

Software Engineering
Software Requirements Specification
(SRS) Document

Cardinal

Project GitHub Repo:
<https://github.com/Alarman-prg/Cardinal>

5/1/2024

1.3

By: Jacob Greene, Christian Wilson, & Andrew Nice

[Our words and actions will reflect Academic Integrity.
We will not cheat or lie or steal in academic matters.
We will promote integrity in the UNCG community.]

Table of Contents

1.	Introduction	4
1.1.	Purpose	4
1.2.	Document Conventions	4
1.3.	Definitions, Acronyms, and Abbreviations	4
1.4.	Intended Audience	5
1.5.	Project Scope	5
1.6.	Technology Challenges	5
1.7.	References	5
2.	General Description	6
2.1.	Product Features	6
2.2.	User Class and Characteristics	6
2.3.	Operating Environment	6
2.4.	Constraints	6
2.5.	Assumptions and Dependencies	6
3.	Functional Requirements	6
3.1.	Primary	6
3.2.	Secondary	7
3.3.	Use-Case Model	7
3.3.1.	Use-Case Model Diagram	7
3.3.2.	Use-Case Model Descriptions	7
3.3.2.1.	Actor: General-User or Trail Guide (Christian Wilson)	8
3.3.2.2.	Actor: Park Operator (Andrew Nice)	8
3.3.2.3.	Actor: Web Admin (Jacob Greene)	8
3.3.3.	Use-Case Model Scenarios	8
3.3.3.1.	Actor: General-User or Trail Guide (Christian Wilson)	8
3.3.3.2.	Actor: Park Operator (Andrew Nice)	9
3.3.3.3.	Actor: Web Admin (Jacob Greene)	10
4.	Technical Requirements	11
4.1.1	Interface Requirements	11
4.1.1.1	User Interfaces	11
4.1.1.2	Hardware Interfaces	11
4.1.1.3	Communications Interfaces	11
4.1.1.4	Software Interfaces	11
5.	Non-Functional Requirements	12
5.1.	Performance Requirements	12
5.2.	Safety Requirements	12

5.3.	Security Requirements	12
5.4.	Software Quality Attributes	12
5.4.1.	Availability	12
5.4.2.	Correctness	12
5.4.3.	Maintainability	12
5.4.4.	Reusability	12
5.4.5.	Portability	12
5.5.	Process Requirements	12
5.5.1.	Development Process Used	12
5.5.2.	Time Constraints	12
5.5.3.	Cost and Delivery Date	13
5.6.	Other Requirements	13
6.	Design Documents	13
6.1.	Software Architecture	13
6.2.	High-Level Database Schema	13
6.3.	Software Design	14
6.3.1.	State Machine Diagram: General User or Trail Guide (Christian Wilson)	
14		
6.3.2.	State Machine Diagram: Park Operator (Andrew Nice)	14
6.3.3.	State Machine Diagram: Web Admin (Jacob Greene)	15
6.4.	UML Class Diagram	15
7.	Scenario	17
7.1.	Brief Written Scenario with Screenshots	17
7.1.1	General User Scenarios	18
7.1.2	Park Operator Scenarios	
7.1.3	Admin Scenarios	

1. Introduction

1.1. Purpose

The goal of Cardinal is for hikers to meet each other while helping them discover new trails in their county to explore. In order to achieve this goal, Cardinal allows users to communicate with other hikers to hike together, allows hikers to post group hikes, and shows updates at trails to help users decide where to explore.

1.2. Document Conventions

The purpose of this Software Requirements Document (SRD) is to describe the general-user, operators, and admin requirement for the Cardinal system. Client-oriented requirements describe the system from the client's perspective. These requirements include a description of the different types of users served by the system. Developer-oriented requirements describe the system from a software developer's perspective. These requirements include a detailed description of functional, data, performance, and other important requirements.

1.3. Definitions, Acronyms, and Abbreviations

- Java
 - A programming language originally developed by James Gosling at Sun Microsystems.
- H2
 - Open-source relational database management system.
- HTML
 - Hypertext Markup Language. This is the code that will be used to structure and design the web application and its content.
- CSS
 - Placeholder
- SpringBoot
 - An open-source Java-based framework used to create a micro Service. This will be used to create and run our application.
- MVC
 - Model-View-Controller. This is the architectural pattern that will be used to implement our system.
- Spring Web
 - Will be used to build our web application by using Spring MVC. This is one of the dependencies of our system.
- Thymeleaf

- A modern server-side Java template engine for our web environment. This is one of the dependencies of our system.
- NetBeans
 - An integrated development environment (IDE) for Java. This is where our system will be created.
- API
 - Application Programming Interface. This will be used to implement a function within the software where the current date and time is displayed on the homepage.

1.4. Intended Audience

The stakeholder(UNCG): 1-5

Project Manager(Ms. Ntini)

Project Developer (Christian Wilson): 3.3.2.1, 3.3.3.1

Project Developer (Andrew Nice): 3.3.2.2, 3.3.3.2

Project Developer (Jacob Greene): 3.3.2.3, 3.3.3.3

1.5. Project Scope

The goal of this software is to provide a platform where casual hikers can find other people through group hikes within their area, allow hikers who know their local trails to meet new hikers who are interested in having a mentor, provide lists of the parks/trails available to a user, as well as allow the local park/trail operators to provide updates and add new parks/trails once they are open to the public.

Benefits of the project include:

- Increased accessibility for the general public to enjoy the beauty of nature while bringing more engagement and therefore more use and knowledge to local parks/trails.
- Increasing safety and proper use of parks/trails through group activities and mentorship.
- Increasing the likelihood of more visitors to parks/trails.
- Allow park/trail operators to post announcements to potential visitors.

1.6. Technology Challenges

1.7. References

2. General Description

2.1. Product Features

For users, the product features include the ability to create and manage an account. This will let them find new trails as well as meet new people through joining and going to group hikes. Park operators will also be able to create an account. Upon approval and account creation park operators can upload and update trails. The web admin will be able to access code, generate status reports, and moderate accounts.

2.2. User Class and Characteristics

Our website application does not expect our users to have knowledge of the parks. Our website application assumes those who have been confirmed as park administrators have knowledge of the parks they create and have the authority to provide updates for those parks.

2.3. Operating Environment

The application is designed to operate on the web across many different devices.

2.4. Constraints

2.5. Assumptions and Dependencies

The software will be dependent on Spring Web, Spring Admin Tools, and Thymeleaf in order to create and execute the MVC architecture that will be developed with NetBeans. The application will use the Leaflet API(<https://leafletjs.com/>) for the interactive map of the Piedmont Region that will show the different counties and lead you to the page of the trails in a specified county. We plan to use a chat api for the chat features but yet to decide on one.

3. Functional Requirements

3.1. Primary

- FR0: The system will allow general users to look up local trails based on a county selection in the piedmont area of North Carolina

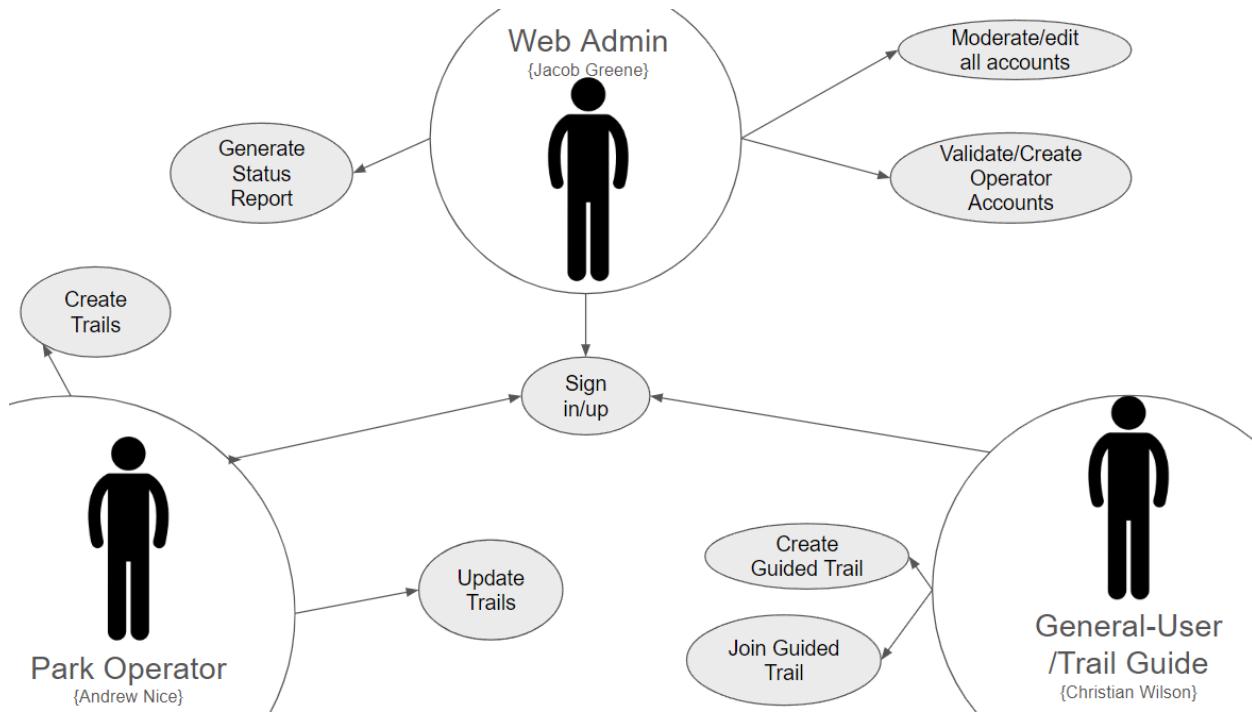
- FR1: The system will allow park operators to upload new trails and post descriptions and updates on those trails
- FR2: The system will allow users to create group hikes at specific trails.

3.2. Secondary

- Password protection for all users.
- Ability to display information looked up.

3.3. Use-Case Model

3.3.1. Use-Case Model Diagram



3.3.2. Use-Case Model Descriptions

3.3.2.1.

- Actor: General-user (Christian Wilson):
 - Sign In/Up: General Users can create an account and sign into said account.
 - Join Group Trail: The General Users can join group hikes set by other users.
 - Create Group Hike: The Users can set time slots to meet with other Users on group hikes through the group hike page.

3.3.2.2.

- Actor: Park Operator (Andrew Nice)
 - Sign In/Up: A Park Operator can create an account by filling out a form and can sign into that account.
 - Create Trail: The Park Operators can create new trails.
 - Update Trails: The Park Operators can update trails and change status of trails to show availability.

3.3.2.3.

- Actor: Web Admin (Jacob Greene)
 - Sign-in to admin account access admin page: The Web Admin is able to sign into their account.
 - Generate Reports: The Web Admin is able to generate a report on the statistics of the website. Reports could contain new users, returning users, list of trails, list of new trails (day, month year drilldown),
 - Moderate/edit all accounts: The Web Admin can delete accounts found to be problematic.
 - Validate/Create park operator accounts: The Web Admin can approve applications for people to become Park Operators after checking their validity with the people running the trail IRL.

3.3.3. Use-Case Model Scenarios

3.3.3.1. Actor: General User (Christian Wilson)

- **Use-Case Model Name: Sign In/Up**
- Initial Assumption: The General Users have an email and can access the website or have already created an account.
- Normal: A preexisting user will enter their information and login, or a new user will click a sign up option and enter an email and password.
- What Can Go Wrong: An user could lose their password or enter an incorrect email.
- Other Activities: A user could switch to the login page if they accidentally selected to sign up for a new account.
- System State on Completion: The User will have logged in and is redirected to the page to select a county.
- **Use-Case Model Name: Join Group Hikes**
- Initial Assumption: The General Users can navigate to specific page that will allow them to join posted hikes.
- Normal: After choosing a trail in their county, a User can join a Group hike which is shown on the group hike page.

- What Can Go Wrong: Users could create Group Hikes and not respond with details or not show up to the hike.
- Other Activities: Users would get information about who is leading the hike to initiate conversation.
- System State on Completion: The User is now attached to the Group Hike in the system, letting the User who made it know that another person intends to join their hike.

- **Use-Case Model Name: Create Group Hike**

- Initial Assumption: Any User who wishes to create group hike paste can navigate to the group hike page.
- Normal: A User navigates to the group page where they want to set up a post and clicks an option to start a group hike.
- What Can Go Wrong:
- Other Activities: N/A
- System State on Completion: A Group Hike is publicly displayed on the trail's description page where others can opt to join. Once the time is reached, the Group Hike is removed.

3.3.3.2. Actor: Park Operator (Andrew Nice)

- **Use-Case Model Name: Sign-in/up**

- Initial Assumption: The park operator will be able to create an account and will see that they must submit a form in order to be approved as a park operator.
- Normal: Park operators will sign up by choosing they'd like to create an account as a park operator. They will sign up by creating an email and password. Upon creation they will have to fill out a google form that will be reviewed by an admin. If approved they will be sent an email saying so, to which they will be able to log into Cardinal as a park operator.
- What Can Go Wrong: They forget their password and must reset it.
- Other Activities: N/A
- System State on Completion: The park operators account will be created and they'll be able to access the website.

- **Use-Case Model Name: Upload Trail**

- Initial Assumption: Park Operators are able to click on a button that lets them create a trail in a county of their choosing.
- Normal: The trail is created by entering the trail name, location, and images. Optionally they can add extra information about the trail.
- What Can Go Wrong: They upload something inappropriate.
- Other Activities: N/A
- System State on Completion: On the county page the new trail is added.

- **Use-Case Model Name: Update Trail**
- Initial Assumption: The park operator is able to modify the information on the trail.
- Normal: They'll click on the edit tool by the trail allowing them to edit the information about the trail.
- What Can Go Wrong:
- Other Activities: They can delete the trails.
- System State on Completion: The updated version of the trail will show in the specified counties page.

3.3.3.3. Actor: Web Admin (Jacob Greene)

- **Use-Case Model Name: Sign-in to admin account access admin page**
- Initial Assumption: User has credentials to input into username/password fields.
- Normal: Admin enters username/password.
- What Can Go Wrong: Admin can forget username/password. Unwanted users accessing admin accounts.
- Other Activities: N/A
- System State on Completion: User selects admin login from homepage sign-in, enters credentials and logs-in, admin is directed to the admin page that contains "tools".
- **Use-Case Model Name: Generate Reports**
- Initial Assumption: Admin account is logged into and currently on page of "tools".
- Normal: Select generate report and report type
- What Can Go Wrong: Admin could not understand how to generate reports.
- Other Activities: N/A
- System State on Completion: Report is printed to the browser window in designated area.
- **Use-Case Model Name: Moderate/Edit All Accounts**
- Initial Assumption: Logged into admin account and currently on page with admin "account tools".
- Normal: Admin enters page of user information with editing authorization, makes changes, saves.
- What Can Go Wrong: Wrong information is edited, changes are not saved, the page is too convoluted and packed with information to find the correct user accounts.
- Other Activities: Filter options?
- System State on Completion: Changes are saved and reflected in user information database.

- **Use-Case Model Name: Validate/Create park operator accounts**
- Initial Assumption: Logged into admin account and currently on page with “account tools”.
- Normal: Admin receives notice of new application for park operator, admin validates information, admin creates new park operator account and enters all information and completes account set-up.
- What Can Go Wrong: Incorrect park operator information entered, account creation failure, park operators lose account login information.
- Other Activities: N/A
- System State on Completion: New park operator account is created.

4. Technical Requirements

4.1. Interface Requirements

4.1.1. User Interfaces

The application will use Java controllers to receive data from HTML inputs.

4.1.2. Hardware Interfaces

The web application will run on any hardware device that has access to the internet, can display web pages, and has the ability to interact with web pages. While this does include things such as smartphones and tablets, the website is designed to work primarily on laptops and pc's so the layout might be different for mobile devices.

4.1.3. Communications Interfaces

It must be able to connect to the internet as well as the local database which will be determined in detail at a later date. The communication protocol HTTP must be able to connect with the Leaflet API to return the map of Piedmont North Carolina.

4.1.4. Software Interfaces

We will use React and Spring Boot ThymeLeaf to help build the frontend, as well as JPA for the backend database functionality. We will also use Spring Boot with Java to connect the frontend to the backend.

5. Non-Functional Requirements

5.1. Performance Requirements

5.2. Safety Requirements

- NFR0(R): 1 on 1 chat should only be seen by those in the chat itself.
- NFR1(R): Passwords should only be able to be accessed to change and recover lost passwords.

5.3. Security Requirements

- NFR2(R): The system will only be usable by authorized users.

5.4. Software Quality Attributes

5.4.1. Availability: Application should be able to be accessed at any time.

5.4.2. Correctness: Details about the Trails should be correct, and false information about said parks should be corrected.

5.4.3. Maintainability: As more trails are added, the website should be able to efficiently retain and maintain all trails.

5.4.4. Reusability: The application should be able to be used multiple times by many people.

5.4.5. Portability: The website should be able to be accessed by any device connected to the internet with an appropriate browser.

5.5 Process Requirements

5.5.1. Development Process Used

Waterfall Model

5.5.2. Time Constraints

Initial Prototype: 2/27/2024

Improved SRS Document: 4/2/2024

Final Prototype: 4/30/2024

Final Documentation: 5/1/2024

5.5.3. Cost and Delivery Date

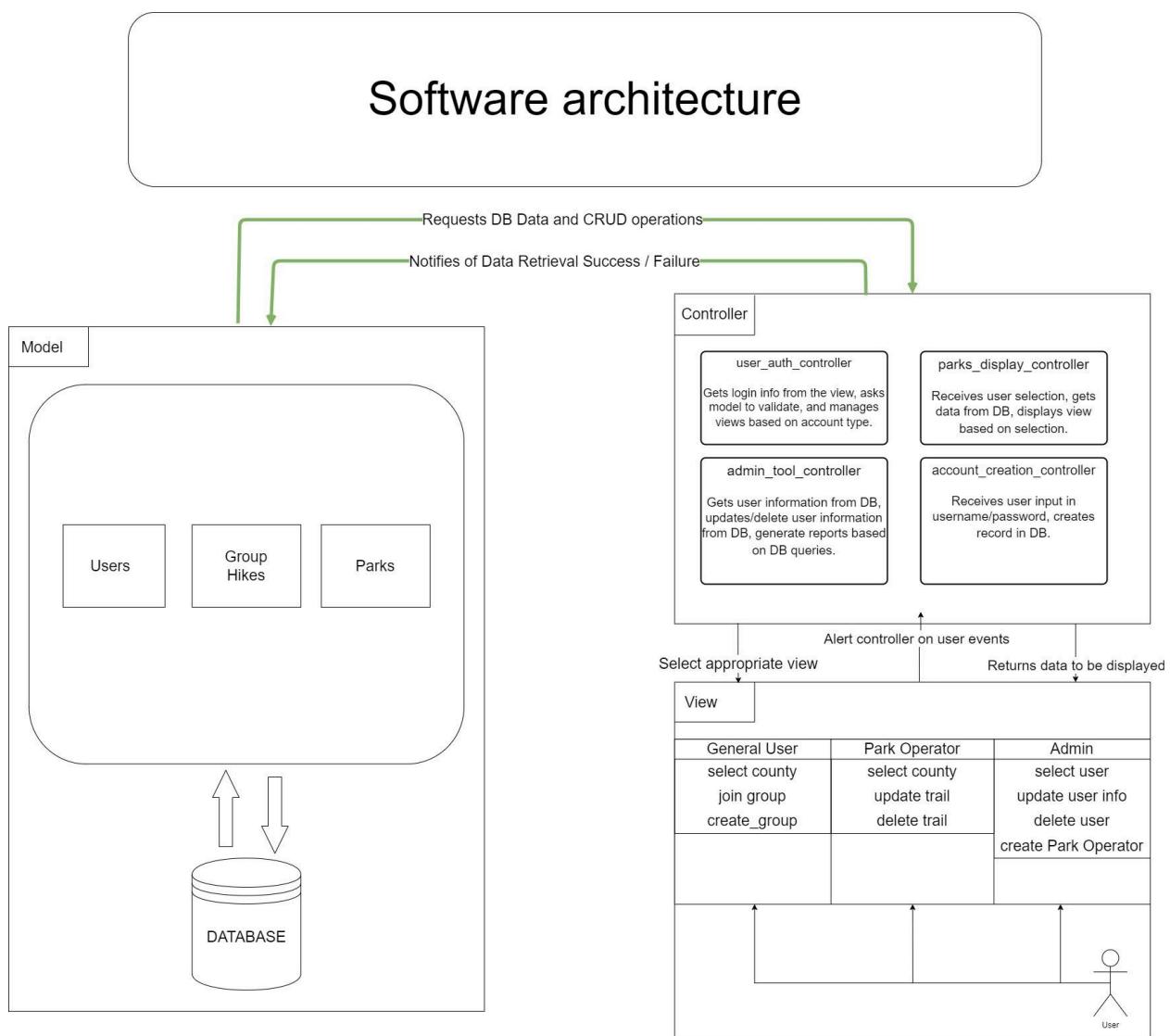
The cost will most likely be close to \$422(rough cost of the class) and the final draft will be submitted by Apr 30, 2024 .

5.6. Other Requirements

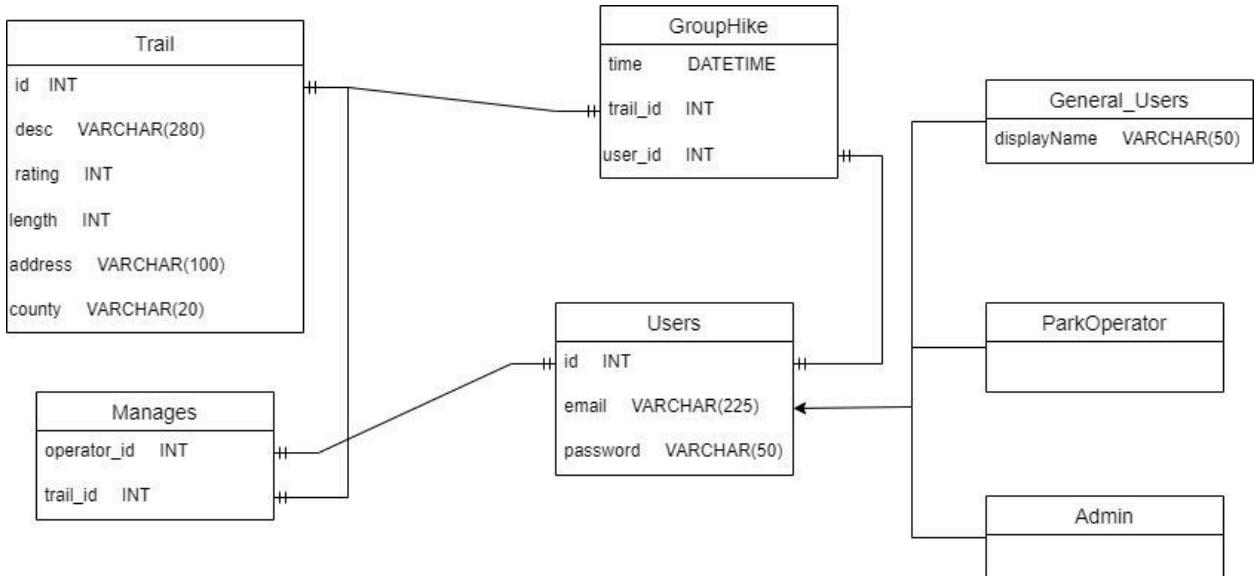
Continued direction from Project Manager about finishing the project (learning in class).

6. Design Documents

6.1. Software Architecture

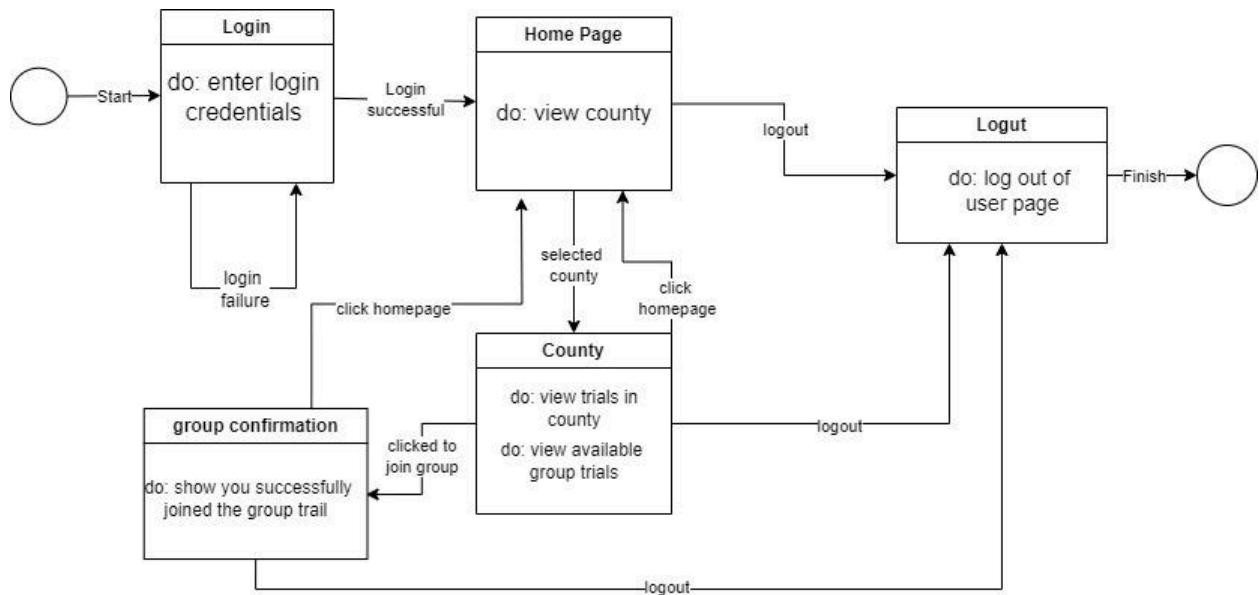


6.2. High-Level Database Schema

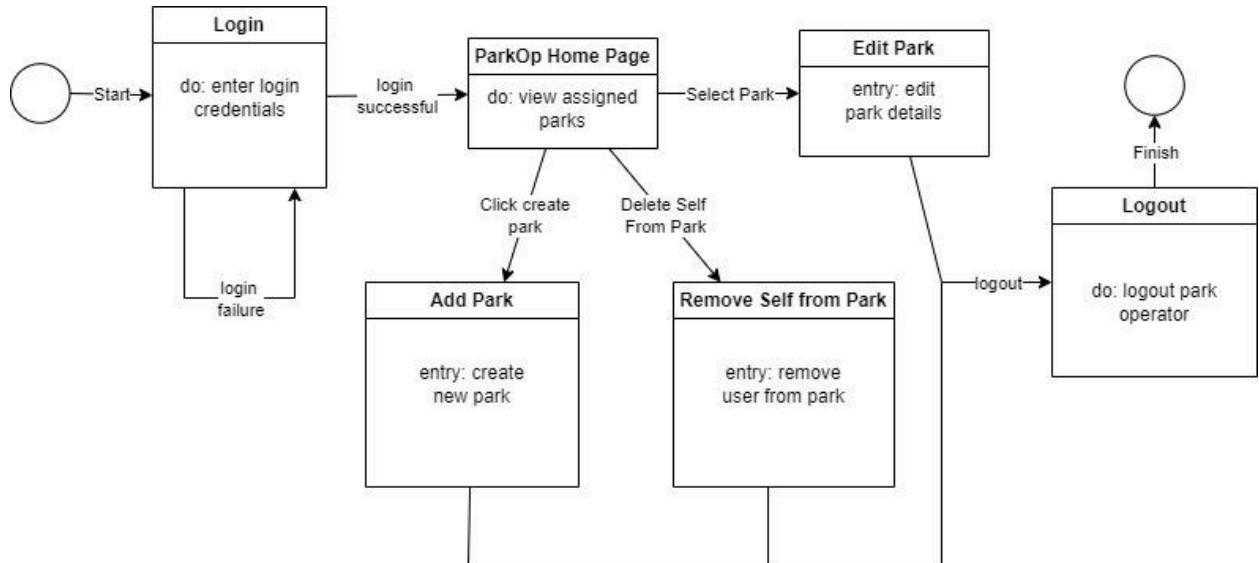


6.3. Software Design

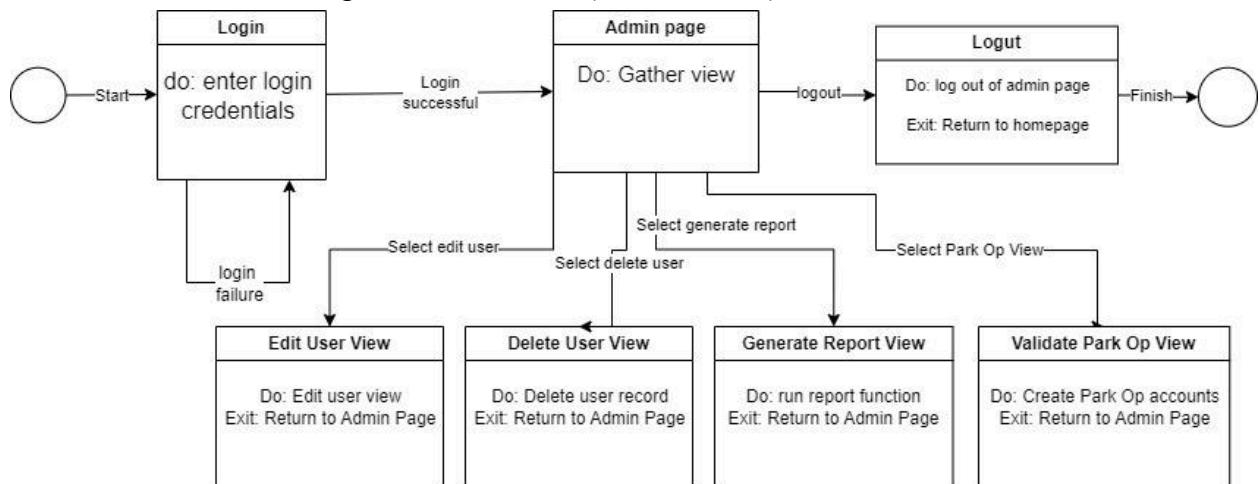
6.3.1. State Machine Diagram: General User (Christian Wilson)



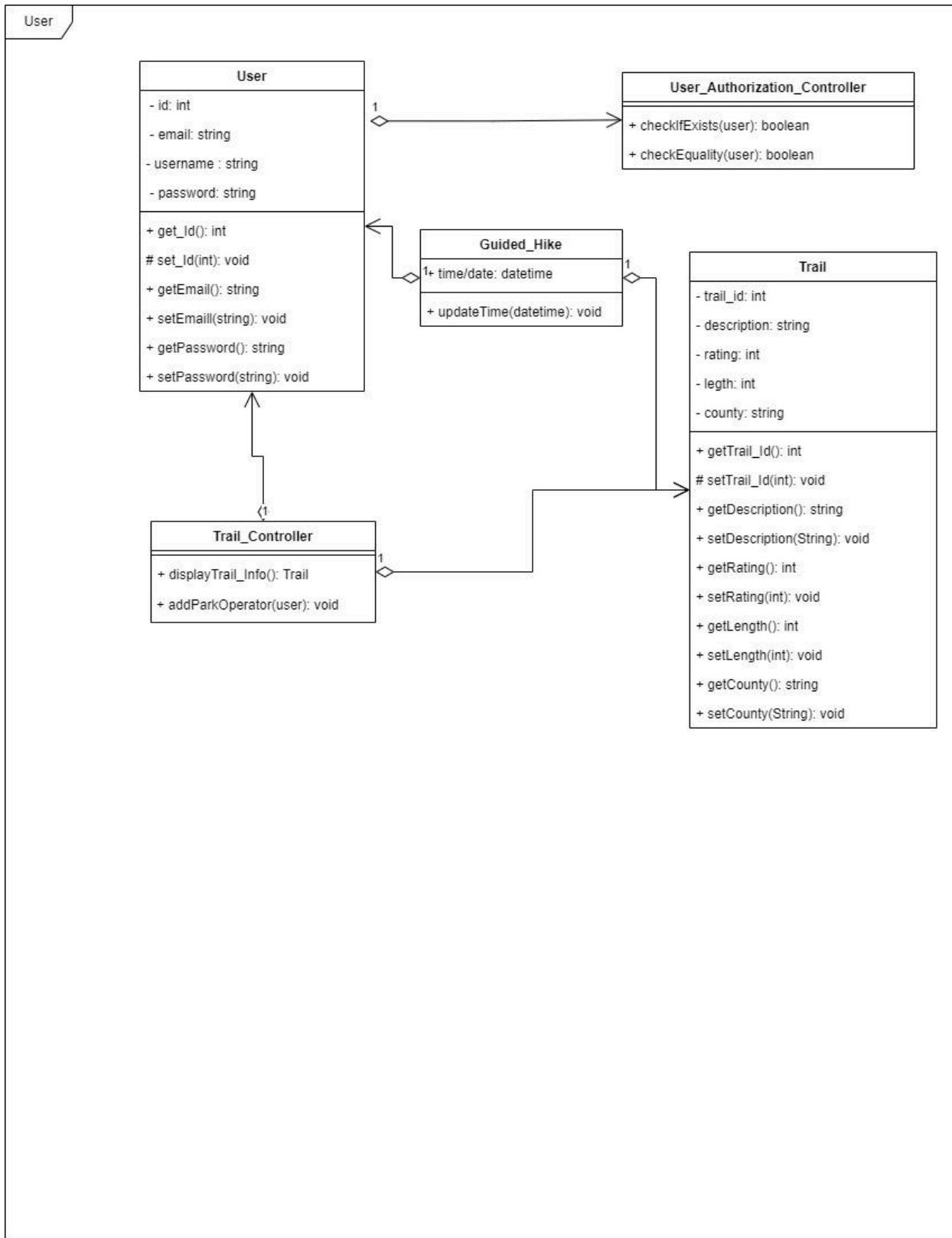
6.3.2. State Machine Diagram: Park Operator (Andrew Nice)

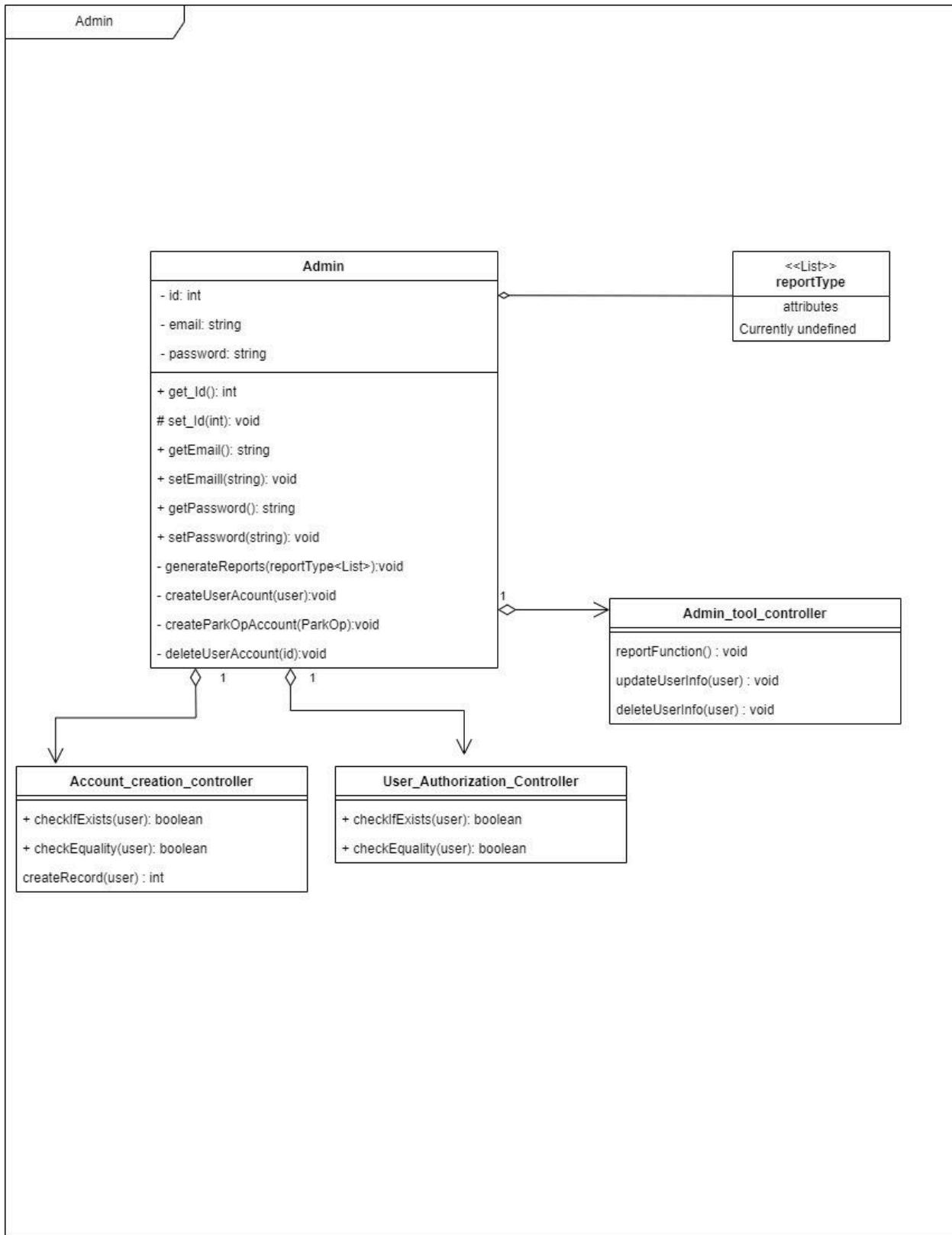


6.3.3. State Machine Diagram: Web Admin (Jacob Greene)



6.4. UML Class Diagram





7. Scenario

7.1. Brief Written Scenario with Screenshots

7.1.1 General User Scenario 1

7.1.1.1 Sign In:

Current user signs into their account(username: christian, password: password) and views the list of parks. They see one that they would like to create a post for and click “Group Hikes” on the navbar.

7.1.1.2 Create Group Hike:

On the page they will see the option to create a group hike post. There they will enter their full name, the name of the county and park, the time, and finally the number of people they would like to have there.

Create your group hike:

Full Name
Christian Wilson

Name of Trail
Cedar Rock Park

County

Time and Date of Hike
04/22/2024 at 10:00 am

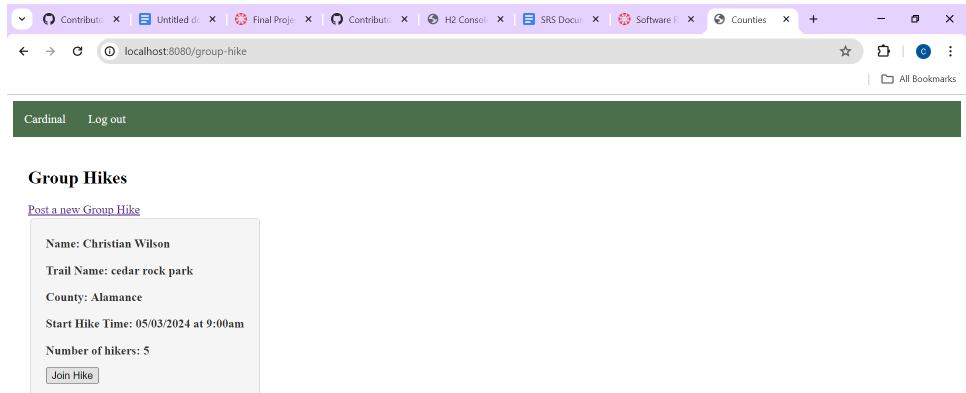
Number of people
5

7.1.1.2 General User Scenario 2

7.1.1.1.2 Sign Up/In:

A new user visits the site and creates a new account by clicking the register button. There they will enter their email, a username, password, first name, and last name. Then they will be able to log in with their new account.

7.1.1.3.2 Join Group Hike: Once logged in the user will press the group hikes tab to view the list of posted group hikes. When they see a hike they would like to join they will click on the “join hike” button(Non functional)



7.1.2 Park Operator Scenario (Andrew Nice)

7.1.2.1 Sign Up/In

A User would enter their information for username and password. This would redirect them to the park operator home page. They also could create a new account by clicking on a button that would lead them to a sign up sheet, where the date entered would be sent to the database.(Non Functional)

7.1.2.2 Create New Park

A park operator would scroll past the already existing parks and click the add park button which would redirect them to another page displaying inputs for park values. Upon clicking submit, once the page is reloaded it would display the new park.(Non Functional)

Create New Trail:

Trail Name

Description

Address

County

Park Operator

7.1.2.3 Edit Parks

A park operator could click a button next to each park to redirect to an editing page where, upon typing in the desired section for new content, only the filled parts would modify the park values, not changing the other parts.

Cardinal

My Parks

Name: Oak Hollow Park

Description: Oak Hollow Park is a spacious park featuring a lake, picnic areas, trails, and sports fields.

Address 1841 Eastchester Dr

County Guilford

Name: Tanglewood Park

Description: Tanglewood Park is a large recreational area with gardens, trails, golf courses, and a campground.

Address 4061 Clemmons Rd

County Forsyth

Name: Hagan Stone Park

Description: Hagan Stone Park offers hiking trails, fishing lakes, picnic areas, and camping facilities

Edit a park below :

Oak Filled Park

Park Description

Address 1841 Westchester

County

Park Operator

Cardinal

My Parks

- Name: Oak Filled Park

Description: A Bad Park

Address 1841 Eastchester Dr

County Guilford
- Name: Tanglewood Park

Description: Tanglewood Park is a large recreational area with gardens, trails, golf courses, and a campground.

Address 4061 Clemmons Rd

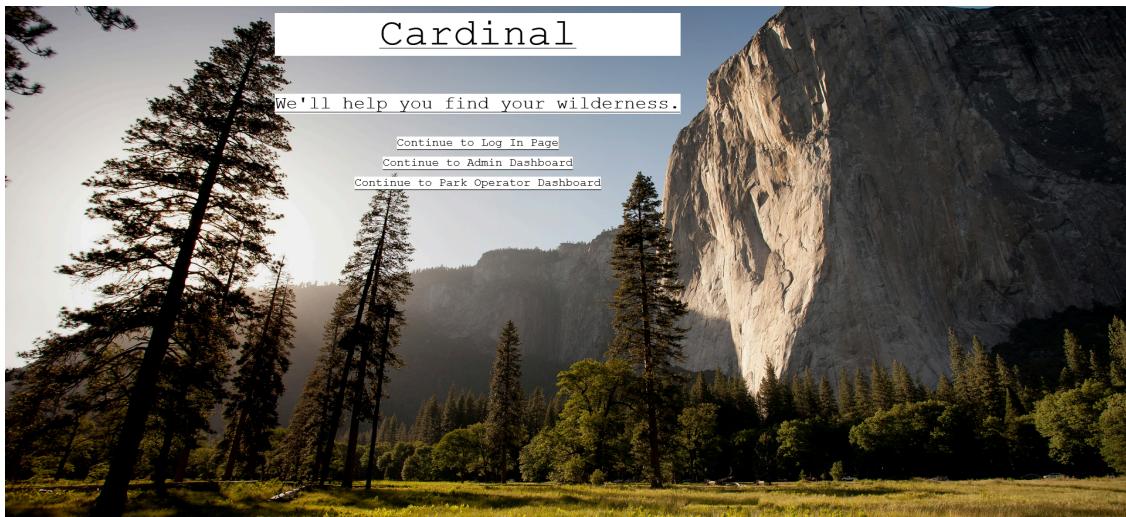
County Forsyth
- Name: Hagan Stone Park

Description: Hagan Stone Park offers hiking trails, fishing lakes, picnic areas, and camping facilities in a serene natural setting.

7.1.3 Admin Scenario

7.1.3.1 Sign in

Admin selects admin dashboard view from landing page `localhost:8080` or `localhost:8080/`.



7.1.3.2 Add user, View Users

Admin can view multiple tools from the admin dashboard. These include, creating a new user, deleting a user through username or database id, and a list of all user records in the user table through the table the admin can delete a record using the trashcan icon or enter the admin-edit view by clicking the pencil icon beside the wanted user record. The admin could also select admin-reports view from the top navbar. The admin could also logout and return to the landing. For the scenario, the admin will create a user, realize there was a mistake and click the edit icon beside that new user.

A screenshot of the Admin Dashboard. The top navigation bar includes "Admin Dashboard", "Admin Reports", and "Logout".

- Add a new user below :** A form with fields for Email address, Username, Password, First Name, and Last Name, each with an "Enter [field]" placeholder. A "Submit" button is at the bottom.
- Delete a user :** Two search forms: "By Username" with "Enter username" placeholder and "Submit" button; and "By Id" with "Enter Id" placeholder and "Submit" button.
- List Users**: A table with columns: Id, Email, Username, Password, First Name, and Last Name. It contains three rows of data:

ID	Email	Username	Password	First Name	Last Name
1	email@email.com	jacob	password	Jacob	Greene
2	email@email.com	andrew	password	Andrew	Nice
3	email@email.com	christian	password	Christian	Wilson

Each row has edit and delete icons in the last two columns.

7.1.3.3 Edit user

The admin could continue with the edit, logout, or go back to admin dashboard.

Admin will edit any of the fields and will be required to change the password.

After clicking submit on the form the view will return to the admin dashboard. Admin will then select generate reports.

A screenshot of a web application titled "Admin Dashboard". At the top right are links for "Logout" and "Admin Reports". The main content area has a dark green background. It displays a form titled "Edit a user below (You MUST create a new password) :". The form contains five input fields: "email@email.com", "jacob", "Password", "Jacob", and "Greene". Below the inputs is a blue "Submit" button.

7.1.3.2 Generate Report

The admin is presented with the total amount of users and the total amount of parks in the database. The admin could click the buttons to remove all records from either table or refresh the view. The admin could return to the admin dashboard or to the landing page.

Admin will select admin dashboard.

A screenshot of the Admin Dashboard. The top navigation bar includes "Admin Dashboard", "Admin Reports", and "Logout". The main content area shows two sections: "You have 3 different records for users!" and "You have 20 different records for parks!". Each section has a yellow "Refresh Records" button. Below each section is a yellow "DELETE ALL RECORDS" button.

7.1.3.3 Delete user

The admin will then delete the created user by either clicking the trashcan icon in list users or the delete a user form by username or Id. The admin will then click the logout button on the navbar to go back to the landing page.

The screenshot shows the Admin Dashboard interface. At the top, there's a navigation bar with 'Admin Dashboard' on the left and 'Admin Reports Logout' on the right. Below the navigation bar, the main content area is divided into three sections:

- Add a new user below :** This section contains fields for Email address, Username, Password, First Name, and Last Name, each with an 'Enter [field]' placeholder. A blue 'Submit' button is located at the bottom.
- Delete a user :** This section has two search options: 'By Username' with a 'Enter username' input field and 'Submit' button, and 'By Id' with a 'Enter Id' input field and 'Submit' button.
- List Users**: This section displays a table with the following data:

Id	Email	Username	Password	First Name	Last Name	Edit	Delete
1	email@email.com	jacob	password	Jacob	Greene		
2	email@email.com	andrew	password	Andrew	Nice		
3	email@email.com	christian	password	Christian	Wilson		