מתודולוגיות מתקדמות לפיתוח תכנה

Camp Management System

Submitted to: Dr. Rubin Cohen

Submitted By:

- Salman Habaka 314704016
- Malak Khateeb 208805739

Submission Date: 23/05/2024

Project Description:

The camping company is interested in developing a dedicated application that will allow them to track activities in the camp for company employees, as well as for campers and their parents. Every year, participants in various camps include children aged 8-16, divided into groups. During the summer months, the company employs around X administrative workers and Y instructors.

The application will manage the registration process for both children and instructors, provide information about the children in the camp, and handle administrative tasks such as scheduling and communication with parents when necessary. Given the responsibilities and workload, the employees are interested in maintaining a task journal.

The application will include personal profiles for each camper and instructor, containing personal information, approvals, health forms, and event logs. This will help instructors know where they need to be and when. Parents of campers also want to stay in touch with the staff and their children.

The application aims to allow instructors to know their assigned groups, tasks, facilitate communication among staff and groups, and enable daily reporting on the activities.

Requirements

Functional Requirements:

1. **User Registration:**

   o Register users as campers or instructors, collecting necessary personal information.

   o Include medical details and emergency contact information for campers.

   o Record qualifications, certifications, and experience for instructors.

2. **Camp Management:**

   o Create and manage multiple camps with specific participants and instructors.

   o Group campers based on age, with the ability to modify group assignments.

3. **Registration Process:**

   o Facilitate a user-friendly registration process for campers and instructors.

   o Include medical forms, emergency contact details, and parental consents in camper registration.

4. **Task Assignment:**

   o Allow instructors to view and manage assigned groups, tasks, and responsibilities.

   o Allocate tasks and duties based on qualifications and expertise.

5. **Daily Reporting:**

   o Enable instructors to submit daily reports on activities, incidents, and noteworthy events.

   o Generate summary reports for administrators and parents.

6. **Accessibility and Multilingual Support:**

   o Ensure the application is accessible to users with diverse abilities and needs.

7. **Financial Management:**

   o Support financial transactions related to camp fees, refunds, and additional charges.

   o Generate financial reports and statements for administrators, with a secure payment gateway.

8. **Feedback and Evaluation:**

   o Allow campers and parents to provide feedback on the camp experience.

   o Evaluate instructors based on performance.

<u>Non-Functional Requirements:</u>

1. **Scalability:**

   o Handle a significant number of users simultaneously, supporting camp enrollment.

2. **Performance:**

   o Ensure low latency in loading pages and processing requests.

   o Handle concurrent user interactions without significant performance degradation.

3. **Security:**

   o Implement robust security measures to protect sensitive information.

   o Have authentication and authorization mechanisms for controlled access.

4. **Reliability:**

   o Minimize downtime and ensure continuous availability during the camp season.

5. **Compatibility:**

   o Be compatible with various devices and browsers for accessibility.

6. **Usability:**

   o Provide an intuitive and user-friendly interface, requiring minimal training.

7. **Maintainability:**

   o Be easily maintainable for updates, bug fixes, and enhancements.

8. **Regulatory Compliance:**

   o Adhere to relevant data protection, child safety, and privacy regulations.

9. **Data Backup and Recovery:**

   o Regularly backup data to prevent loss and have a robust recovery plan.

10. **Response Time:**

    o Maintain a short response time during peak usage periods.

Use Case Diagram
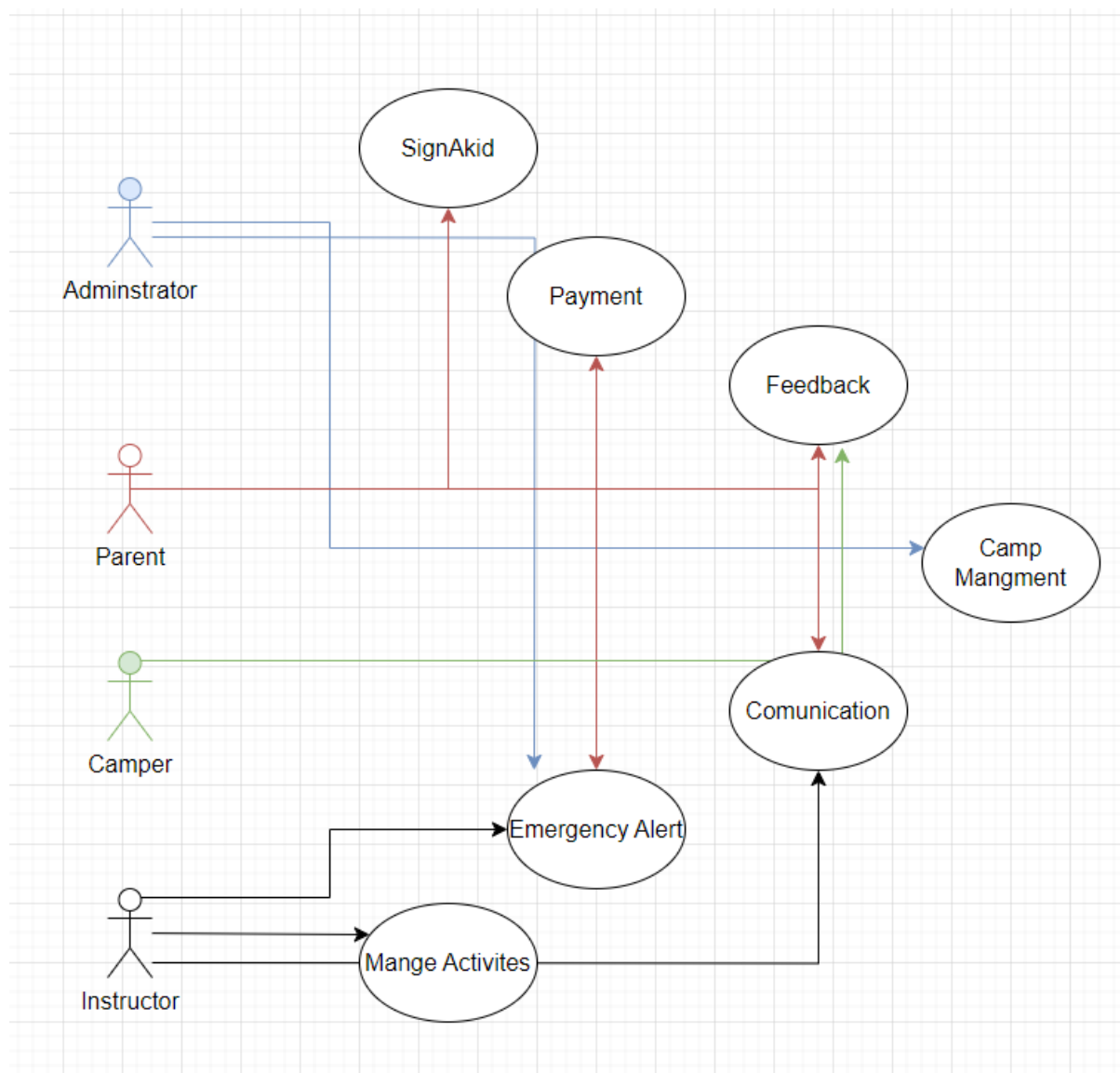
Use Cases Description

Use case 1: Register Campers and Instructors

Actors: administrative workers

Description: The use case describes the process of registering campers and instructors in the camping company's dedicated application

Main flow:

- Administrative workers input personal information, approvals, health forms, and events logs for each camper and instructor.
- the application generates a unique identifier for each camper and instructor.

Alternative flow:

- If the application detects an error in the input data, it displays an error message and prompts the user to correct it.

Use case 2: Assign campers to group

Actors: administrative workers

Description: this use case describes the process of assigning campers to group based on age.

Main flow:

- Administrative workers select the appropriate group for each camper.
- the application update the camper's profile with their assigned group.

Alternative flow:

- *if the application detects a conflict in the group assignments, it displays a warning messages and prompts the user to resolve the conflict.

Use case 3: View assigned groups

Actors: instructors

Description: this use case describes the process of viewing the groups assigned to an instructor**.**

Main flow:

- instructors log in to the application
- the application displays the group assigned to the instructor.

Use case 4: schedule tasks

Actors: instructors

Description: this use case describes the process of scheduling tasks for a group.

Main flow:

- instructors input the task details, including start time, end time and location.
- the applications store the task schedule.

Alternative flow:

- if the application detects a conflict in the task scheduling, it displays a warning messages and prompts the user to resolve the conflict.
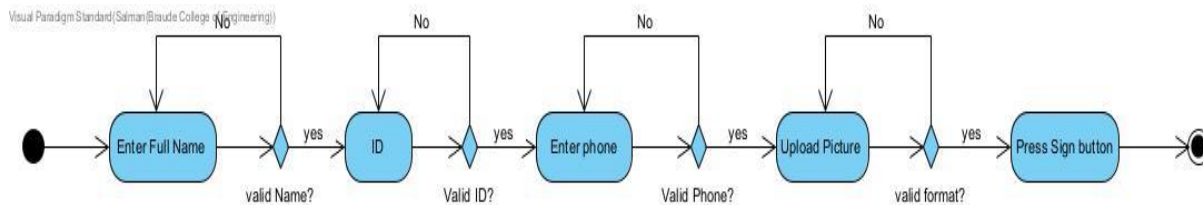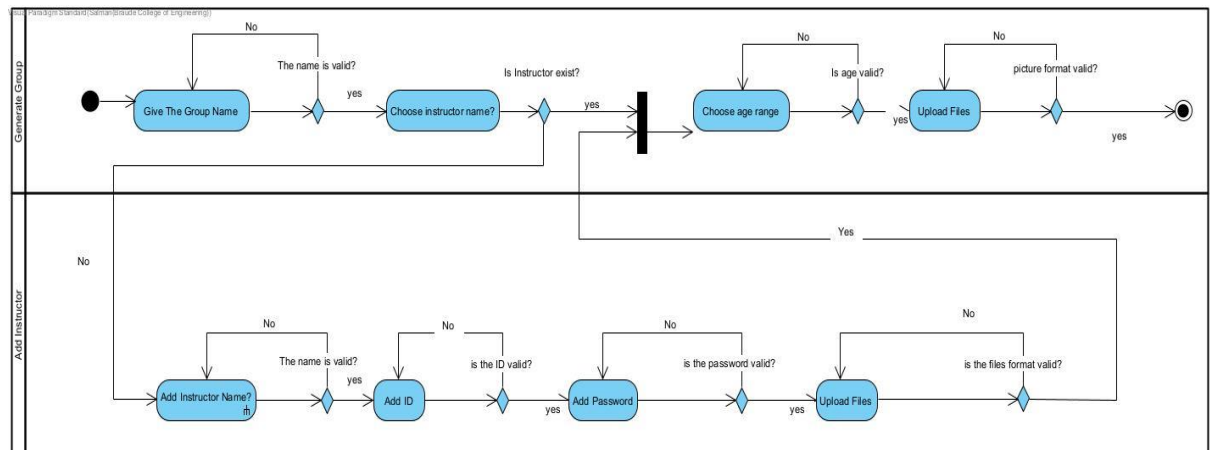
Activity Diagrams

Feedback:
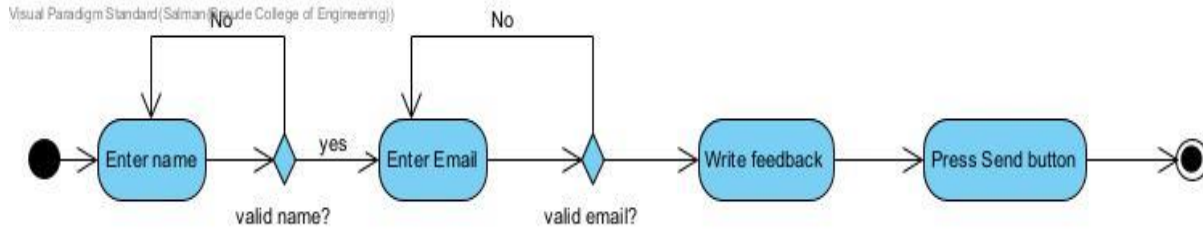
Description: Getting feedback from the parent or the son about the camp, we ask for there name, email and then we add the feedback to the system.
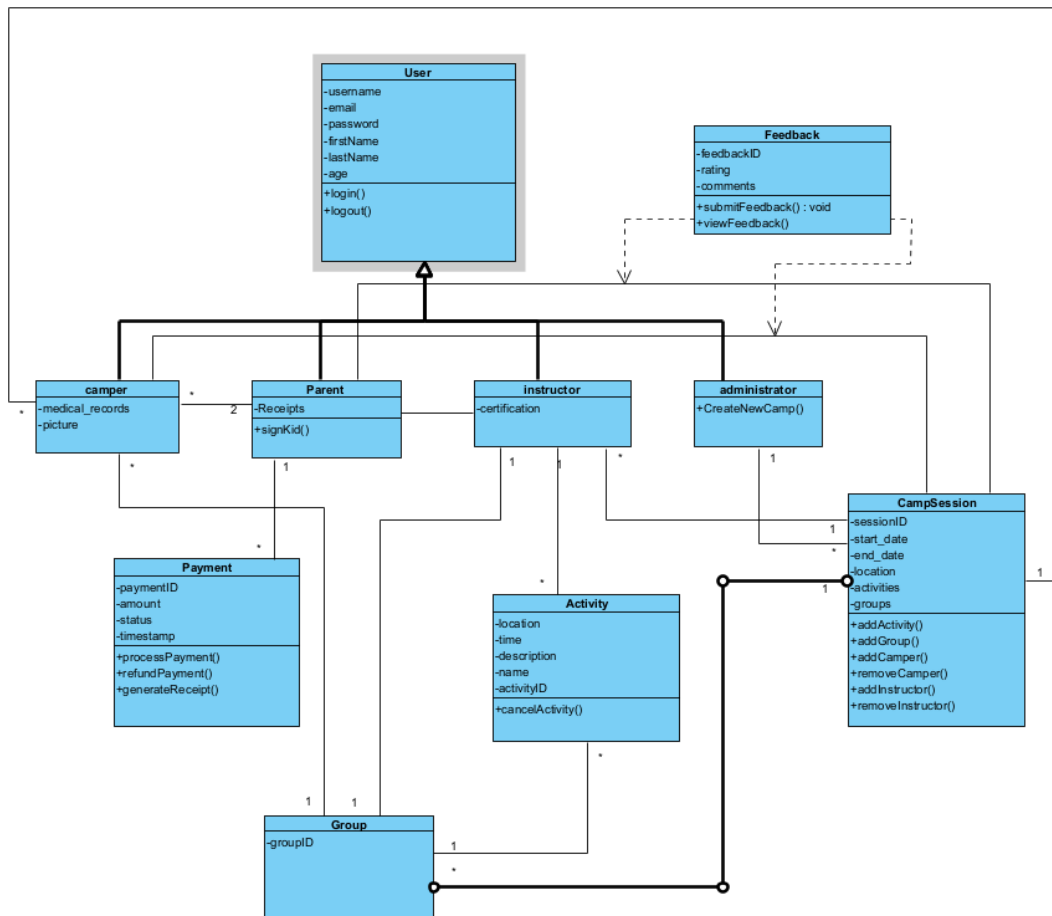
Sign Kid:

Description: Each parent enter their son details to sign him to specific camp.
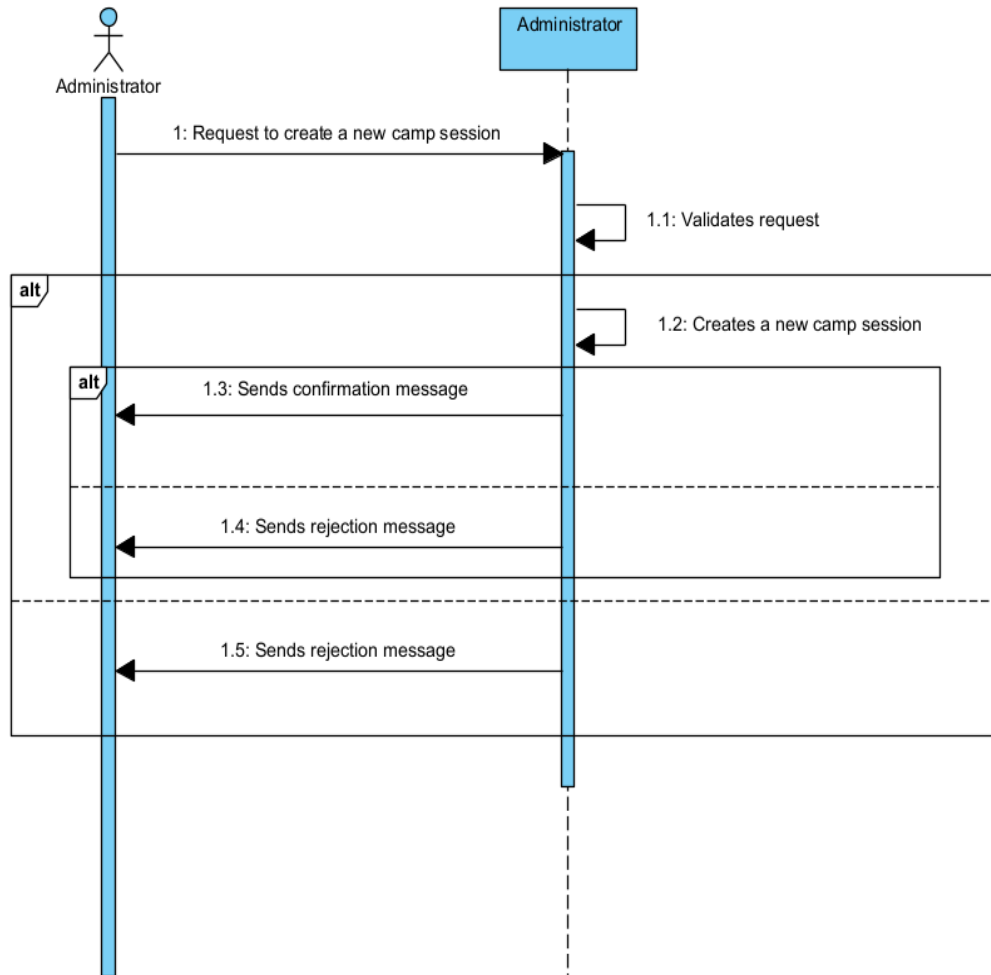
Generate Group:

Description: the admin of the camp generate groups after checking that the instructor exist to mange the group, after the generation process, the group assigned to instructor.
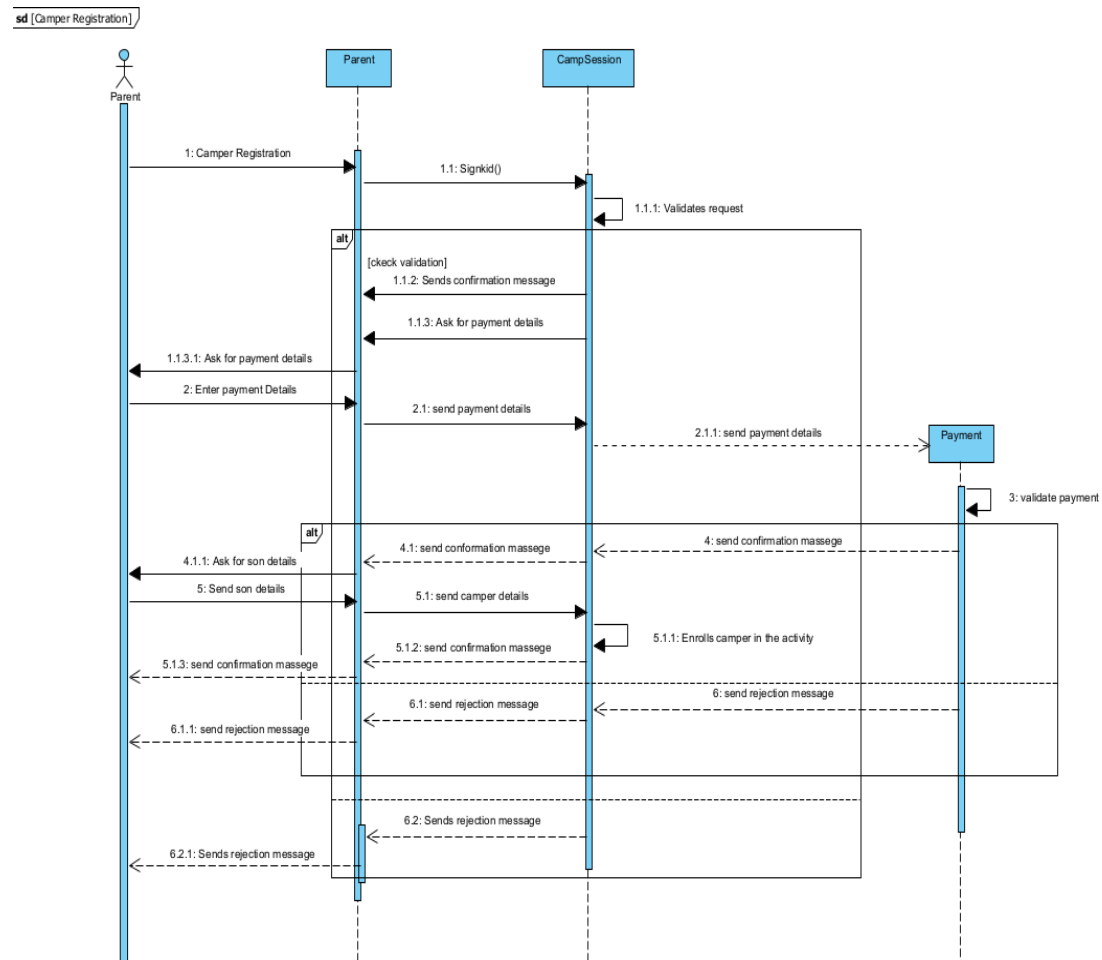
Class Diagram

Sequence Diagram

Create Camp Session

Description: The Admin of the Camp connect with the camps admin company to create new camp

Camper Registration

sd [Camper Registration]

Parent — Parent — CampSession — Payment

1: Camper Registration
1.1: Signkid()
1.1.1: Validates request

alt
[ckeck validation]
1.1.2: Sends confirmation message
1.1.3: Ask for payment details
1.1.3.1: Ask for payment details
2: Enter payment Details
2.1: send payment details
2.1.1: send payment details
3: validate payment

alt
4: send confirmation massege
4.1: send conformation massege
4.1.1: Ask for son details
5: Send son details
5.1: send camper details
5.1.1: Enrolls camper in the activity
5.1.2: send confirmation massege
5.1.3: send confirmation massege
6: send rejection message
6.1: send rejection message
6.1.1: send rejection message

6.2: Sends rejection message
6.2.1: Sends rejection message

Description: the parent chose to sign there son in the application, the enter there son details wait for confirmation after that they have to enter the payment details and when the payment approved, the process end and confirmation message sent.

Black Box Test Plan

To perform black-box testing on your camping application project, we will create test cases based on the requirements and functionalities you've provided. Here's a breakdown of the test cases for each feature:

1. Check Feedback

Test Case 1.1: Submit Feedback

- Input: User submits feedback through the application.
- Expected Output: Feedback is successfully submitted and stored in the system.

Test Case 1.2: View Feedback

- Input: User requests to view submitted feedback.
- Expected Output: The system displays all feedback relevant to the user (e.g., parents can see feedback about their child).

2. Payment

Test Case 2.1: Process Payment

- Input: User submits payment information (credit card, amount).
- Expected Output: Payment is processed successfully, and a confirmation message is displayed.

Test Case 2.2: View Payment History

- Input: User requests to view payment history.
- Expected Output: The system displays the user's payment history accurately.

3. Generate Group

Test Case 3.1: Automatic Group Generation

- Input: Admin triggers the group generation process.
- Expected Output: Groups are generated according to predefined criteria (age, skill level, etc.), and each camper is assigned to a group.

Test Case 3.2: View Generated Groups

- Input: User requests to view the generated groups.
- Expected Output: The system displays the groups and their members correctly.

4. Enroll Campers and Instructors

Test Case 4.1: Camper Registration

- Input: Parent submits a registration form for their child.

- Expected Output: The system registers the camper successfully and provides a confirmation message.

Test Case 4.2: Instructor Registration

- Input: Instructor submits a registration form.

- Expected Output: The system registers the instructor successfully and provides a confirmation message.

5. Update List

Test Case 5.1: Update Camper Information

- Input: Admin updates the camper's personal information.

- Expected Output: The system saves the updated information and reflects the changes accurately.

Test Case 5.2: Update Instructor Information

- Input: Admin updates the instructor's personal information.

- Expected Output: The system saves the updated information and reflects the changes accurately.

General Test Cases

Test Case 6.1: Login Functionality

- Input: User submits login credentials.

- Expected Output: The system authenticates the user and grants access to their profile.
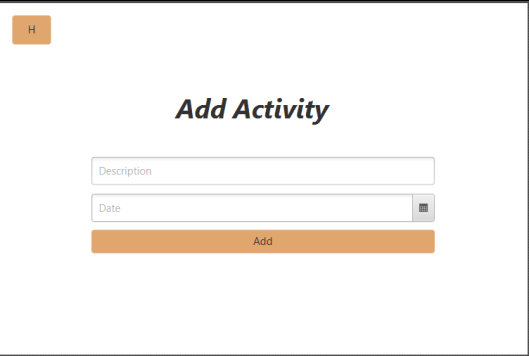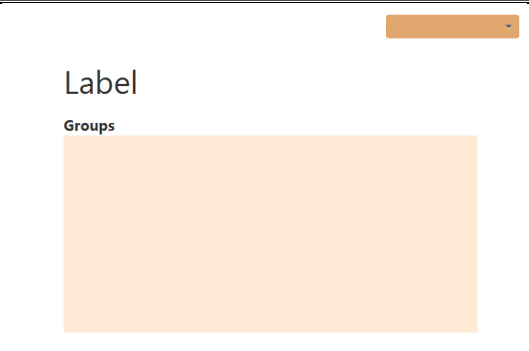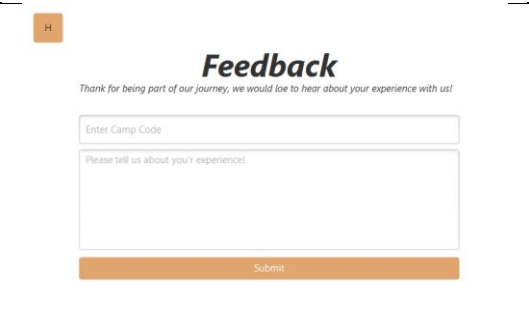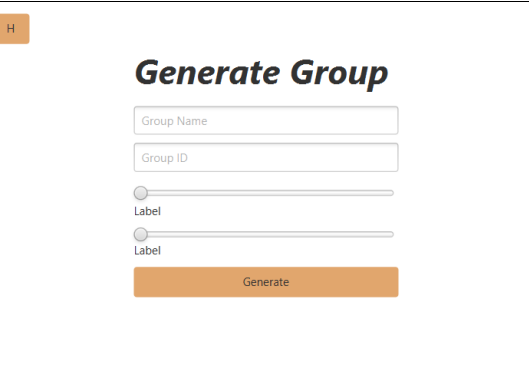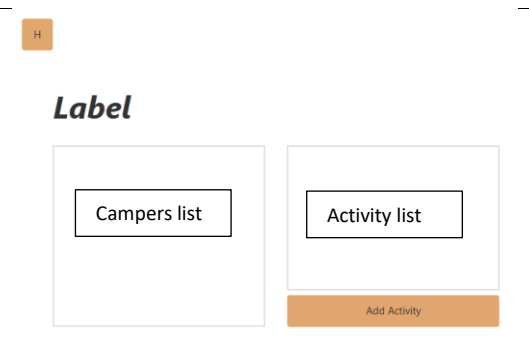
Test Case 6.2: Profile Access

- Input: User accesses their profile.

- Expected Output: The system displays the user's profile with all relevant information.

Test Case 6.3: Task Journal

- Input: User accesses the task journal.

- Expected Output: The system displays the task journal with all entries.

GUI

Description	Gui
Add activity to the group activity list with dates.	**Add Activity** Description Date Add
Admin Page: we have drop menu that contain generate group and add new instructor and we have list that containe all the groups	Label **Groups**
Feedback page the help users give there opinion about the camp and the management use for future analysis.	**Feedback** Thank for being part of our journey, we would loe to hear about your experience with us! Enter Camp Code Please tell us about you'r experience! Submit
Generate New Group to specific camp	**Generate Group** Group Name Group ID Label Label Generate
Instructor home page that help him see all the campers and the future activity and add activity if its Nassery.	**Label** Campers list / Activity list Add Activity

Parent home page that they can mange all there kids, they can see al there kids and the drop down menu contain payment, sign kid and see receipts.	**HELLO**
Pay for the camp	**Payment** amount Kid ID ☐ *Accept terms and conditions* Pay
Parent can see all there receipts or click pay to go to the payment page.	**Receipts** Pay
Sign in page with option to log in as admin or instructor if you have permission.	**Sign In** ID Password ☐ I am admin ☐ I am Instrucor SignIn SignUp
Sign new instructor to the camp.	**Sign Instructor** Instructor name Instructor ID Password Group ID Bank Details ☐ *Let the instructor accept the terms and the conditions* Sign Instructor

Sign new kid to the camp.		
	15	**Sign A Kid** Name ID Phone Code ☐ Accept terms and conditions Sign
Sign Up page.		Register Name ID Phone Password Email ☐ Accept terms and confitions SignUp

Summary:

- The camping company aims to develop a dedicated application to streamline operations and enhance communication between campers, parents, and staff. The application will manage registration, provide information about campers, handle administrative tasks, and facilitate communication. It will include personal profiles for campers and instructors, enabling efficient management of tasks and groups. The goal is to improve the overall camping experience while ensuring safety and efficiency.

Difficulties:

- Using JavaFx for the GUI.
- Work as a team.