
***Software Engineering
Software Requirements Specification
(SRS) Document***

<Puzzle>

<11/29/2022>

<Version 1.0>

<By: Sofie Muska, Stuart Bridges, and Benjamin Woods>

**<We have abided by the UNCG Academic Integrity Policy on this
assignment.>**

Table of Contents

1. Introduction (Sofie Muska)	2
1.1 Purpose:	2
1.2 Document Conventions	2
1.3 Definitions, Acronyms, and Abbreviations	2
1.4 Intended Audience	4
1.5 Project Scope	4
1.6 Technology Challenges	4
1.7 References	4
2. General Description (Sofie Muska)	4
2.1 Product Perspective	4
2.2 Product Features	4
2.3 User Class and Characteristics	4
2.4 Operating Environment	4
2.5 Constraints	5
2.6 Assumptions and Dependencies	5
3. Functional Requirements (Sofie Muska)	5
3.1 Primary	5
3.2 Secondary	5
4. Technical Requirements (Benjamin Woods)	5
4.1 Operating System and Compatibility	5
The application will be compatible with any operating system that is able to view and to interact with traditional web pages.	5
4.2 Interface Requirements	5
We will use necessary internet protocols built into Spring Boot to connect users to our web applications.	6
5. Non-Functional Requirements (Stuart Bridges)	6
5.1 Performance Requirements	6
6. Use Case Model (Benjamin Woods)	8
7. Software Architecture (Stuart Bridges, Sofie Muska)	12
7.1 MVC Architecture OOP	12
8. Software Design (Sofie Muska, Benjamin Woods, Stuart Bridges)	13
8.1 State Diagrams	13
8.2 UML Class Diagram	17
9. Scenario (Benjamin Woods)	18

1. Introduction (Sofie Muska)

1.1 Purpose:

The goal of the Puzzle application is to enable rock climbers to effortlessly keep track of their climbing progress and to help them find the motivation they need to achieve their rock climbing goals by connecting them to the rock-climbing community through climbing gyms.

1.2 Document Conventions

The purpose of this Software Requirements Document (SRD) is to describe the climber view, the climbing gym view, and the administrator view for the Puzzle web application as well as the developer-oriented requirements. The climber-oriented requirements frame the web app from the climber's perspective, and the same applies, respectively, to the climbing gym requirements, to the administrator requirements, and to the developer requirements. The requirements describe the different types of climber, climbing gym, and administrative scenarios served by the application, and they include a detailed description of functional, data, performance, and other important requirements.

1.3 Definitions, Acronyms, and Abbreviations

1.3.1 Generic Acronyms

Term	Definition. Acronym, Abbreviation
SQL	A domain-specific language for managing relational databases.
MySQL	Open-source relational database management system
API	An application programming interface is a software intermediary that allows two applications to talk to each other.
Java	The programming language we will use to build the backend of this web application.
JPA	A Java Persistence API is a specification of Java that is used to access, manage, and persist data between Java objects and a relational database.
JavaScript	JavaScript is a scripting or programming language that allows the implementation of complex features on web pages.
HTML	The web development language we will use to build the frontend of this web application.

CSS	Cascading Style Sheets is a stylesheet language used to describe the presentation of a document written in HTML.
Spring Boot	A Java-based framework used to help create the standalone web application.
Inspirational Quotes API	An API that returns an array of inspirational quotes in JSON format.
CRUD	Refers to the four basic operations a software application should be able to perform – Create, Read, Update, and Delete . In such apps, users must be able to create data, have access to the data in the UI by reading the data, update or edit the data, and delete the data.
JDBC	Java database connectivity (JDBC) is the JavaSoft specification of a standard application programming interface (API) that allows Java programs to access database management systems.

1.3.1 Spring Boot Based Acronyms / Dependencies

Term	Definition. Acronym, Abbreviation
Spring Security	Spring Security is a powerful and highly customizable authentication and access-control framework. It is the de-facto standard for securing Spring-based applications.
Thymeleaf	Thymeleaf is modern server-side Java template engine for both web and standalone environments.
Spring Data JPA	Spring Data JPA, part of the larger Spring Data family, makes it easy to easily implement JPA based repositories. This module deals with enhanced support for JPA based data access layers. It makes it easier to build Spring-powered applications that use data access technologies.
Object Relational Mapping	ORM stands for Object-Relational Mapping (ORM) and is a programming technique for converting data between relational databases and object-oriented programming languages such as Java, C#, etc.
Hibernate	Java Object-relational Mapper (ORM) software used for converting data between relational databases and object-oriented programming (persistent framework ORM service)

1.4 Intended Audience

The Introduction, the General Description, and the Functional Requirements sections of this documentation are meant for all users to view, whereas the Technical and Non-Functional Requirements sections are meant for developers.

Project Stakeholders: developers, administrators, climbing gyms, and climbers.

1.5 Project Scope

Our goal is to make the tracking of climbing progress easier for climbers and to make it easier for climbing gyms and climbers to connect. This aligns with our overall business goals: to make everyday actions simpler and the path to connections between people clearer. This project will cement Caffeinated Foxes in the minds of climbers as individuals who seek to inspire them to become better versions of themselves through determination and consistent effort.

1.6 Technology Challenges

1. The product is in source code format with currently no executable format.
2. The product GUI does not resize to fit the user's screen.
3. The climber user cannot view a gym's route and event page due to search bar implementation issues.

1.7 References

N/A

2. General Description (Sofie Muska)

2.1 Product Perspective

Puzzle found its origins in a climber's desire for a simpler way to track their climbing progress. The idea was originated by a climber for climbers.

2.2 Product Features

The product features include the ability for individual climbers and climbing gyms to create accounts and the ability for administrators to manipulate those accounts. Climbers can also add climbing routes to their profiles, where they can track their climbing progress, with different tracking options based on climbing style. For gyms, the functionality also includes the ability to create climbing routes. They can also create events with a title and information. For administrators, the functionality also includes the possibility to delete and view accounts.

2.3 User Class and Characteristics

Our web application requires basic knowledge of how to interact with traditional web pages and a general understanding of rock climbing terminology.

2.4 Operating Environment

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

2.5 Constraints

Due to our introductory knowledge of project development and limited resources, we had to develop with a small selection of web environments in mind, which may have an undesirable effect on the user interface.

2.6 Assumptions and Dependencies

We assume that only climbers and those with climbing knowledge will be using the application. We also assume that climbers will likely be using the web application on a mobile device in an indoor rock-climbing gym while climbing gyms and administrators will likely be accessing the web application via laptop or desktop. The reliability of Spring Boot may affect the project.

3. Functional Requirements (Sofie Muska)

3.1 Primary

FR0: The system will allow the user to create a “climber” account.

FR1: The system will allow the user to create a “gym” account.

FR2: The system will allow the user to log into their account.

FR3: The system will allow a climber account to add a climbing route to their profile.

FR4: The system will allow the climber to track their climbing progress on a specific route. Progress tracking will be done by number of falls.

FR5: The system will store progress information in a database.

FR6: The system will display progress information to the climber in table format.

FR7: The system will allow the climber to input their number of falls or climbing time into a table and to select the date on which they attempted the route in the table’s date column.

FR11: The system will allow the climbing gym to update their homepage with event information.

FR12: The system will allow the climbing gym to add climbing routes to their catalog.

FR13: The system will allow the climbing gym to remove climbing routes from their catalog.

FR14: The system will allow administrators to view a list of user accounts.

FR15: The system will allow administrators to delete user accounts.

3.2 Secondary

N/A

4. Technical Requirements (Benjamin Woods)

4.1 Operating System and Compatibility

The application will be compatible with any operating system that is able to view and to interact with traditional web pages.

4.2 Interface Requirements

4.2.1 User Interfaces

The first screen will be the login/register home screen, where there will be two buttons—for the user to either register or to log in—and a short description of the web app, as well as an inspirational quote from the Inspirational Quotes API.

If the user hits the “Register” button, they will be taken to a page where they will fill out their information—name, email address, type (climber or gym), and password. The email address will function as their username. Once they fill out their information, they will click on the “Create Account” button to create their account.

If the user hits the “Log in” button, they will be taken to a screen where they will see a box with two fields to fill out, “Username” and “Password.” Once they fill out the fields, they will hit the “Log in” button under the “Password” field, which will take them to their homepage. The layout of their homepage will depend on the type of user they are.

The “administrator” homepage will include a list of user accounts. Next to each account, there will be a trash can symbol. Clicking this symbol will allow the administrator to delete the specific user account.

The “climbing gym” main page will be the gym routes page. Here, the climbing gym will see a list of its current routes. Clicking on a route name will show route info, a delete button, and a modify button. Above the list of routes, there will be a “Create New Route” button, which will allow the climbing gym to add a new route to their catalog. The gym will be able to name the route and to add relevant information below. When it is done creating the route, the gym will hit the “Create” button to post the new route. There will also be an events tab with an “Events” field, where the gym can inform climbers of upcoming events and other happenings. Clicking the “Post” button will post the message. The “climbing gym” homepage will also include a “Climbing Routes” tab.

The “climber” homepage will display a history of the climber’s attempted routes. Clicking on the name of the route will take the climber to a new page, where they will see the name of the route. Beneath the name of the route will be the route information. Finally, there will be a table. This table will have two columns: “Date” and either “Number of Falls” or “Time.” Above the table will be a button with the option to “Add New Attempt.” Clicking this button will add a new row to the table, where the climber will be able to input the relevant information.

The “climber” homepage will also include another tab, called “Climbing Gyms.” Here, the climber will see a list of the available gyms.

The style guides used will be IDE dictated, specifically the IntelliJ and Netbeans style guides.

4.2.2 Hardware Interfaces

The web application will run on any hardware device that has access to the internet, the ability to display webpages, and the ability to interact with web pages. This includes, but is not limited to, smartphones, tablets, desktop computers, and laptops.

4.2.3 Communications Interfaces

We will use necessary internet protocols built into Spring Boot to connect users to our web applications.

4.2.4 Software Interfaces

We will use React and Spring Boot 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 (Stuart Bridges)

5.1 Performance Requirements

NFR0(R): The user will be able to create a “climber” account in under 3 minutes.

NFR1(R): The user will be able to create a “gym” account in under 3 minutes.
 NFR2(R): The user will be able to log into their account in under 1 minute.
 NFR3(R): A “climber” user will be able to add a route to their profile in under 3 minutes.
 NFR4(R): A “climber” user will be able to view their climbing progress in under 1 minute.
 NFR5(R): The local copy of the climbing progress database will consume less than 20 MB of memory.
 NFR6(R): The table and graph will consume less than 20 MB of memory.
 NFR7(R): A “climber” user will be able to input their progress data into the route progress table in under 1 minute.
 NFR8(R): A “climbing gym” user will be able to add a new route to their route catalog in under 5 minutes.
 NFR9(R): A “climbing gym” user will be able to remove a route from their route catalog in under 1 minute.
 NFR10(R): The local copy of the list of user accounts will take up less than 50 MB of memory.
 NFR11(R): An administrator will be able to delete a user account in under 1 minute.

5.1 Safety requirements

Do not allow any user, besides the admin, to access another user’s personal data such as their password or email.

5.2 Security requirements

NFR12(R): The system will only be usable by users that have an account.

5.3 Software quality attributes

- 5.4.1. Availability
Should be available to any web users with access to a web browser.
- 5.4.2. Correctness
The code should adhere to our specified style guides and pass the number of unit and system tests we create.
- 5.4.3. Maintainability
The code and system should be written and developed with developers in mind, who may or may not be a part of the original team. The functions and web app behavior should leave clear space for updates.
- 5.4.4. Reusability
Users should have all their data stored when they log in and should not have to worry about any of that data being lost. The code should also be readable to developers who have never seen it.
- 5.4.5. Portability
The project being built on the web should make it extremely portable and accessible to nearly anyone who wants to use it. Using Java for the backend should also increase the portability of the software.

Detailing on the additional qualities that need to be incorporated within the software:

1. Dependency Inversion (part of the SOLID principles)
2. Adherence to the Style Guide
3. Proper use of Enumeration
4. MVC Architecture
5. Separation of Concerns
6. Loose Coupling

5.4 Process Requirements

- 5.5.1. Development Process Used: The project is built on the incremental development model with specification, development, and validation activities interleaved (with feedback).
- 5.5.2. Time Constraints: The overall time constraint is the UNCG Fall Semester 2022, and implementation/development of the final project is constrained to November 22, 2022.
- 5.5.3. Cost and Delivery Date: There is no associated cost with development of Puzzle.

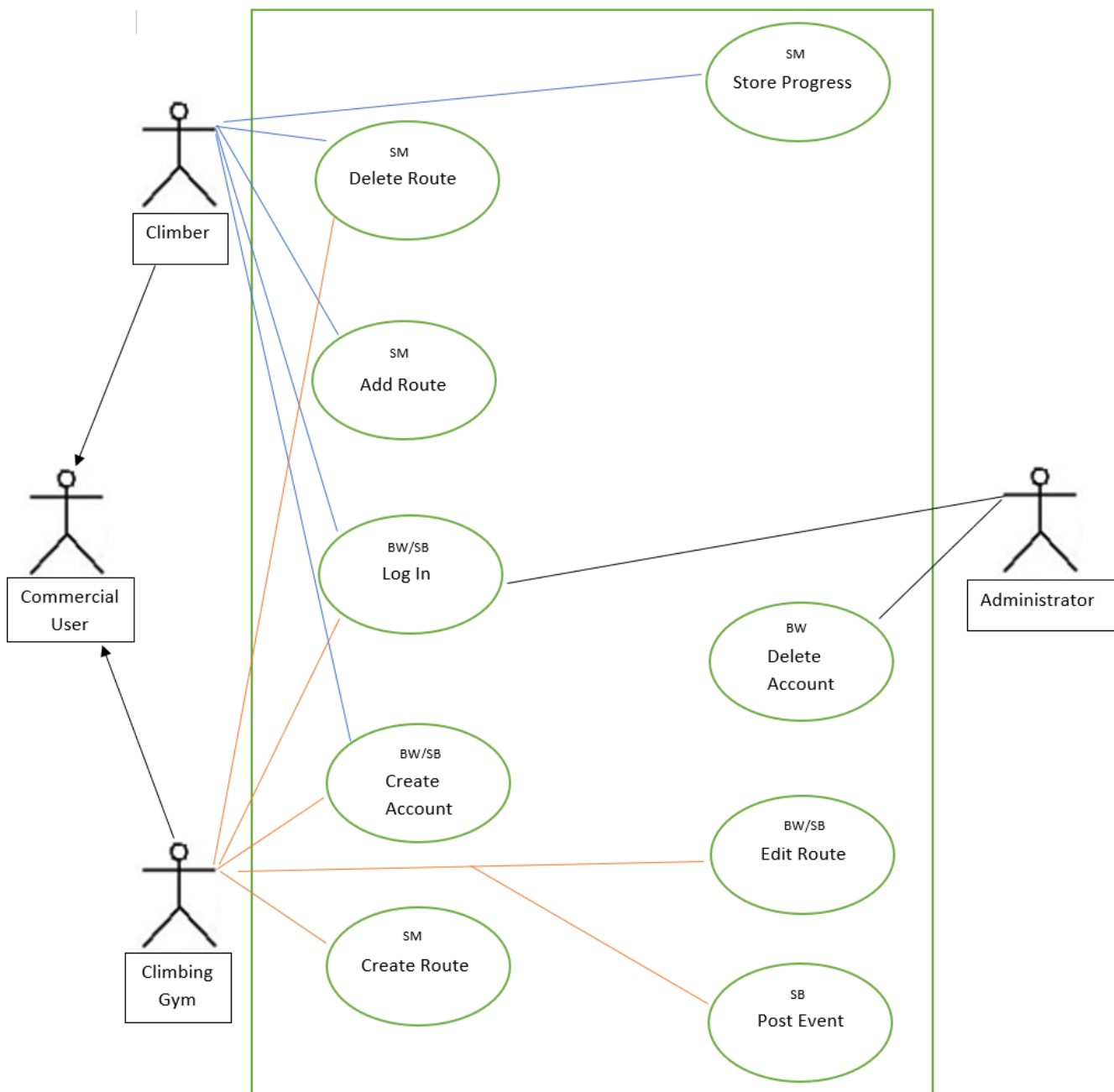
5.5 Other requirements

NFR13(R): UNCG Honor Code adherence.

All SRS/SRD should be:

- **Correct:** A method of analysis that ensures that the software meets the requirements identified.
- **Unambiguous:** There is only one interpretation of what the software will be used for and it is communicated in a common language.
- **Complete:** There is a representation for all requirements for functionality, performance, design constraints, attributes, or external interfaces.
- **Consistent:** Must be in agreement with other documentation, including a systems requirements specification and other documents.
- **Ranked for Importance and/or Stability:** Since all requirements are not of equal weight, you should employ a method to appropriately rank requirements.
- **Verifiable:** Use measurable elements and defined terminology to avoid ambiguity.
- **Modifiable:** A well-defined organizational structure of the SRS document that avoids redundancies can allow easy adaptation.
- **Traceable:** Ability to trace back to the origin of development and move forward to the documents produced from the SRS.
- **Legible and Professionally Presented:** Must use a consistent font and style. Must have proper formatting of tables and charts. Must be grammatically correct. Use active tense and concise sentences.

6. Use Case Model (Benjamin Woods)



Descriptions

I. Brief Description

- **Create Account**

The commercial user shall be able to create an account. After creating an account, they shall be able to log in to view their homepage and their other tabs.

Responsible Team Member: Benjamin Woods, Stuart Bridges

- **Log in**

The commercial user and administrator shall be able to log into their account via their log-in credentials. After logging in, they will be able to view their homepage and if applicable, their other tabs.

Responsible Team Member: Benjamin Woods, Stuart Bridges

- **Add Route**

The climber shall be able to add a climbing route on which they would like to track their progress to their profile. After adding the route, the climber shall be able to input their climbing data under it.

Responsible Team Member: Sofie Muska

- **Create Route**

The climbing gym user shall be able to create a new climbing route for their climbing route catalog. After adding the route, they will be able to view it on their route page.

Responsible Team Member: Sofie Muska

- **Edit Route**

The climber and climbing gym users shall be able to edit any route they have added to their climbing route catalog. The updated information will be displayed to the user.

Responsible Team Member: Benjamin Woods, Stuart Bridges

- **Delete Route**

The commercial user shall be able to delete a route from their profile. The profile will be immediately updated to reflect the change.

Responsible Team Member: Sofie Muska

- **Store Progress**

The climber user shall be able to enter their progress data into the table associated with their added route. Their progress will be displayed to them in table format.

Responsible Team Member: Sofie Muska

- **Post Event**

The climbing gym user shall be able to post event information on their homepage. The information will be displayed to the user.

Responsible Team Member: Stuart Bridges

- **Delete Account**

The administrator shall be able to select an account from the list of accounts and delete it. After account deletion, the commercial user to whom the account belonged shall no longer be able to log into the application with their credentials.

Responsible Team Member: Benjamin Woods

II. Scenarios:

- **Create Account**

- **Initial Assumption:** The commercial user has a valid email address to create their account with, and they use unique credentials.
- **Normal:** The commercial user will fill out their username and password and select whether they are creating a climber account or a climbing gym account, which will prompt them for additional information.
- **Potential Problems:** The commercial user tries to use a password that does not meet the password requirements. If they do so, they shall be prompted to choose a different password.
- **Other Activities:** N/A
- **System State on Completion:** The commercial user's account is stored in the account database, and they can now log in.

- **Log in**

- **Initial Assumption:** The user has a registered account for logging into the system. This account is stored in the account database.
- **Normal:** The user will enter their username and password, which will log them into their account and allow them to view their homepage and if applicable, their other tabs.
- **Potential Problems:** The user's credentials are not accepted because either the username or password is incorrect. If either is incorrect, the user will be prompted to reenter.
- **Other Activities:** N/A
- **System State on Completion:** The user is logged into their account and can view their homepage and if applicable, their other tabs.

- **Add Route**

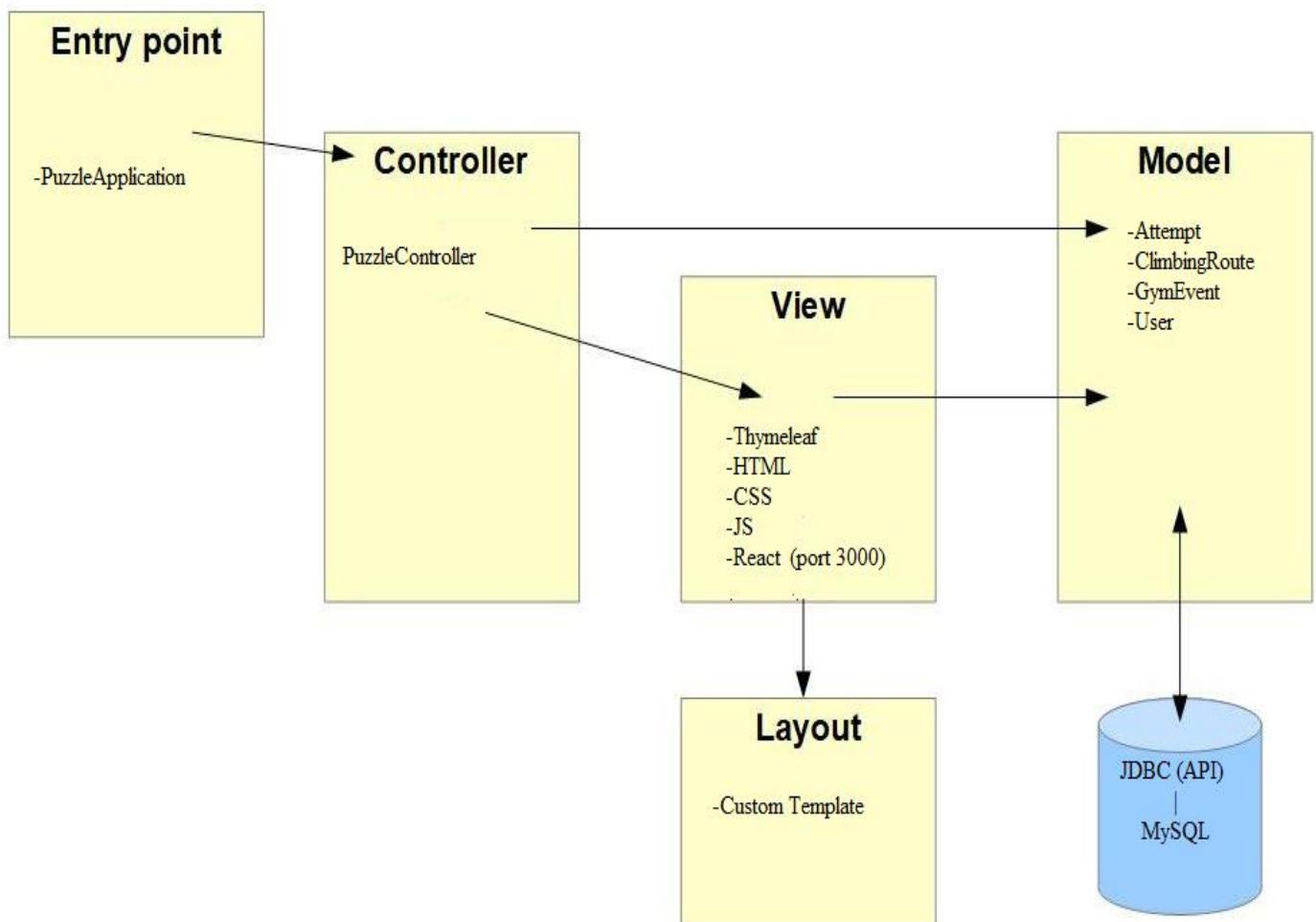
- **Initial Assumption:** The climber user has a registered account and has logged into it.
- **Normal:** The climber user will click on the add route button on their profile to add a climbing route to it.

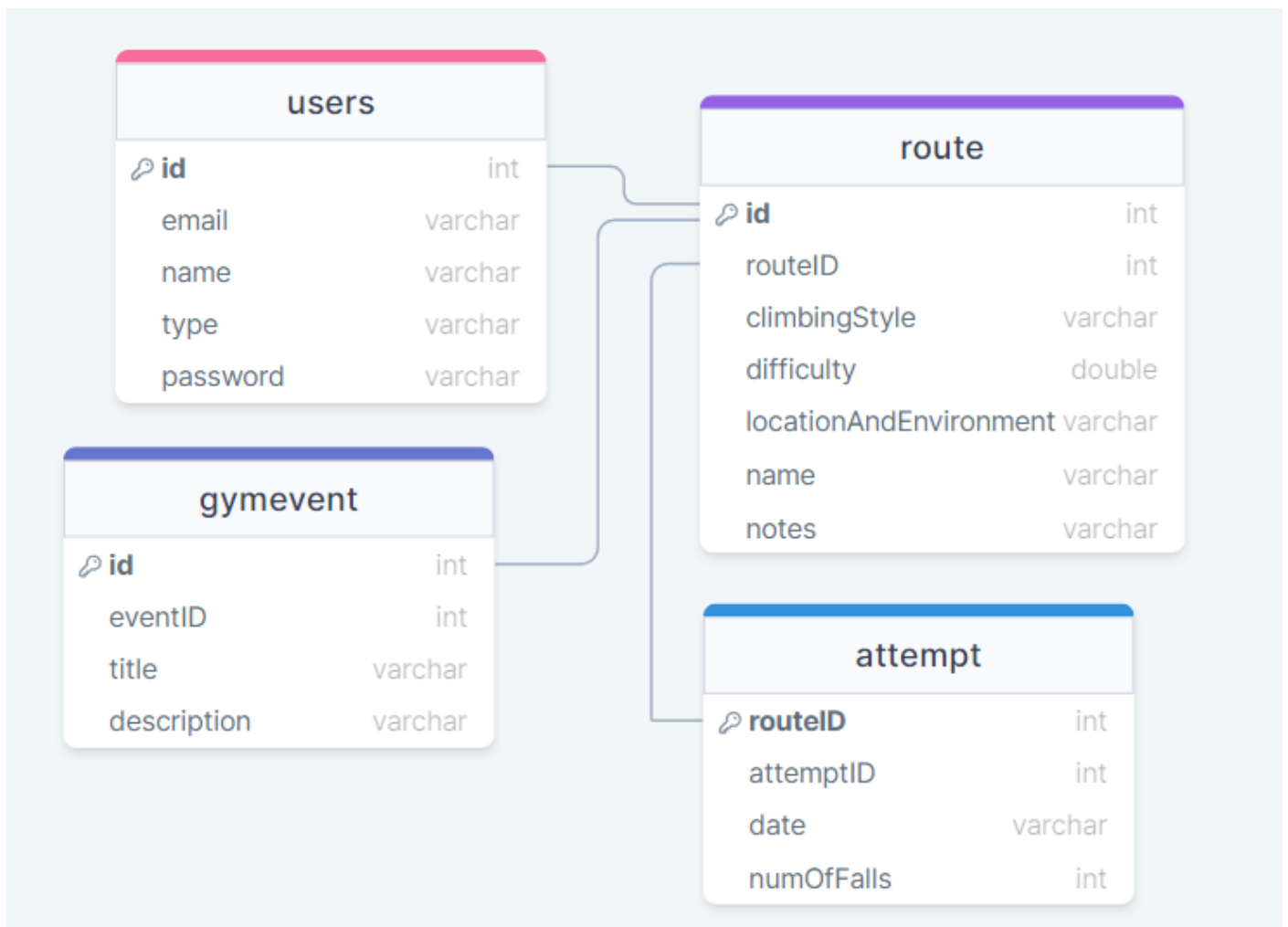
- **Potential Problems:** The climber user enters invalid input into the data fields. The climber user should be prompted to reenter if this occurs.
- **Other Activities:** N/A
- **System State on Completion:** The climbing route will appear on the climber user's list of tracked routes.
- **Create Route**
 - **Initial Assumption:** The climbing gym user has successfully logged into their account and has navigated from the homepage to the route catalog tab.
 - **Normal:** The climbing gym user will press a button to create a new route, which will prompt them to fill out several data fields with pertinent information. After filling out the information, they will hit another button to save it.
 - **Potential Problems:** The climbing gym user enters invalid input into the data fields. The climbing gym user should be prompted to reenter if this occurs.
 - **Other Activities:** N/A
 - **System State on Completion:** The new route will be added to the climbing gym's database of climbing routes, and it will become visible in the climbing gym's route catalog.
- **Edit Route**
 - **Initial Assumption:** The climbing gym user has logged into their account and has successfully navigated to the route catalog tab.
 - **Normal:** The climbing gym user will click on the edit icon next to the route name, which will prompt them to edit the route information. After editing the route information, the climbing gym user will save their updates.
 - **Potential Problems:** The climbing gym user enters invalid input into the data fields while updating the route information. The climbing gym user should be prompted to reenter if this occurs.
 - **Other Activities:** N/A
 - **System State on Completion:** The updated route information will be stored in the climbing gym user's climbing route database. The climbing gym user's route catalog will now display the updated information.
- **Delete Route**
 - **Initial Assumption:** The commercial user has logged into their account, navigated to their active climbing routes, and has at least one active climbing route.
 - **Normal:** The commercial user will click the delete icon next to the climbing route name to delete it from their list of active climbing routes.
 - **Potential Problems:** N/A
 - **Other Activities:** N/A
 - **System State on Completion:** The climbing route will be removed from the commercial user's database of active climbing routes, and it will no longer be visible on their profile.
- **Store Progress**
 - **Initial Assumption:** The climber user has logged into their account and added a route on which they would like to track their progress to their profile.
 - **Normal:** The climber user will input the climbing data from their climbing attempts on this route into the route table.
 - **Potential Problems:** The climber user enters invalid input into the data fields while updating their progress information. The climber user should be prompted to reenter if this occurs.
 - **System State on Completion:** The progress data will be stored in a database and used to generate or update the climber user's progress table.
- **Post Event**
 - **Initial Assumption:** The climbing gym user has logged into their account and can see their homepage.
 - **Normal:** The climbing gym user will press a button to post an event, which will display a text box where they can enter the event information. After entering the event information, they will press another button to save it.

- **Potential Problems:** The climbing gym user could attempt to post an event without entering any text. In this case, they should be notified that the text box is empty and not allowed to post until some text is entered.
- **Other Activities:** N/A
- **System State on Completion:** The event will be stored in the climbing gym user's event database and become visible to the climbing gym user.
- **Delete Account**
 - **Initial Assumption:** The administrator has successfully logged into their account and can view the list of active accounts from their homepage.
 - **Normal:** The administrator will click the delete icon next to the commercial user's account name (i.e., their email address) to delete the account.
 - **Potential Problems:** The administrator could accidentally delete the wrong account type with the same information. To mitigate the risk of this, the administrator will be able to see the user's account type.
 - **Other Activities:** N/A
 - **System State on Completion:** The commercial user account will be removed from the database of user accounts. The administrator will no longer be able to see the account on the list of active accounts on their homepage.

7. Software Architecture (Stuart Bridges, Sofie Muska)

7.1 MVC Architecture OOP

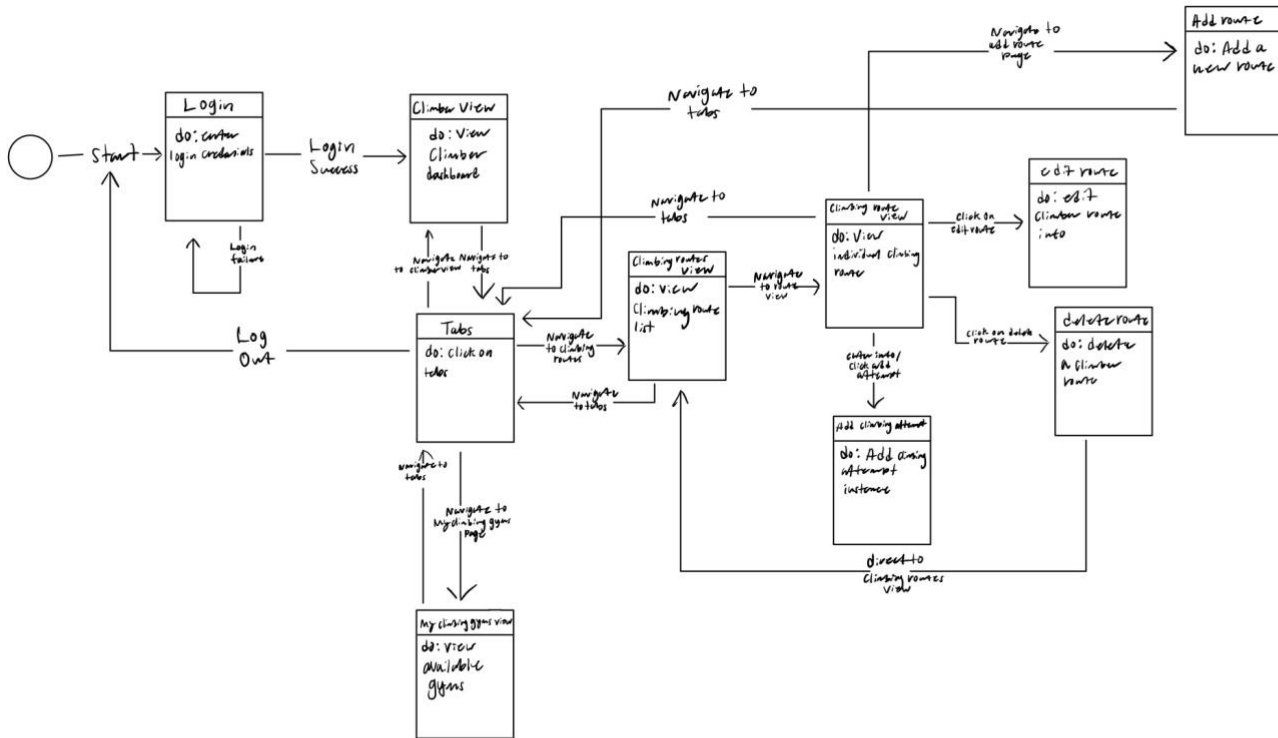




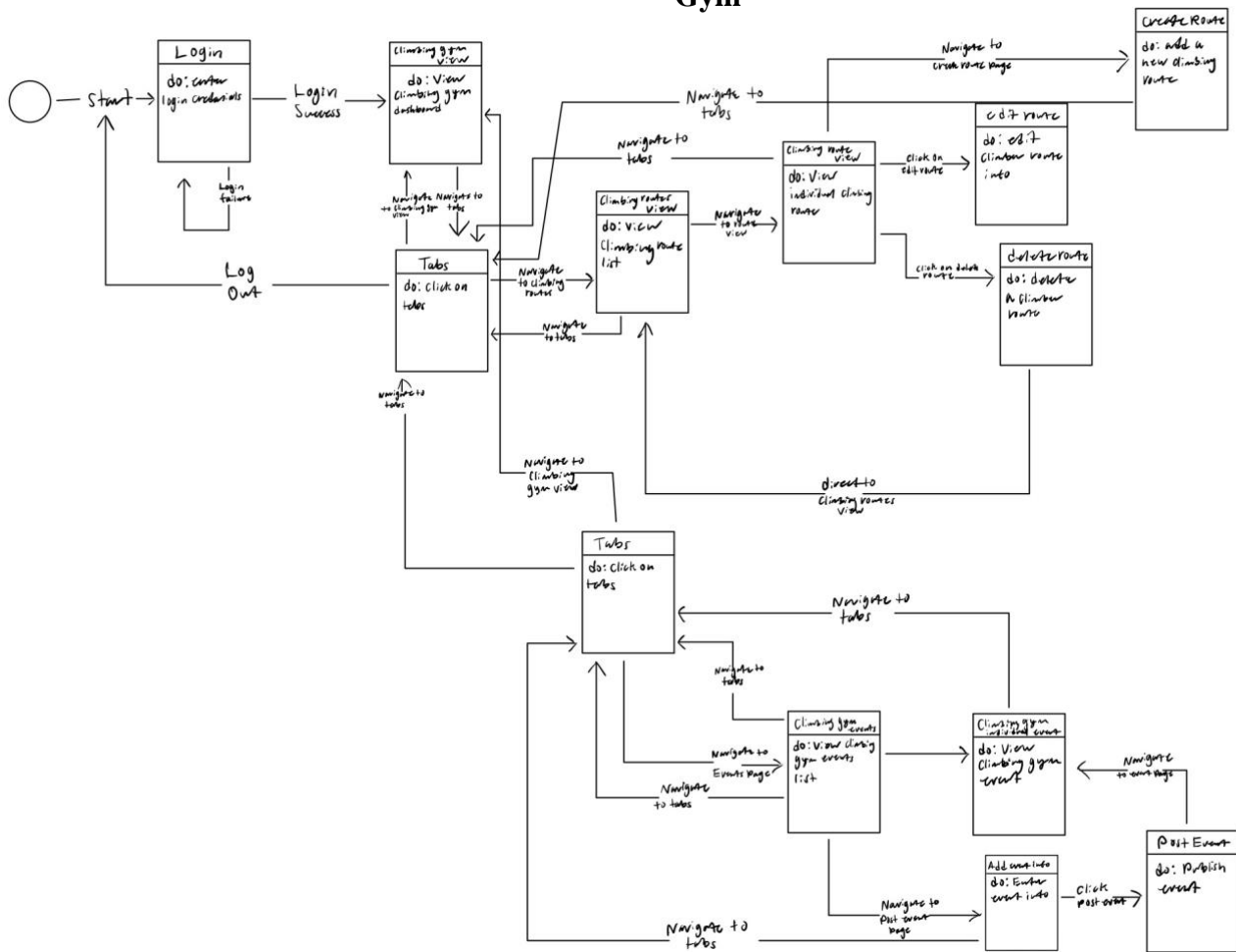
8. Software Design (Sofie Muska, Benjamin Woods, Stuart Bridges)

8.1 State Diagrams

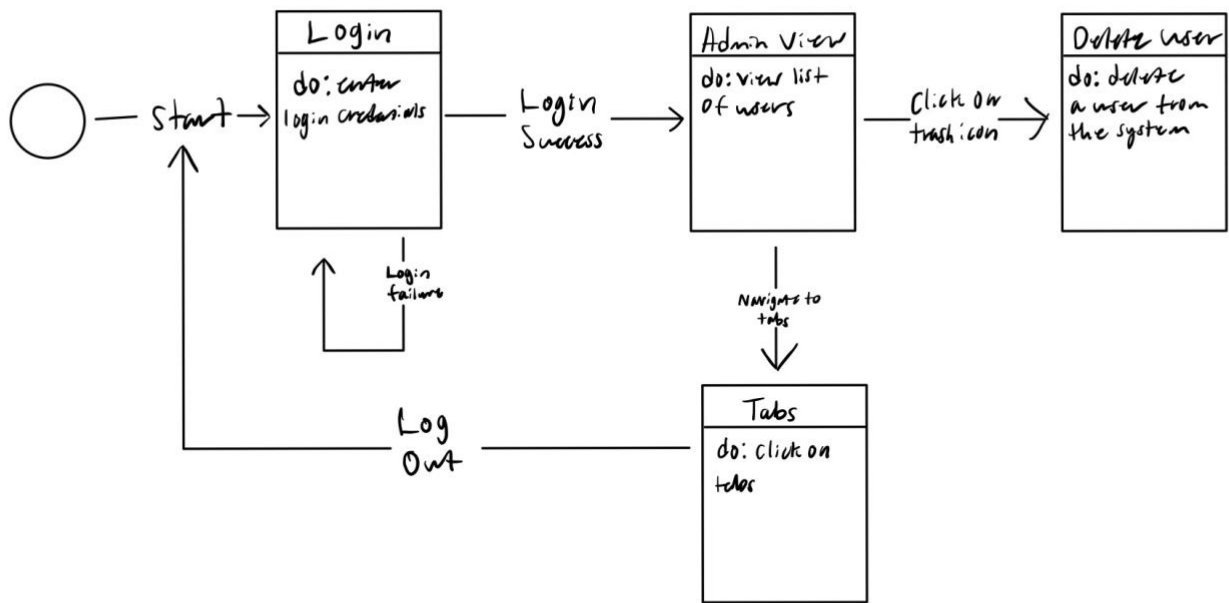
Climber



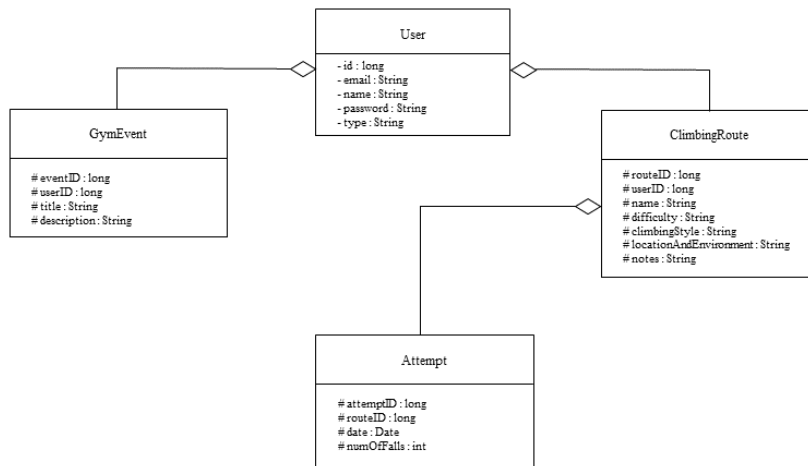
Gym



Admin



8.2 UML Class Diagram



9. Scenario (Benjamin Woods)

Climber: Create climber profile use case:

Climber C1 creates a profile including email, password, name, and type. C1 can now log in.
Climber C2 creates a profile including email, password, name, and type. C2 can now log in.

Gym: Create gym profile use case:

Gym G1 creates a profile including email, password, name, and type. G1 can now log in.
Gym G2 creates a profile including email, password, name, and type. G2 can now log in.

The screenshot shows the 'PUZZLE' application interface. On the left, a vertical sidebar contains 'Homepage' (highlighted in brown) and 'Log In' (in light blue). The main content area has a blue background and features a white 'Registration' form. The form includes fields for 'E-mail:', 'Password:', 'Name:', and 'Type:'. The 'Type:' dropdown menu is open, showing options: 'climber' (selected), 'climber', 'gym', and 'admin'. Below the dropdown, there is a link 'Already registered? Click here'. At the bottom of the form is a blue 'Sign Up' button.

Climber: Login, View Route List, View Available Gyms, View Route Details, Create Route, Modify Route, Delete Route use cases:

1. Climber C1 logs in for the first time and creates a route. After creating the route, they view it in the route list.

The screenshot shows the 'PUZZLE' application interface. On the left, a vertical sidebar contains 'Homepage' (highlighted in brown) and 'Register' (in light blue). The main content area has a blue background and features a white 'Login' form. The form includes fields for 'E-mail:' (with the value 'climber@climber.com') and 'Password:' (with masked characters '*****'). There is a 'Submit' button at the bottom left of the form and a user icon at the bottom right of the password field.

PUZZLE

Home

My Climbing Routes

My Climbing Gyms

Log Out

Add Route

General Information

Name:
Kelp Forest

Difficulty Level:
10

Climbing Style:
Top-Rope

Location/Environment:
Indoor

Notes:
Climb Fast!

Add Route

PUZZLE

Home

My Climbing Routes

My Climbing Gyms

Log Out

My Climbing Routes

Add Route

Kelp Forest

2. C1 views the route details, modifies the created route, and saves. Then, they add a few route attempts.

PUZZLE

Home

My Climbing Routes

My Climbing Gyms

Log Out

Kelp Forest

General Information

Name:
Kelp Forest

Difficulty Level:
10

Climbing Style:
Top-Rope

Location/Environment:
Indoor

Notes:
Climb Fast!

Save Changes

PUZZLE

Home

My Climbing Routes

My Climbing Gyms

Log Out

Dry Forest

Edit Route

Delete Route

- General Information
 - Difficulty: 10
- Climbing Style: Top-Rope
- Location and Environment: Indoor
- Notes: Climb Fast!

Date: Number of Falls:

Add Attempt

Date	Number of Falls

PUZZLE

Home

My Climbing Routes

My Climbing Gyms

Log Out

Kelp Forest

Edit Route

Delete Route

- General Information
 - Difficulty: 10
- Climbing Style: Top-Rope
- Location and Environment: Rocky
- Notes: Climb Fast!

Date: Number of Falls:

Add Attempt

Date	Number of Falls
2022-12-15	0

- C1 creates a second route, views it on the route list. C1 logs out.
- C2 logs in for the first time and creates a route. C2 logs out.
- C1 logs in again and shows that the route is still different from C2. Then, C1 deletes a route.

PUZZLE

Home

My Climbing Routes

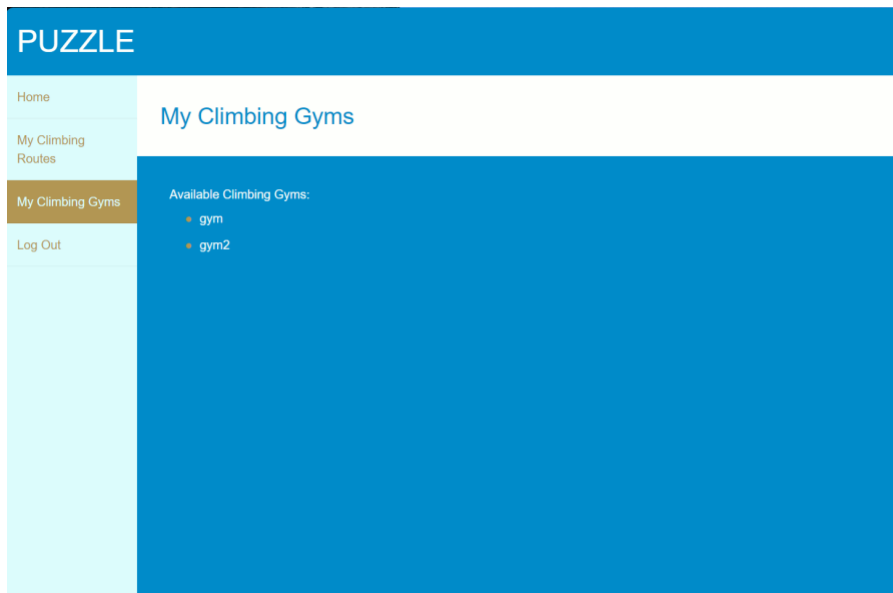
My Climbing Gyms

Log Out

My Climbing Routes

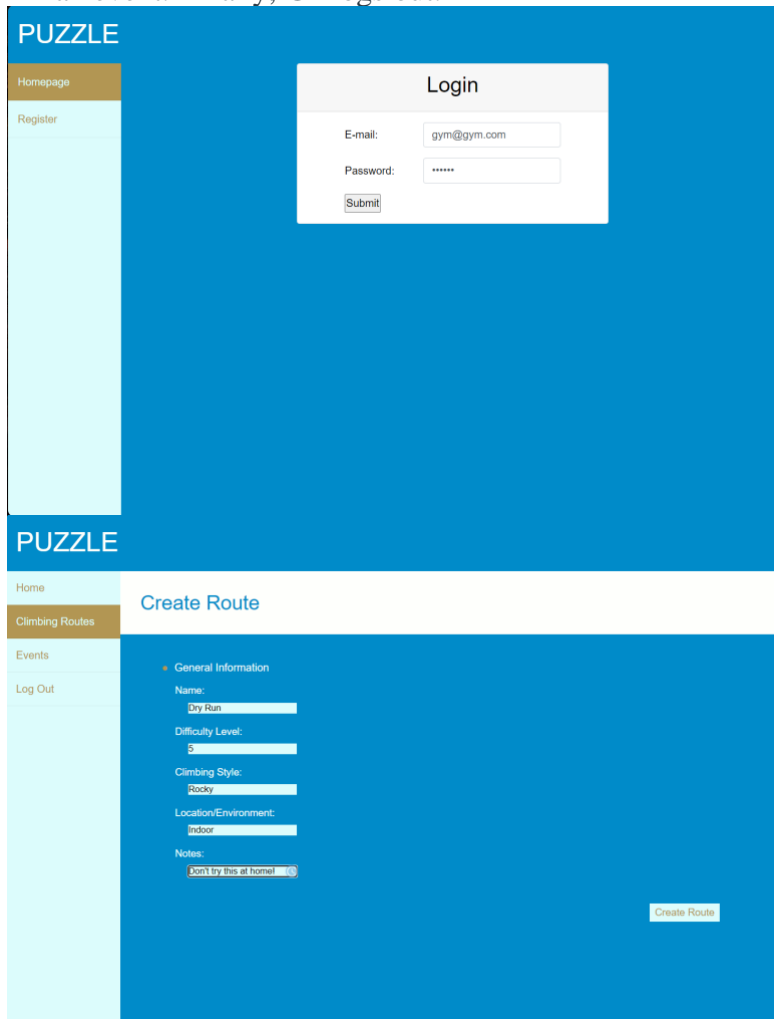
Add Route

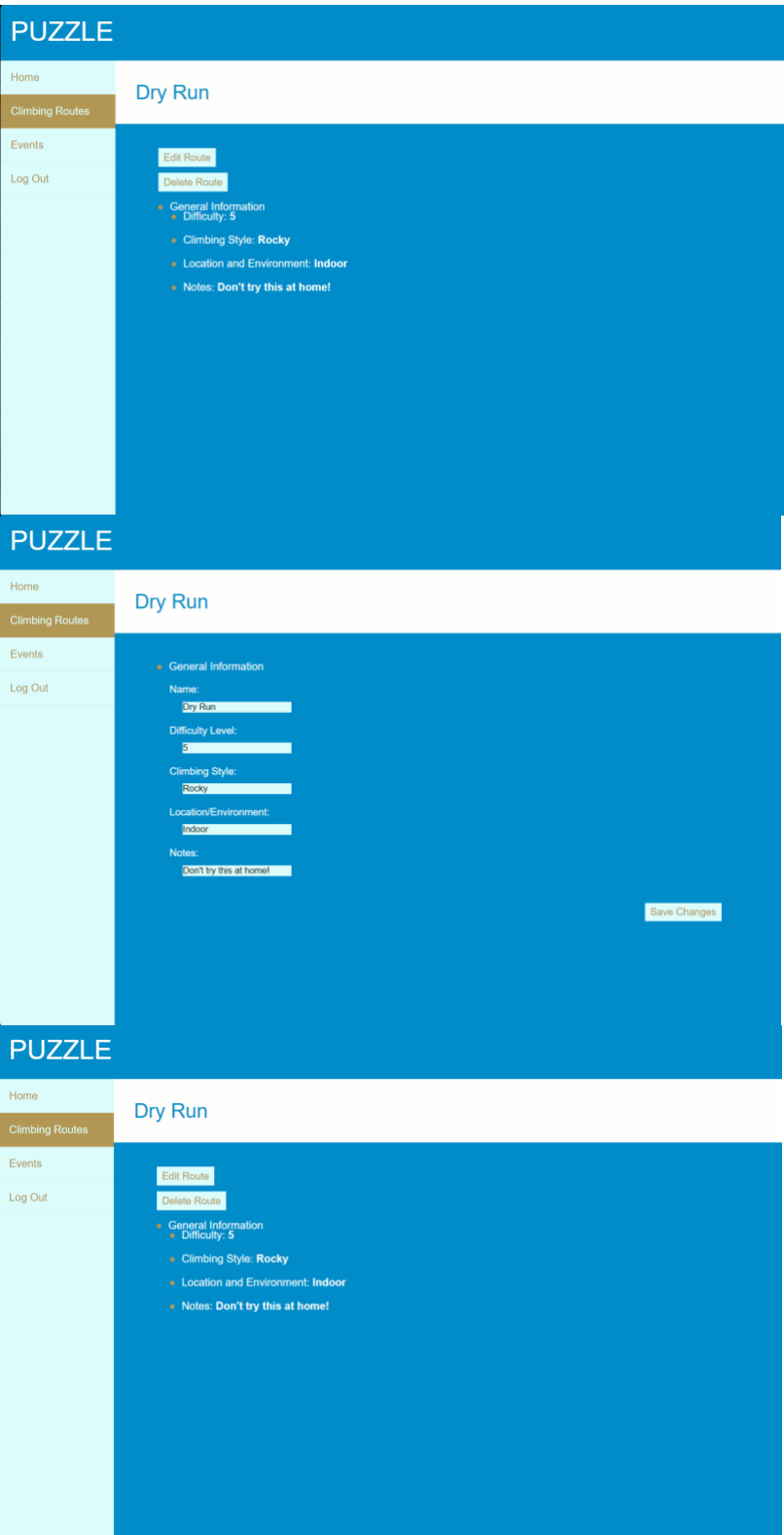
- C1 navigates to their climbing gyms and shows the available climbing gyms.

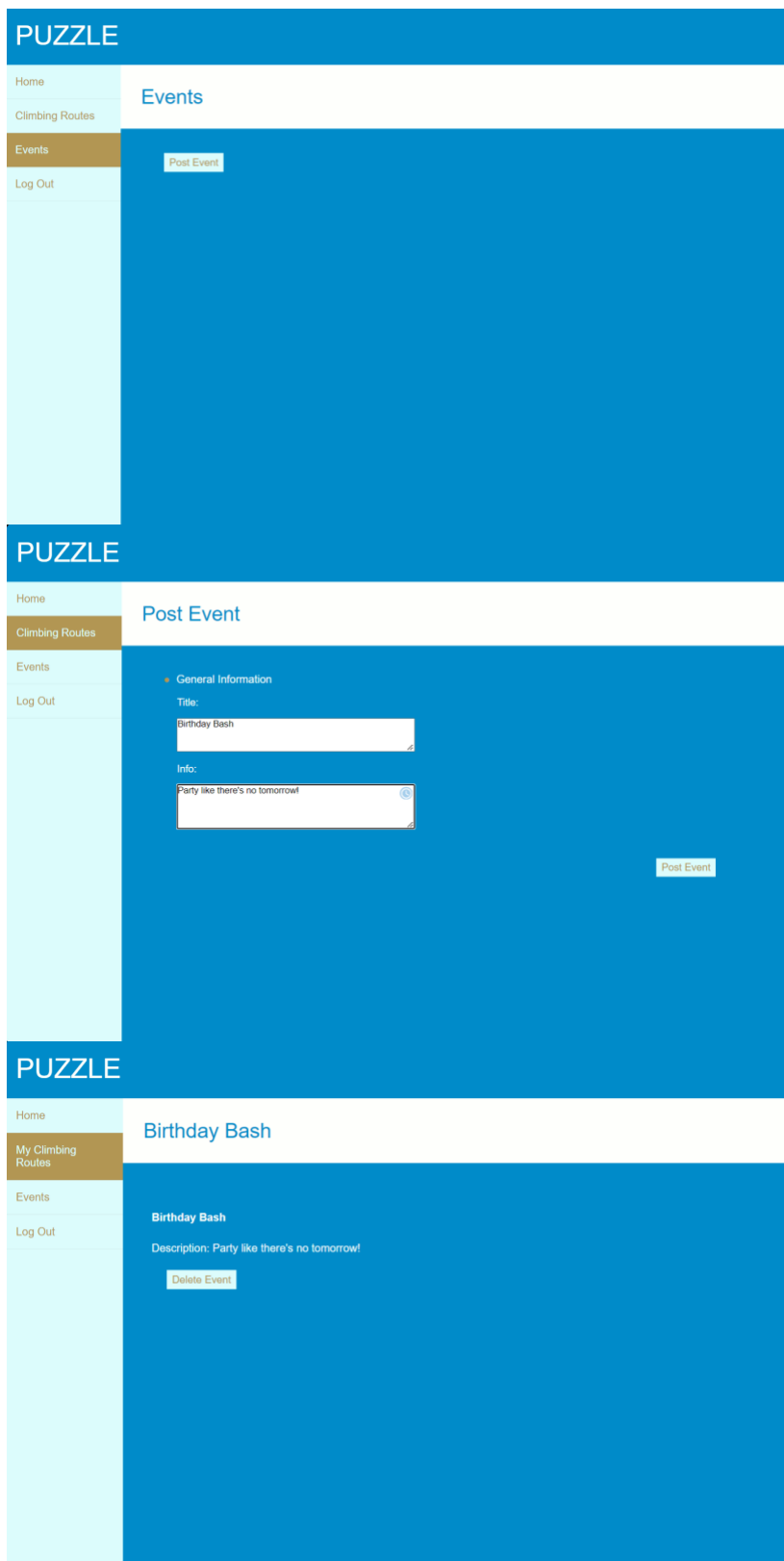


Gym: Login, Create Route, Modify Route, View Route List, View Route Details, Delete Route, Create Event, and Delete Event use cases:

1. Gym G1 logs in for the first time and creates a route. G1 then modifies the route. G1 then creates an event. Finally, G1 logs out.







2. Gym G2 logs in for the first time and creates a route. G2 logs out.
3. G1 logs in and shows their route list is still the same. G1 deletes a route and an event.

Admin: Login, View Users, and Delete Users use cases:

1. Admin A1 logs in for the first time and views the list of users—their email, password, name, and type.

PUZZLE

Homepage

Register

Login

E-mail:

Password: 

PUZZLE

User ID	E-mail	Password	Name	Type	
1	benewoods02@gmail.com	puzzledb	Ben	admin	<input type="button" value="Delete"/>
2	climber@climber.com	climber	climber	climber	<input type="button" value="Delete"/>
3	gym@gym.com	gymgym	gym	gym	<input type="button" value="Delete"/>
4	gym2@gym.com	gymgym	gym2	gym	<input type="button" value="Delete"/>
5	climber2@climber.com	climber2	Climer2	climber	<input type="button" value="Delete"/>

2. A1 deletes C2 and logs out.

PUZZLE

Log Out

User ID	E-mail	Password	Name	Type	
1	benewoods02@gmail.com	puzzledb	Ben	admin	Delete
2	climber@climber.com	climber	climber	climber	Delete
3	gym@gym.com	gymgym	gym	gym	Delete
4	gym2@gym.com	gymgym	gym2	gym	Delete

3. C2 tries to login and cannot because their account is deleted.