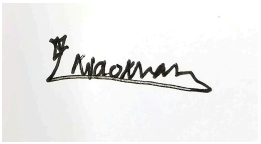
Declaration of Original Work for SC2002 Assignment

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 honoured the principles of academic integrity and have upheld the 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	Lab Group	Signature / Date
Goh Pei Han	SC2002	SCEA	 17/11/2025
Lai Kah Seng	SC2002	SCEA	 16/11/2025
Samuel Chan	SC2002	SCEA	 5/11/2025
Lim Xiao Xuan	SC2002	SCEA	 16/11/2025
Emmanuel Matthew Christopher Loho	SC2002	SCEA	 17/11/2025

Important notes:

1. Name must EXACTLY MATCH the one printed on your Matriculation Card.
2. The Student Code of Academic Conduct includes the latest guidelines on the usage of Generative AI and any other guidelines as released by NTU.

1. Introduction

https://github.com/saamuul/internship-placement-management-system

The Internship Placement Management System (IPMS) is designed to manage key internship placement operations such as user roles, applying, approving and rejecting internships using Object-Oriented and Design Programming (OODP) principles. The system supports multiple user roles, including students, company representatives and career staff, each with tailored functionalities.

2. Design Consideration and Approach

This section outlines the core design principles and rationale behind our Internship Placement Management System, highlighting how various components and classes work together to ensure reusability, extensibility, maintainability, modularity, and scalability. The main components of the system are listed below, along with explanations.

2.1. Primary Components

The system is separated into these few components. Each class is classified as one of these components:

1. **Authentication**
2. **User Roles (Entity/Model)**
3. **User Interface Menus (Boundary/View)**
4. **Controllers (Controller)**
5. **User-Role Based Services**
6. **Repository**
7. **Utility**
8. **DTO**
9. **Config**

2.1.1. Authentication

When the application starts, the user is greeted with a login prompt where they enter their credentials. The **Authenticator** validates these credentials against records stored in the database. In our system, we use HashMaps with keys (user ID) and values (passwords) extracted from the **database** (a documented .csv file) to authenticate login credentials. Once authenticated, the application determines the user's role (**STUDENT**, **COMPANY_REP**, **CAREER_STAFF**) and displays the relevant interface from **User Interface Menus** (i.e. **StudentView**, **CompanyRepView,** and **CareerStaffView**). This separation of concerns ensures that user

authentication and role-based access are clear and secure from each other, laying the foundation for the system's structure.

2.1.2. User Roles (Entities)

The system employs inheritance to model different user roles through the **User** abstract class. The three roles of our system: **Student**, **Company Representative**, and **Career Staff** extend this class, inheriting common attributes such as userID, name, password, and email, hence minimising code duplication. It also allows the system to scale easily when introducing new user roles in the future, such as **Mentors** or **Professors**, promoting code reusability.

2.1.3. User Interface Menus

Each **User Interface Menu** class is dedicated to a specific user role. By separating user interaction logic from the core functionality (e.g., database management or internship services), we ensure a clear separation of concerns. For example, the **StudentView** class interacts with the **Student** class and **StudentService** class through the intermediary class **StudentController** to allow students to view internship applications or internship opportunities, while the **CompanyRepView** focuses on outputting operations done from Company Representative operations (i.e. viewing created Internship Opportunities). This approach leads to easier debugging and cleaner code, as the **User Interface Menu** class focuses solely on user interaction, while the **User** class encapsulates the data.

2.1.4. Controllers

To achieve looser coupling and higher cohesion, intermediary classes are created between each User Interface Menu class and their respective User Role and Services classes. The controllers are responsible for the following:

- Business logic: Calling appropriate methods to handle data based on user requests.
- Control hub of classes: Intermediate between UI Menu & Service classes for ease of debugging and maintainability

Example:

CompanyRepView receives the user's input choice, then passes the choice to **CompanyRepController** to handle business logic. **CompanyRepController** then proceeds to invoke the function based on the user's request by calling the respective function in **CompanyRepService**, before finally passing the output to **CompanyRepController** again for **CompanyRepView** to display. By using Controllers as the intermediary, the UI Menus are kept focused on user interaction, and the Logics & Database access is handled separately by

the Controllers, ensuring that each part of the system only deals with what it's supposed to do. This approach keeps the code modular and easy to maintain. This approach also aligns with SOLID design principles, which will be discussed later.

2.1.5. User-Role Based Services

This is where all the logic for the core functions of the system is located. It is designed as a core component of our system to define a clearer line between function logic and the database. This is to enhance the readability of code for ease of debugging and to facilitate extensibility in the foreseeable future. It is also the method that invokes **Repository** classes to read and save information into the database (this will also be discussed later). This approach further eases coding and enforcing the **Single Responsibility Principle** in **SOLID**. Each **User-Role Based Service** class is invoked when the **Controller** class calls for them. The class will then retrieve relevant information from the **User Role (Entity) and Repository** classes before logic processing, and then finally return the processed information to the **Controllers**.

Example:
In the method *handleWithdrawApplication(User user)*, the **StudentController** class passes the input from **StudentView** to the **StudentService** class. The **StudentService** class fetches the relevant data from the **WithdrawalRepository** class, does the logic needed to allow students to withdraw from a certain internship, before saving the database with the **WithdrawalRepository** class and passing the processed output back to the **StudentController**. As you can see, each class has a well-defined responsibility. The classes are able to work with high cohesion with as little coupling as possible with this approach.

2.1.6. Repository

These instances of classes are where the fetching and saving of the database happen. They are always called at the start and end of each method of **User-Role Based Services**. This is to ensure real-time updates of the database so that the **User Interface Menus** will have up-to-date information of the database. These classes also have an additional feature of **filtering** by certain attributes (e.g. findById, findByStatus, etc.)

Example:
In the method *acceptInternship(String studentId, String applicationId)* from class **StudentService**, it can be seen that the method calls for the **ApplicationRepository** class to fetch data from. The class does so by iterating through the CSV file and adding and filtering the relevant information to the result to be passed back to **StudentService.** In the end, once the function and logic have reached their end in *acceptInternship,* **ApplicationRepository** saves the new data into the CSV file.

2.1.7. Utility

These classes serve no purpose other than making the coding and debugging process easier. It is used extensively in the **Repository** classes.

Example:

The class CSVUtil has the responsibility to read and write data from and into the database, in our case, the CSV file.

2.1.8. DTO

These classes replace the conventional method of returning 'true' or 'false' from a certain method based on function results. They are used mainly to ease the process of displaying function outputs to the user. (e.g. OperationResult, AuthResult)

2.1.9. Config

This architecture of a class is only applied to the class DataConfig. It stores the file paths of our database (CSV files).

2.2. SOLID Design Principles

Each class is designed such that the system adheres to SOLID design principles to ensure that it is understandable, flexible, and maintainable. These principles guide our system architecture, ensuring that each component has only one clear responsibility, leading to new features and roles being able to be easily added.

2.2.1. Single Responsibility Principle (SRP)

Each class in the system has one distinct responsibility, and the main ones are:

1. Model class - Stores data of the object classes, consisting of simple getters and setters used by other classes
2. Controller class - Controls the communication between the user and the service
3. View class - Display code output and receive inputs for functions of the code
4. Service class - Handles the necessary business logic of the model classes
5. Repository class - Handles data access logic for the different model classes

Each class is also further broken down into appropriate class types. For example, Service classes have StudentService, CompanyRepService, and CareerStaffService, ensuring that each class only handles logic related to the corresponding user type, without affecting the other class logic.

2.2.2. Open-Closed Principle (OCP)

The Open/Closed Principle states that software entities should be open for extension but closed for modification. In our Internship System, this principle is applied through the design of separate service classes for each user type (StudentService, CompanyRepService, CareerStaffService) and the use of interfaces. This allows new functionality, such as additional types of users, new internship validations, or reporting features, to be added by extending the system with new classes or methods without altering existing, tested code. As a result, the system remains stable, maintainable, and adaptable to future requirements.

2.2.3. Liskov Substitution Principle (LSP)

The Liskov Substitution Principle dictates that objects of a superclass should be interchangeable with objects of its subclasses without affecting the program's behaviour. For our Internship System, we apply this to our User class, which provides essential attributes and methods across all user types: Student, CompanyRepresentative, and CareerStaff. For example, each subclass inherits functions such as Login() and various getters and setters for information shared across all users, such as name and password. This ensures that any code written to handle User objects, such as authentication or retrieving basic profile information, will work correctly regardless of the specific user type, allowing the system to remain flexible, maintainable, and scalable.

2.2.4. Interface Segregation Principle (ISP)

The Interface Segregation Principle (ISP) suggests that instead of forcing classes to implement large, all-encompassing interfaces, it is better to create smaller, specific interfaces that provide only the methods a particular client needs. We have done this through the use of user-specific interfaces, such as ICareerStaffService, which interfaces methods only required by the CareerStaffService class. Same for the CompanyRepService and StudentService, each interfacing their own ICompanyRepService and IStudentService, which interfaces only methods specific to the respective user types, ensuring no clients see/use methods they don’t need.

2.2.5. Dependency Inversion Principle (DIP)

The Dependency Inversion Principle states that the high-level modules should not depend upon the low-level modules, and both should depend upon abstractions. Abstractions should not depend upon details, and instead, details should depend upon abstractions.

The system follows the Dependency Inversion Principle by ensuring that higher-level modules (controllers and services) depend on abstractions rather than concrete implementations, while lower-level modules (services and repositories) implement the abstractions.

All service classes accept repository interfaces (IUserRepository, IInternshipRepository, IApplicationRepository) in their constructors instead of directly instantiating concrete CSV-based repositories, while lower-level modules (concrete repository classes) implement the repository interfaces. This makes the business logic independent of the underlying data storage mechanism.

Likewise, controllers depend on service interfaces (ICompanyRepService, IStudentService, etc.) rather than specific implementations. All concrete dependencies are created and injected at the application's composition root (InternshipSystem.main()), which provides the actual implementations. This decouples module layers, increases testability, and allows repository implementations to be replaced (e.g., swapping CSV files for SQL or JSON) without modifying business logic or controllers by only making a new SQL or JSON modifier class to implement the same said interfaces.

2.3. Assumptions Made

1. Only 1 user is logged in at a time
2. Since students and career staff are auto-registered, it is assumed that all their info is already registered within the database (CSV file).
3. Students use StudentID as login credentials, and Company Representatives use email as login credentials, while the CareerStaff class uses StaffID as login credentials.
4. Users are responsible for maintaining the confidentiality of their passwords to prevent unauthorised access. No additional multi-factor authentication will be implemented.
5. All individual persons may have only one account at any given time.
6. All company representatives have unique names.
7. All company representatives will have the default password of ‘password’ upon registering and logging in until they manually change their passwords using the changePassword() method.

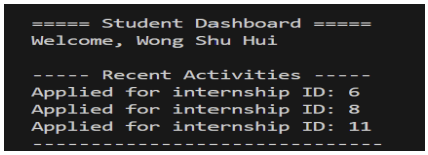
2.4. Additional Features Implemented

1. All role and internship data is persisted even after exiting the system.
2. All changes happen in real time, so after updating anything, the user is able to view the change.
3. The system will automatically delete internship opportunities passed the closing date.
4. View Internships Filter - Each user can use appropriate filters to filter their respective view of all

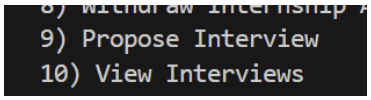
internships based on their roles, and it is cached and remembered even after switching menu pages.

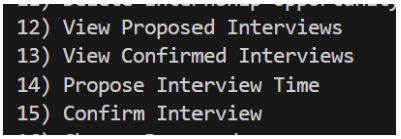
   a. Student and company representative is shown as filtering internship opportunities or applications
   b. Career staff is shown as generate report

5. Recent activity log - Show users a list of their recent actions (e.g. applications submitted, opportunities created) within their login session.



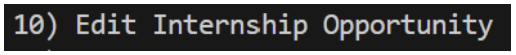
6. Interview Scheduler - Allow students and company representatives to propose and confirm interview times, after the internship is accepted.





Student Company Rep

7. Editing of Internship Opportunity - Company representatives have the ability to modify and remove existing internship opportunity details.



3. Diagrams

3.1. UML Sequence Diagram

https://drive.google.com/file/d/1BKSFqFZkmg1KNwdsEDGK18smDp0dZ62p/view?usp=sharing

3.2. UML Class Diagram

https://drive.google.com/file/d/1g1T6HQwnm3MoIlZhG4aGI8Qa2vXcH74n/view?usp=sharing

4. Test Cases

4.1. Student Operations

```
===== Student Dashboard =====
Welcome, Wong Shu Hui

----- Recent Activities -----
No recent activities.
------------------------------
1) View Available Internships
2) Filter Internships
3) Clear Filters
4) Apply for Internship
5) View Internship Application(s)
6) View Accepted Internship
7) Accept Internship
8) Withdraw Internship Application(s)
9) Change Password
10) Logout
Choose: 3
SUCCESS: Filters cleared. Viewing all available internships.

=== Available Internships ===
```

ID	Title	Company	Description	Open Date	Close Date	Slots	Level
6	Mobile App Developer	Emily Tan	Build iOS and Android applications	2025-11-18	2025-12-27	5	INTERMEDIATE
8	Cybersecurity Analyst Intern	Brandon Yap	Monitor security systems and res...	2025-11-20	2026-01-10	2	ADVANCED
11	Backend Developer Intern	Emily Tan	Build REST APIs and microservices	2025-11-18	2025-12-24	4	INTERMEDIATE
17	Quality Assurance Intern	Alice Smith	Test software and report bugs	2025-11-23	2025-12-29	3	BASIC
18	Blockchain Developer Intern	Marcus Ong	Develop smart contracts and DApps	2025-11-20	2026-01-12	2	ADVANCED
22	Full Stack Developer Intern	Emily Tan	Work on both frontend and backen...	2025-11-22	2026-01-06	3	INTERMEDIATE
25	Game Development Intern	Brandon Yap	Create games using Unity or Unre...	2025-11-23	2025-12-31	3	INTERMEDIATE
29	Enterprise Software Intern	Alice Smith	Develop enterprise resource plan...	2025-11-22	2026-01-05	3	INTERMEDIATE

Clear Filter

```
===== Student Dashboard =====
Welcome, Wong Shu Hui

----- Recent Activities -----
No recent activities.
------------------------------
1) View Available Internships
2) Filter Internships
3) Clear Filters
4) Apply for Internship
5) View Internship Application(s)
6) View Accepted Internship
7) Accept Internship
8) Withdraw Internship Application(s)
9) Change Password
10) Logout
Choose: 4

=== Available Internships ===
```

ID	Title	Company	Description	Open Date	Close Date	Slots	Level
17	Quality Assurance Intern	Alice Smith	Test software and report bugs	2025-11-23	2025-12-29	3	BASIC

```
Enter Internship ID to apply: 17
SUCCESS: Tan Wei Ling successfully applied for internship ID: 17
```

Apply for Internship

```
===== Student Dashboard =====
Welcome, Wong Shu Hui

----- Recent Activities -----
Applied for internship ID: 18
Applied for internship ID: 6
Applied for internship ID: 8
------------------------------
1) View Available Internships
2) Filter Internships
3) Clear Filters
4) Apply for Internship
5) View Internship Application(s)
6) View Accepted Internship
7) Accept Internship
8) Withdraw Internship Application(s)
9) Change Password
10) Logout
Choose: 4

=== Available Internships ===
```

ID	Title	Company	Description	Open Date	Close Date	Slots	Level
6	Mobile App Developer	Emily Tan	Build iOS and Android applications	2025-11-18	2025-12-27	5	INTERMEDIATE
8	Cybersecurity Analyst Intern	Brandon Yap	Monitor security systems and res...	2025-11-20	2026-01-10	2	ADVANCED
11	Backend Developer Intern	Emily Tan	Build REST APIs and microservices	2025-11-18	2025-12-24	4	INTERMEDIATE
17	Quality Assurance Intern	Alice Smith	Test software and report bugs	2025-11-23	2025-12-29	3	BASIC
18	Blockchain Developer Intern	Marcus Ong	Develop smart contracts and DApps	2025-11-20	2026-01-12	2	ADVANCED
22	Full Stack Developer Intern	Emily Tan	Work on both frontend and backen...	2025-11-22	2026-01-06	3	INTERMEDIATE
25	Game Development Intern	Brandon Yap	Create games using Unity or Unre...	2025-11-23	2025-12-31	3	INTERMEDIATE
29	Enterprise Software Intern	Alice Smith	Develop enterprise resource plan...	2025-11-22	2026-01-05	3	INTERMEDIATE

```
Enter Internship ID to apply: 25
ERROR: You already have 3 active internship applications.
```

Apply for Internship(Max 3)

```
===== Student Dashboard =====
Welcome, Wong Shu Hui

----- Recent Activities -----
No recent activities.
------------------------------
1) View Available Internships
2) Filter Internships
3) Clear Filters
4) Apply for Internship
5) View Internship Application(s)
6) View Accepted Internship
7) Accept Internship
8) Withdraw Internship Application(s)
9) Change Password
10) Logout
Choose: 5

=== Your Internship Applications ===
```

App ID	Internship Title	Company Representative	Level	Status
1	Blockchain Developer Intern	Marcus Ong	ADVANCED	SUCCESSFUL
2	Mobile App Developer	Emily Tan	INTERMEDIATE	PENDING
3	Cybersecurity Analyst Intern	Brandon Yap	ADVANCED	PENDING

View Internship Applications(regardless of visibility)

```
===== Student Dashboard =====
Welcome, Wong Shu Hui

----- Recent Activities -----
Accepted internship application ID: 1
------------------------------
1) View Available Internships
2) Filter Internships
3) Clear Filters
4) Apply for Internship
5) View Internship Application(s)
6) View Accepted Internship
7) Accept Internship
8) Withdraw Internship Application(s)
9) Change Password
10) Logout
Choose: 6

=== Your Accepted Internship ===
------------------------------
Application ID: 1
Status: SUCCESSFUL
Internship ID: 18
Title: Blockchain Developer Intern
Company Representative: Marcus Ong
Level: ADVANCED
------------------------------
```

View accepted Internships

```
===== Student Dashboard =====
Welcome, Wong Shu Hui

----- Recent Activities -----
Accepted internship application ID: 1
------------------------------
1) View Available Internships
2) Filter Internships
3) Clear Filters
4) Apply for Internship
5) View Internship Application(s)
6) View Accepted Internship
7) Accept Internship
8) Withdraw Internship Application(s)
9) Change Password
10) Logout
Choose: 6

=== Your Accepted Internship ===
------------------------------
Application ID: 1
Status: SUCCESSFUL
Internship ID: 18
Title: Blockchain Developer Intern
Company Representative: Marcus Ong
Level: ADVANCED
------------------------------
```

Accept Internships

```
=== Your Internship Applications ===
```

App ID	Internship Title	Company Representative	Level	Status
1	Blockchain Developer Intern	Marcus Ong	ADVANCED	SUCCESSFUL

Applications removed after acceptance

```
===== Student Dashboard =====
Welcome, Wong Shu Hui

----- Recent Activities -----
Applied for internship ID: 18
Applied for internship ID: 6
Applied for internship ID: 8
------------------------------
1) View Available Internships
2) Filter Internships
3) Clear Filters
4) Apply for Internship
5) View Internship Application(s)
6) View Accepted Internship
7) Accept Internship
8) Withdraw Internship Application(s)
9) Change Password
10) Logout
Choose: 8

=== Your Internship Applications ===
```

App ID	Internship Title	Company Representative	Level	Status
1	Blockchain Developer Intern	Marcus Ong	ADVANCED	PENDING
2	Mobile App Developer	Emily Tan	INTERMEDIATE	PENDING
3	Cybersecurity Analyst Intern	Brandon Yap	ADVANCED	PENDING

```
Enter Application ID for withdrawal: 1
SUCCESS: Withdrawal request submitted for Application 1. Awaiting Approval.
```

Withdraw Internships(before or after accepting)

4.2. Company Representative Operations

```
=== Internship Placement Management System ===
1) Login
2) Register as Company Representative
3) Exit
Select option: 2

-- Company Rep Registration --
Name: Lai Kah Seng
Email: LAIK0012@e.ntu.edu.sg
Company: NVIDIA
Department: Hardware Engineering
Position: CEO
SUCCESS: Registration successful. Awaiting approval.

=== Internship Placement Management System ===
1) Login
2) Register as Company Representative
3) Exit
Select option: 1
User ID: LAIK0012@e.ntu.edu.sg
Password: password
ERROR: Account pending approval.
```

New Company Representative creation

```
Choose: 1
Enter title: Hardware Development Intern
Enter the description: Develop Hardware and Learn
Enter preferred Major: Computer Engineering
Enter Level (BASIC/INTERMEDIATE/ADVANCED): NA
ERROR: Invalid level. Please try again.
Enter Level (BASIC/INTERMEDIATE/ADVANCED): ADVANCED
Enter Open Date (YYYY-MM-DD): 2020-11-20
Enter Close Date (YYYY-MM-DD): 2020-11-19
ERROR: Close date must be AFTER open date. Please try again.
Enter Close Date (YYYY-MM-DD): 2021-11-20
Enter Number Of Slots: 11
ERROR: Value must be between 0 and 10. Please try again.
Enter Number Of Slots: 3
SUCCESS: Internship opportunity created with ID: 32
```

Create Internship Opportunity

```
=== Company Representative Dashboard ===
Welcome, Emily Tan
1) Create Internship Opportunity
2) View Created Opportunities
3) Filter Opportunities
4) Clear Filters
5) View Internship Applications
6) Filter Applications
7) Clear Application Filters
8) Review Internship Applications
9) Toggle Internship Visibility
10) Change Password
11) Logout
Choose: 2

=== Your Created Internship Opportunities ===
```

ID	Title	Description	Preferred Major	Open Date	Closing Date	Number of Slots	Visibility	Status	Level
2	Software Development Intern	Develop web applications us...	Computer Science	2025-11-18	2025-12-25	1	false	SUCCESSFUL	BASIC
6	Mobile App Developer	Build iOS and Android appli...	Computer Science	2025-11-18	2025-12-27	5	true	SUCCESSFUL	INTERMEDIATE
11	Backend Developer Intern	Build REST APIs and microse...	Computer Science	2025-11-18	2025-12-24	4	true	SUCCESSFUL	INTERMEDIATE
22	Full Stack Developer Intern	Work on both frontend and b...	Computer Science	2025-11-22	2026-01-06	3	true	SUCCESSFUL	INTERMEDIATE
32	Hardware Development Intern	Develop Hardware and Learn	Computer Engineering	2020-11-20	2021-11-20	3	true	PENDING	ADVANCED

View Created Internship Opportunities

Maximum 5 Opportunities allowed

```
Choose: 9

=== Your Created Internship Opportunities ===
```

ID	Title	Description	Preferred Major	Open Date	Closing Date	Number of Slots	Visibility	Status	Level
2	Software Development Intern	Develop web applications us...	Computer Science	2025-11-18	2025-12-25	1	false	SUCCESSFUL	BASIC
6	Mobile App Developer	Build iOS and Android appli...	Computer Science	2025-11-18	2025-12-27	1	true	SUCCESSFUL	INTERMEDIATE
11	Backend Developer Intern	Build REST APIs and microse...	Computer Science	2025-11-18	2025-12-24	4	true	SUCCESSFUL	INTERMEDIATE
22	Full Stack Developer Intern	Work on both frontend and b...	Computer Science	2025-11-22	2026-01-06	3	true	SUCCESSFUL	INTERMEDIATE
32	Hardware Developer	develop hardware	Computer Engineering	2025-11-20	2025-11-30	1	true	PENDING	ADVANCED

```
Enter Internship Opportunity ID: 2
Toggle visibility (true/false): true
SUCCESS: Internship opportunity is now visible.
```

Toggle Internship Opportunity visibility

```
Choose: 5

=== Internship Applications ===
```

ID	Student ID	Internship ID	Status
2	U2310001A	2	PENDING
1	U2310005E	6	SUCCESSFUL

View Internship Applications

```
Choose: 6

=== Filter Applications ===
Select filters to apply (type numbers separated by commas):
1) Status (PENDING/SUCCESSFUL/UNSUCCESSFUL)
2) Student ID
Or press ENTER for no filters.
Your choice: 2
Enter Student ID: U2310001A

=== Internship Applications ===
```

ID	Student ID	Internship ID	Status
2	U2310001A	2	SUCCESSFUL

```
SUCCESS: Filter applied. Showing 1 application(s).
```

Filter Internship Applications

```
Choose: 4
SUCCESS: Filters cleared.

=== Your Created Internship Opportunities ===
```

ID	Title	Description	Preferred Major	Open Date	Closing Date	Number of Slots	Visibility	Status	Level
2	Software Development Intern	Develop web applications us...	Computer Science	2025-11-18	2025-12-25	0	true	FILLED	BASIC
6	Mobile App Developer	Build iOS and Android appli...	Computer Science	2025-11-18	2025-12-27	1	true	SUCCESSFUL	INTERMEDIATE
11	Backend Developer Intern	Build REST APIs and microse...	Computer Science	2025-11-18	2025-12-24	4	true	SUCCESSFUL	INTERMEDIATE
22	Full Stack Developer Intern	Work on both frontend and b...	Computer Science	2025-11-22	2026-01-06	3	true	SUCCESSFUL	INTERMEDIATE
32	Hardware Developer	develop hardware	Computer Engineering	2025-11-20	2025-11-30	1	true	PENDING	ADVANCED

Clear Filters applied

```
=== Company Representative Dashboard ===
Welcome, Emily Tan
1) Create Internship Opportunity
2) View Created Opportunities
3) Filter Opportunities
4) Clear Filters
5) View Internship Applications
6) Filter Applications
7) Clear Application Filters
8) Review Internship Applications
9) Toggle Internship Visibility
10) Change Password
11) Logout
Choose: 8

=== Internship Applications ===
```

ID	Student ID	Internship ID	Status
2	U2310001A	2	PENDING

```
Enter Internship Application ID to approve: 2
Enter decision (SUCCESSFUL to approve, UNSUCCESSFUL to reject): successful
SUCCESS: Application reviewed successfully.
```

Review Internship Applications

```
Choose: 3

=== Filter Opportunities ===
Select filters to apply (type numbers separated by commas):
1) Level (BASIC/INTERMEDIATE/ADVANCED)
2) Preferred Major
3) Status (PENDING/SUCCESSFUL/UNSUCCESSFUL/FILLED)
4) Visibility (true/false)
5) Closing Date (YYYY-MM-DD)
Or press ENTER for no filters.
Your choice: 5
Enter Closing Date (YYYY-MM-DD): 2025-12-25

=== Your Created Internship Opportunities ===
```

ID	Title	Description	Preferred Major	Open Date	Closing Date	Number of Slots	Visibility	Status	Level
2	Software Development Intern	Develop web applications us...	Computer Science	2025-11-18	2025-12-25	0	true	FILLED	BASIC

```
SUCCESS: Filter applied. Showing 1 opportunity(s).
```

Filter Internship Opportunities

4.3. Career Staff Operation

```
=== Internship Placement Management System ===
1) Login
2) Register as Company Representative
3) Exit
Select option: 1
User ID: sng001
Password: password

=== Career Staff Dashboard ===
Welcome, Dr. Sng Hui Lin
1) View Pending Company Representatives
2) Approve Company Representative
3) View All Internship Opportunities
4) Approve Internship Opportunity
5) View Withdrawal Requests
6) Approve Withdrawal Request
7) Create Filter and Generate Report
8) Change Password
9) Logout
Choose:
```

Career Staff login

```
Choose: 1

=== Pending Company Representatives ===

UserID                          Name                Company
-------------------------------------------------------------------
alex.ho@primespark.com          Alex Ho             PrimeSpark Retail
ben@gmail.com                   ben                 Google
goh@gmail.com                   Goh Loh             theCompany
jonathan.wong@aerolink.com      Jonathan Wong       AeroLink Systems
newrep@company.com              NewRep              NewCompany
nicole.chua@silverpeak.com      Nicole Chua         SilverPeak Finance
password                        crep001             2
Terenceloh@gmail.com            Terence Loh         Tech tree Pte Ltd
test@example.com                Test User           Test Corp
test@testcompany.com            Test Rep            Test Company
```

View pending Company Representative

```
Choose: 2

=== Pending Company Representatives ===

UserID                          Name                Company
-------------------------------------------------------------------
alex.ho@primespark.com          Alex Ho             PrimeSpark Retail
ben@gmail.com                   ben                 Google
goh@gmail.com                   Goh Loh             theCompany
jonathan.wong@aerolink.com      Jonathan Wong       AeroLink Systems
newrep@company.com              NewRep              NewCompany
nicole.chua@silverpeak.com      Nicole Chua         SilverPeak Finance
password                        crep001             2
Terenceloh@gmail.com            Terence Loh         Tech tree Pte Ltd
test@example.com                Test User           Test Corp
test@testcompany.com            Test Rep            Test Company
Enter Company Representative ID to approve: ben@gmail.com
Approve Company Representative? (Y/N): Y
SUCCESS: Company representative approved successfully.
```

Approve Company Representative Registration

```
Choose: 7
Select filters to apply (type numbers separated by commas):
1) Level
2) Preferred Major
3) Representative Name
4) Status
5) Visibility
Or press ENTER for no filters.
Your choice: 5
Enter Visibility (true/false):
false
========== Internship Report ==========
Report ID: 8a21687b-f611-4339-9436-044a5669b46f
Generated Date: 2025-11-18T23:35:02.513530100
Generated By: Dr. Sng Hui Lin

----- Filter Criteria -----
Internship Level: N/A
Preferred Major: N/A
Company Representative: N/A
Internship Status: N/A
Closing Date: N/A
Visibility: false

----- Internship Opportunities -----
Title: Software Development Intern
Representative: Emily Tan
Level: BASIC
Preferred Major: Computer Science
Application Dates: 2025-11-18 to 2025-12-25
Number of Slots: 1
Status: SUCCESSFUL
Visibility: false
---------------------------------------
Title: Frontend Developer Intern
Representative: Daniel Lee
Level: BASIC
Preferred Major: Computer Science
Application Dates: 2025-11-19 to 2025-12-28
Number of Slots: 5
Status: SUCCESSFUL
Visibility: false
---------------------------------------
=======================================
```

Filter and generate report

```
5) View Withdrawal Requests
6) Approve Withdrawal Request
7) Create Filter and Generate Report
8) Change Password
9) Logout
Choose: 5

=== Pending Withdrawal Requests ===

Application ID      Student ID          Internship ID
-------------------------------------------------------
2                   U2310002B           3
```

View Internship Application withdrawal requests

```
== All Internship Opportunities ==

ID  Title                       Description                     Preferred Major          Open Date   Closing Date  Number of Slots  Visibility  Status      Level
1   Test Internship 4           This is test 4                  Data Science & AI        2025-12-01  2025-12-31    5                true        SUCCESSFUL  BASIC
2   Software Development Intern Develop web applications using m... Computer Science       2025-11-18  2025-12-25    1                false       SUCCESSFUL  BASIC
3   Data Analytics Intern       Analyze large datasets and creat... Data Science & AI        2025-11-20  2025-12-28    4                true        SUCCESSFUL  BASIC
4   Network Engineer Intern     Configure and maintain network i... Computer Engineering     2025-11-19  2025-12-30    2                true        SUCCESSFUL  INTERMEDIATE
5   AI Research Intern          Research machine learning algori... Data Science & AI        2025-11-21  2026-01-05    3                true        SUCCESSFUL  ADVANCED
6   Mobile App Developer        Build iOS and Android applications  Computer Science         2025-11-18  2025-12-27    5                true        SUCCESSFUL  INTERMEDIATE
7   Cloud Solutions Intern      Work with AWS and Azure cloud pl... Computer Engineering     2025-11-22  2025-12-29    3                true        SUCCESSFUL  INTERMEDIATE
8   Cybersecurity Analyst Intern Monitor security systems and res... Computer Science       2025-11-20  2026-01-10    2                true        SUCCESSFUL  ADVANCED
9   UI/UX Design Intern         Design user interfaces for web a... Information Engineerin... 2025-11-19  2025-12-26    4                true        SUCCESSFUL  BASIC
10  Database Administrator Intern Manage SQL and NoSQL databases    Computer Engineering     2025-11-23  2025-12-31    3                true        SUCCESSFUL  INTERMEDIATE
11  Backend Developer Intern    Build REST APIs and microservices   Computer Science         2025-11-18  2025-12-24    4                true        SUCCESSFUL  INTERMEDIATE
12  Machine Learning Engineer   Develop ML models for production... Data Science & AI        2025-11-21  2026-01-08    2                true        SUCCESSFUL  ADVANCED
13  DevOps Engineer Intern      Automate deployment pipelines an... Computer Engineering     2025-11-20  2025-12-30    3                true        SUCCESSFUL  INTERMEDIATE
14  Frontend Developer Intern   Create responsive web interfaces... Computer Science         2025-11-19  2025-12-28    5                false       SUCCESSFUL  BASIC
15  Video Production Intern     Edit and produce multimedia content Information Engineerin... 2025-11-22  2025-12-27    3                true        SUCCESSFUL  BASIC
16  Systems Integration Intern  Integrate enterprise software sy... Computer Engineering     2025-11-18  2026-01-05    2                true        SUCCESSFUL  INTERMEDIATE
17  Quality Assurance Intern    Test software and report bugs       Computer Science         2025-11-23  2025-12-29    3                true        SUCCESSFUL  BASIC
18  Blockchain Developer Intern Develop smart contracts and DApps   Computer Science         2025-11-20  2026-01-12    2                true        SUCCESSFUL  ADVANCED
19  IoT Solutions Intern        Build Internet of things applica... Computer Engineering     2025-11-21  2025-12-31    3                true        SUCCESSFUL  INTERMEDIATE
20  Business Intelligence Intern Create dashboards and reports fo... Data Science & AI        2025-11-19  2025-12-26    4                true        SUCCESSFUL  BASIC
```

View All Internship Opportunities

```
Choose: 6

=== Pending Withdrawal Requests ===

Application ID      Student ID          Internship ID
-------------------------------------------------------
2                   U2310002B           3

Enter the ID of the internship application to approve withdrawal: 2
Approve Application Withdrawal? (Y/N): Y
SUCCESS: Withdrawal approved. Application removed.
```

Approve/Reject Internship Application withdrawal requests

```
=== Pending Internship Opportunities ===

ID  Title                       Description                     Preferred Major          Open Date   Closing Date  Number of Slots  Visibility  Status      Level
31  Hardware Testing Intern     Test and validate computer hardw... Computer Engineering     2025-11-18  2025-12-30    4                true        PENDING     BASIC
Enter the ID of the internship opportunity to approve: 31
Approve Career Opportunity? (Y/N): N
SUCCESS: Internship opportunity rejected successfully.
```

Approve/Reject Internship Opportunites

5. Group Reflection

5.1. Difficulties Encountered and Overcoming It

One of our early challenges was designing the class diagram to follow the UML standard and our algorithm to comply with the SOLID principles. We had to rework our UML diagram multiple times because the first version did not include many classes we needed. After every redesign, we had to redesign the layout to make it look clean, neat, and easy to read. Coding the actual application using the diagram itself, however, was the biggest challenge, and it was far from straightforward as we constantly bumped into errors and bugs. Fixing one bug sometimes resulted in another bug, which should not be the case if we complied with the SOLID principles. As such, fixing one bug sometimes involved rethinking the structure and inheritance, resulting in restructuring part of the project, which required testing all the functions again and again. However, we are glad that we managed to do the whole thing.

5.2. Knowledge Gained

We gained the ability to utilise the SOLID principles by doing our object-oriented application project. We learned how a large project should be scalable, extendable with new features in the future, without modifying what should not be affected, a practice that we will use in the professional field after university. We also learned the hard way how to choose a “difficult but good” design as opposed to an “easy but bad” design, an important experience for us, who often choose to ignore the important principles and pack our classes with too many functions for the sake of simplicity and ease of coding. We also learned the importance of coordination and collaboration, making sure that everyone was on the same page at the same time while working on each one’s portion, and fixing the bugs that could affect each other’s work.

5.3. Areas of Improvement

We had a fairly limited time when working on this project. However, had we had more time, we would have liked to give this project a proper graphical user interface to make it more intuitive and easier to use for those not familiar with using a command-line interface. We believe that our application, with a well-designed UI, is fit for actual real-world use for universities and businesses alike.