)

[Edit Post](#) [Delete Post](#)

The islets of Ksamil, consist of four rocky islets located in the direct proximity to the Ionian Sea in Southern Albania. The village of Ksamil, after whom the islets are named, is located to the east of the islets. Furthermore, the islands are situated within the boundaries of the Butrint National Park.

Search results page:

- will search for any post that has the searched keyword either in the title or the description of the post.

Home Filter by Region ▾

Castle Add new post Login/Signup

Search Result: Castle

The Castle of Berat

[The Castle of Berat](#)
Remains of a sprawling 13th-century fortress that once housed 20 churches & a mosque.

Lekursi Castle

[Lekursi Castle](#)
Lekursi Castle is a castle on Lekursi Hill just behind Saranda with an amazing view over the town a...

Saranda

[Saranda](#)
Sarandë is a resort on the Albanian Riviera in southern Albania. Sandwiched between the Ionian Sea...

Aerial view of a coastal area

Clock tower in a hillside town

Aerial view of a coastline

Filter by region results page:

- as visitor

Berat



Gorica Bridge, Berat

[Gorica Bridge over the Osum river is a landmark in the city of Berat, Albania. It is one of the oldest...](#)



The Castle of Berat

[Remains of a sprawling 13th-century fortress that once housed 20 churches & a mosque.](#)



Osumi Canyon

[The Osum Gorge is a river gorge in southern Albania, near the town of Corovode. The river Osum, whic...](#)



APPENDIX D: PROGRESS MONITORING

Velocity

The iteration velocity is calculated by the number of the tasks. In Iteration 1, we have completed 7 user stories which contain 33 tasks, all accounting for 81 story points. Therefore, the velocity of Iteration 1 is 81 points. At the end of Iteration 2, we have completed 6 user stories, accounting for 48 story points. Therefore, the velocity of Iteration 2 was 48 points.

All iterations

The following is a table illustrating progress and changes for all iterations per Release and Iteration Breakdown Chart. The story points at start of the iteration 1 were 129. At the end of that iteration, we completed almost half of the stories, but certainly the ones with the highest amount of story points, finishing with a velocity of 81 points per iteration. Additionally, we did not have any new stories added, so iteration 2 is starting at 48 story points left. Since the new iteration velocity can't exceed first iteration's, the team is in position to execute the tasks in the remaining time, given that a 48 points velocity is needed.

Table 27: Progress and changes for a release with two iterations.

| | Iteration 1 | Iteration 2 |
|--------------------------|-------------|-------------|
| Story points at start of | 129 | 48 |

| iteration | | |
|----------------------------------|----|----|
| Completed during iteration | 81 | 48 |
| Changed estimates | 2 | 0 |
| Story points from new stories | 0 | 0 |
| Story points at end of iteration | 48 | 0 |

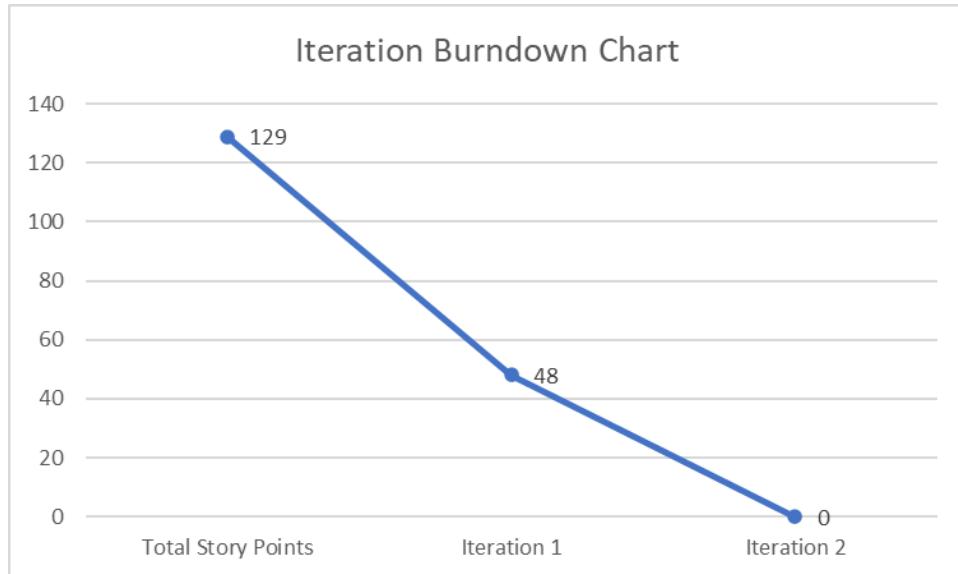


Figure 64: Iteration Burndown Chart.

Total estimated and actual hours for developers

The following table shows the total estimation hours for each developer. In the

table, it is shown that Chung Yin Tsang has extremely high estimated and actual hours in Iteration 1. The main reason is that some developers did not do their assigned tasks as mentioned in the above paragraphs, and Chung Yin Tsang volunteered for those not-done tasks because she had completed all her assigned tasks earlier and was available to take more tasks. In Iteration 2, the estimated hours for each developer were closer to the hours projected, and work distribution was more balanced than in Iteration 1.

Table 28: Total Estimated and Actual Hours for Each Developer.

| Developer | Iteration 1 | | Iteration 2 | |
|----------------------------------|-----------------|--------------|-----------------|--------------|
| | Estimated Hours | Actual Hours | Estimated Hours | Actual Hours |
| Adityasinh Makwana | 5 | 4 | 4 | 3 |
| Arvin Almario | 5 | 3 | 4 | 5 |
| Chung Yin Tsang | 23 | 19 | 4 | 5 |
| Egor Shabalkin | 5 | 4 | 4 | 4 |
| Gustavo Gomes M L R F Almeida | 1 | 1 | 2 | 3 |
| Karanjot Singh | - | - | 1 | 4 |
| Kristi Goxhaj | 6 | 5 | 4 | 4 |
| Saima Rafik Patel | 7 | 8 | 2.5 | 6 |

APPENDIX E: PROJECT PROPOSAL

ExploreBooking: A Web Application for Travel

A. Executive Summary

With an increasing number of traveling tourists from around the world to Albania, more and more people are looking for travel recommendations to Albania. Current websites or applications are offering limited travel ideas for travelers and detailed information for many places in Albania is hard to find. Therefore, the team decides to create an application catering to tourists wanting to explore unfamiliar spots that are only known to locals. The solution the team proposed is a web application called ExploreBooking. The application allows registered users to log in to the system and create posts to share their recommended travel spots in any region in Albania. Users can view posts, filter posts by region, and make comments under each post. The application also provides links to other websites for booking flights and hotels.

B. Business Objectives

1. Business Need/Opportunity/Objectives

Recently, a Virtuoso study reveals that the number of travelers booking trips for 2023 on average is far more than that in 2019. According to the article on Euractiv and Global Voices, 2022 was a bumper year of tourism

for Albania, and there is a raising trend in tourism. However, there are difficult to gain travel detailed recommendations for Albania due to the lack of applications focusing on a travel guide to Albania. The current solutions provide limited travel recommendations for Albania and lack of comments from travelers and local people. To facilitate the development of tourism in Albania, the government and travel-related companies may want a solution to promote different attractions in Albania. It brings a vast opportunity to connect adventurous tourists with the potential but not-widely-known touristic attractions recommended by locals.

2. Product Description (Solution)

The solution proposed here is a web application sharing travel recommendations from locals in Albania. The application will offer a platform for locals to share and promote their recommended places, and for travelers to gain up-to-date and detailed travel information as well as real comments from other travelers. The project will benefit travelers that they can get detailed travel recommendations easily. It will also benefit the travel industry of Albania as the project will promote potential attractions in Albania. The project will be successful if at least 1% of tourism to Albania obtained travel information from it.

3. Deliverables

This project has several deliverables when it is completed. Deliverables include:

- An automated posting system.
- A frontend web application.
- A backend web application.

Deliverables exclude:

- Test data.
- Database (uses current legacy database).
- Servers.

C. Project Description

1. Scope

ExploreBooking includes several functions such as the sign-up and login functions. Users can create an account and log in to the system. The main feature of the web application is the posting function for travel recommendations. Logged-in users can click the ‘add a new post’ button to create a new post. They can input the content of the post which includes the title, the text content, and an image. The web application also has the view post summary, filtering post, view post details, and comment

function. The users can view the overviews of all posts on the homepage. They can filter the posts by selecting a specific region in Albania. When they click on a post, they can view the details of the post and all comments on the post. They can also add a comment under the post no matter whether they are logged in or not. If they are logged in, the first and last names of the user will be shown beside the comment. If they are not logged in, the text 'anonymous user' will be shown. There are functions showing flight and hotel information. Users can view information about websites that are related to flights to and hotels in Albania. There are buttons linking to those websites. Users will be directed to the corresponding website when they click on the button.

2. Completion Criteria

This project is considered complete when the web application performs the following functions:

- (1) The users are able to sign up for an account and log into the system.
- (2) The logged-in users can add posts for writing travel recommendations for travelers.
- (3) All users can view the details of the posts and add comments to the post.
- (4) There are buttons linking to corresponding booking websites that direct users to that website when they are clicked.

3. Measures of Project Success

3.1 According to the Albanian Institute of Statistics ([Press release \(Professional design\) \(instat.gov.al\)](#)), in 2022 Albania recorded an affluence of foreign visitors of 7.5 million. Our platform is aiming to capture the attention of at least 1% of the people visiting the country, so the yearly hits target for the site is 75000.

3.2 One of the main drivers of the project is local involvement in recommending alternative destinations for tourism, so the success of it is linked to the engagement of locals to the platform. To measure the success in this aspect, we are expecting at least 100 alternative destinations suggested from people living on the country.

Project Approval Form

I, Jake Nesovic, have reviewed the information contained in this document and approve the project proposal.

Signature

Date