Campdays

Requirements Specification and Analysis

Version 1.0

16.03.2018

Orhan Aykut Kardeş

Muhammed Ali Altınel

Hüseyin Haluk Nurbaki

Berke Kağan Yıldız

Batuhan Kurt

Mert Günay

Prepared for

SE302 Software Engineering Practice



Table of Contents

[1. Introduction 1](#_Toc498723792)

[1.1. Purpose of the System 1](#_Toc498723793)

[1.2. Scope of the System 1](#_Toc498723794)

[1.3. Objectives and Success Criteria of the Project 1](#_Toc498723795)

[1.4. Definitions, Acronyms, and Abbreviations 1](#_Toc498723796)

[1.5. Overview 2](#_Toc498723797)

[2. Current System 2](#_Toc498723798)

[3. Proposed System 2](#_Toc498723799)

[3.1. Overview 3](#_Toc498723800)

[3.2. Functional Requirements 3](#_Toc498723801)

[3.3. Nonfunctional Requirements 6](#_Toc498723802)

[Usability: 6](#_Toc498723803)

[Reliability: 6](#_Toc498723804)

[Performance: 6](#_Toc498723805)

[Supportability: 6](#_Toc498723806)

[Implementation: 6](#_Toc498723807)

[Interface: 7](#_Toc498723808)

[Packaging: 7](#_Toc498723809)

[Legal: 7](#_Toc498723810)

[3.4. System Models 7](#_Toc498723811)

[Scenarios 7](#_Toc498723812)

[Use Case Model 15](#_Toc498723813)

[Object model 29](#_Toc498723814)

[Dynamic model 30](#_Toc498723815)

[3.5. Project Schedule 39](#_Toc498723816)

[4. Glossary 42](#_Toc498723817)

[5. References 43](#_Toc498723818)

REQUIREMENTS ANALYSIS DOCUMENT

# Introduction

## Purpose of the System

This project is a camping place reservation web-site where users can find the camp places according to their requirements. This idea is unique for our country, people who love camping can easily reach the camping places. The main purpose of this project is to help the people who want to find real camping places. This website contains several quality camping places. Camp owners can easily add their camp page to the system with admin approval. This system also supplies that every user can make advance search. Advance search ensures finding places that near to user who uses it. In addition, users can make a reservation from a camping place with using the pinned map which is on main page. As a result, this system is unique and, useful for finding a camping place.

## Scope of the System

Our system is called Campdays. Campdays can be used anywhere anytime as it is a web-based application. It is so simple to use; register to the system and start to search camping place instantly. This site is also an efficient sales platform for many camp owners. Any camp owner that owns a quality camping place can show to there with creating a camp page. This system collects the people who want to be together in a camping place with ease.

## Objectives and Success Criteria of the Project

The success of the system depends on providing the given main set of arguments:

* Use GitHub repository effectively with your teammates.
* The design of register and login systems.
* Association between database and system design.
* The general design of system in shop to have fast, efficient system.
* The demo of system should ensure good success rate.
* Implementation should be understandable, clear and basic and efficient.

## Definitions, Acronyms, and Abbreviations

The abbreviations and definitions contained in the document are given below:

* Campdays: Location based camping area finder
* Admin: System Admin
* Owner: Owner of a camping place on the site
* User: Uses the site
* Model: A schematic description of a system that accounts for its known or inferred properties.
* System: Any interacts by the application are considered to be done by the system.
* Efficiency: The properties of an algorithm, which is the amount of computational resources used by the algorithm.
* Service: Service is a keyword. Purpose of the service is to provide the customer with a secure payment system.

## Overview

This document contains sections;

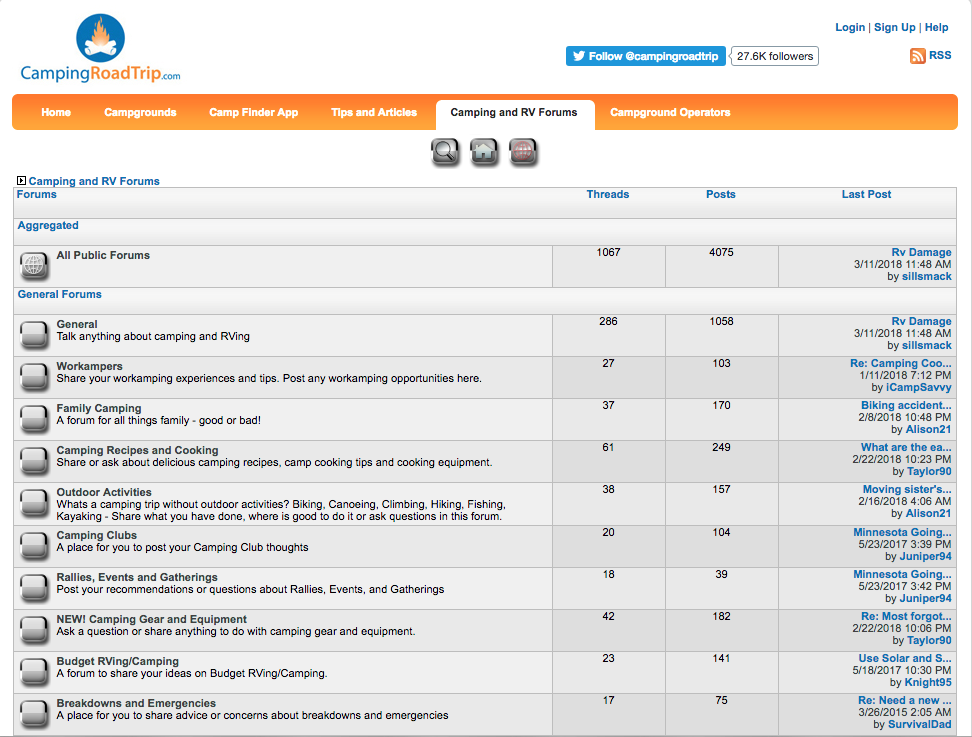
* Introduction
* Current System
* Functional Requirements
* Non-Functional Requirements
* Scenarios
* Use Case Model
* Object Model
* Dynamic Model
* User Interface
* Glossary
* References

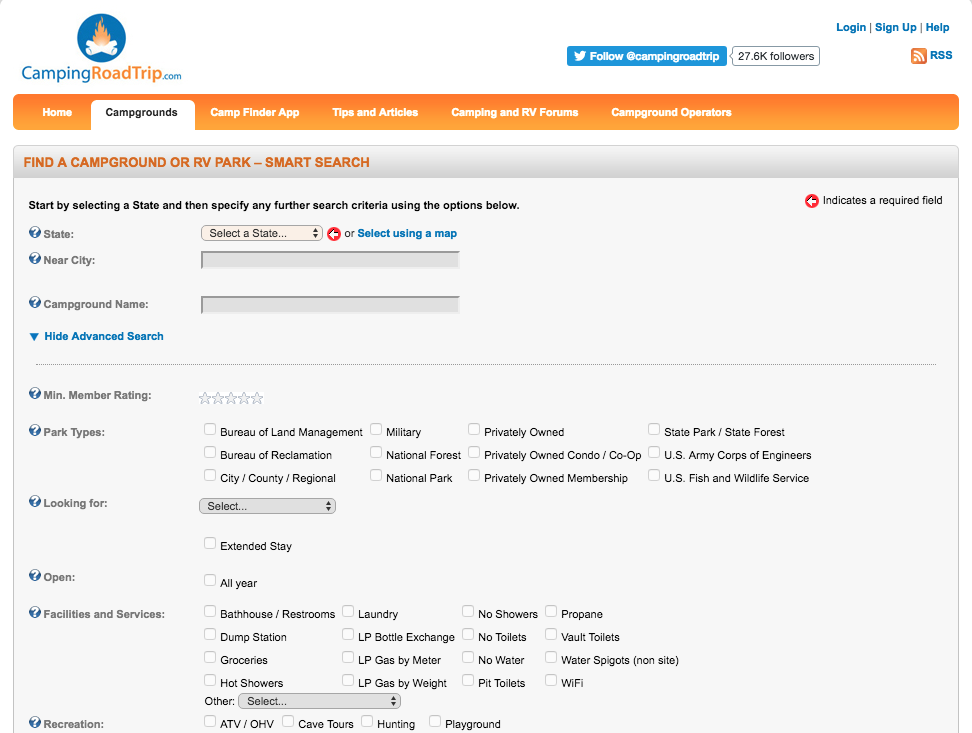
In introduction, we have described a good explanation of scope purpose, criteria, definitions and other explanations which help other people who want to read document. In proposed system, we have described the advantages of Campdays and what makes Campdays be an innovative and unique site. In functional requirement, we have described that the functions are open to end users. Functional requirements states for explaining scope of the system.

In non-functional requirement section, we have described our system’s reliability, performance and others to make the system accessible to end users. In Scenarios section, we have described Use Case ’s descriptions and explanations. In Use Case model, we have stated or Use Case diagram. In Object model, we have described the programming objects which will implemented later on. In Dynamic model, we have described the flow of system in time with operations and function of actors. In glossary section, we have described dictionary and unknown words in RAD.

# Current System

<http://www.campingroadtrip.com/> is a web application provides a list of camping areas in U.S.A online. Screenshots of the website are available at the end of this document. Search & Advanced search sections of the website are extremely poor. You can list the camp areas by their names without seeing no other information about them. Name of the camp must be provided for detailed search which is very bad for beginners and foreigners. The website is not user friendly at all. Design is very complicated. Also there is no map support. Forum section includes non-related topics.





# Proposed System

## Overview

Web based platform Campdays simply provides map supported location based advance search of camp areas. Campdays intends to bring camp owners and campers together. Amateur campers, professional campers, adventurers, vacationers etc. anyone wishes to camp, Campdays.com is the solution. Campdays is not just a bot searching camp areas. Platform allows people to make/cancel reservations for desired camp areas online. Camp areas are highly detailed, map supported and rated by campers. This is the key point of the platform. It leads campsite owners and people to never forget the fact that they always must protect their ratings. This is one of our quality assurances. People can share their experiences, thoughts, memories, comments or warnings. People can also follow each other. User friendly Campdays platform is separated from current systems especially by its well-thought features. People need more than camp names. Ratings, experiences will become decision mechanism for campers to decide. Map support simplifies location based searching. Finding camp areas will be easy as pointing. Campdays aims to become campers’ number one application

## Functional Requirements

***Customer Functional Requirements***

* **Sign Up**

The user registers on the system to make reservation through the site

* **Sign In**

The user logs in to the system with the username and password from the homepage.

* **Set Profile**

The user sets their information and profile picture.

* **Show Map**

The user sees maps with camping area pins on it.

* **Show Location**

The user sees his/her current location.

* **Search Location**

The user search camping area with specific location.

* **Show Camping Info**

The user sees selected camping area with entire information.

* **Search Near**

The user search closest camping areas to his/her current location.

* **Booking**

The user sees available quota of selected camping place and picks one.

* **Follow Follower System**

The user views all users and camping places, follows which his/her wants.

* **Rating**

The user rates camping place with limited rate scale.

* **Select Type**

The user views all camps from the camps page.

* **Show Reservation Detail**

The user views all camps from the camps page.

* **Edit/Cancel Reservation**

The user views all camps from the camps page.

* **Advanced Search**

The user makes extended searches by name, place and address to make it easier to find a camping place.

***Admin Functional Requirements***

* **Sign In;**

Admin part is different from CostumerSign in and Distributor Sign in. Admin should enter the different UML. In this page, admin can login on system with username and password. Password is given by the system database.

* **Set Camp Page**

Admin views all camping place decides coming request what it will happened.

* **Accept Request**

Admin accepts coming camping place requests.

* **Reject Request**

Admin rejects coming camping place requests.

* **Set Statistic**

Admin views all statistics.

* **Set User Account**

Admin manipulates user accounts.

* **Control Comment**

Admin views all comments and manipulates them.

* **User Control**

Admin views all users and manipulates them.

***Owner Functional Requirements***

* **Sign Up**

Owner registers to the system to share camping places.

* **Set Camp Page**

Admin prepares a camp page request.

* **Set Camp Detail**

Admin adds camping details.

* **Set Location**

Admin adds location in to their camping page.

* **Show Reservations**

Admin view all coming reservations request

* **Edit Reservations**

Admin deletes reservation

## Nonfunctional Requirements

Usability: Using the Campsday website is quite simple, there is a design that everyone can understand and a global interface. Our interface is based on camping and pictures; our customers are our priority to use our system without any confusion. Campdays continues to work to create a more effective interface.

Reliability: Campdays system has almost all security requirements that is %90.

Performance: Campdays uses SQLite as its database system, which has a permanent storage space. So, it reacts quite fast and at the same time about 3000 parallel users can enter our system.

Supportability: The maintenance of the Campdays system is a secure, short-lived portal that is open only for managers.

Implementation: The Campdays system is implemented in the Django and Python programming language with excellent OOP and POP features using ATOM and SQLite programs. Campdays uses SQLite database system platform in back-end, In-Front-End Campdays uses Django and its libraries. You can access it via [www.Campdays.com](http://www.campdays.com).

Interface: The interface of the Campdays system is designed using Django's interface implementation and HTML, CSS files. In addition, despite the use of Interfaces, third party applications and services used in the Campdays system and disclosed to customers may not be used. There are no inherited projects that present services or interfaces.

Packaging:The system packaging will be with future versions.

Legal:Campdays is licensed with MIT License. It is provided "as is" without warranty of any kind, including, but not limited to, marketability, manufacturability, software, certain fitness for purpose, and any infringement of rights. In any unusual circumstances, administrators shall not be liable for any damage or other liability arising out of the author, authors or copyright holders, from a software contract, tort or otherwise.

## System Models

### Scenarios

**Scenario 1**

|  |
| --- |
| **Scenario name:** User Rating and Comment |
| **Participant actor instances:**  Mert: User |
| **Flow of events:**   * Firstly, Mert enters the Campdays web-site as a user. * Mert went to a camp place before which is on the Campdays web-site. * So, he wants to rate or comment to this place. * Mert makes an advanced search for this camp page, he clicks on “Advance Search“ button, he finds and displays the page. * He can rate to clicks on “Rate button”, and he give stars 1 to 5 according to his satisfaction. * Also, He can click on “Comment button” which is on relevant camp place page for commenting. |

**Scenario 2**

|  |
| --- |
| **Scenario name:** User Follow-Follower System |
| **Participant actor instances:**  Mert: User, Berke: Owner |
| **Flow of events:**   * Mert visits the Campdays web-site. * He logins and wants to follow a user or camp page. * Mert types a name on search tab and, he clicks on “Advance Search” button. * After the Campdays system lists the results, he chooses a user or camp page. * Then he clicks to it, so he can easily follow a user or camp page easily to clicks on   “Follow“ button which is on their pages. |

**Scenario 3**

|  |
| --- |
| **Scenario name:** User Search Location |
| **Participant actor instances:**  Mert: User |
| **Flow of events:**   * Mert visits the Campdays web-site, and logs in. * He wants to find a camp place, and click on “Search Location“ button. * Mert chooses the location that he wants, he can filter and order the his searching to click on “Advanced Search “ button. * He can search and order the camp places with name, region, camp type, and price. * Also, he can search the places nearby with clicking on the “ Search Near “ button. * Finally, he can find a camp easily. |

**Scenario 4**

|  |
| --- |
| **Scenario name:** User Show Location |
| **Participant actor instances:** Mert: User |
| **Flow of events:**   * Mert visits the Campdays web-site. * He logins, and he wants to know his location. * Then, he clicks on “ Show Location “ button. * Mert can see his location. |

**Scenario 5**

|  |
| --- |
| **Scenario name:** User Show Camping Info |
| **Participant actor instances:**  Mert: User |
| **Flow of events:**   * Firstly, Mert enters the Campdays web-site, and logins. * He made a camp place reservation before. * He wants to show his camping information. * Mert clicks on “Show Camping Info“ button. * Then, he can see the all information about his reservation such as date, camping type, price, camp owner information. |
|  |

**Scenario 6**

|  |
| --- |
| **Scenario name:** User Show Map |
| **Participant actor instances:**  Mert: User |
| **Flow of events:**   * Firstly, Mert enters the Campdays web-site. * If he wants to look up a camp place, he logins. * When he clicks on " Show Map " button, he can see the full map which includes camp places with pins. * Mert can see the camp places clicking to pins which represent the camp pages. * Shortly, Mert chooses the region and click on the pin. * When he clicks on a pin, camp pages are shown on, he can easily find information about the camp places. |

**Scenario 7**

|  |
| --- |
| **Scenario name:** User Booking |
| **Participant actor instances:**  Mert: User |
| **Flow of events:**   * Firstly, Mert has to logs into the Campdays for booking. * After, he wants to book a camp place. * Mert clicks on " Booking " button. * Then, he can edit his reservation with click on “ Edit Reservation “ button ,   Also he can cancel his reservation with click on “ Cancel Reservation “ button.   * Mert can select the his camp type, for instance; he wants to camp with 2 people he chooses the camp place for 3 people. He clicks on “ Select Type “ button then he choose the camp type. * Finally he can see to his reservation details to clicks on “ Show Reservation Details “. |

**Scenario 8**

|  |
| --- |
| **Scenario name:** User Set Profile |
| **Participant actor instances:**  Mert: User |
| **Flow of events:**   * Firstly, Mert logs into the Campdays. * After that, he wants to update his profile. * He clicks on the ” Set Profile “ button. * He can update his profile in this page such as he can change his profile photo, can change e-mail , address , telephone number , shortly he can update all information about himself. |

**Scenario 9**

|  |
| --- |
| **Use Case name:** Owner Set Camp Page |
| **Participant actor instances:**  Berke: Owner |
| **Flow of events:**   * Firstly, Berke logs into the Campdays, he logins as an owner. * After that, he wants to a camp page profile for his camping place. * He clicks on the” Set Camp Page “button which is on his main page. * Then he has to give some information about his camping page to clicks on   “Set Camp details “button.   * Berke sets own camping page information with uploading photos of his camping place, specifying an address, giving the telephone number and identifying the camping type’s information. * Also, he has to set the location of his camping place to clicks on “Set Location “button. * As a result, he chooses the region of his camping place, and marks to there on map. |

**Scenario 10**

|  |
| --- |
| **Scenario name:** Owner Show Reservations |
| **Participant actor instances:**  Berke: Owner |
| **Flow of events:**   * After Berke logs as an owner on Campdays web-site. * He can see the reservations which are made by users. He can see names of users who make reservations and dates. * After he checks the reservations, if there is a bad situation about camp place, he can cancel the reservations to clicks on “Cancel Reservation” button. |

**Scenario 11**

|  |
| --- |
| **Scenario name:** Admin Set User Account |
| **Participant actor instances:**  Batuhan: Admin |

**Scenario 12**

|  |
| --- |
| **Scenario name:** Admin Set Camp Page |
| **Participant actor instances:**  Batuhan: Admin, Berke: Owner |
| **Flow of events:**   * Firstly, Batuhan logs into the Campdays as an admin. * If Berke sets a page for his camp, he has to send it to Batuhan for accepting. * After Berke sent his page to him, Batuhan sees the Berke’s request. * Then, Batuhan looks up to Berke’s camp page, and if there are no missing information or unsuitability, he approves the Berke’s page with clicking on “Accept Request “button. * Otherwise he clicks on “Reject Request “button. |
|  |

**Scenario 13**

|  |
| --- |
| **Use Case name:** Admin Set Statistic |
| **Participant actor instances:**  Batuhan: Admin**,** Berke: Owner |
| **Flow of events:**   * Batuhan enters the Campdays web-site as an admin. * Then if he receives, he wants to evaluate them. * So, Batuhan clicks on “Set Statistic “button on administrator page. * After he clicks, he can see the statistics that shows the user numbers who stay   in camps.   * When he sees the statistic, he can evaluate these statistics for example;   - Batuhan can share the information to owners if they request them. |

**Scenario 14**

|  |
| --- |
| **Scenario name:** Sign up |
| **Participant actor instances:**  Mert: User, Berke: Owner, Batuhan: Admin |
| **Flow of events:**   * Firstly, if a visitor enters to Campdays web-site. * Then, he wants to shop or wants to register Campdays-web site. * He clicks on "Sign Up" button on the main page. * He fills the all fields and clicks Sign Up. * If information is correct, and the passwords that are entered by him matches,   He can access his account.   * Finally, user account is created. |

**Scenario 15**

|  |
| --- |
| **Scenario name:** Sign In |
| |  | | --- | | **Participant actor instances:**  Mert: User, Berke: Owner, Batuhan: Admin | |
| **Flow of events:**   * A visitor enters to Campdays web-site. * Then, he may be user, owner or admin. * If he wants to only visit the web-site he doesn't have to be login, but he wants to   Login as an admin, user or owner he has to be login.   * So, He clicks on "Sign In" button on the main page. * He fills the username and password fields, then clicks sign in. * If information is correct, he will sign in, buy if the password is not correct, he can   Clicks on “Remember Password “button to reset his password.   * Finally, he signs in and if he wants to log out, he clicks on "Log out " button to log   On main page. |

### Use Case Model

### **Use Case 1**

|  |
| --- |
| **Use Case Name:** Set Profile |
| **Participant actor instances:** User |
| **Flow of events:**   * This event starts with User's click to the "Set Profile" button. * User is directed to the "Profile Settings Page". * System shows the settings of user on this page, and user can change the changeable fields here. * User clicks done * The updated settings send to database, and system shows the approve message. |
| **Entry Condition:** User should be signed-in. |
| **Exit Condition:** Settings updated or canceled. |
| **Quality Requirement:**  **1.** If the fields are filled as they shouldn't, then system shows an error message for the correct type. |

**Use Case 2**

|  |
| --- |
| **Use Case Name:** Show Map |
| **Participant actor instances:** User |
| **Flow of events:**   * This event starts with User's click to the Map on the Main Page. * User is directed to the Map page. * System shows map |
| **Entry Condition:** User should be in the "Main Page". |
| **Exit Condition:** Map Opened. |
| **Quality Requirement:**  **1.** Map shows the camping areas, but doesn't get the location first to perform faster, if user wants to see his/her location, Use Case 3 starts. |

**Use Case 3**

|  |
| --- |
| **Use Case Name:** Show Location |
| **Participant actor instances:** User |
| **Flow of events:**   * This event starts with User's click to the "Show Location" button on the Map page. * System shows the location of user in the "Map Page". |
| **Entry Condition:** User should be in the "Map Page". |
| **Exit Condition:** System gets the location and shows or can't get the location and shows an error. |
| **Quality Requirement:**  **1.** If location sharing data is not available then system shows the error and tips user if the location data is available in their device. |

**Use Case 4**

|  |
| --- |
| **Use Case Name:** Search Location |
| **Participant actor instances:** User |
| **Flow of events:**   * This event starts with User's click to the "Search Location" button on the Map page. * System shows the "district" field for the search. * User fills the required "district" fields for the search. * System opens the location of the district that is given. |
| **Entry Condition:** User should be in the "Map Page". |
| **Exit Condition:** System opens the given location in the map or can't get the location and shows an error. |
| **Quality Requirement:**  **1.** If the given location is not available then system shows the error and tips user for the typo errors. |

**Use Case 5**

|  |
| --- |
| **Use Case Name:** Advanced Search |
| **Participant actor instances:** User |
| **Flow of events:**   * This event starts with User's click to the "Advanced Search" button after the "Search Location" click on the "Map Page". * System shows the fields for the advanced search. * User fills the required fields for the search. * System opens the location of the place that is given. |
| **Entry Condition:** User should click to the "Search Location" button in the "Map Page". |
| **Exit Condition:** System opens the given location or camping place in the map or can't get the location or camping place and shows an error. |
| **Quality Requirement:**  **1.** If user only searches for a place then system shows the place in the map and shows other available places near to that place. |

**Use Case 6**

|  |
| --- |
| **Use Case Name:** Search Near |
| **Participant actor instances:** User |
| **Flow of events:**   * This event starts with User's click to the "Search Near" button after the "Search Location" click on the "Map Page". * System shows the camping places near to the User's location. |
| **Entry Condition:** User should click to the "Search Location" button in the "Map Page". |
| **Exit Condition:** System opens the user's location or can't get the location and shows an error. |
| **Quality Requirement:**  **1.** If location sharing data is not available then system shows the error and tips user if the location data is available in their device. |

**Use Case 7**

|  |
| --- |
| **Use Case Name:** Show Camping Info |
| **Participant actor instances:** User |
| **Flow of events:**   * This event starts with User's click to the any Camping Place on the "Map Page". * System opens the page of the camping place. |
| **Entry Condition:** User should be on the "Map Page". |
| **Exit Condition:** System opens the page of the camping place that user clicked. |

**Use Case 8**

|  |
| --- |
| **Use Case Name:** Booking |
| **Participant actor instances:** User |
| **Flow of events:**   * This event starts with User's click to the "Make a Reservation" button in the "Camping Place Page" * System shows the Calendar for check-in and check-out dates. * User checks the dates in the calendar. * System checks if the given date is available. * System shows the requirement fields for booking if the given date is available. * User fills the required fields. * System saves the reservations and shows the "Thank You" message. |
| **Entry Condition:** User should be signed-in and should be in the "Camping Place Page" |
| **Exit Condition:** User booked the reservation or cancelled. |
| **Quality Requirement:**  **1.** If the given dates are not available system shows nearest dates that is available to the given date. |

**Use Case 9**

|  |
| --- |
| **Use Case Name:** Select Type |
| **Participant actor instances:** User |
| **Flow of events:**   * This event starts with User's click to the "Select Type" button in the booking page. * System shows the types that is available for the place. * User selects a type. * System saves the type. |
| **Entry Condition:** User should booked a reservation |
| **Exit Condition:** User selected the type or cancelled. |
|  |

**Use Case 10**

|  |
| --- |
| **Use Case Name:** Show Reservation Detail |
| **Participant actor instances:** User |
| **Flow of events:**   * This event starts with User's click to the "Show Reservation Detail" button in the "Reservations Page". * System shows the reservations that are made by the user. |
| **Entry Condition:** User should be signed-in and should be in the "Reservations Page" |
| **Exit Condition:** System opens the detail of reservation. |
| **Quality Requirement:**  **1.** System shows if the reservation is approved by the place owner or not. |

**Use Case 11**

|  |
| --- |
| **Use Case Name:** Edit/Cancel Reservation |
| **Participant actor instances:** User |
| **Flow of events:**   * This event starts with User's click to the "Edit/Cancel Reservation" button in the "Reservation Detail Page" which is in the "Reservations Page". * System shows the available reservations. * User selects the desired reservation. * System shows the reservation with editable fields and "Cancel Reservation" button. * User edits the desired fields or presses "Cancel Reservation" button. * System shows a confirmation message box. * User clicks yes. * System updates edited or canceled reservation in the database. |
| **Entry Condition:** User should be signed-in and should be in the "Reservation Detail Page" |
| **Exit Condition:** User edited or canceled the reservation. |
| **Quality Requirement:**  **1.** If user wants the change the date, system automatically shows the available dates, not the whole calendar. |

**Use Case 12**

|  |
| --- |
| **Use Case Name:** Booking |
| **Participant actor instances:** User |
| **Flow of events:**   * This event starts with User's click to the "Make a Reservation" button in the "Camping Place Page" * System shows the Calendar for check-in and check-out dates. * User checks the dates in the calendar. * System checks if the given date is available. * System shows the requirement fields for booking if the given date is available. |
| **Entry Condition:** User should be signed-in and should be in the "Camping Place Page" |
| **Exit Condition:** User booked the reservation or cancelled. |
| **Quality Requirement:**  **1.** If the given dates are not available system shows nearest dates that is available to the given date. |

**Use Case 13**

|  |
| --- |
| **Use Case Name:** Follow Follower System |
| **Participant actor instances:** User |
| **Flow of events:**   * This event starts with User's click to the "Follow" button in the any camping page or user page. * System shows a confirmation message for following the desired user or place. * User clicks yes. * System saves the information in the database. |
| **Entry Condition:** User should be signed-in and should be in the any "Camping Place Page" or in the any user's "User Page" except for his/her own page. |
| **Exit Condition:** User followed a place or user. |
| **Quality Requirement:**  **1.** If the following user shares new comment or rating, then it notifications automatically to user.  2. If the following camping place shares new information about its place, then it notifications automatically to user. |

**Use Case 13**

|  |
| --- |
| **Use Case Name:** Follow Follower System |
| **Participant actor instances:** User |
| **Flow of events:**   * This event starts with User's click to the "Follow" button in the any camping page or user page. * System shows a confirmation message for following the desired user or place. * User clicks yes. * System saves the information in the database. |
| **Entry Condition:** User should be signed-in and should be in the any "Camping Place Page" or in the any user's "User Page" except for his/her own page. |
| **Exit Condition:** User followed a place or user. |
| **Quality Requirement:**  **1.** If the following user shares new comment or rating, then it notifications automatically to user.  2. If the following camping place shares new information about its place, then it notifications automatically to user. |

**Use Case 14**

|  |
| --- |
| **Use Case Name:** Rating |
| **Participant actor instances:** User |
| **Flow of events:**   * This event starts with User's click to the "Give Rating" button in the "Camping Place Page". * System shows the required field for the comment and stars for rating. * User gives the rating and comment. |
| **Entry Condition:** User should be signed-in and should already have past booking on the place. |
| **Exit Condition:** User rated and commented. |
| **Quality Requirement:**  **1.** If the page of the camping place is not booked by the user before, system shows an error message.  **2.** If the page of the camping place is booked but the date of reservation is not past, then system again shows an error message.  **3.** In this way only people who already stayed in the place gives ratings and comments. |

**Use Case 15**

|  |
| --- |
| **Use Case Name:** Set Camp Page |
| **Participant actor instances:** Owner |
| **Flow of events:**   * This event starts with Owner's click to the "Set a New Camp Place" button in the main page. * System shows the required fields for the setting for a new camping page in a new page. * Owner fills all the required fields. * System checks if all required fields are filled * System shows "Approval Message". * System sends the new page to the admin approval. |
| **Entry Condition:** Owner should be signed-in. |
| **Exit Condition:** Owner created a new Camping Page. |
| **Quality Requirement:**  **1.** After creating a new camping page owner can see approval status. |

**Use Case 16**

|  |
| --- |
| **Use Case Name:** Set Camp Details |
| **Participant actor instances:** Owner |
| **Flow of events:**   * This event starts with User's click to the "Set Details" button in the "Camping Place Page" of his/her own. * System shows the fields for the details that could change. * Owner fills the desired fields of details. * System shows "Approved" message. |
| **Entry Condition:** Owner should be signed-in and already have a Approved Camping Page |
| **Exit Condition:** Owner sets or changes the details of the camping page. |
| **Quality Requirement:**  **1.** System checks if any changes are made. If not it shows a message "No changes made". |

**Use Case 17**

|  |
| --- |
| **Use Case Name:** Set Location |
| **Participant actor instances:** Owner |
| **Flow of events:**   * This event starts with User's click to the "Set Location" button in the "Camping Place Page" of his/her own. * System shows the map to the owner. * Owner pins a place in the map. * System shows a confirmation message. * Owner clicks yes. |
| **Entry Condition:** Owner should be signed-in and already have a Approved Camping Page |
| **Exit Condition:** Owner sets or changes the location of the camping page. |
| **Quality Requirement:**  **1.** System opens the map with current location of the owner, then owner can click done with the current location signals taken from him/her, or can pin the map by his/her own. |

**Use Case 18**

|  |
| --- |
| **Use Case Name:** Show Reservations |
| **Participant actor instances:** Owner |
| **Flow of events:**   * This event starts with Owner's click to the "Show Reservation" button in the "Owner Page". * System shows the all reservations for the owner's place in the new page called "Reservations Page". * Owner can select any of it. |
| **Entry Condition:** Owner should be signed-in and already have a Approved Camping Page |
| **Exit Condition:** Owner sees the all reservations. |
| **Quality Requirement:**  **1.** If owner selects any of the reservations then the "Edit/Cancel Reservation" Use Case starts from there. |

**Use Case 19**

|  |
| --- |
| **Use Case Name:** Edit/Cancel Reservation |
| **Participant actor instances:** Owner |
| **Flow of events:**   * This event starts with Owner's selection of any reservation in the "Reservation Page" * Details of the selected reservations displays. * Owner can cancel a reservation from here. |
| **Entry Condition:** Owner should be signed-in and already have a Approved Camping Page |
| **Exit Condition:** Owner cancels a reservation. |
| **Quality Requirement:**  **1.** If a reservation cancels then the system immediately shows notification and sends mail to the user who has the reservation. |

**Use Case 20**

|  |
| --- |
| **Use Case Name:** Set Camp Page |
| **Participant actor instances:** Admin |
| **Flow of events:**   * This event starts with Admin's click to the "Set Camp Page" button in the "Admin Page". * System shows the all the requested camping pages to admin. * Admin sees the details of the pages and can select any of it. |
| **Entry Condition:** Admin should be signed in. |
| **Exit Condition:** Admin sees the pending pages. |
| **Quality Requirement:**  **1.** If admin selects any of it then "Accept Request" and "Reject Request" Use Case s starts after it. |

**Use Case 21**

|  |
| --- |
| **Use Case Name:** Accept Request |
| **Participant actor instances:** Admin |
| **Flow of events:**   * This event starts with Admin's click to the any pending camp page "Set Camp Page". * System shows the all the details of the pending camping page. * Admin clicks "Accept Request" button. |
| **Entry Condition:** Admin should be signed in and should be in the "set camp page" |
| **Exit Condition:** Admin accepts the camping. |
| **Quality Requirement:**  **1.** System immediately notifications the owner after the accepting is done. |

**Use Case 21**

|  |
| --- |
| **Use Case Name:** Accept Request |
| **Participant actor instances:** Admin |
| **Flow of events:**   * This event starts with Admin's click to the any pending camp page "Set Camp Page". * System shows the all the details of the pending camping page. * System opens a message box for admin * Admin types a message to the owner for explaining about rejection. * Admin clicks "Reject Request" button. |
| **Entry Condition:** Admin should be signed in and should be in the "set camp page" |
| **Exit Condition:** Admin accepts the camping. |
| **Quality Requirement:**  **1.** System immediately gives notification to the owner about the message that admin types for the owner. |

**Use Case 22**

|  |
| --- |
| **Use Case Name:** Set Statistic |
| **Participant actor instances:** Admin |
| **Flow of events:**   * This event starts with Admin's click to the "Statistic Page" * System shows the all the statistic to admin in one page. |
| **Entry Condition:** Admin should be signed in. |
| **Exit Condition:** Statistic information shows to the admin. |
|  |

**Use Case 23**

|  |
| --- |
| **Use Case Name:** Set User Account |
| **Participant actor instances:** Admin |
| **Flow of events:**   * This event starts with Admin's click to the "Set User Account" page. * System displays user list to the admin. * Admin clicks to the any user in the list. * System shows details of the user. |
| **Entry Condition:** Admin should be signed in. |
| **Exit Condition:** User list displayed to the admin. |
| **Quality Requirement:**  **1.** After admin clicks to the user then "User Control" or "Watch Comment" Use Case s start. |

**Use Case 24**

|  |
| --- |
| **Use Case Name:** Watch Comment |
| **Participant actor instances:** Admin |
| **Flow of events:**   * This event starts with Admin's click "Watch Comment" button in the user details of "Set User Account" Page's list of users. * System displays comments written by user. * Admin clicks to a comment. * System shows the comment with the edit box. * Admin edits the comment with the edit box. |
| **Entry Condition:** Admin should be signed in and should select a user in the list in the "Set User Account" page. |
| **Exit Condition:** Admin edits a comment or cancels. |
|  |

**Use Case 25**

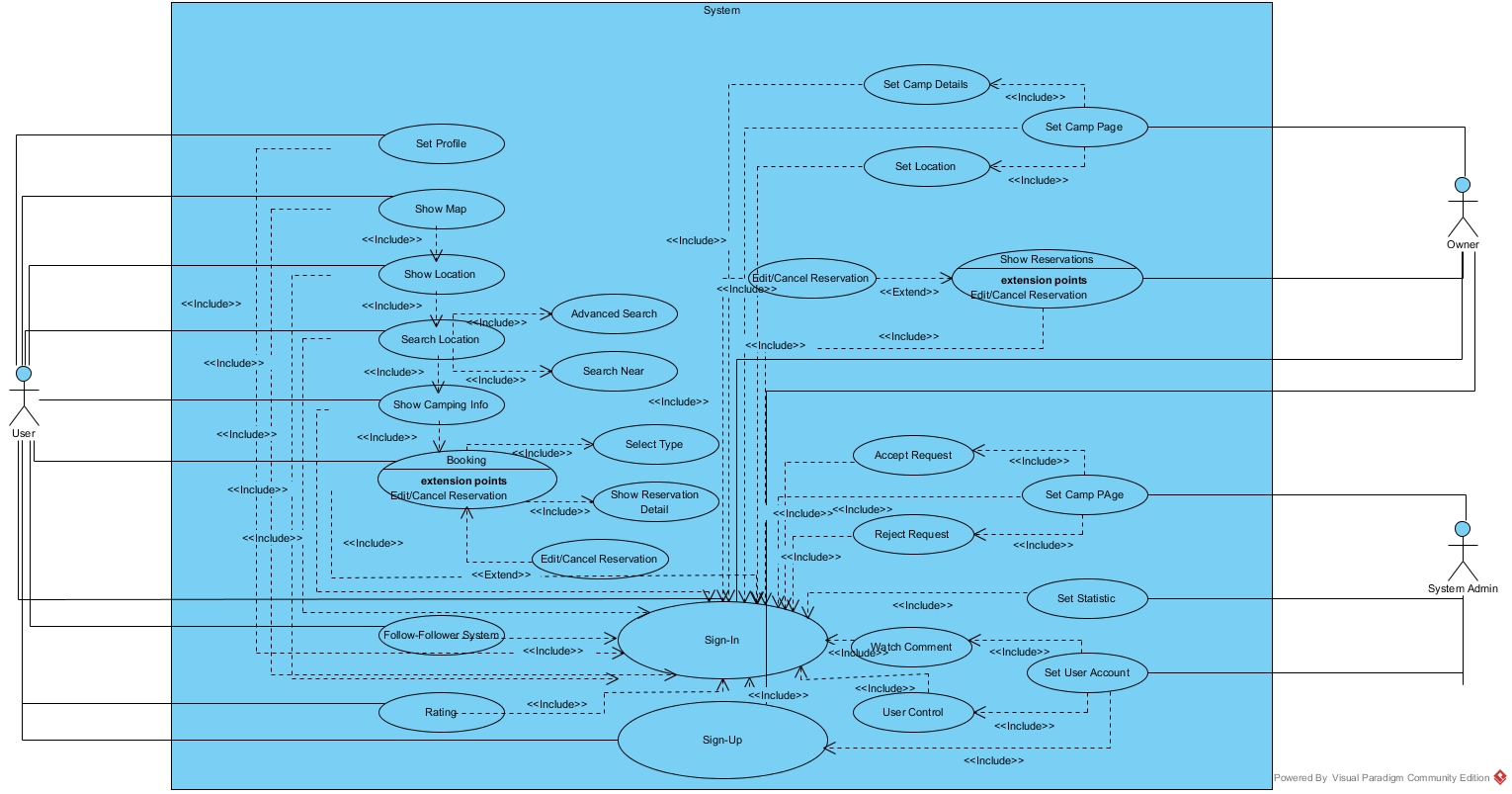
|  |
| --- |
| **Use Case Name:** User Control |
| **Participant actor instances:** Admin |
| **Flow of events:**   * This event starts with Admin's click "User Control" button in the user details of "Set User Account" Page's list of users. * System shows all the details of the user. * System displays a "Ban User" button here. * Admin clicks "Ban User" button. * System shows a confirmation message box. * Admin clicks yes |
| **Entry Condition:** Admin should be signed in and should select a user in the list in the "Set User Account" page. |
| **Exit Condition:** Admin bans a user or cancels. |
|  |

**Use Case 26**

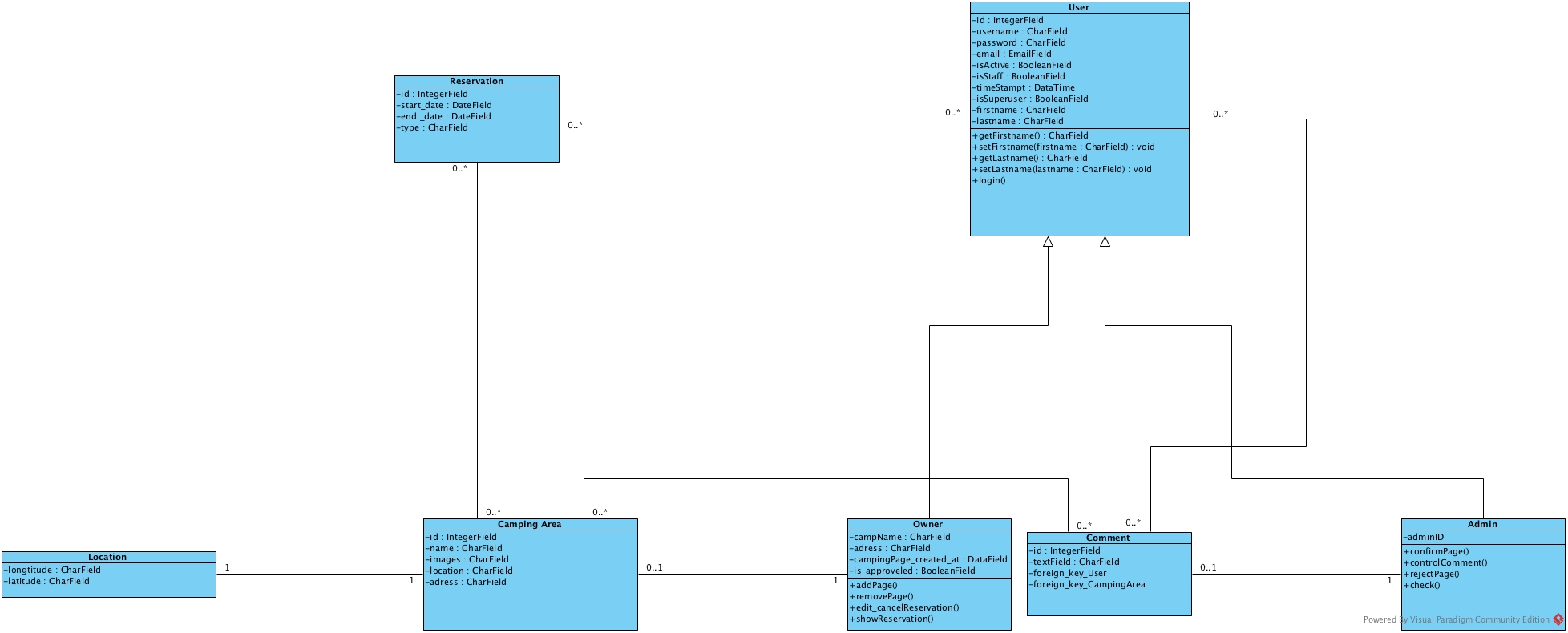
|  |
| --- |
| **Use Case name:** Sign-up |
| **Participant actor instances:** All actors. |
| **Flow of events:**   * Actor enters to the site. * Actor clicks "Sign-up" button. * Site responds by displaying the Sign-up screen. * Actor enters all required fields. * Actor clicks "Sign-up" button. * System inserts the information to the database which is provided by the actor. * System redirecting actor to the login page. |
| **Entry Condition:** Actor visits the URL page of the system. |
| **Exit Condition:** Actor clicks "Sign-up" button. |
| **Quality Requirements:**  **1.** If Actor leaves one or more fields empty, system displays a warning message, like  "This area cannot be empty.”  2. The information that user entered is checking and verified by the system in real-time |

**Use Case 27**

|  |
| --- |
| **Use Case name:** Sign-in |
| **Participant actor instances:** All actors. |
| **Flow of events:**   * Actor goes on sign-in page. * System responds by displaying the login screen. * Actor enters username and password to the login fields. * Actor clicks "Sign-in" button * If the username and password match with the username and password in the database, then user authenticates. * System opens the main page for the actor. |
| **Entry Condition:** Actor visits the URL page of the system. |
| **Exit Condition:** Actor clicks "Sign-in" button. |
| **Quality Requirements:**  **1.** If Actor leaves one or more fields empty, system displays a warning message, like  "This area cannot be empty.”  **2.** If the information’s checked from database are not true, system displays a warning  Message, like "Wrong username or password, please retry.” |

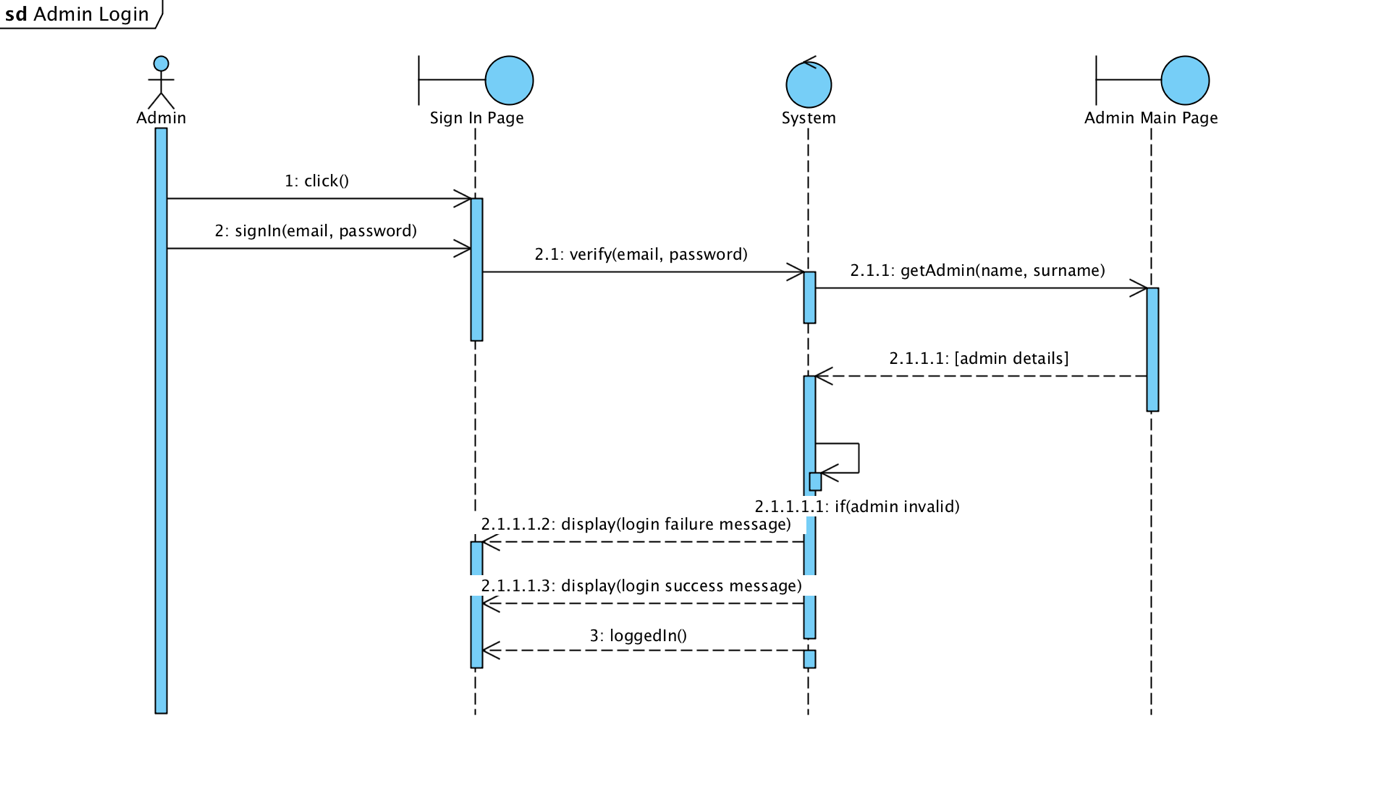


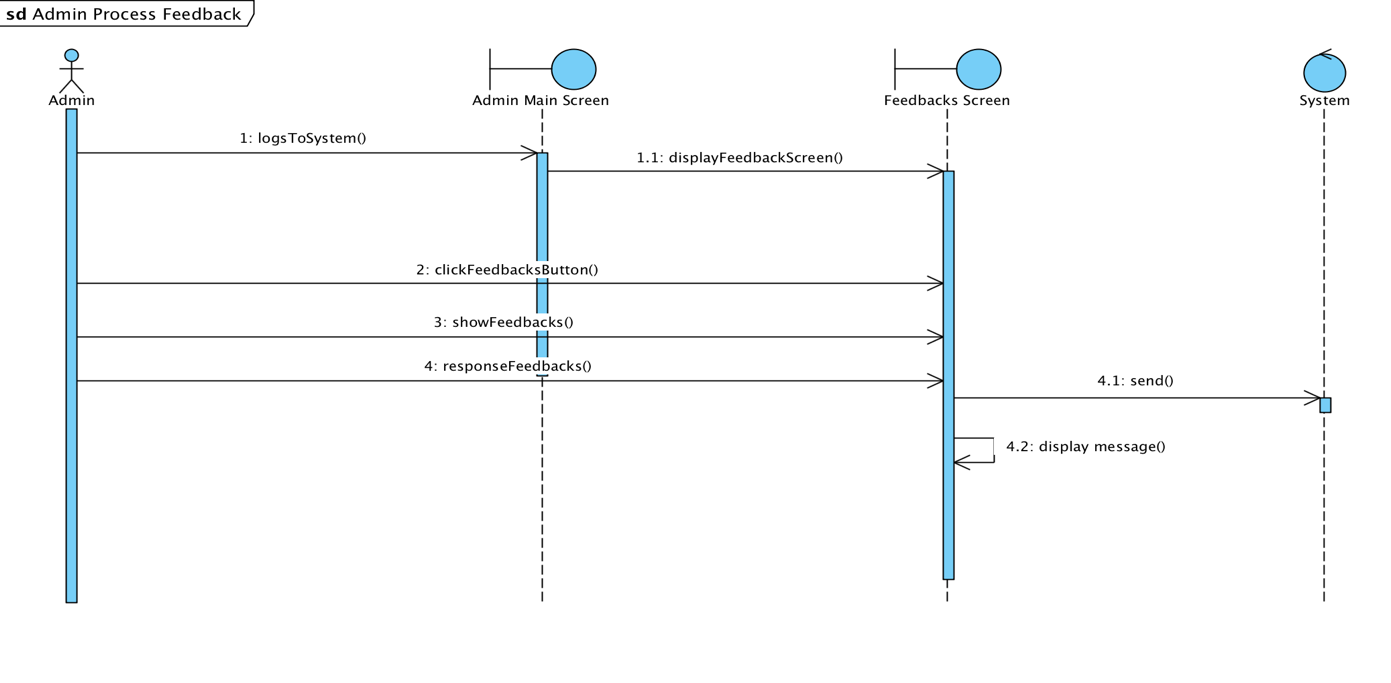
### Object model

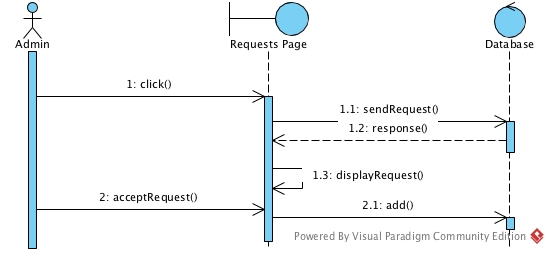


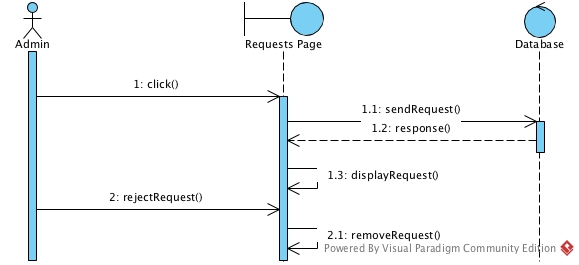
### Dynamic model

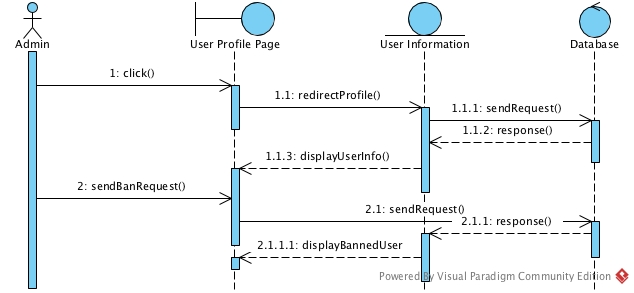
Admin Dynamic Model



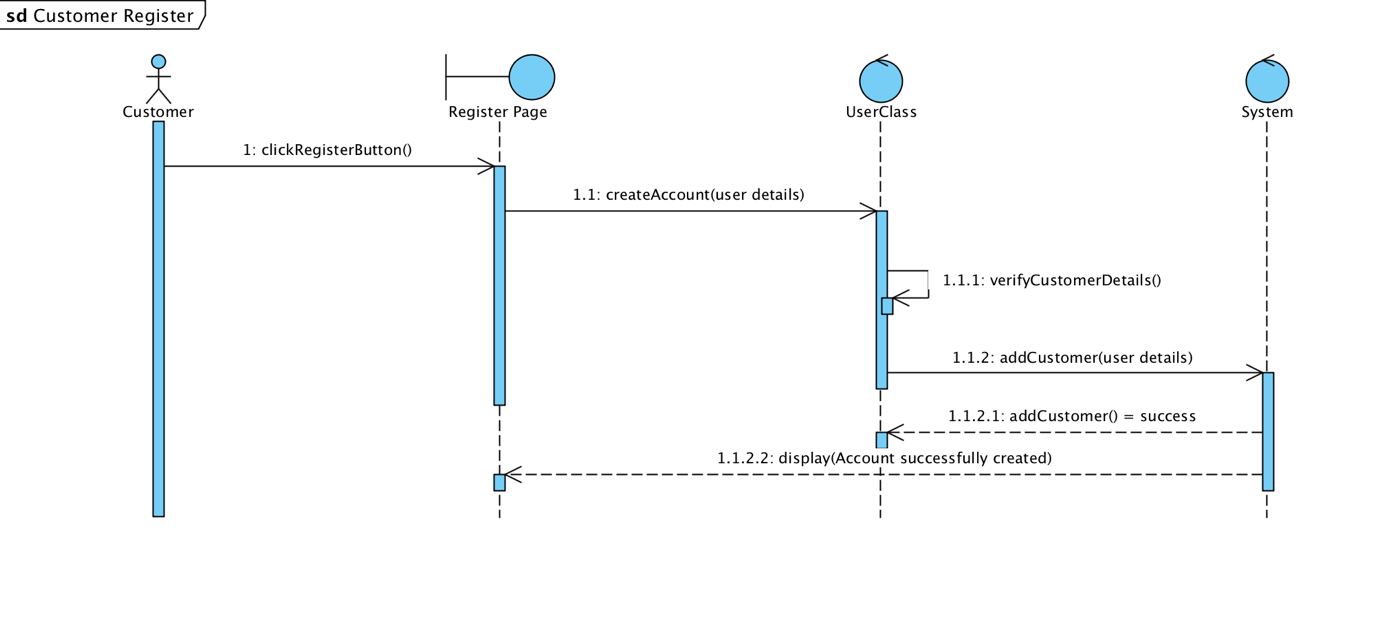


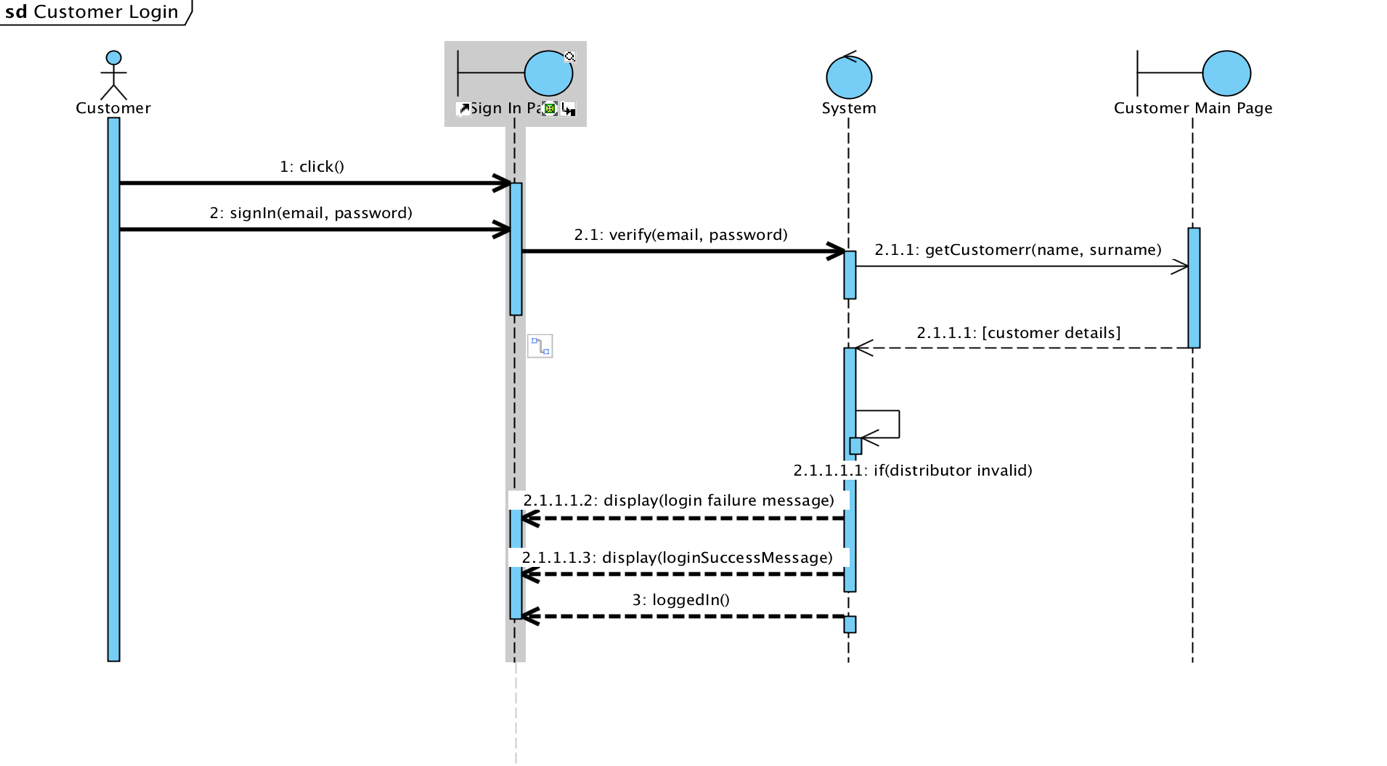


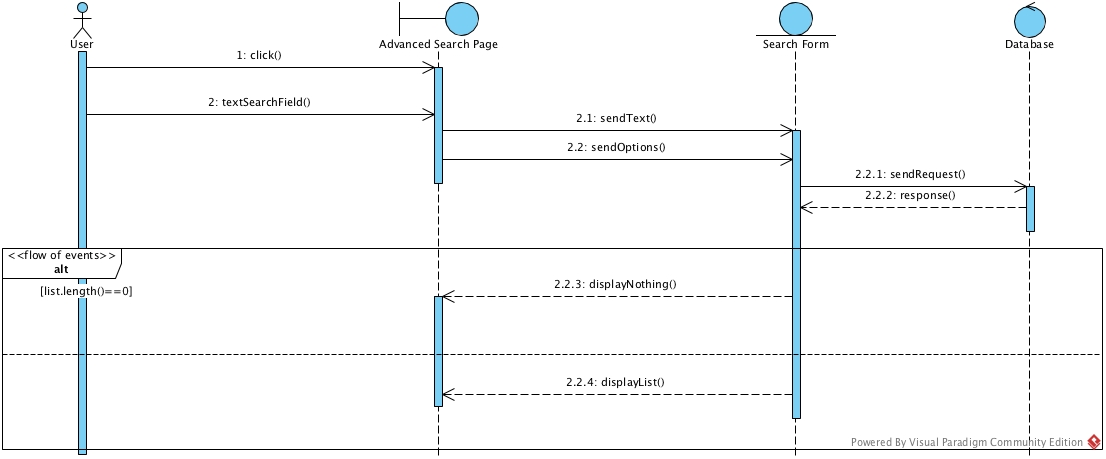


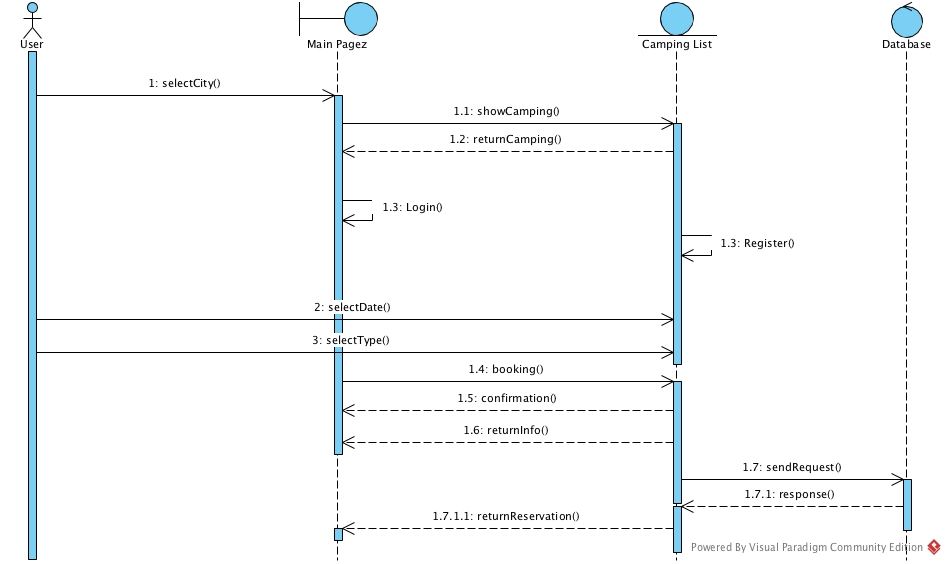


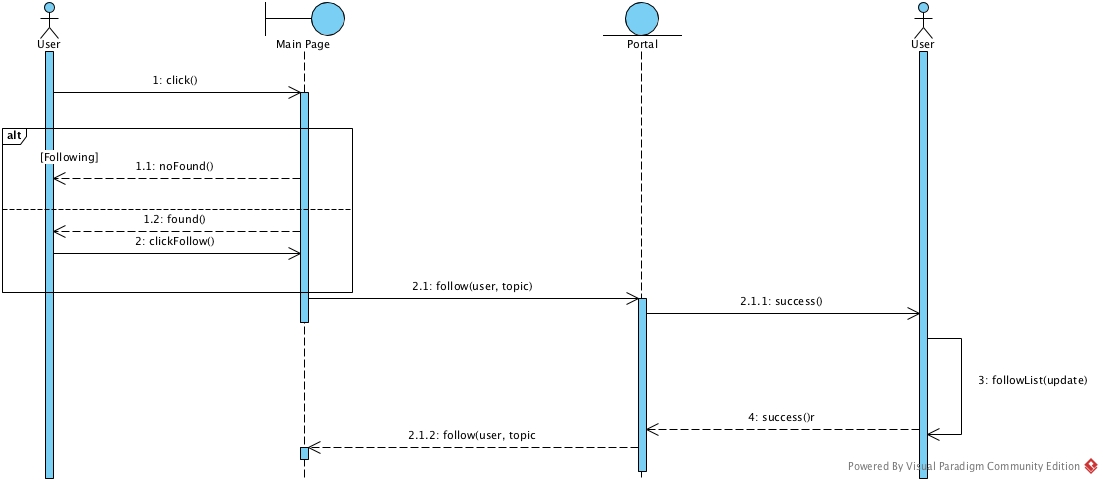
User Dynamic Model

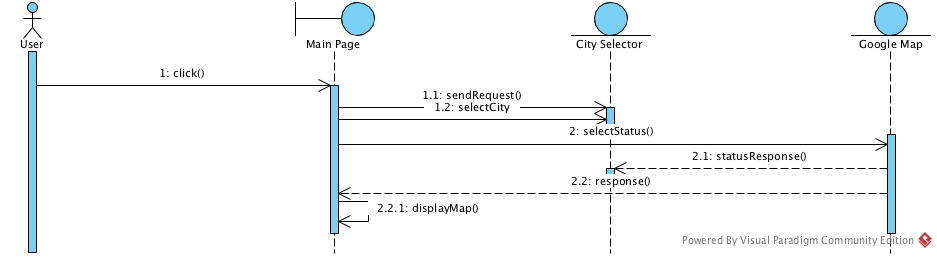


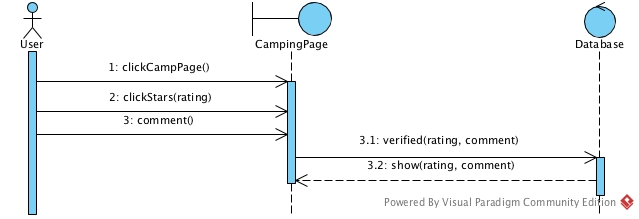




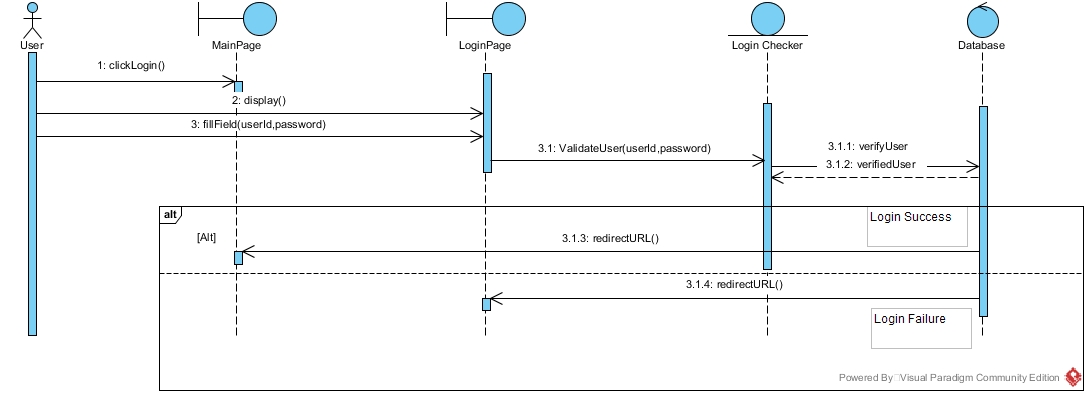


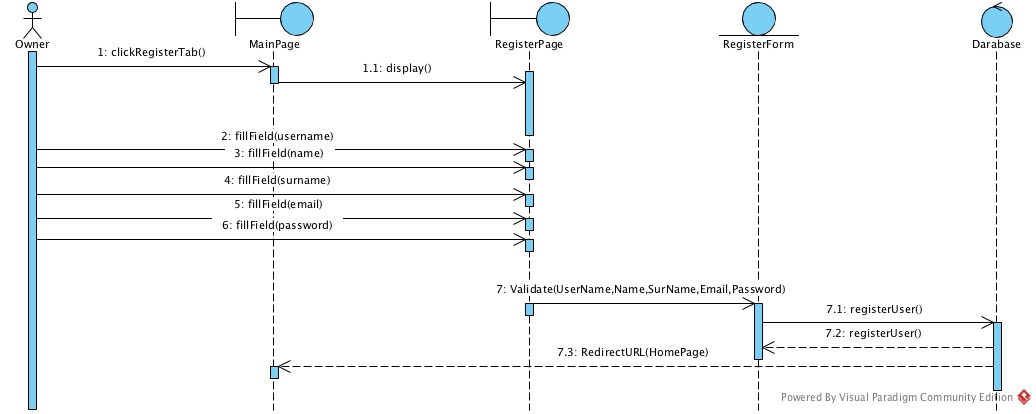


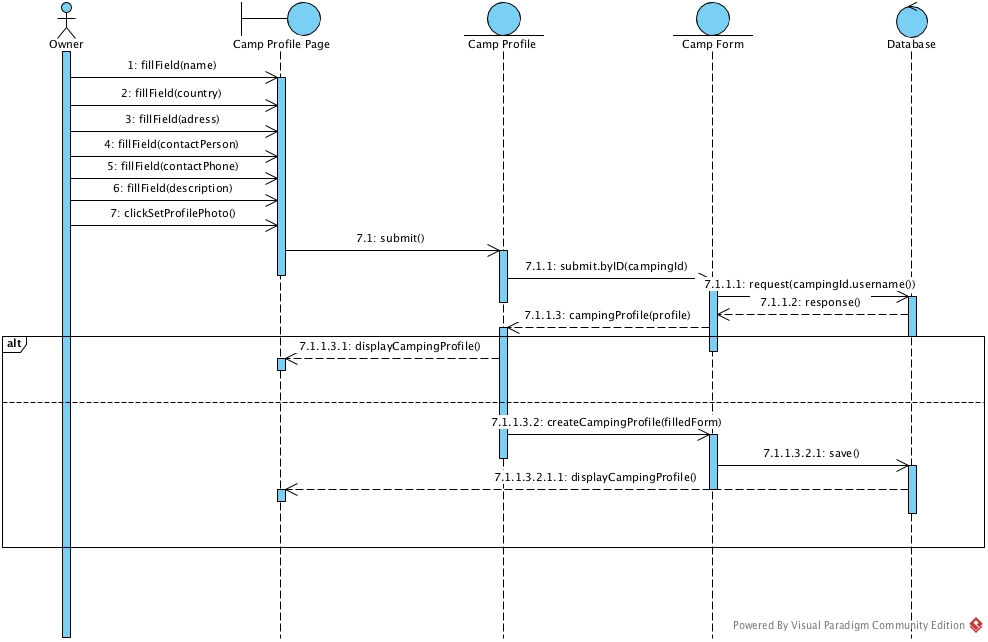


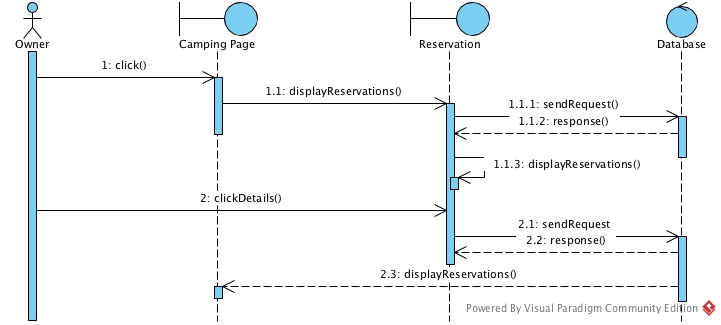


Owner Dynamic Model

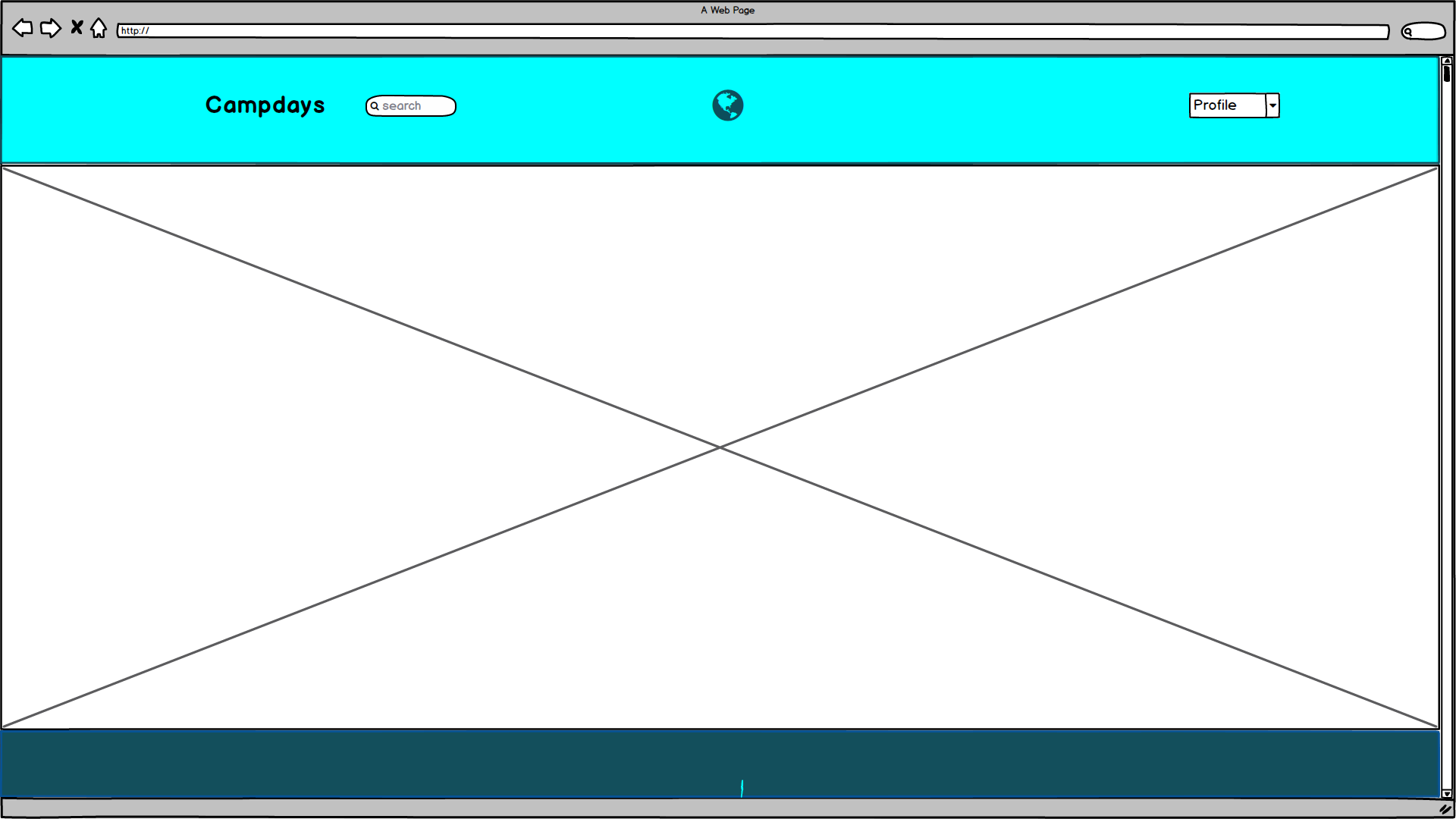


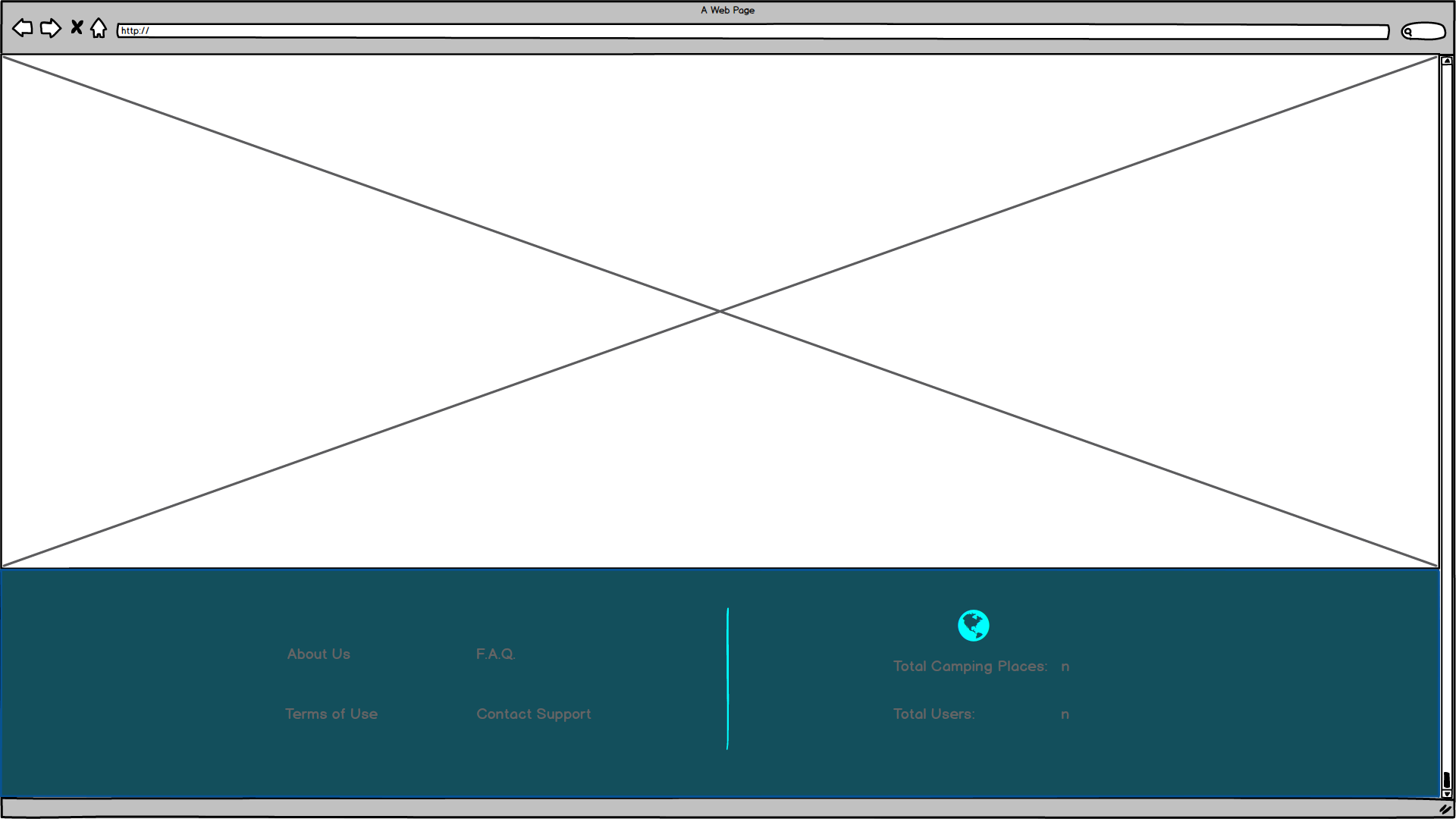


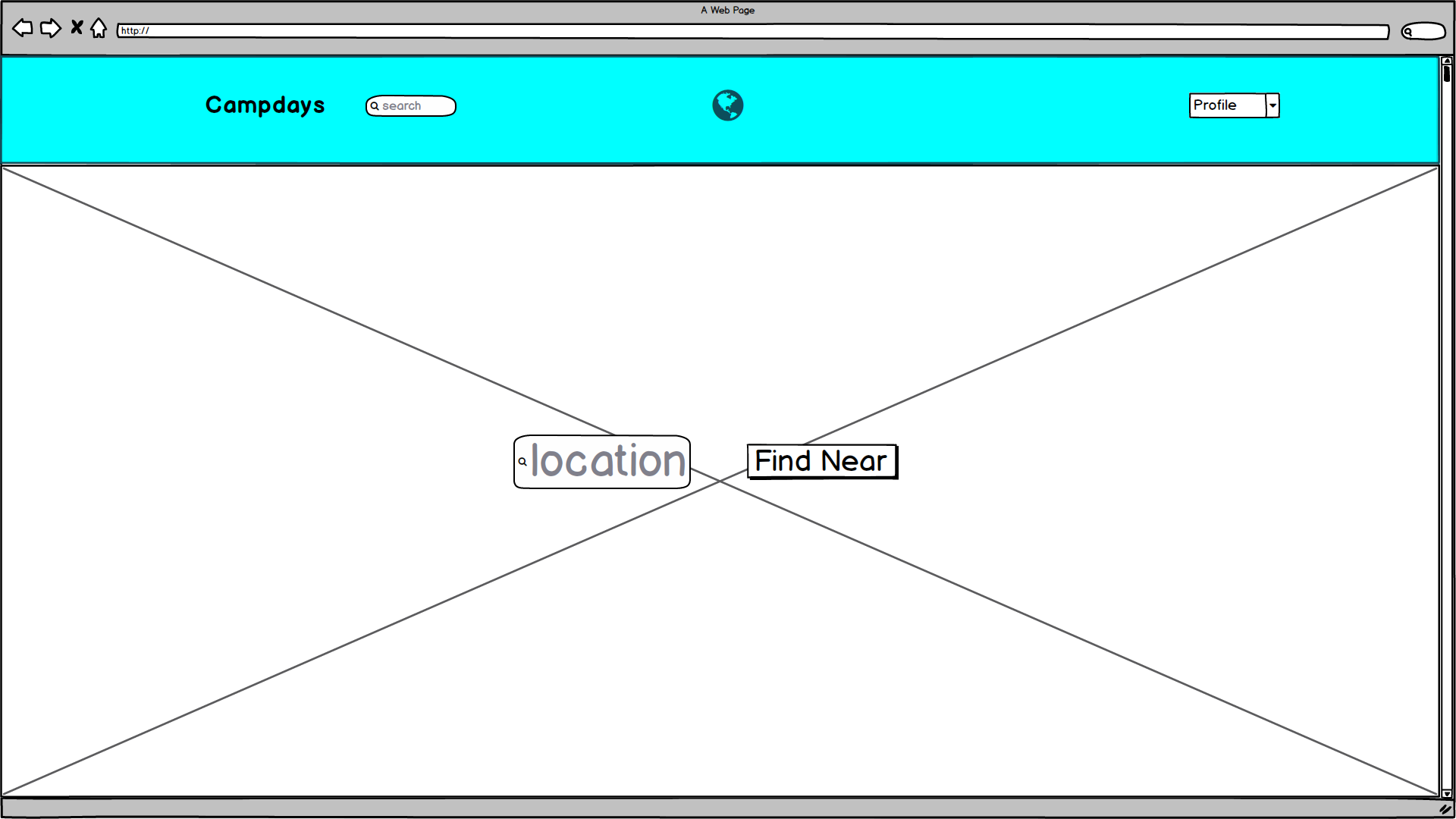


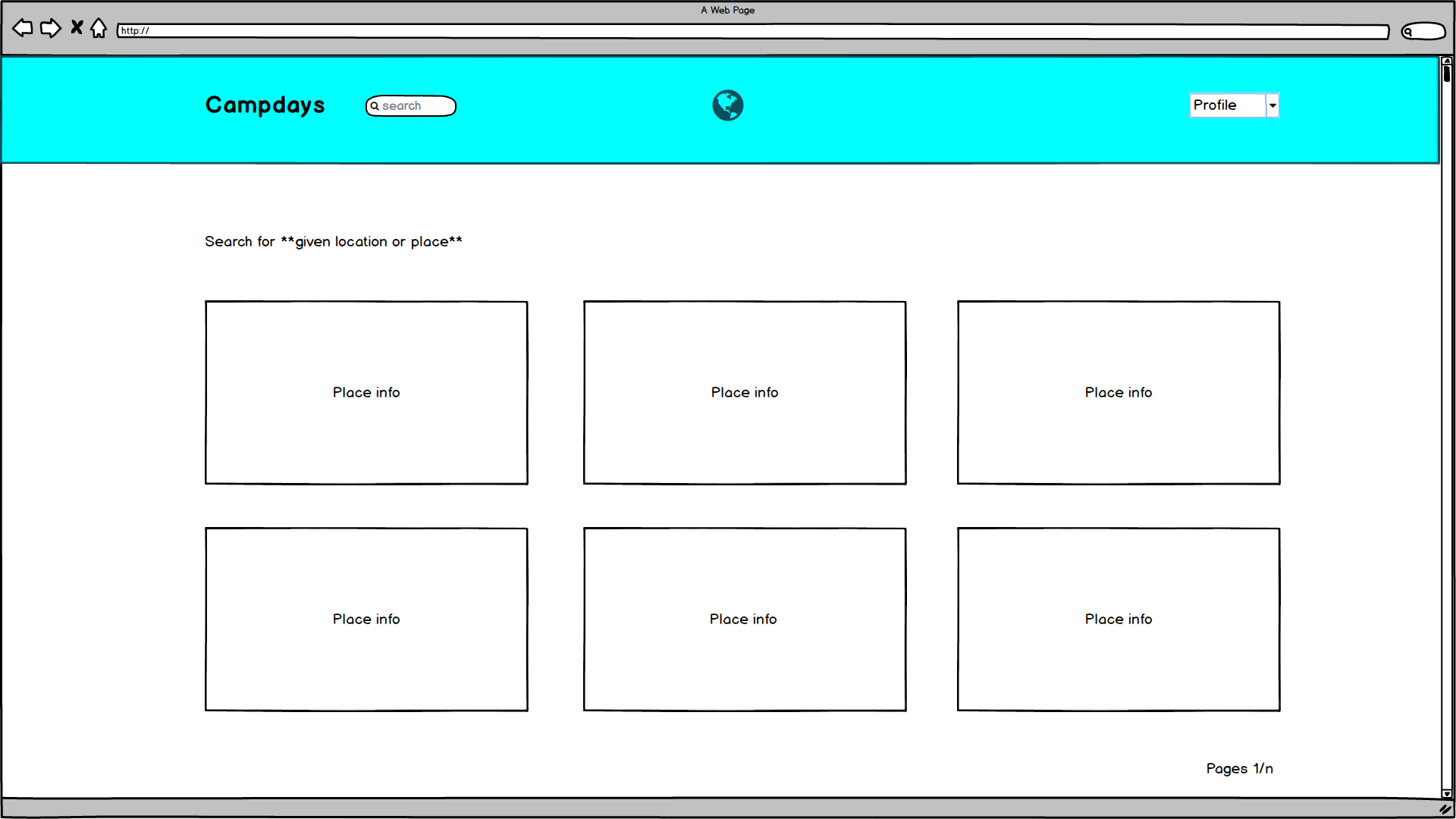


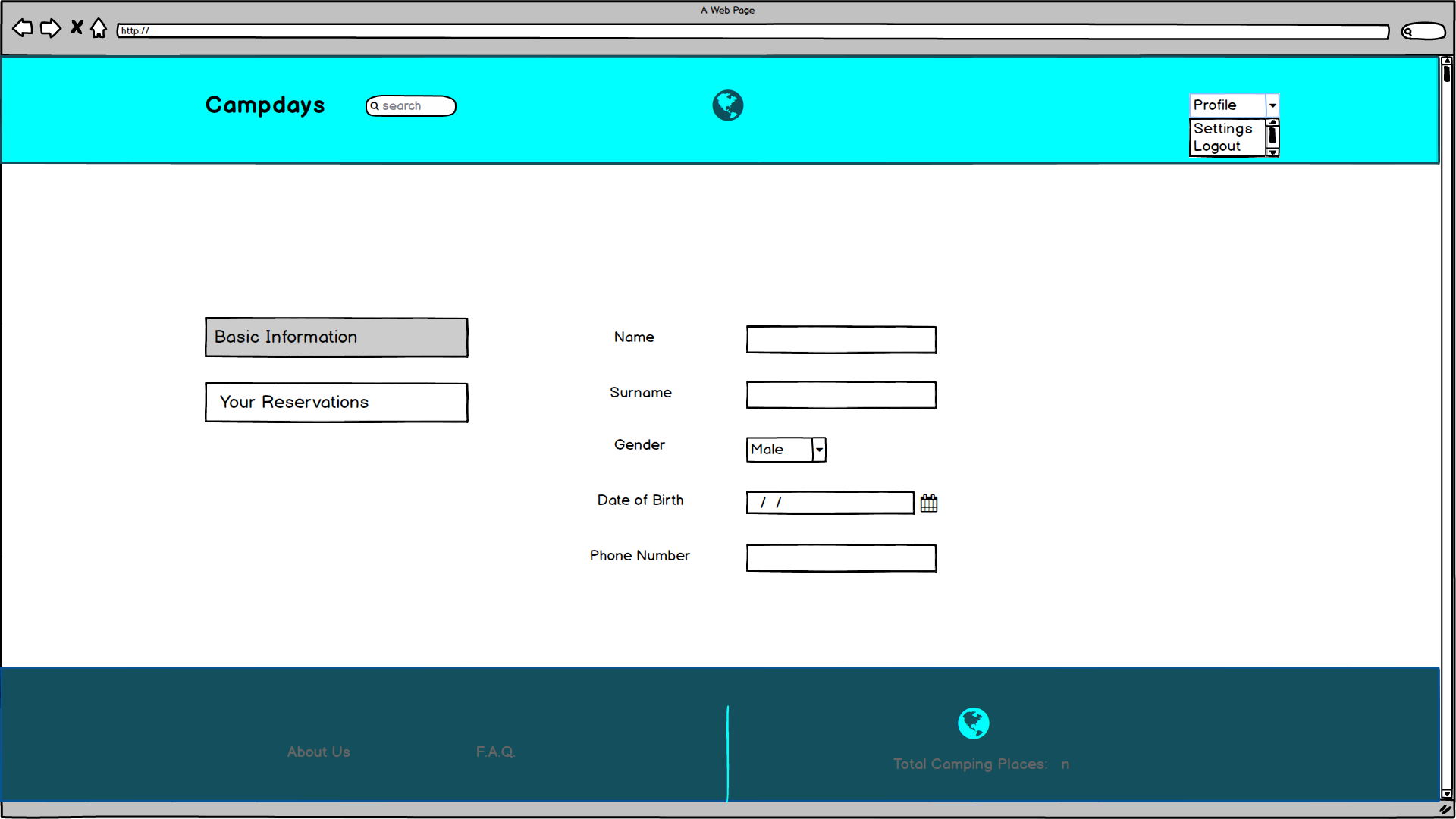
**User Interfaces**

****

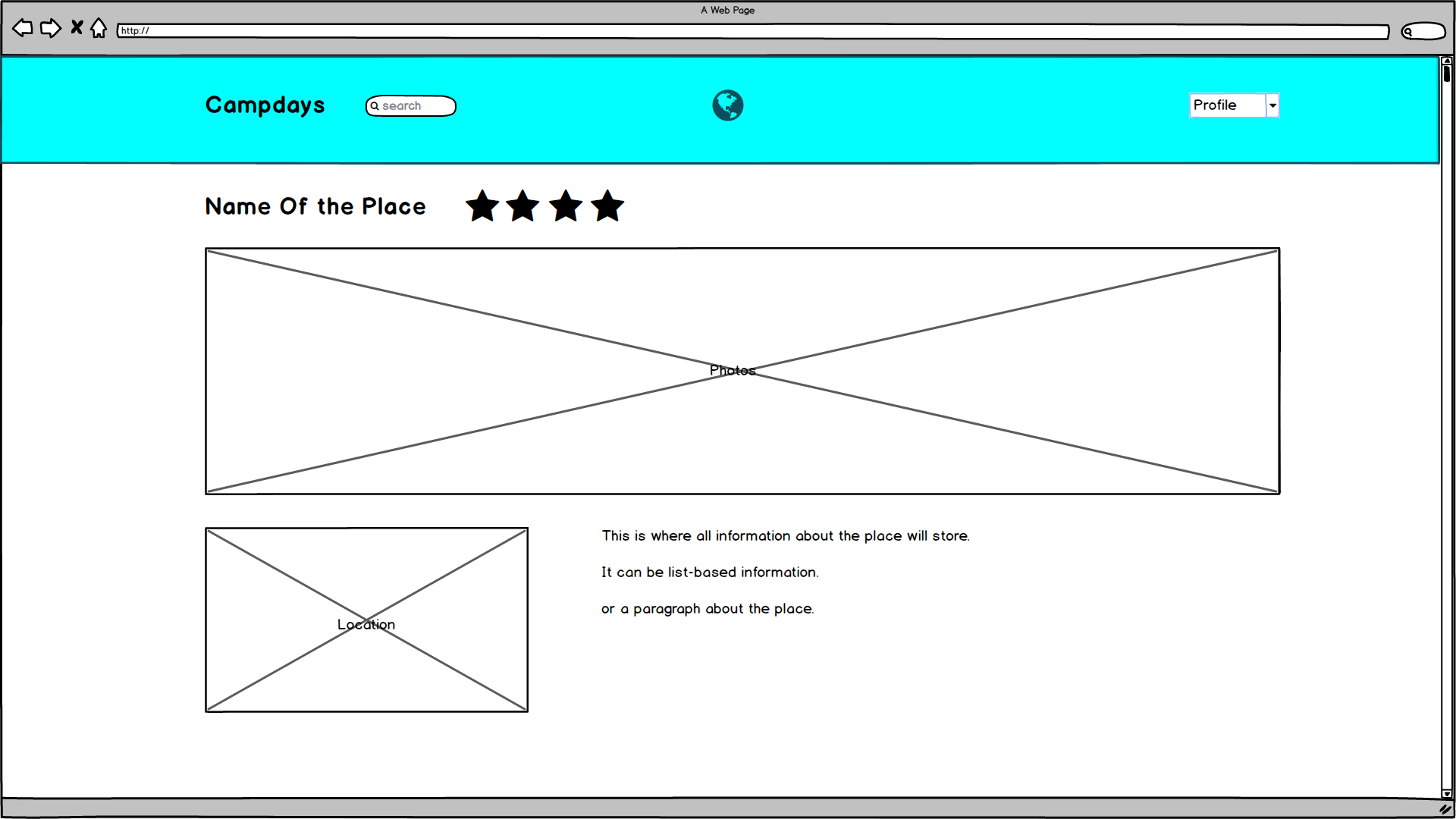
****

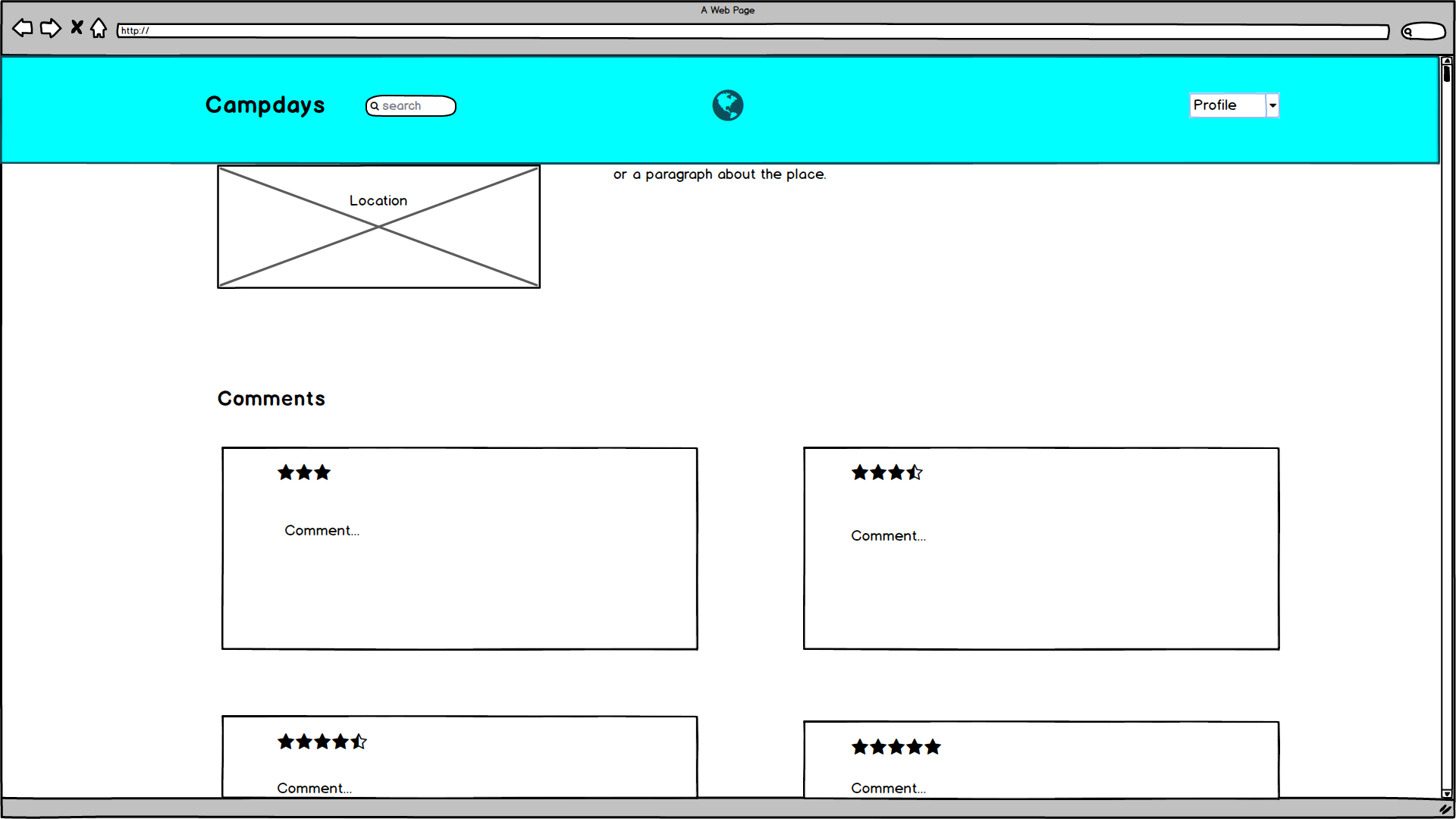
****

****

****

****

****

****

****

# Glossary

**User:** People who can make reservation, follow other users and places or comment or rate the places.

**Admin:** Person who have judgment in the system in order to prevent bad usage, spam and taking under the control owners and users.

**Owner:** Person who have camping place and taking reservations on the online system

**Booking**: Process that is provide to users make reservation

**Rating:** Scale for reliability and quality of the camping places by the users

**Set Statistics:** It gives information usage of places and users to administrator

**Search Near:** Process that provides to user optionally find camping places using their GPS

**Watch Comment:** This part is specialized for the Admin to prevent appropriate comments and spam with” delete” option and take action for spam and robot

**User Control:** Provides to Administrator to prevent robot users and bad using by deleting user or blocking

**Select Type:** Provides to user what kind of place or concepts they want to choose

# References

1. Bruegge B. & Dutoit A.H.. (2010). *Object-Oriented Software Engineering Using UML, Patterns, and Java*, Prentice Hall, 3rd ed.
2. <https://www.uml-diagrams.org/class-diagrams-overview.html>
3. <https://www.uml-diagrams.org/class-reference.html>
4. <https://www.ibm.com/developerworks/architecture/library/ar-analpat/> (IBM)