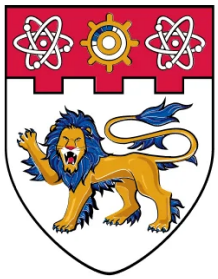


NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

SC2002: Object Oriented Design and Programming

Project Title: Camp Application and Management System (CAMs)

Lab Group: SCSZ

Assignment Group 1

Name	Matriculation Number
Chan Hin Wai Howell	U2221335D
Cheng Lin	U2222079E
Lee Ern Qi Eunice	U2220339J
Tay Wei Hong	U2220777G
Yong Shun Jie	U2221938C

1. Introduction

This report delves into the design considerations implemented in our Camp Application and Management System (CAMs) project. The overall architecture of our project is the **Model-View-Controller (MVC) design pattern**. The main application calls upon the classes, which are segmented into Model, View, Controller and Database. The design principles, particularly the SOLID principles are used to design the codes to allow reusability, extensibility and maintainability of the code structure. Additionally, the project applies the four pillars of Object-Oriented Programming (OOP) to ensure a cohesive and efficient software structure. Our source code can be found here.

Term	Definition
Student	A user that studies at the school that can register and withdraw from camps.
Staff	A user that is responsible for creating, managing, and reporting on camps, as well as responding to and processing student enquiries and committee suggestions.
Attendee	Both camp committee and attendees
CampCommittee	Camp committee members, a role taken by students, who contribute to camp management.

Figure 1: Data Dictionary

2. Design Considerations

2.1 Design Patterns

Model-View-Controller (MVC)

In our project, we adopted the Model-View-Controller (MVC) architectural pattern, which divides the program into three distinct components: the model, view, and controller (Figure 2.1.1). This separation facilitates the development process by ensuring that each component addresses specific aspects of the application. **High cohesion and low coupling** are achieved as classes are designated to individual responsibilities. The **Model** component is tasked with representing the data or state of an object. The Model component works together with the **Database** classes to store User, Camp, Enquiry and Suggestion data. The **View** is the main UI of the application. The **Controller** serves as the intermediary, coordinating between the View and Model. It updates and retrieves information from the Model while also handling input and data from the View.

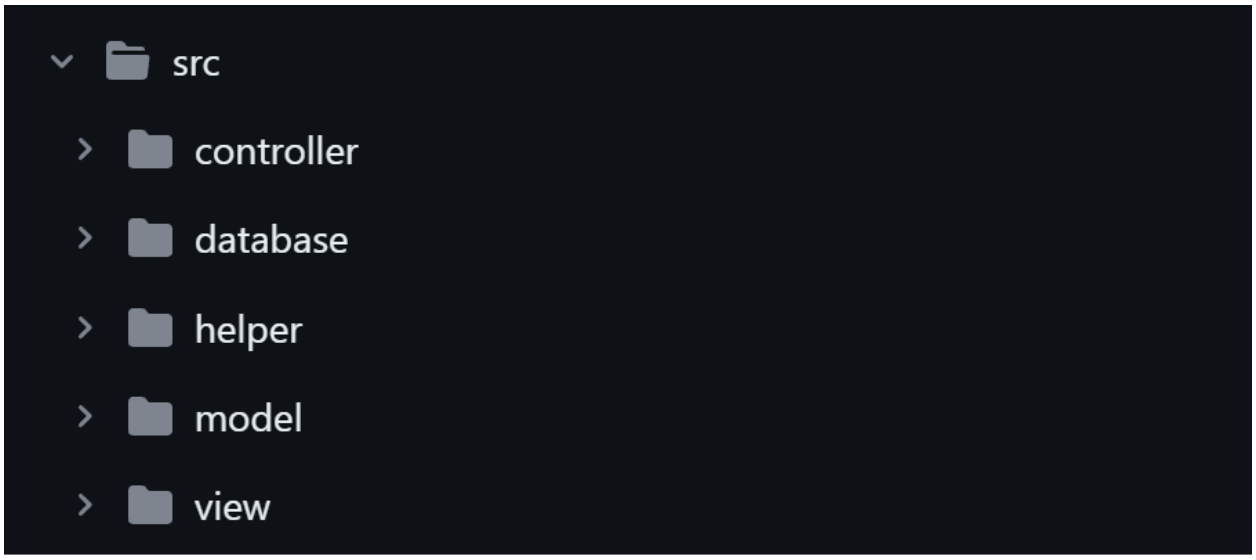


Figure 2.1.1: MVC pattern in our code

Initially, we adopted the Entity, Boundary and Control (EBC) Class pattern, but have made the switch to the MVC design pattern due to its distinct advantages over EBC. **(1)** MVC has a **clear separation of codes** by dividing the application into the user interface

(view), the data layer (Model and the control logic (Controller). This leads to a more organised and manageable codebase. **(2)** Additionally, MVC's modular structure improves the **scalability and maintainability** of the CAM application. Each component can be independently modified or scaled. This is in contrast to the tightly coupled entities and boundaries in EBC. **(3)** MVC facilitates improved **testability**. It allows us to conduct unit testing on individual components without interdependencies.

Singleton

The Singleton design pattern is a software design principle that ensures a class has only one instance. This pattern is useful when an object is needed to coordinate actions across the system. This is applied in the database classes (E.g. **EnquiryDB)**. This class manages a collection of enquiries with the system. A Singleton provides a global point of access to it. This helps to prevent conflicts that could arise from having multiple instances managing different sets of enquiry data.

```
23        // Static method to create instance of Singleton(CampDB) class
24        public static synchronized EnquiryDB getInstance()
25        {
26            if (enquiryDB == null)
27                enquiryDB = new EnquiryDB();
28
29            return enquiryDB;
30        }
```

Figure 2.1.2: Codes of singleton design pattern in EnquiryDB

2.2 Design Principles (S,O,I of SOLID)

Single Responsibility Principle

| "A class should have one, and only one, reason to change"

This design principle is implemented throughout all components and classes. Each class is aimed to achieve only one purpose. This approach results in low coupling, meaning that changes in one class do not necessitate large modifications in others. For instance, in the View component, there are multiple view classes such as **LoginView**

and **ReportView** which are dedicated to displaying the UI of the specific functions (e.g: login, report, etc.) of the application (Figure 2.2.1).

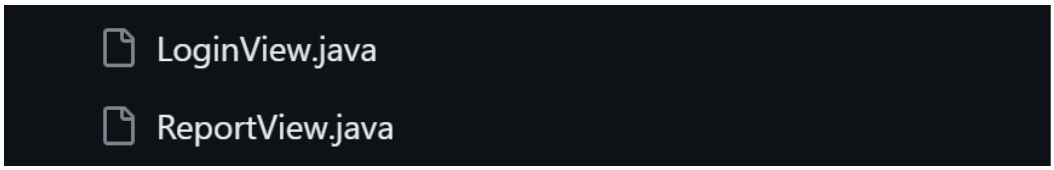


Figure 2.2.1: Classes of the LoginView and ReportView

Another example will be the **various helper functions** in the helper folder. Each of the helper functions is responsible for a specific function. For instance, the **FilerHelper** class reads and writes a file, and the UserDB database uses the FilerHelper class to read the student and staff text files (Figure 2.2.2: Codes of the FilerHelper function).

```
14  public class FileHelper {
15
16      /**
17       * Reads student/staff txt file
18       * @param filepath The filepath of student/staff txt file
19       * @return List of strings, each element containing [name email faculty]
20       */
21      public List<String> readFile(String filepath) {
22          List<String> lines = new ArrayList<>();
23          // Construct the absolute path based on the working directory and the relative path
24          String absolutePath = System.getProperty("user.dir") + File.separator + filepath;
25
26          try (FileReader fr = new FileReader(absolutePath);
27               BufferedReader br = new BufferedReader(fr)) {
28
29              String line;
30
31              while ((line = br.readLine()) != null) {
32                  lines.add(line);
33              }
34          } catch (IOException e) {
35              System.err.println("An error occurred while reading the file: " + e.getMessage());
36          }
37          return lines;
38      }
```

Figure 2.2.2: Codes of the FileHelper function

Open Closed Principle

|“Open for extension, closed for modification”

Currently, the program is working with two users, Student and Staff. However, as the project grows, new users may need to be added like camp trainers. Adhering to the OCP ensures that the system is able to accommodate new types of users without needing significant changes to the existing codebase.

In addition, in the **FileHelper** class, it is utilised for reading and writing files. OCP is being implemented here as this class is able to read new files if needed, like a file of camp trainers. Thus, the existing code structure does not need extensive modification whenever a new user type is added.

Interface Segregation Principle

| “Clients should not be forced to depend upon interfaces that they do not use”

Interface segregation principle means that an interface serves a single specific purpose. Consequently, the classes that implement this interface do not inherit methods that they do not use. In our example, our **BaseController** interface serves a singular purpose, which is to set the master variables (Figure 2.2.3). In our context, the master variables are variables that are utilised in the controller classes.

```
1   package controller;
2
3   /**
4    * The {@code BaseController} interface is responsible for setting master variables for all controllers.
5    * It is implemented by all controllers.
6    * @author Chan Hin Wai Howell
7    * @version 1.0
8    * @since 2023-11-02
9    */
10
11   // This interface is used to ensure that all controllers have a setMasterVariables() method.
12  public interface BaseController {
13      /***
14       * Abstract function to set the master variables for a controller.
15       */
16      void setMasterVariables();
17  }
```

Figure 2.2.3: Codes of the BaseController class

Dependency Injection Principle implemented with Controllers (Outside of SOLID)

Dependency injection is a pattern whereby an object is provided with their dependencies instead of creating or asking for dependencies. In our system, the controllers initialise dependencies in their respective constructors. This ensures decoupling as the controller would not need to call for other dependencies, instead they are directly injected into the controller.

2.3 Object Oriented Concepts

Abstraction

Abstraction is where a programmer hides all but the relevant data about an object in order to reduce complexity and efficiency. This is achieved by creating a **BaseController** interface, which abstracts away the implementation details of how **setMasterVariables()** works. This is such that when a user works on classes that implement the **BaseController** interface, the user need not be aware of the implementation details.

```
// This interface is used to ensure that all controllers have a setMasterVariables() method.
public interface BaseController {
    /***
     * Abstract function to set the master variables for a controller.
     */
    void setMasterVariables();
}
```

Figure 2.3.1: Private attributes of the Student class

Encapsulation

Encapsulation is used to build a barrier around an object's data, preventing the unauthorized access of data. This is achieved using the private access modifiers in the classes. For instance, in the **Camp** class, the HashMaps of attendees and committees are set to private, and in the **User** class, attributes like userID and name are private (Figure 2.3.1). These data can only be accessed using the public methods, get and set methods. These methods allow the data to be accessible and modified.

```java
public class User {
    private final String userID;
    private final Faculty faculty;
    private String password = "password";
    private final String name;
    private boolean firstTimeRegistered;
```

Figure 2.3.1: Private attributes of the Student class

Inheritance

Inheritance allows us to derive new classes from existing classes. The child class inherits data attributes and methods from the parent class. This “is-a” relationship between the child class and parent class can be visualized in our class diagram. Inheritance is helpful as it reduces developer effort and code length, as it enables the reuse of code when implementing new classes. For example, in the model component, the **Student** and **Staff** classes are inheriting methods from the **User** class (Figure 2.3.2). This allows common methods such as getID and getFaculty to be inherited and reused.

```java
public class Student extends User {
    private final HashMap<String, Camp> previouslyRegisteredCamps;
    private final HashMap<String, Camp> registeredCamps; //Camps where student is a member
    private Camp registeredCommitteeCamp;
    private boolean isCampCommitteeMember = false;
    private int points = 0;
```

Figure 2.3.2: Student class is inheriting from user class

Polymorphism

Polymorphism allows objects of different types to be treated as objects of a common type. Polymorphism can be implemented through overriding of methods. When implementing the **BaseController** in the various controller classes, each controller class overrides the **setMasterVariables()** method by creating objects essential for its class. Hence, everytime a controller object is created, the overridden method in the specific controller class will be invoked, allowing each controller to initialize its essential objects. This is a powerful mechanism that promotes code reuse, flexibility, and maintainability.

```java
// This interface is used to ensure that all controllers have a setMasterVariables() method.
public interface BaseController {
    /**
     * Abstract function to set the master variables for a controller.
     */
    void setMasterVariables();
}
```

3. UML Class Diagram

Please refer to the “Class Diagram.png” for our UML Class Diagram.

4. Reflection

Understanding Requirements:

We first prioritized a comprehensive understanding of the camp management system's requirements. We understood that the students and staff have different responsibilities and levels of accessibility to information. We are aware that the students have different roles like camp committee member or participant of a camp. We had to draw out an initial conceptual class diagram to sort out overlapping and complex functionalities. Making use of agile methodology where program specification, design and implementation are interleaved allowed for an iterative process that enabled us to refine and clarify specifications along the way. There were frequent deliveries of new versions for evaluation and the outputs from the development process were decided through a process of negotiation during the software development of the camp management system,leading to a more accurate representation of the system's functionalities.

Areas of improvement:

A feature that could be implemented is to allow the camp committee members and staff to edit their replies to enquiries. Currently, once an enquiry is replied to, nothing can be changed to it. Hence, in the event the response to the enquiry needs to be updated, it cannot be done. To counter this, we could use similar methods to that of editing enquiries and suggestions before they are replied to and accepted respectively.

We can have another function in the menu bar where users of the camp management system can provide feedback for the system. This can help us gather useful feedback from the customers’ perspective. This can help to identify pain points efficiently and respond to such areas of improvement immediately.

Currently, we are only testing the camp management system with a small number of users. We understand that in a real-life system, a large amount of data will be handled. Therefore, it is important to select a database system to help store our information in the future. NoSQL databases like MongoDB and Cassandra can be used.

We can have a more comprehensive filter system that filters out specific camps or participants. Currently, the filter system can only filter students according to whether they are camp committee members or attendees. Hence, we can implement more filters (dates, location, etc) to help expedite the process of pulling desired information from our camp management system.

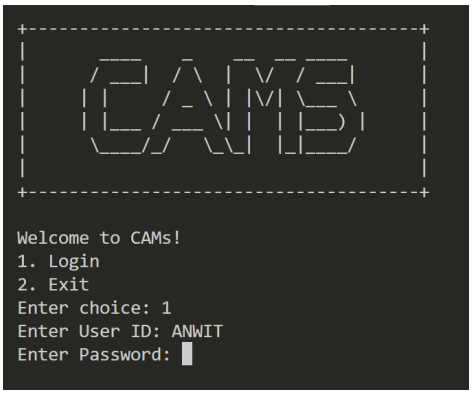
We can deploy an entire web app for this instead of compromising to CLI. This means we can track a full user session and all the data will be kept in a single session instead of having to restart the terminal every time. This brings greater convenience and is more user friendly.

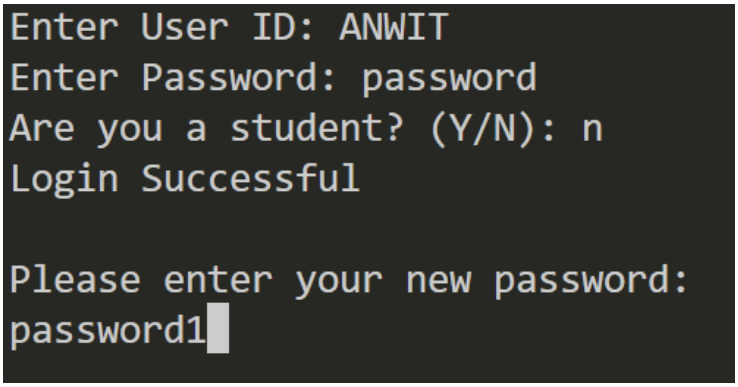
Lastly, we could potentially even develop a recommendation system that can recommend camps to students based on their search history and previous camps.

5. Test Cases

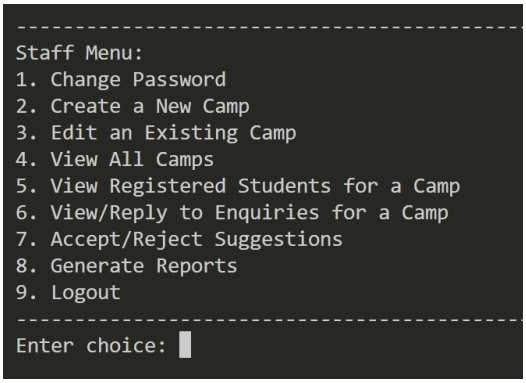
5.1 Login

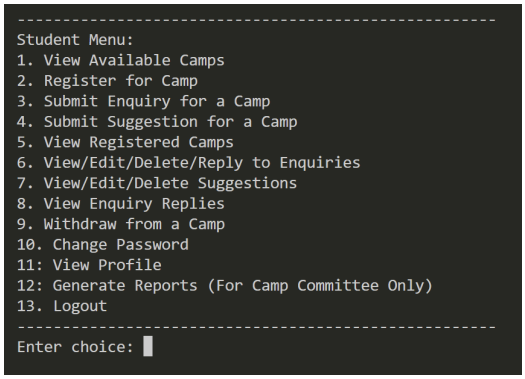
Login Page (Successful)





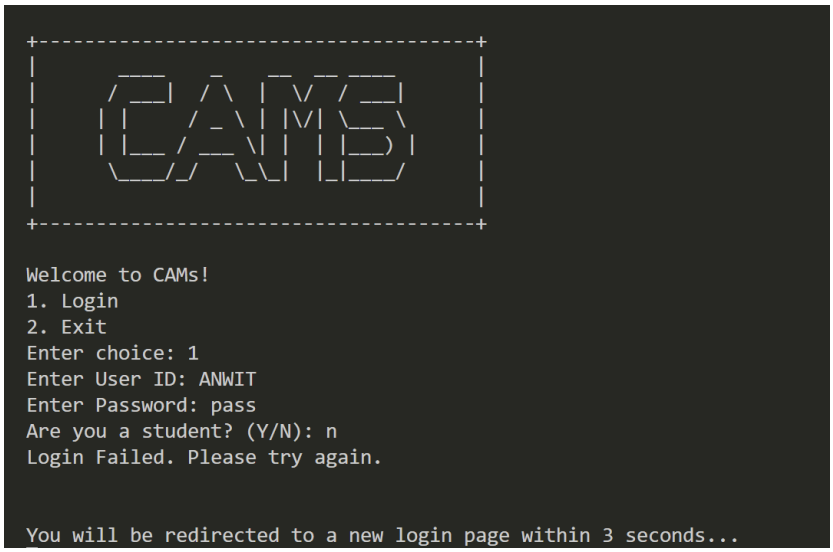
Upon entering the system, the user can choose to either **login** or **exit**. The user will also then indicate whether they are a student. When logged in successfully, the user is prompted to **change password**.





Here we present the **Staff Menu** (left) and **Student Menu** (right) as examples.

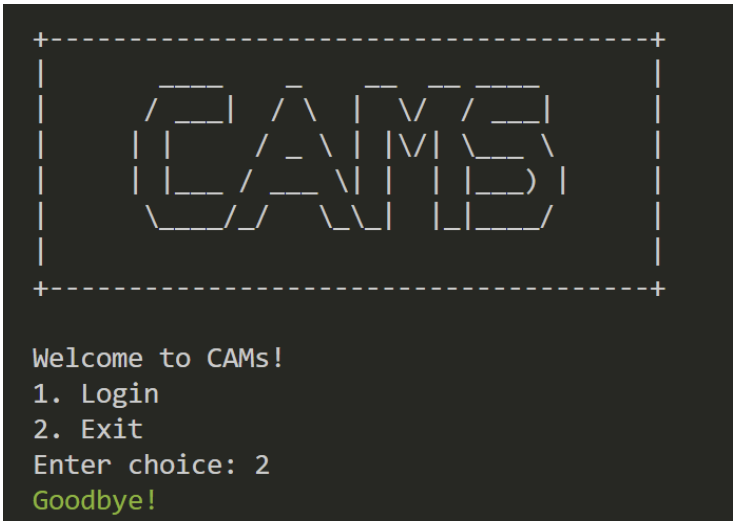
Login Page (Failed)



If the user types in the **wrong user credentials**, they are prompted to attempt logging in again.

The user is also **redirected** back to the login page after 3 seconds.

Logout



5.2 Staff Camp Management Options

Staff Creates A Camp

```
+------------------------------------------------------------+
|                                                            |
|                    CAMP CREATION PORTAL                    |
|                                                            |
|                                                            |
|        To return to the main menu, simply enter '0'.       |
|                                                            |
+------------------------------------------------------------+

Enter Camp Name: ori
Enter start date (dd-mm-yyyy): 21-12-2023
Enter end date (dd-mm-yyyy): 23-12-2023
Enter Registration Closing Date (dd-mm-yyyy): 18-12-2023
Enter Faculty: SCSE
Enter Location: NTU
Enter Total Slots: 10
Enter Committee Slots: 8
Enter Description: Free Food and Drinks!
Is it visible? (true/false): true
```

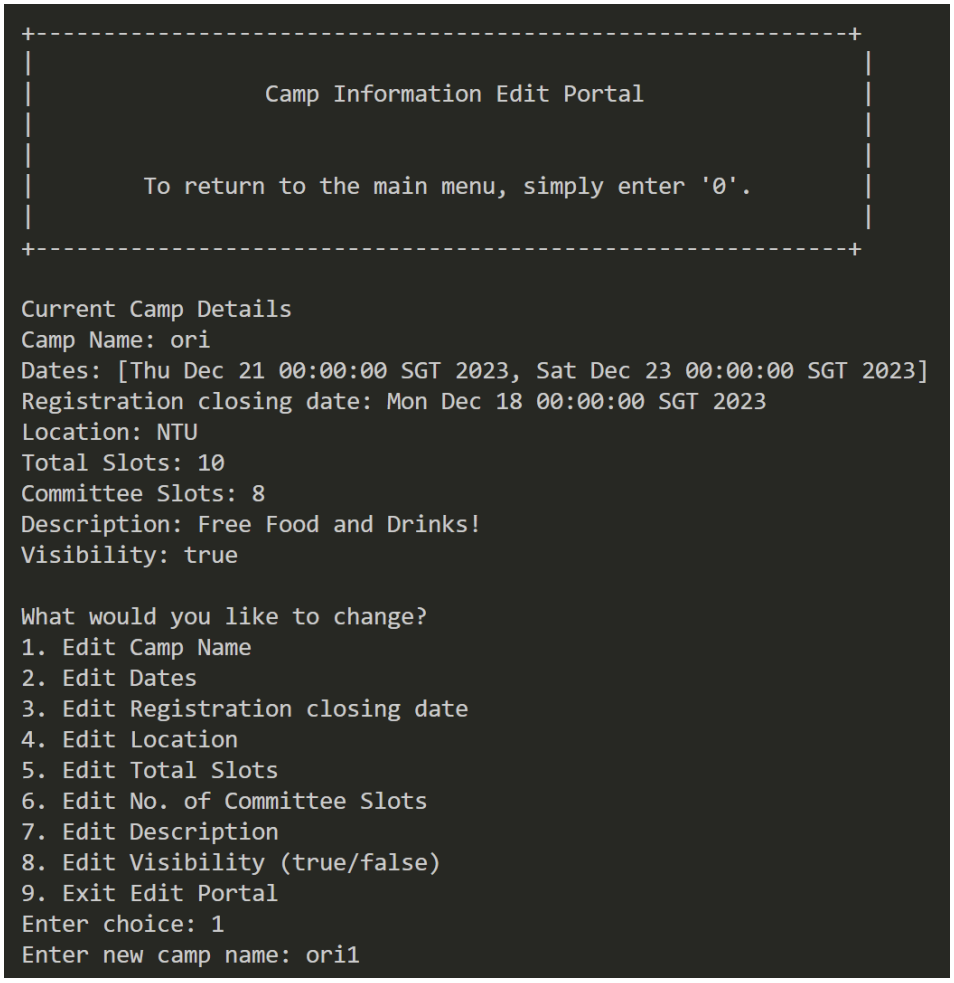
```
+------------------------------------------------------------+
|                                                            |
|                    CAMP DETAILS PORTAL                     |
|                                                            |
|                                                            |
|        To return to the main menu, simply enter '0'.       |
|                                                            |
+------------------------------------------------------------+

Camp Details
Camp Name: ori
Dates: [Thu Dec 21 00:00:00 SGT 2023, Sat Dec 23 00:00:00 SGT 2023]
Registration closing date: Mon Dec 18 00:00:00 SGT 2023
Location: NTU
Total Slots: 10
Remaining Slots: 10
Description: Free Food and Drinks!
Staff In Charge: Datta
Visibility: true
```

The staff may choose to **create a camp** with the information above. **Confirmed camp details** are later displayed after the camp is created.

Staff Edits A Camp

```
+------------------------------------------------------------+
|                                                            |
|                    LIST OF CAMPS You Manage                |
|                                                            |
|                                                            |
|        To return to the main menu, simply enter '0'.       |
|                                                            |
+------------------------------------------------------------+

List of Camps:

1. Camp Name: ori
   Total Slots: 10
   Remaining Slots: 10
   Remaining Committee Slots: 8
Which camp would you like to edit?
```

The staff may choose to **edit a camp** that they had previously created.



The staff may select the exact type of **camp information** to edit.

Staff Can View All Camps

```
+------------------------------------------------------------+
|                                                            |
|                       LIST OF ALL CAMPS                    |
|                                                            |
|                                                            |
|        To return to the main menu, simply enter '0'.       |
|                                                            |
+------------------------------------------------------------+

List of Camps:

1. Camp Name: SCSE Camp
   Total Slots: 9
   Remaining Slots: 9
   Remaining Committee Slots: 7
2. Camp Name: ori1
   Total Slots: 10
   Remaining Slots: 10
   Remaining Committee Slots: 8
```

Staff can view a **list of all camps**, including those created by other staff.

Staff Can View All Registered Students for a Camp

```
+------------------------------------------------------------+
|                                                            |
|                       LIST OF ALL CAMPS                    |
|                                                            |
|                                                            |
|        To return to the main menu, simply enter '0'.       |
|                                                            |
+------------------------------------------------------------+

List of Camps:

1. Camp Name: Ori
   Total Slots: 10
   Remaining Slots: 8
   Remaining Committee Slots: 7
Select a camp to view registered students for:
Ori

List of Camp Committee Members:
1. CHERN

List of Other Camp Attendees:
1. LEE
```

Staff can view a **list of all registered students in a camp**.

5.3 Student Camp Management Options

Student Registers for a Camp

```
+------------------------------------------------------------+
|                                                            |
|                    LIST OF CAMPS YOU ARE IN                |
|                                                            |
|                                                            |
|        To return to the main menu, simply enter '0'.       |
|                                                            |
+------------------------------------------------------------+

List of Camps:

1. Camp Name: SCSE Camp
    Total Slots: 9
    Remaining Slots: 9
    Remaining Committee Slots: 7
2. Camp Name: ori1
    Total Slots: 10
    Remaining Slots: 10
    Remaining Committee Slots: 8
```

Students may **view all the camps** that have been created and **register** for the one they desire. Note that number of remaining slots and remaining committee slots will be reduced by 1 each as a result.

```
Type the number of the camp you would like to register for!
1

Would you like to register as a participant or a committee member?
1. Participant
2. Committee Member
2
You are registered!
```

Student may choose to register as either a participant or a committee member.

Student Withdraws from a Camp

```
+------------------------------------------------------------+
|                                                            |
|                    LIST OF CAMPS YOU ARE IN                |
|                                                            |
|                                                            |
|        To return to the main menu, simply enter '0'.       |
|                                                            |
+------------------------------------------------------------+

List of Camps:

1. Camp Name: SCSE Camp
    Total Slots: 9
    Remaining Slots: 8
    Remaining Committee Slots: 6
    Role: Camp Committee
Type the name of the camp you would like to withdraw from.
SCSE Camp
You have withdrawn!
```

Students may **view all the camps** they are in and select the one they wish to withdraw from.

Student Views User Profile

```
+------------------------------------------------------------+
|                                                            |
|                      YOUR USER PROFILE                     |
|                                                            |
|                                                            |
|        To return to the main menu, simply enter '0'.       |
|                                                            |
+------------------------------------------------------------+

User Profile
User ID: YCHERN
Faculty: SCSE
Committee Member Status: True
```

Students may **view their user profiles**..

5.3 Student Camp Enquiry/Suggestion Submission

Student Submits Enquiry

```
+------------------------------------------------------------+
|                                                            |
|                    LIST OF CAMPS YOU ARE IN                |
|                                                            |
|                                                            |
|        To return to the main menu, simply enter '0'.       |
|                                                            |
+------------------------------------------------------------+

List of Camps:

1. Camp Name: SCSE Camp
    Total Slots: 9
    Remaining Slots: 8
    Remaining Committee Slots: 7
2. Camp Name: ori1
    Total Slots: 10
    Remaining Slots: 9
    Remaining Committee Slots: 8
Type the name of the camp you would like to submit an enquiry for!
SCSE Camp
Type the enquiry you would like to make!
Are we to stay overnight?
Enquiry submitted successfully.
```

Students may **submit an enquiry**.

Student Submits Suggestion

```
+-------------------------------------------------------------+
|                                                             |
|                       SUGGESTIONS PORTAL                    |
|                                                             |
|                                                             |
|       To return to the main menu, simply enter '0'.         |
|                                                             |
+-------------------------------------------------------------+

Your committee camp name is: SCSE Camp

Type the number of the information you would like to make a suggestion for:
1. Current Camp Name: SCSE Camp
2. Current Camp Description: For SCSE students only
3. Current Camp Location: NTU
4. Current Camp Start Date: Sun Dec 10 00:00:00 SGT 2023
5. Current Camp End Date: Tue Dec 12 00:00:00 SGT 2023
6. Current Camp Registration Closing Date: Thu Nov 30 00:00:00 SGT 2023
7. Current Camp Total Slots: 9
8. Current Camp Committee Slots: 7
2
Please enter your suggestions for the camp description:
For all students interested in CS
Suggestion successfully submitted!
+1 point for making a suggestion
```

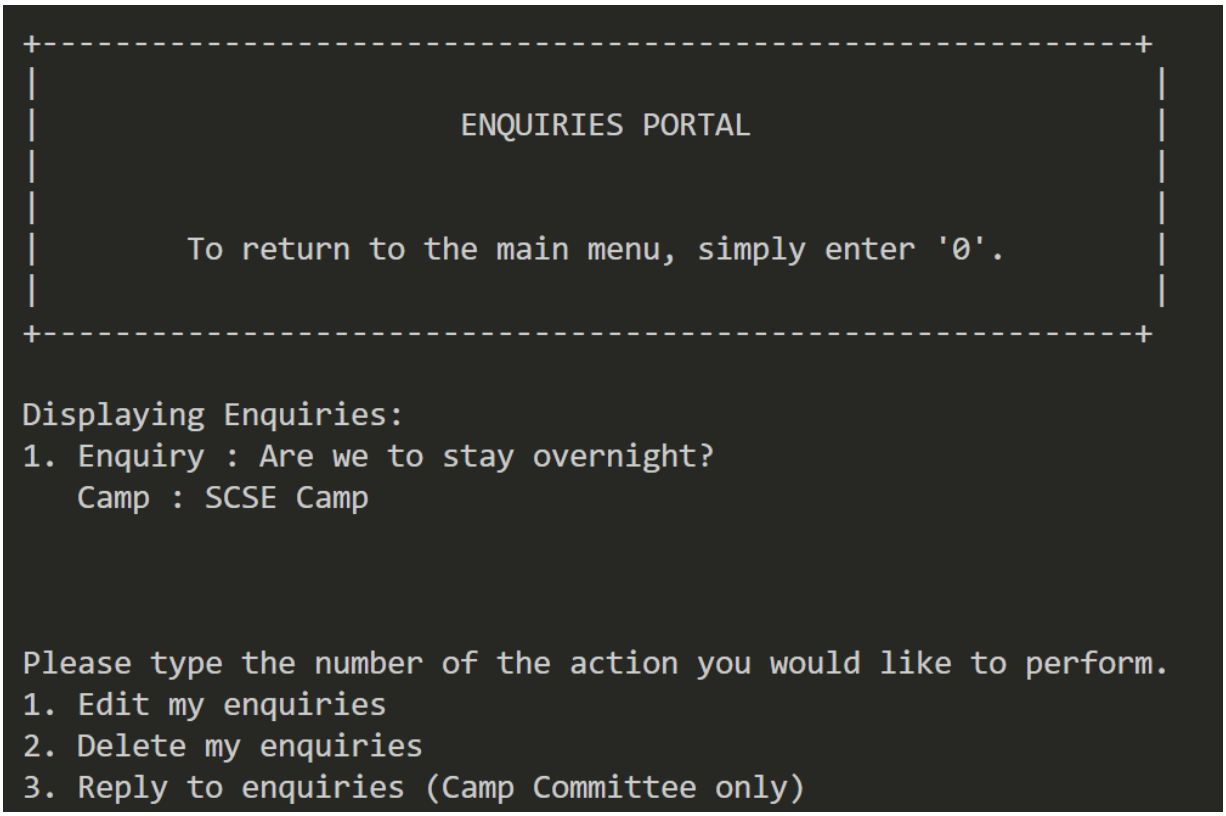
Camp committee members may **submit suggestions** to change any of the **camp information** for each **respective camp**.

```
You are not a committee member, so you can't make suggestions!
```

Participants are not allowed to make suggestions, so they received the above message if they attempt to do so.

5.3 Student Enquiries Portal

Enquiries Portal for Students



Students may choose to edit, delete or reply to enquiries. However, only camp committee members can choose to reply to enquiries

Student Can Edit Enquiries

```
+------------------------------------------------------------+
|                                                            |
|                       ENQUIRIES PORTAL                     |
|                                                            |
|                                                            |
|       To return to the main menu, simply enter '0'.        |
|                                                            |
+------------------------------------------------------------+

Displaying Enquiries:
1. Enquiry : Are we to stay overnight?
   Camp : SCSE Camp



Type the enquiry you would like to edit!
1

Type the new enquiry you would like to make!
What are we doing at night?
Enquiry edited successfully.
```

Students can **select the enquiry they wish to edit** and **type the new one they want to make**. They then receive a message that it is edited successfully.

Student Can Delete Enquiries

```
+------------------------------------------------------------+
|                                                            |
|                       ENQUIRIES PORTAL                     |
|                                                            |
|                                                            |
|       To return to the main menu, simply enter '0'.        |
|                                                            |
+------------------------------------------------------------+

Displaying Enquiries:
1. Enquiry : What are we doing at night?
   Camp : SCSE Camp



Type the number of the enquiry you would like to delete!
1
Enquiry deleted successfully.
```

Students can **select the enquiry they wish to delete.** They then receive a message that it is deleted successfully.

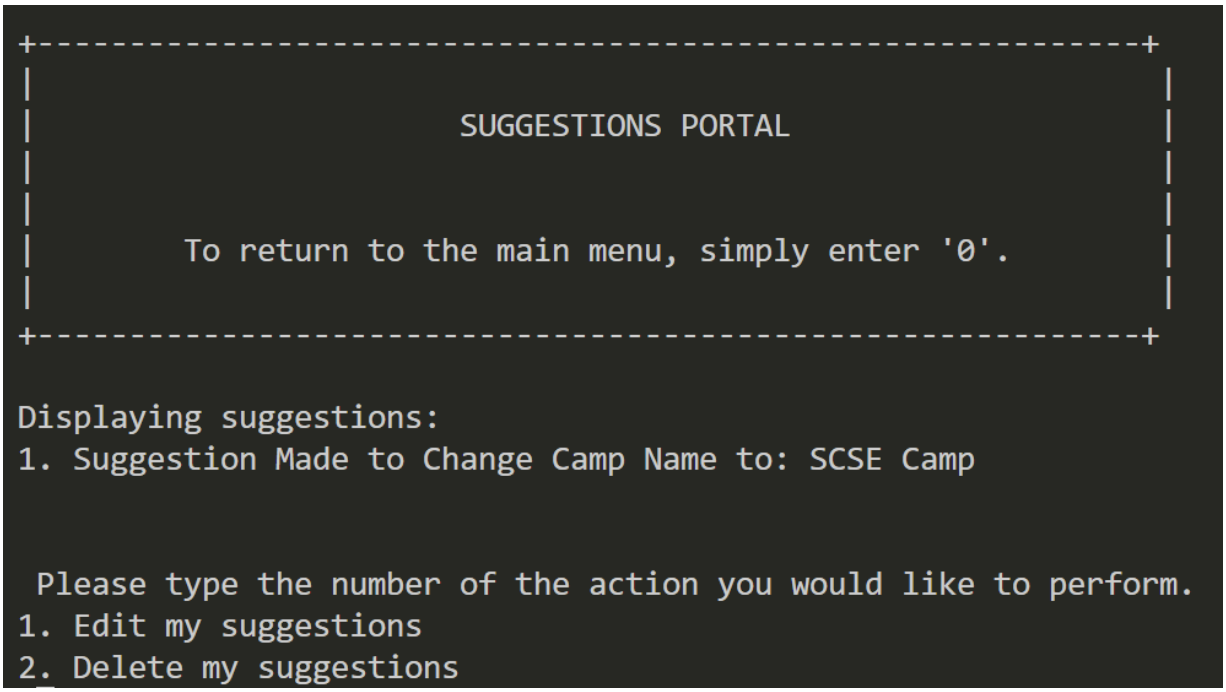
Camp Committee Member Can Reply to Enquiries

```
Please type the number of the action you would like to perform.
1. Edit my enquiries
2. Delete my enquiries
3. Reply to enquiries (Camp Committee only)
3
Your committee camp name is: SCSE Camp

+-------------------------------------------------------------+
|                                                             |
|                       ENQUIRIES PORTAL                      |
|                                                             |
|                                                             |
|        To return to the main menu, simply enter '0'.        |
|                                                             |
+-------------------------------------------------------------+

Displaying Enquiries:
1. Enquiry : Is it held outdoors?
   Camp : SCSE Camp



Type the number of the enquiry you would like to reply to!
1

Type the reply you would like to make!
Activities are all outdoors except meals!

Enquiry replied successfully.
+1 point for replying to an enquiry!
```

Camp committee members can **select the enquiry they wish to reply to.** They can then type the reply and receive a point for replying to the enquiry.

5.3 Student Suggestions Portal

Suggestions Portal for Students



Camp committee members may choose to edit or delete suggestions.

Camp Committee Members Can Edit Suggestions

```
+------------------------------------------------------------+
|                                                            |
|                       SUGGESTIONS PORTAL                   |
|                                                            |
|                                                            |
|        To return to the main menu, simply enter '0'.       |
|                                                            |
+------------------------------------------------------------+

Displaying suggestions:
1. Suggestion Made to Change Camp Name to: SCSE Camp

Enter the number of the suggestion you want to edit:
1
You Suggested to Change Camp Name to: SCSE Camp

Type edited suggestion:
Tech Camp
Suggestion successfully updated!
```

Students may choose to **edit suggestions**.

Camp Committee Members Can Delete Suggestions

```
+------------------------------------------------------------+
|                                                            |
|                       SUGGESTIONS PORTAL                   |
|                                                            |
|                                                            |
|        To return to the main menu, simply enter '0'.       |
|                                                            |
+------------------------------------------------------------+

Displaying suggestions:
1. Suggestion Made to Change Camp Name to: Tech Camp

Enter the number of the suggestion you want to delete:
1
Suggestion successfully deleted!
```

Students may choose to **delete suggestions**.

5.4 Staff Enquiries Portal

Staff can view and reply to enquiries for the camps they created

```
+------------------------------------------------------------+
|                                                            |
|                       ENQUIRIES PORTAL                     |
|                                                            |
|                                                            |
|        To return to the main menu, simply enter '0'.       |
|                                                            |
+------------------------------------------------------------+

Displaying Enquiries:
1. Enquiry : Is this outdoors?
   Camp : Ori
   Response : Yes everything is outdoors

2. Enquiry : Is there food provided?
   Camp : Ori



Type the number of the enquiry you wish to reply to.
2
Type the reply you would like to make!
Yes meals r provided
Enquiry replied successfully.
```

Staff may choose to reply to enquiries made by students.

5.5 Staff Suggestions Portal

Staff can view and reply to suggestions for the camps they created

```
+------------------------------------------------------------+
|                                                            |
|                  LIST OF CAMPS You Manage                  |
|                                                            |
|                                                            |
|        To return to the main menu, simply enter '0'.       |
|                                                            |
+------------------------------------------------------------+

List of Camps:

1. Camp Name: Ori
   Total Slots: 10
   Remaining Slots: 8
   Remaining Committee Slots: 7
Enter the number of the camp you want to view suggestions for:
1

+------------------------------------------------------------+
|                                                            |
|                      SUGGESTIONS PORTAL                    |
|                                                            |
|                                                            |
|        To return to the main menu, simply enter '0'.       |
|                                                            |
+------------------------------------------------------------+

Displaying suggestions:
1. Suggestion Made to Change Camp Name to: SCSE Camp

Enter the number of the suggestion you want to approve/reject:
1
Enter 1 to approve, 2 to reject: 1
Suggestion successfully approved!
```

Staff may choose to **approve or reject suggestions** made by students.

5.6 Report

Camp Committee can generate report of list of students of each camp they oversee

```
Camp Information of Ori

Camp Name: Ori
Camp Description: time to have fun
Camp Location: NTU

Camp Dates: [Wed Dec 20 00:00:00 SGT 2023, Sat Dec 23 00:00:00 SGT 2023]
Camp Registration Closing Date: Fri Dec 15 00:00:00 SGT 2023

Camp Faculty: SCSE

Camp Total Slots: 10
Camp Committee Slots: 2

Camp In Charge: Datta

Camp Attendee Details:
CALVIN(User ID: CT113)| Attendee, CHERN(User ID: YCHERN)| Committee Member , LEE(User ID: LE51)| Attendee

```

Camp Committee members can generate a report of the list of attendees of the camp they oversee

```
Camp Information of Ori

Camp Name: Ori
Camp Description: time to have fun
Camp Location: NTU

Camp Dates: [Wed Dec 20 00:00:00 SGT 2023, Sat Dec 23 00:00:00 SGT 2023]
Camp Registration Closing Date: Fri Dec 15 00:00:00 SGT 2023

Camp Faculty: SCSE

Camp Total Slots: 10
Camp Committee Slots: 2

Camp In Charge: Datta

Camp Attendee Details:
CHERN(User ID: YCHERN)| Committee Member
```

Camp Committee members can generate a report of the list of committee members of the camp they oversee

Staff can generate report of list of students of each camp they created

```
Camp Information of Ori

Camp Name: Ori
Camp Description: time to have fun
Camp Location: NTU

Camp Dates: [Wed Dec 20 00:00:00 SGT 2023, Sat Dec 23 00:00:00 SGT 2023]
Camp Registration Closing Date: Fri Dec 15 00:00:00 SGT 2023

Camp Faculty: SCSE

Camp Total Slots: 10
Camp Committee Slots: 1

Camp In Charge: Datta

Camp Attendee Details:
CALVIN(User ID: CT113)| Attendee
CHERN(User ID: YCHERN) | Committee Member
LEE(User ID: LE51) | Committee Member

```

Staff can generate a report of the list of attendees of the camp they created

```
Camp Information of Ori

Camp Name: Ori
Camp Description: time to have fun
Camp Location: NTU

Camp Dates: [Wed Dec 20 00:00:00 SGT 2023, Sat Dec 23 00:00:00 SGT 2023]
Camp Registration Closing Date: Fri Dec 15 00:00:00 SGT 2023

Camp Faculty: SCSE

Camp Total Slots: 10
Camp Committee Slots: 1

Camp In Charge: Datta

Camp Attendee Details:
CHERN(User ID: YCHERN) | Committee Member
LEE(User ID: LE51) | Committee Member
```

Staff can generate a report of the list of committee members of the camp they created

Staff can generate a performance report of the camp committee members

```
Camp Performance Report for Ori

List of Committee Members:

Student ID: YCHERN
Name: CHERN
Points: 2
----------------------------------------

Student ID: LE51
Name: LEE
Points: 1
----------------------------------------
```

Staff can generate a performance report of the committee members in the camp they created

Staff can generate an Enquiry Report of each camp they created

```
Student Enquiry Report for Ori

List of Enquiries:

Student ID: CALVIN
Enquiry Text: Is the camp outdoors?
Response: Yes some of the activities are!
----------------------------------------

Student ID: LEE
Enquiry Text: is food provided?
Response: No response yet
----------------------------------------
```

Staff can generate am enquiry report of the camp they created