SC2002: : Object-Oriented Design & Programming
AY 2023/2024 SEMESTER 1

SC2002 Report

Group: 2

Team Members	Matriculation Number
Tian Qiuzaicheng	U2221288C
Yves Samson Li	U2222560G
Chong Huai Zhi	U2221000J
Lee Jing Yang, Joshua	U2230288D
Tan Jing Han Chad	U2222125G

Date: 15 November 2023

<u>Declaration of Original Work for CE/CZ2002 Assignment</u>

We hereby declare that the attached group assignment has been researched, undertaken, completed, and submitted as a collective effort by the group members listed below.

We have honored the principles of academic integrity and have upheld Student Code of Academic Conduct in the completion of this work.

We understand that if plagiarism is found in the assignment, then lower marks or no marks will be awarded for the assessed work. In addition, disciplinary actions may be taken.

Name	Course (CE2002 or CZ2002)	Lab Group	Signature /Date
Chad Tan	SC2002	SCSZ	[illegible]
Chong Huai Zhi	SC2002	SCSZ	[illegible]
Tian Qiuzaicheng	SC2002	SCSZ	[illegible]
Lee Jing Yang Joshua	SC2002	SCSZ	[illegible]
Yves Samson Li	SC2002	SCSZ	[illegible]

Table of Contents

Application Overview

The Camp Application and Management System (CAMs), is an application designed to facilitate camp-related activities for both staff and students. The application will act as a centralized hub which provides users a range of functionalities including camp management, viewing, and registration. The main system consists of different interfaces that can be navigated using command-line interface (CLI). The user interface will make use of user subclasses, which implements an abstract method prompt that guides users to perform specific actions, all userAction subclasses implement the act method for user classes to use, and only allow related information to be displayed. Various roles will have their own display subclasses that are managed by respective display manager/controller subclasses.

Design Considerations

Requirements

1. Every user will have a unique user ID, only 1 user can access the system at one time.
2. Each camp will consist of 1 supervising staff, name, descriptions, slot number, attendee list, duration of camp.
3. Staff can also view and reply to enquiries from students to the camp(s) they created.
4. Students are able to withdraw from the camp but will not be able to rejoin it.
5. A committee member can read and reply to suggestions written by attendees.
6. Committee members can write and edit enquiries to staff.
7. User ID would be NTU network userID.
8. Upon login all users will be set a default password which they can change later.
9. Users can be initiated through files uploaded into the system at initialization.
10. Staff will be able to create, edit and delete camps and have the options to toggle visibility of the camps for students.

11. A staff can view and approve suggestions to changes to camp details from camp committee.
12. A staff member can generate a report of the list of students attending each camp that his/her has created. The list will include details of the camp as well as the roles of the participants. (generated in either txt or csv format).
13. A staff can also generate a performance report of the camp committee members.
14. A student can only view the list of camps that are open to his/her user group (SCSE,
15. whole NTU etc.) and if their visibility has been toggled "on".

Use of Object Oriented Concepts

The following paragraph addresses how the CAMs design applies some principles of OOP.

1. Encapsulation

A fundamental principle of object-oriented programming, which involves hiding a class's internal data from external classes, only allowing them to access this information via public methods. In line with this, all classes in the CAMs have variables with private access modifiers and corresponding public getter and setter methods that ensure controlled access to internal data.

2. Polymorphism

Another principle of OOP which describes how subclasses can be treated as instances of their base class. This has various benefits, including code maintenance and readability. Examples would be staff account classes extending staff class.

3. Abstraction

Abstraction models real-world objects, in the sense that an entity would contain attributes and actions that describe the state and behavior of the entity respectively. Camp.java contains the details of a camp, whilst having its own methods that handle functions such as getTotalSlots() etc.

The front-end classes/controllers such as account_manager.java are able to use the methods of the back-end classes such as Student_User.java without knowledge of the implementations of the methods within Student_User.java. Likewise, Student_User.java only knows the existence of the various Users through account_manager.java.

4. Inheritance

   Inheritance allows us to derive new classes from existing classes, thereby making it easier to implement new classes and enables code reusability. Since a committee member is also a student. Committee.java inherits from Student_User.java all its attributes and methods, and further implements a point system which is unique only to committee members.

SOLID design principles

The following paragraph addresses how the CAMs takes SOLID design principles into consideration.

1. Single Responsibility Principle

   This principle recommends that a class should have a single responsibility, so that it will have only one reason to change. We have adhered to this principle in many classes within the module. For example, the *CSVWriter* class is solely responsible for adding new students and their associated information to an external .csv file. As such, the class will only be modified if there are changes to methods of persisting data on the external file. The *Enquiry* and *Suggestion* class are entity classes solely responsible for representing enquiry and suggestion-related information respectively, and providing getter and setter methods for other classes to interact with them. As a result, these classes will only be modified when changes need to be made to enquiry or suggestion representations. By increasing cohesion and reducing coupling, we have improved the extensibility and maintainability of our design.

2. Liskov-Substitution Principle (LSP)

   According to this principle, a subtype should be substitutable for its base type, so a user class expecting a base type should experience no difference when receiving a subtype. This is generally assured through Design by Contract, where a subtype does not require more preconditions (input) and less postconditions (output) than its base type. In our design, we represented student attendee and committee members as *Attendee* and *Committee* subclasses that both extend from a *Student_User* parent class. Both subclasses can be used interchangeably with *Student_User*, such as when a student (*Attendee* or *Committee*) calls the *withdrawfromCamp* method, thus adhering to the LSP.

3. Interface Segregation Principle (ISP)

According to this principle, interfaces should be client-specific and not be designed for general-purpose. In other words, interfaces should not have a large range of methods, such that some classes that extend them do not provide implementations for all methods. This promotes reduced coupling and enhanced readability. Furthermore, smaller interfaces comply with the Single Responsibility Principle. In our design, the application needed to be able to display camp lists and even output lists modified by various sorting options or keyword querying. While both functionalities similarly output modified lists, we decided to create separate abstract classes for sort and search options respectively to comply with the ISP. The *DisplaybySort* abstract class provides an abstract *Sorting* method and is only implemented by sorting option subclasses that extend the class, such as *SortByLocation*. The *DisplaybySearch* abstract class provides an abstract *Searching* method and is only implemented by searching option subclasses that extend the class, such as *SearchByLocation*.

4. Open-Closed Principle

This principle suggests that the module should be closed for modification but open for extension. In our design of *SortApp*, which guides the user in selecting a preferred camp sorting option and presents a sorted list of camps in accordance with user choice, we adhered to this principle. Within the method *startSorting*, a switch statement takes user choice, indicated by an integer from 1 to 7, and instantiates a *sorter* variable of DisplayBySort type with a subclass indicated by user choice. The method then returns a sorted ArrayList of camp by calling the *DisplayBySort* method *Sorting* on the sorter variable. In the event that CAMs requires changes to sorting options, this can take place with minimal modification to the *SortApp*, thus improving the code's extensibility and maintainability.

5. Dependency Inversion Principle

According to this principle, higher-level classes should not depend on lower-level classes and both should depend instead on an abstraction layer (eg. interfaces, abstract classes). Details should depend on abstractions. This has the benefit of reducing coupling and promoting reusability, extensibility and maintainability.

We applied this to the generate report function of our project. A report generated by a staff would have more information than that of a similar report generated by a committee member for a camp. In particular, both Staff and Committee Member can generate a report on the committee members list of the camp, but information on the performance points of each committee member is withheld, and only made available in the Staff's version of the committee report. Hence, Staff and Committee class are separated from CommitteeGenerateReport and StaffGenerateReport class with a GenerateReport interface in order to account for this feature without having to manually change the Staff and Committee class.

Scalability

1. Extensibility

By applying different OOP design principles, new features such as new user action, new requests or role type could be easily added by inheriting from existing relevant base classes and implementing the necessary method. Therefore requiring only minimal changes to the main code.

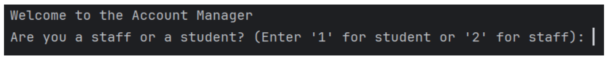
2. Maintainability

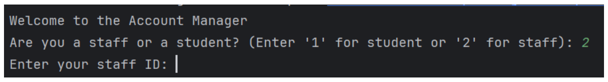
Through the fulfillment of various SOLID principles, our code has benefitted from improved maintainability. Our code is easily understandable and classes could be debugged independently by applying the OOP approach. It also allows high reusability of code.

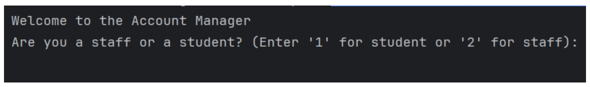
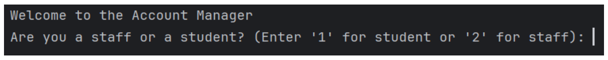
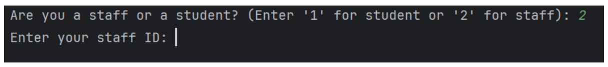
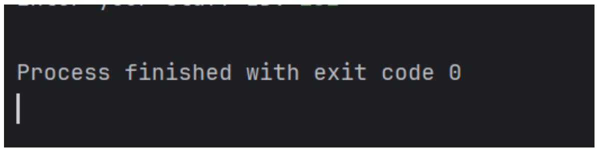
Test Cases

Please note that these test cases are not exhaustive.

Staff

Function	User Input	Program output
Staff Registration	Nil (start of program)	 `Welcome to the Account Manager` `Are you a staff or a student? (Enter '1' for student or '2' for staff):`

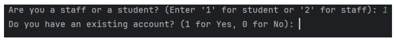
	Enter “2” for staff	 `Welcome to the Account Manager`
		`Are you a staff or a student? (Enter '1' for student or '2' for staff): 2`
		`Enter your staff ID:`

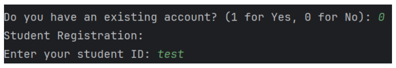
Start menu	Nil	 `Welcome to the Account Manager` `Are you a staff or a student? (Enter '1' for student or '2' for staff):`
Login for Staff	Nil (start of Program)	 `Welcome to the Account Manager` `Are you a staff or a student? (Enter '1' for student or '2' for staff):`
	Select “1” for Staff Login	 `Are you a staff or a student? (Enter '1' for student or '2' for staff): 2` `Enter your staff ID:`
	Enter Invalid userID	 `Process finished with exit code 0`
	Enter Valid userID	Ask password
	Enter Invalid Password	Ask for password again
	Enter Valid Password	Log in successful, ask if user wants to update profile information
Staff menu	Nil (staff menu)	
Invalid menu option	Select invalid menu option “10”	Should loop back to menu with warning
View Camps	Select “1” from staff menu	Show camp list
View Created Camps	Follow instructions	Shows camp list created by the staff account

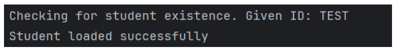
	from terminal	
Edit Created Camps	Follow instructions from terminal	Prompts staff to choose which camp to select
Delete Created Camp	Follow instructions from terminal	Asks for confirmation
View Suggestions	Follow instructions from terminal	Shows suggestion list from camp committee members
Generate Report	Follow instructions from terminal	Asks if the user wants the format in CSV or TXT
Log out (Staff)	Follow instructions from terminal	Go back to Account Manager page to choose between student and staff to log in

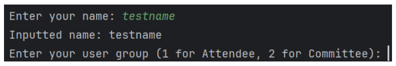
Student Camp Attendee

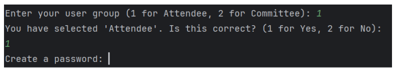
Function	User Input	Program output

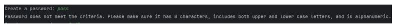
Registration for student	Select “1” for Student	
		`Are you a staff or a student? (Enter '1' for student or '2' for staff): 1`
		`Do you have an existing account? (1 for Yes, 0 for No):`

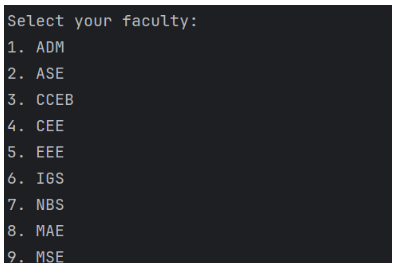
	Select “0” for No	
		`Do you have an existing account? (1 for Yes, 0 for No): 0`
		`Student Registration:`
		`Enter your student ID: test`

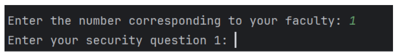
	Enter valid Student ID	
		`Checking for student existence. Given ID: TEST`
		`Student loaded successfully`

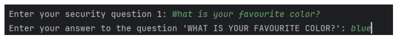
	Enter name “test”	
		`Enter your name: testname`
		`Inputted name: testname`
		`Enter your user group (1 for Attendee, 2 for Committee):`

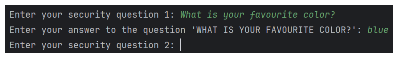
	Select “1” for Attendee	
		`Enter your user group (1 for Attendee, 2 for Committee): 1`
		`You have selected 'Attendee'. Is this correct? (1 for Yes, 2 for No):`
		`1`
		`Create a password:`

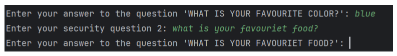
	Enter invalid password “pass”	
		`Create a password: pass`
		`Password does not meet the criteria. Please make sure it has 8 characters, includes both upper and lower case letters, and is alphanumeric.`

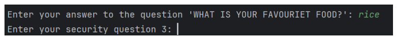
	Enter valid password “Passw0rd”	
		`Select your faculty:`
		`1. ADM`
		`2. ASE`
		`3. CCEB`
		`4. CEE`
		`5. EEE`
		`6. IGS`
		`7. NBS`
		`8. MAE`
		`9. MSE`

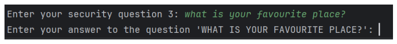
	Select “1” for ADM faculty	
		`Enter the number corresponding to your faculty: 1`
		`Enter your security question 1:`

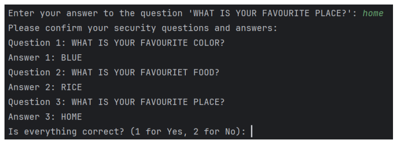
	Enter security question 1: “what is your favorite color?	
		`Enter your security question 1: What is your favourite color?`
		`Enter your answer to the question 'WHAT IS YOUR FAVOURITE COLOR?': blue`

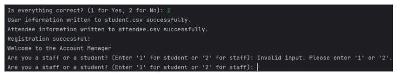
	Enter security question answer 1 : “blue”	
		`Enter your security question 1: What is your favourite color?`
		`Enter your answer to the question 'WHAT IS YOUR FAVOURITE COLOR?': blue`
		`Enter your security question 2:`

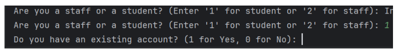
	Enter security question 2: “what is your favorite food?	
		`Enter your answer to the question 'WHAT IS YOUR FAVOURITE COLOR?': blue`
		`Enter your security question 2: what is your favouriet food?`
		`Enter your answer to the question 'WHAT IS YOUR FAVOURIET FOOD?':`

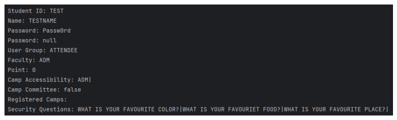
	Enter security question answer 2 : “rice”	
		`Enter your answer to the question 'WHAT IS YOUR FAVOURIET FOOD?': rice`
		`Enter your security question 3:`

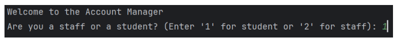
	Enter security question 3: “what is your place?	
		`Enter your security question 3: what is your favourite place?`
		`Enter your answer to the question 'WHAT IS YOUR FAVOURITE PLACE?':`

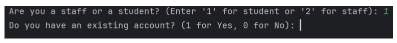
	Enter security question answer 3 : “home”	
		`Enter your answer to the question 'WHAT IS YOUR FAVOURITE PLACE?': home`
		`Please confirm your security questions and answers:`
		`Question 1: WHAT IS YOUR FAVOURITE COLOR?`
		`Answer 1: BLUE`
		`Question 2: WHAT IS YOUR FAVOURIET FOOD?`
		`Answer 2: RICE`
		`Question 3: WHAT IS YOUR FAVOURITE PLACE?`
		`Answer 3: HOME`
		`Is everything correct? (1 for Yes, 2 for No):`

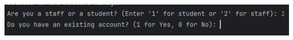
	Select “1” to confirm	
		`Is everything correct? (1 for Yes, 2 for No): 1`
		`User information written to student.csv successfully.`
		`Attendee information written to attendee.csv successfully.`
		`Registration successful!`
		`Welcome to the Account Manager`
		`Are you a staff or a student? (Enter '1' for student or '2' for staff): Invalid input. Please enter '1' or '2'.`
		`Are you a staff or a student? (Enter '1' for student or '2' for staff):`

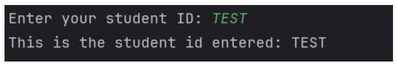
	Select “1” for Student	
		`Are you a staff or a student? (Enter '1' for student or '2' for staff): In`
		`Are you a staff or a student? (Enter '1' for student or '2' for staff): 1`
		`Do you have an existing account? (1 for Yes, 0 for No):`

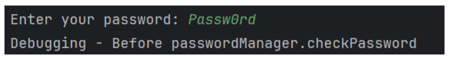
	Select “1” for existing	
		`Student ID: TEST`
		`Name: TESTNAME`
		`Password: Passw0rd`
		`Password: null`
		`User Group: ATTENDEE`
		`Faculty: ADM`
		`Point: 0`
		`Camp Accessibility: ADM|`
		`Camp Committee: false`
		`Registered Camps:`
		`Security Questions: WHAT IS YOUR FAVOURITE COLOR?|WHAT IS YOUR FAVOURIET FOOD?|WHAT IS YOUR FAVOURITE PLACE?|`

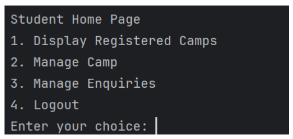
Start menu	Nil	
		`Welcome to the Account Manager`
		`Are you a staff or a student? (Enter '1' for student or '2' for staff): 1`

Login For student (Attendee)	Select “1” for student	
		`Are you a staff or a student? (Enter '1' for student or '2' for staff): 1`
		`Do you have an existing account? (1 for Yes, 0 for No):`

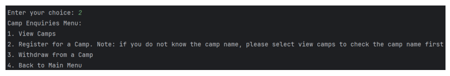
	Select “1” for existing account	
		`Are you a staff or a student? (Enter '1' for student or '2' for staff): 1`
		`Do you have an existing account? (1 for Yes, 0 for No):`

	Enter “TEST” for Student ID	
		`Enter your student ID: TEST`
		`This is the student id entered: TEST`

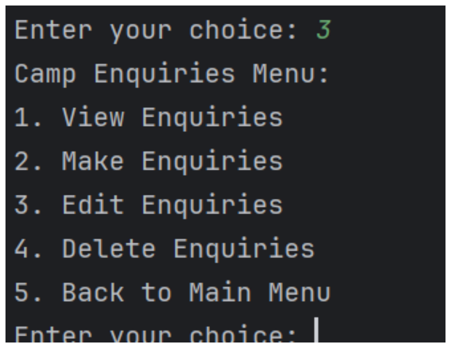
	Enter “Passw0rd” for password	
		`Enter your password: Passw0rd`
		`Debugging - Before passwordManager.checkPassword`

	Nil (Student Menu page)	
		`Student Home Page`
		`1. Display Registered Camps`
		`2. Manage Camp`
		`3. Manage Enquiries`
		`4. Logout`
		`Enter your choice:`

Invalid menu option	Select invalid menu option “10”	Should loop back to menu with warning
Display Registered Camps	Select “1” from student menu	List of camps attendee registered for If attendee has not registered for anything, display “You have not registered for any camps”

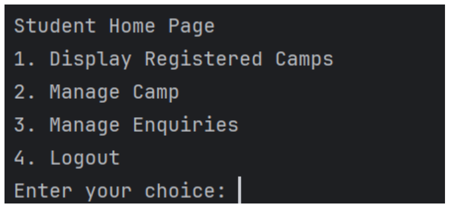
Manage Camp	Select “2” from student menu	
		`Enter your choice: 2`
		`Camp Enquiries Menu:`
		`1. View Camps`
		`2. Register for a Camp. Note: if you do not know the camp name, please select view camps to check the camp name first`
		`3. Withdraw from a Camp`
		`4. Back to Main Menu`

Join Camps	Follow instructions from terminal	Asks if user knows the exact name of the camp they want to join
Withdraw from Camp	Follow instructions from terminal	Asks for confirmation
Rejoin Camp	Follow instructions from terminal	Displays not allowed

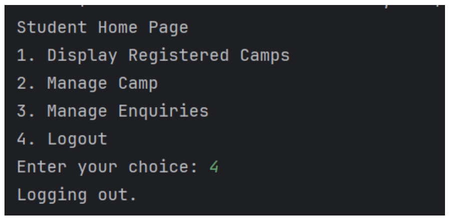
Manage Enquiries	Select “3” from student menu	
		`Enter your choice: 3`
		`Camp Enquiries Menu:`
		`1. View Enquiries`
		`2. Make Enquiries`
		`3. Edit Enquiries`
		`4. Delete Enquiries`
		`5. Back to Main Menu`
		`Enter your choice:`

View Enquiries	Select “1” from Student Enquiry menu	Shows list of enquiries attendee has made for the camps they have joined If attendee has not registered for anything or made any enquiries, display “You have not made any enquiries”
Submit Enquiries	Select “2” from Student Enquiry menu	Prompted to write enquiry content to submit

Edit Enquiries	Select “3” from Student Enquiry menu	Choose which enquiry
Delete Enquiries	Select “4” from Student Enquiry menu	Asks for confirmation

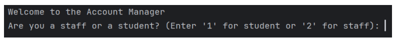
Back to Main Menu	Select “5” from Student Enquiry menu	
		`Student Home Page`
		`1. Display Registered Camps`
		`2. Manage Camp`
		`3. Manage Enquiries`
		`4. Logout`
		`Enter your choice:`

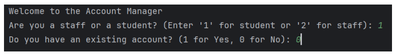
Change Password (Student)	Select “__” from student menu	Show changing of password here

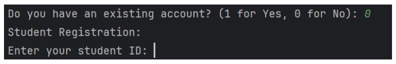
Log out	Select “4” from student menu	
		`Student Home Page`
		`1. Display Registered Camps`
		`2. Manage Camp`
		`3. Manage Enquiries`
		`4. Logout`
		`Enter your choice: 4`
		`Logging out.`

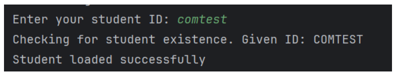
Student Camp Committee Member

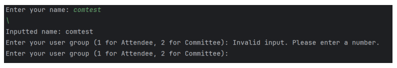
Function	Expected output	Program output

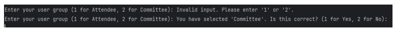
Registration Student (Committee)	Nil (program start)	
		`Welcome to the Account Manager`
		`Are you a staff or a student? (Enter '1' for student or '2' for staff):`

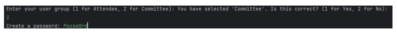
	Select “1” for student	
		`Welcome to the Account Manager`
		`Are you a staff or a student? (Enter '1' for student or '2' for staff): 1`
		`Do you have an existing account? (1 for Yes, 0 for No): 0`

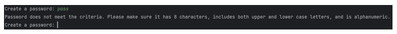
	Select “0” for new account	
		`Do you have an existing account? (1 for Yes, 0 for No): 0`
		`Student Registration:`
		`Enter your student ID:`

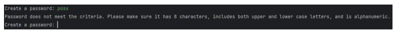
	Enter UserID “comtest”	
		`Enter your student ID: comtest`
		`Checking for student existence. Given ID: COMTEST`
		`Student loaded successfully`

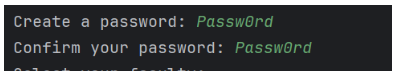
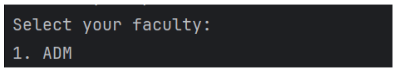
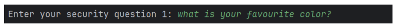
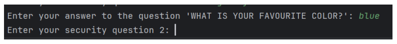
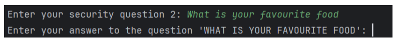
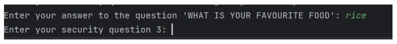
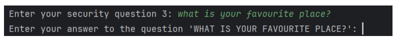
	Enter name “comtest”	 `Enter your name: comtest`
		`\`
		`Inputted name: comtest`
		`Enter your user group (1 for Attendee, 2 for Committee): Invalid input. Please enter a number.`
		`Enter your user group (1 for Attendee, 2 for Committee):`

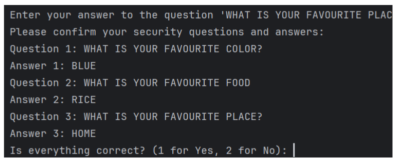
	Select “2” for Committee role	 `Enter your user group (1 for Attendee, 2 for Committee): Invalid input. Please enter '1' or '2'.` `Enter your user group (1 for Attendee, 2 for Committee): You have selected 'Committee'. Is this correct? (1 for Yes, 2 for No):`

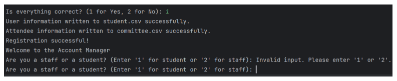
	Select “1” for confirmation	 `Enter your user group (1 for Attendee, 2 for Committee): You have selected 'Committee'. Is this correct? (1 for Yes, 2 for No):`
		`1`
		`Create a password: Passw0rd`

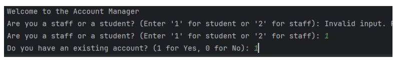
	Enter invalid password : “pass”	 `Create a password: pass`
		`Password does not meet the criteria. Please make sure it has 8 characters, includes both upper and lower case letters, and is alphanumeric.`
		`Create a password:`

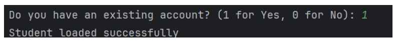
	Enter valid password : Passw0rd	 `Create a password: pass`
		`Password does not meet the criteria. Please make sure it has 8 characters, includes both upper and lower case letters, and is alphanumeric.`
		`Create a password:`

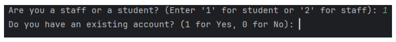
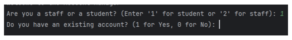
		 `Create a password: Passw0rd` `Confirm your password: Passw0rd`
	Select “1’ for faculty	 `Select your faculty:` `1. ADM`
	Enter security question 1: “what is your favorite color?	 `Enter your security question 1: what is your favourite color?`
	Enter security question answer 1 : “blue”	 `Enter your answer to the question 'WHAT IS YOUR FAVOURITE COLOR?': blue` `Enter your security question 2:`
	Enter security question 2: “what is your favorite food?	 `Enter your security question 2: What is your favourite food` `Enter your answer to the question 'WHAT IS YOUR FAVOURITE FOOD':`
	Enter security question answer 2 : “rice”	 `Enter your answer to the question 'WHAT IS YOUR FAVOURITE FOOD': rice` `Enter your security question 3:`
	Enter security question 3: “what is your place?	 `Enter your security question 3: what is your favourite place?` `Enter your answer to the question 'WHAT IS YOUR FAVOURITE PLACE?':`

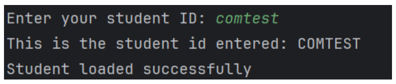
	Enter security question answer 3 : “home”	 `Enter your answer to the question 'WHAT IS YOUR FAVOURITE PLAC`
		`Please confirm your security questions and answers:`
		`Question 1: WHAT IS YOUR FAVOURITE COLOR?`
		`Answer 1: BLUE`
		`Question 2: WHAT IS YOUR FAVOURITE FOOD`
		`Answer 2: RICE`
		`Question 3: WHAT IS YOUR FAVOURITE PLACE?`
		`Answer 3: HOME`
		`Is everything correct? (1 for Yes, 2 for No):`

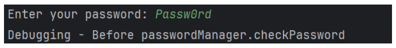
	Select “1” to confirm	 `Is everything correct? (1 for Yes, 2 for No): 1`
		`User information written to student.csv successfully.`
		`Attendee information written to committee.csv successfully.`
		`Registration successful!`
		`Welcome to the Account Manager`
		`Are you a staff or a student? (Enter '1' for student or '2' for staff): Invalid input. Please enter '1' or '2'.`
		`Are you a staff or a student? (Enter '1' for student or '2' for staff):`

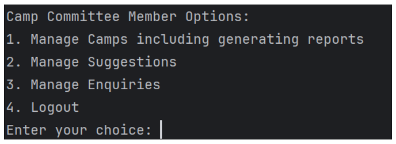
	Select “1” for student	 `Welcome to the Account Manager`
		`Are you a staff or a student? (Enter '1' for student or '2' for staff): Invalid input.`
		`Are you a staff or a student? (Enter '1' for student or '2' for staff): 1`
		`Do you have an existing account? (1 for Yes, 0 for No): 1`

	Select “1” for existing account	 `Do you have an existing account? (1 for Yes, 0 for No): 1` `Student loaded successfully`
Student menu (committee)		

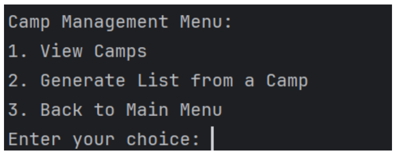
Login for student (Committee)	Nil (program start)	
	Select “1” for student	 `Are you a staff or a student? (Enter '1' for student or '2' for staff): 1` `Do you have an existing account? (1 for Yes, 0 for No):`
	Select “1” for existing account	 `Are you a staff or a student? (Enter '1' for student or '2' for staff): 1` `Do you have an existing account? (1 for Yes, 0 for No):`

	Enter valid user ID “comtest”	 `Enter your student ID: comtest`
		`This is the student id entered: COMTEST`
		`Student loaded successfully`

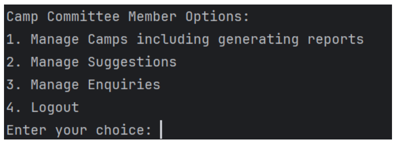
	Enter valid password “Passw0rd”	 `Enter your password: Passw0rd` `Debugging - Before passwordManager.checkPassword`

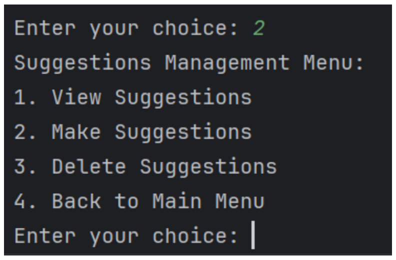
Committee menu	Nil (Committee Menu)	 `Camp Committee Member Options:`
		`1. Manage Camps including generating reports`
		`2. Manage Suggestions`
		`3. Manage Enquiries`
		`4. Logout`
		`Enter your choice:`

Invalid menu option	Select invalid menu option “10”	Should loop back to menu with warning

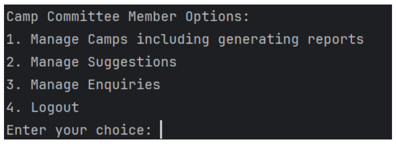
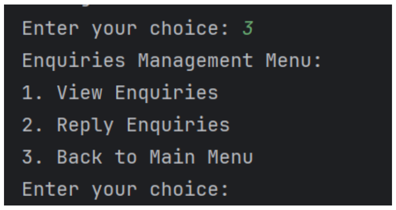
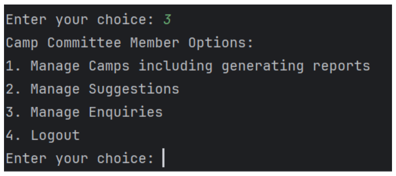
Manage Camp + Generating reports	Select “1” for manage camps	 `Camp Management Menu:`
		`1. View Camps`
		`2. Generate List from a Camp`
		`3. Back to Main Menu`
		`Enter your choice:`

View camps	Select “1” for view camps	Displays list of camps that can be joined as attendee
Generate list from camps	Select “2” for generate list of camps	Generate list of attendees from the camp they are a committee in

Back to committee menu	Select “3” to return to Committee menu	 `Camp Committee Member Options:`
		`1. Manage Camps including generating reports`
		`2. Manage Suggestions`
		`3. Manage Enquiries`
		`4. Logout`
		`Enter your choice:`

Suggestions Management Menu	Nil (Suggestion menu)	 `Enter your choice: 2`
		`Suggestions Management Menu:`
		`1. View Suggestions`
		`2. Make Suggestions`
		`3. Delete Suggestions`
		`4. Back to Main Menu`
		`Enter your choice:`

View suggestions	Select “1” to view suggestions	Show suggestions

Make suggestions	Select “2” to make suggestions	Show Make suggestions
Delete Suggestions	Select “3” to delete suggestions	Show Delete suggestions
Back to Main menu	Select “4” to return to main menu	
	Nil (Committee Menu)	 Camp Committee Member Options: 1. Manage Camps including generating reports 2. Manage Suggestions 3. Manage Enquiries 4. Logout Enter your choice:
Manage Enquiries menu	Nil (Enquiries Menu)	 Enter your choice: 3 Enquiries Management Menu: 1. View Enquiries 2. Reply Enquiries 3. Back to Main Menu Enter your choice:
View Enquiries	Select “1” to View Enquiries	View list of enquiries from attendees for the camp they are a committee in
Reply Enquiries	Select “2” to Reply to Enquiries	Choose which enquiry
Return to committee menu	Select “3” to return to Committee menu	
	Nil (Committee menu)	 Enter your choice: 3 Camp Committee Member Options: 1. Manage Camps including generating reports 2. Manage Suggestions 3. Manage Enquiries 4. Logout Enter your choice:

Reflection

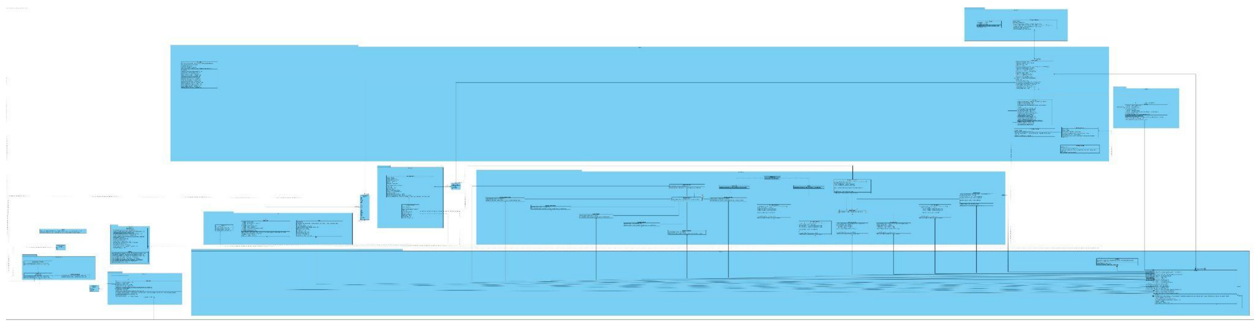
Designing the CAM system posed initial challenges as we transitioned from a familiar top-down procedural approach to a bottom-up paradigm, incorporating object-oriented programming principles. Embracing this shift required a mindset adjustment, but it became evident that OOP offers invaluable advantages, especially for complex systems. The use of UML diagrams proved instrumental in visualizing and understanding the system's intricate structure.

The development phase reinforced the benefits of the OOP approach. Breaking down the system into modules allowed for independent development and debugging, enhancing overall efficiency. Notably, the avoidance of redundant implementation through shared methods across different classes streamlined our coding process.

This project served as a practical exploration of an alternative and effective approach to system design and implementation. The gained knowledge and experience are valuable assets for tackling future complex or large-scale projects. As a collaborative effort, we successfully navigated the learning curve, demonstrating the significance of good design and implementation practices. Looking ahead, continued refinement of our OOP skills will undoubtedly contribute to even more effective and seamless project development.

UML Diagram

The .vpp and .jpg file for the UML are also made available in the project submission folder.



Other Resources

Other resources can be found in the project submission folder along with the Github repository at (https://github.com/huaizhic/OOP-camp-management-system)