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 honored 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
Li Shuhui	CSC	FCSH	Li Shuhui 17/4/2025
Pong Pip Hui	CSC	FCSH	17/4/2025
Shannon Wong Tze Shian (Huang Zhixuan)	CSC	FCSH	17/4/2025
Tan Jia Le, Caleb	CSC	FCSH	17/4/25

Project Title: Build-To-Order (BTO) Management System

Group Members:

Name	**Matriculation Number**
Shannon Wong Tze Shian (Huang Zhixuan)	U2421914B
Pong Pip Hui	U2422311F
Li Shuhui	U2422454A
Tan Jia Le, Caleb	U2422263C

Design Considerations and Use of OO Concepts

User class follows Open-Closed Principle (OCP) as it is closed for modification and open for extension such as by Applicant, HDBOfficer and HDBManager classes. Methods under User class (login and change password) are required by all different types of users using this system. Furthermore, the functions of this class do not have to be changed if a new type of User (new subclass) is introduced since they follow a general logic which is unlikely to change over time. This allows new subclasses of User to be created and extend the User class. Liskov Substitution Principle (LSP) can also apply as the functions in User are not overridden by its subclasses, thus subtypes are substitutable for their base types without altering program functionality.
The new excel sheets of the new subclasses can also easily be exported into the code using UserDatabase class functions without modifying most of its underlying code. UserDatabase class displays Single Responsibility Principle (SRP) as it handles the conversion of excel sheet data to create HashMaps containing users' details.

We decided to create "__Manager" control classes for Application, Project and Enquiry entity classes to separate their roles. This allows the entity classes to follow SRP as they mainly contain attributes, setter and getter methods, and their single focus is to hold the information of the particular class. This ensures code reusability and readability. Their respective control classes manage the logic flow of functions involving the entity class, which allows for better coordination and accessibility when modifying or using ArrayLists of objects of entity classes. One trade-off was that the control classes have many functions, which causes the code to be lengthy. However, the logic of functions such as approving HDBOfficer application in

ProjectManager class and booking flat function in ApplicationManager classes are unlikely to change as well, thus these control classes are unlikely to be modified and the length of code will not unlikely raise an issue. Splitting into entity and control classes also reduces code dependency as changing of a function in a class is unlikely to affect functions in other classes.

We also created "__Display" classes for Applicant, HDBOfficer and HDBManager instead of putting all codes in one Main.java file, ensuring code readability. Since the subclasses of User have different capabilities, the different "__Display" classes ensure minimal changes to the Main class if new subclasses are introduced in the future, ensuring code reusability. These "__Display" classes are also boundary classes as they interact with external users, receiving input from users and passing information to control logic, and printing output according to resulting logic.

We considered applying Interface Segregation Principle (ISP) to the EnquiryInterface interface. However, we realised that too many interfaces will be created since our interface holds multiple abstract functions. However, future enquiry-related classes (apart from EnquiryManager) implementing this interface will need to implement all the functions declared under Interface as they are the basic functions required for handling enquiries of the Enquiry class in this system, thus there is no point in splitting the interface into multiple specific interfaces.

Our code also demonstrates the use of modern Java features such as enhanced for loops, .stream() and .filter(). These tools enabled us to write cleaner codes by abstracting iteration and filtering into more concise expressions, allowing clarity and improved code readability.

UML Class Diagram

[NOTE: the Class Diagram image file is found in the zip folder]

- **How are the main classes identified?**

  Based on the diagram in the file, different roles and objects we need in the system. We also added some small classes in the progress to complete the system.

- **What are the responsibilities of class?**

- User, Applicant, HDBOfficer and HDBManager : entity classes, to create different types of users for our system; UserManager: to store all the users and their information based on their identities; ApplicantDisplay, HDBOfficerDisplay, HDBManagerDisplay :

boundary class, to interact with the user, and call the related methods he needs; Application: entity class, representing a flat application submitted by an applicant; Project: entity class, representing a project created by HDB Manager; Enquiry: entity class, representing a enquiry submitted by the applicant including the messages and the replies; ApplicantManager, ProjectManager, EnquiryManager: control class, performing all the methods related to applications, projects and enquiries; UserDatabase: reading and structuring raw user data from Excel files; ProjectDatabase: creating and storing lists of user objects for the entire system; Filter: encapsulating filtering criteria used to narrow down project listings; Main: reading user data, structuring it using helper functions, and then launching the appropriate interface (Display class) based on the user's role.

- **How were relationships determined?**

  We first assumpted the relationships based on their interactions and behavior. Then when implementing the code, we thought over and improved it. Some examples:

  - Generalization: we capture the common attributes and behaviors of all the users
  - Association: we use it when a class need to have a reference to another class
  - Dependency: we use it when a class like manager class needs to pass a class as a parameter.

UML Sequence Diagrams

[NOTE: the Sequence Diagrams image files are found in the zip folder]

• How were the sequence diagrams decided upon?

- We determined which processes were the most important and central to our system's functionality. This led us to decide on the process for applicants to submit an application for a project, as well as officers to register to be HDB officers in charge of a particular project. We believed that these 2 processes were more complicated to understand as they involve interactions between multiple classes and objects, and we aimed to illustrate these 2 scenarios to showcase the flow of control and data within our system. This ensures clarity in our design and enables programmers to understand our system better.

Analysing the Project and Requirements

We began by carefully reviewing the BTO project document, going through the individual roles of Applicant, HDB Officer and HDB Manager and the system requirements. From this, we collated a summary of possible functions based on the specific abilities of the different users, as well as other essential features of the system. For each function, we wrote brief descriptions on their expected behaviours according to the requirements provided.

Next, we brainstormed on possible classes to create and analysed their possible relationships to represent these entities within the system. We also examined the test cases provided to further solidify our understanding of how the system should flow. We then considered object-oriented design principles to guide the creation of additional classes. After thorough discussions and evaluations, we finalised the list of classes we thought were most suitable to support the HDB booking system. We then mapped the identified functions to the appropriate classes, ensuring a logical and cohesive distribution of responsibilities, while keeping design principles in mind. Finally, we reviewed our list of functions and classes and identified additional functions necessary to enhance overall system flow and usability.

Deciding on Features and Scope

We decided to focus on implementing all core features of the project provided in the requirements. The core features include:

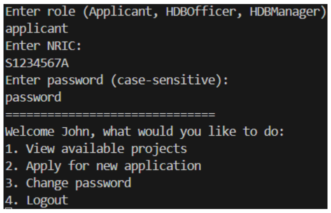
- logging in and changing password (by all users); applying for project, withdrawing application (by Applicant, HDB Officers); submit, view, edit enquiries (Applicant, HDB Officers); replying enquiries (HDB Officers); registering to be a HDBOfficer of a particular project; creating, editing, deleting project (by HDB Managers); viewing projects (differ for Applicants - according to their user group, HDB Officers, HDB Manager) - include filters; approving, disapproving HDB Officer registration (HDB Managers); approving, rejecting applications; approving, rejecting withdrawal requests (HDB Manager); booking a flat upon successful application (HDB Officer);

We did not add in the feature to save new projects created into an excel sheet as the way our program is implemented is such that the system continues running until we enter "ENDPROGRAM", which represents the shutting down of this booking system in real life application. As such, we loaded the existing user information and project information from the excel sheets provided, but in real life contexts, our system will run continuously and the projects created will be saved to the specific HDBManager object loaded in the system. The system resets to the original state once the program ends. If given more time, we would like to implement such that data on new projects, applications, officer registration (pending or successful), as well as flat booked can be saved into an excel sheet or a cloud to store permanently, instead of getting lost when the system ‘shuts down’.

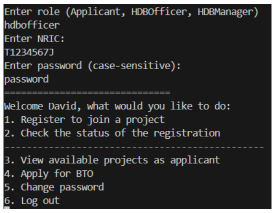
Testing

1. Valid User Login

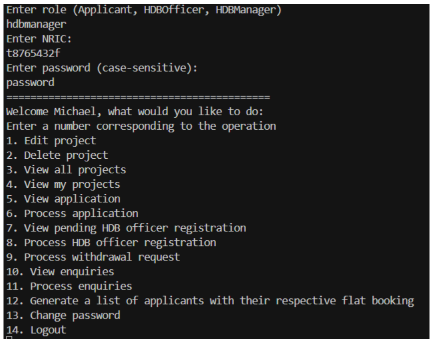
(1: Applicant) (2: HDBOfficer) (3: HDBManager)



```
Enter role (Applicant, HDBOfficer, HDBManager)
applicant
Enter NRIC:
S1234567A
Enter password (case-sensitive):
password
==============================
Welcome John, what would you like to do:
1. View available projects
2. Apply for new application
3. Change password
4. Logout
```

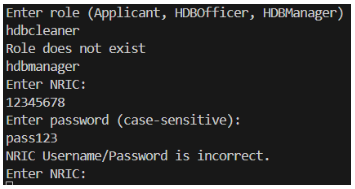


```
Enter role (Applicant, HDBOfficer, HDBManager)
hdbofficer
Enter NRIC:
T1234567J
Enter password (case-sensitive):
password
==============================
Welcome David, what would you like to do:
1. Register to join a project
2. Check the status of the registration
-----------------------------------------------
3. View available projects as applicant
4. Apply for BTO
5. Change password
6. Log out
```



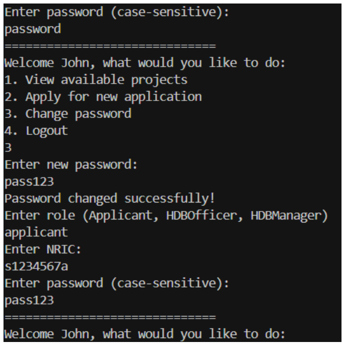
```
Enter role (Applicant, HDBOfficer, HDBManager)
hdbmanager
Enter NRIC:
t8765432f
Enter password (case-sensitive):
password
============================================
Welcome Michael, what would you like to do:
Enter a number corresponding to the operation
1. Edit project
2. Delete project
3. View all projects
4. View my projects
5. View application
6. Process application
7. View pending HDB officer registration
8. Process HDB officer registration
9. Process withdrawal request
10. View enquiries
11. Process enquiries
12. Generate a list of applicants with their respective flat booking
13. Change password
14. Logout
```

2. Invalid NRIC Format + 3. Incorrect Password



```
Enter role (Applicant, HDBOfficer, HDBManager)
hdbcleaner
Role does not exist
hdbmanager
Enter NRIC:
12345678
Enter password (case-sensitive):
pass123
NRIC Username/Password is incorrect.
Enter NRIC:
```

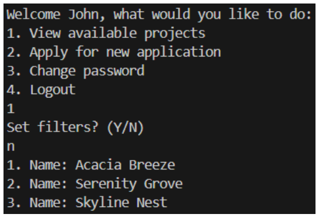
4. Password Change Functionality



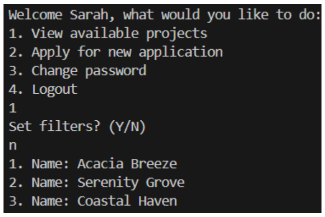
```
Enter password (case-sensitive):
password
==============================
Welcome John, what would you like to do:
1. View available projects
2. Apply for new application
3. Change password
4. Logout
3
Enter new password:
pass123
Password changed successfully!
Enter role (Applicant, HDBOfficer, HDBManager)
applicant
Enter NRIC:
s1234567a
Enter password (case-sensitive):
pass123
==============================
Welcome John, what would you like to do:
```

5. Project Visibility Based on User Group and Toggle

(1: single) (2: married) (3: filtered)



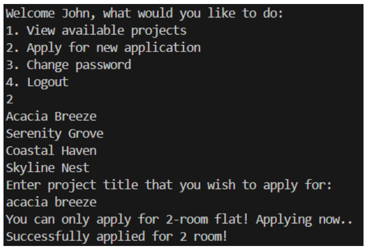
```
Welcome John, what would you like to do:
1. View available projects
2. Apply for new application
3. Change password
4. Logout
1
Set filters? (Y/N)
n
1. Name: Acacia Breeze
2. Name: Serenity Grove
3. Name: Skyline Nest
```



```
Welcome Sarah, what would you like to do:
1. View available projects
2. Apply for new application
3. Change password
4. Logout
1
Set filters? (Y/N)
n
1. Name: Acacia Breeze
2. Name: Serenity Grove
3. Name: Coastal Haven
```

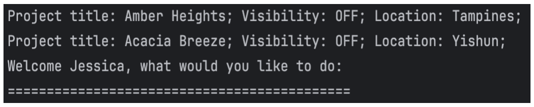
Coastal Haven only have 3 rooms, thus this project is not viewable or appliable by singles; Skyline Nest only have 2 rooms thus we do not let married couples apply for it (for simplicity in this system); Acacia Breeze and Serenity Grove are the only projects in Yishun.

6. Project Application

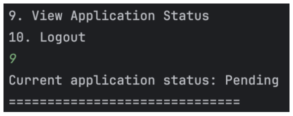


```
Welcome John, what would you like to do:
1. View available projects
2. Apply for new application
3. Change password
4. Logout
2
Acacia Breeze
Serenity Grove
Coastal Haven
Skyline Nest
Enter project title that you wish to apply for:
acacia breeze
You can only apply for 2-room flat! Applying now..
Successfully applied for 2 room!
```

7. Viewing Application Status after Visibility Toggle Off



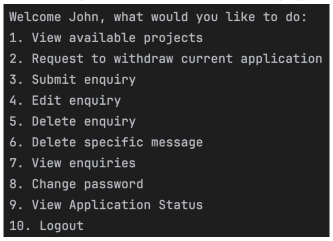
```
Project title: Amber Heights; Visibility: OFF; Location: Tampines;
Project title: Acacia Breeze; Visibility: OFF; Location: Yishun;
Welcome Jessica, what would you like to do:
============================================
```



```
9. View Application Status
10. Logout
9
Current application status: Pending
==============================
```

We manually off the visibility of the project (acacia breeze) that John (applicant) applied for, then check his application status

8. Single Flat Booking per successful Application

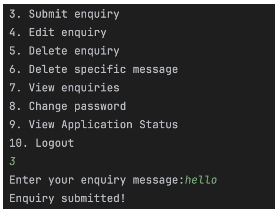


```
Welcome John, what would you like to do:
1. View available projects
2. Request to withdraw current application
3. Submit enquiry
4. Edit enquiry
5. Delete enquiry
6. Delete specific message
7. View enquiries
8. Change password
9. View Application Status
10. Logout
```

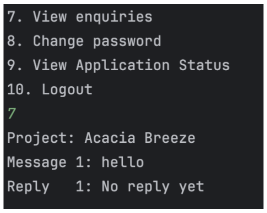
John (applicant) can no longer apply for another project as we removed it as an option under his updated dashboard.

9. Applicant’s enquiries management

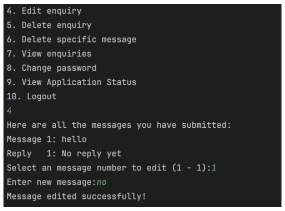
(1: submitted) (2: displayed) (3: edited) (4: check replies) (5-6: deleted)



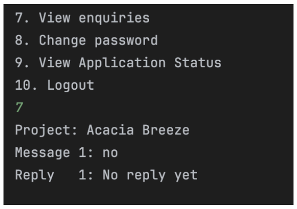
```
3. Submit enquiry
4. Edit enquiry
5. Delete enquiry
6. Delete specific message
7. View enquiries
8. Change password
9. View Application Status
10. Logout
3
Enter your enquiry message:hello
Enquiry submitted!
```



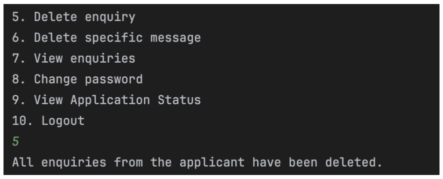
```
7. View enquiries
8. Change password
9. View Application Status
10. Logout
7
Project: Acacia Breeze
Message 1: hello
Reply   1: No reply yet
```



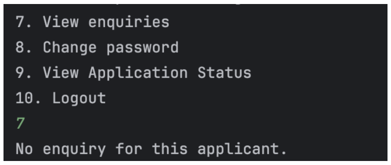
```
4. Edit enquiry
5. Delete enquiry
6. Delete specific message
7. View enquiries
8. Change password
9. View Application Status
10. Logout
4
Here are all the messages you have submitted:
Message 1: hello
Reply   1: No reply yet
Select an message number to edit (1 - 1):1
Enter new message:no
Message edited successfully!
```



```
7. View enquiries
8. Change password
9. View Application Status
10. Logout
7
Project: Acacia Breeze
Message 1: no
Reply   1: No reply yet
```

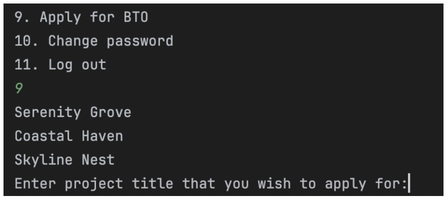


```
5. Delete enquiry
6. Delete specific message
7. View enquiries
8. Change password
9. View Application Status
10. Logout
5
All enquiries from the applicant have been deleted.
```



```
7. View enquiries
8. Change password
9. View Application Status
10. Logout
7
No enquiry for this applicant.
```

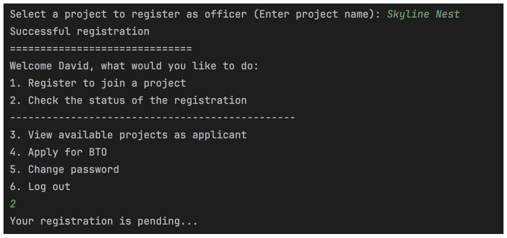
10. HDBOfficer Registration Eligibility



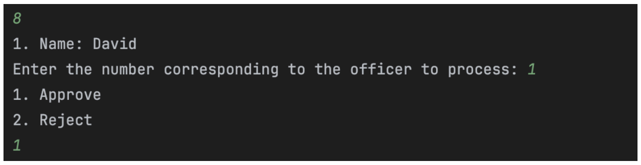
```
9. Apply for BTO
10. Change password
11. Log out
9
Serenity Grove
Coastal Haven
Skyline Nest
Enter project title that you wish to apply for:
```

The system will not show the project the officer is handling when the officer wishes to apply for a BTO project.

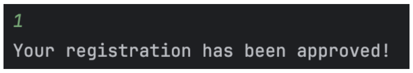
11. HDBOfficer Registration Status



```
Select a project to register as officer (Enter project name): Skyline Nest
Successful registration
==============================
Welcome David, what would you like to do:
1. Register to join a project
2. Check the status of the registration
---------------------------------------------
3. View available projects as applicant
4. Apply for BTO
5. Change password
6. Log out
2
Your registration is pending...
```



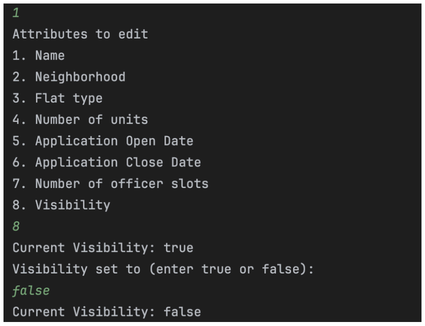
```
8
1. Name: David
Enter the number corresponding to the officer to process: 1
1. Approve
2. Reject
1
```



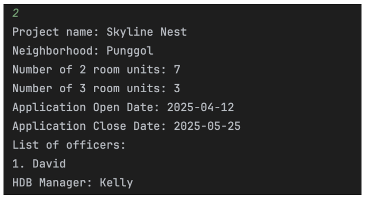
```
1
Your registration has been approved!
```

After David (HDB Officer) registers for a project, registration will be shown as pending before approval or rejection. We login as the HDBManager in-charge of the project he applied for and approved his application here, allowing him to see his status as approved.

12. Project Detail Access for HDBOfficer



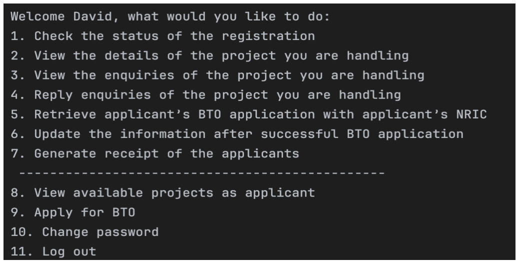
```
1
Attributes to edit
1. Name
2. Neighborhood
3. Flat type
4. Number of units
5. Application Open Date
6. Application Close Date
7. Number of officer slots
8. Visibility
8
Current Visibility: true
Visibility set to (enter true or false):
false
Current Visibility: false
```



```
2
Project name: Skyline Nest
Neighborhood: Punggol
Number of 2 room units: 7
Number of 3 room units: 3
Application Open Date: 2025-04-12
Application Close Date: 2025-05-25
List of officers:
1. David
HDB Manager: Kelly
```

We login as Kelly (HDBManager) to manually turn off visibility of the project (skyline nest), and re-login as David (HDBOfficer) in charge of this project; David is still able to view details of this project.

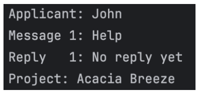
13. Restriction on Editing Project Details



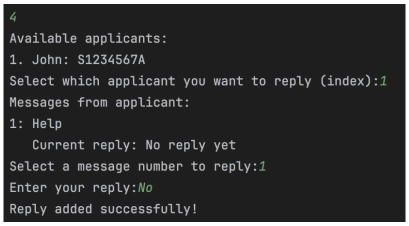
```
Welcome David, what would you like to do:
1. Check the status of the registration
2. View the details of the project you are handling
3. View the enquiries of the project you are handling
4. Reply enquiries of the project you are handling
5. Retrieve applicant's BTO application with applicant's NRIC
6. Update the information after successful BTO application
7. Generate receipt of the applicants
 ----------------------------------------------
8. View available projects as applicant
9. Apply for BTO
10. Change password
11. Log out
```

David (HDB Officer) is not given the option to edit projects in his dashboard.

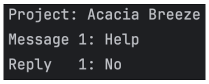
14. Response to Project Enquiries



```
Applicant: John
Message 1: Help
Reply   1: No reply yet
Project: Acacia Breeze
```



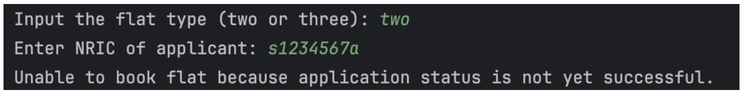
```
4
Available applicants:
1. John: S1234567A
Select which applicant you want to reply (index):1
Messages from applicant:
1: Help
   Current reply: No reply yet
Select a message number to reply:1
Enter your reply:No
Reply added successfully!
```



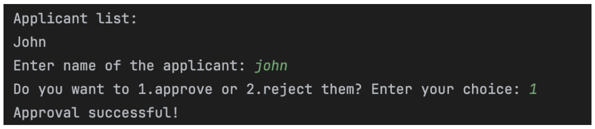
```
Project: Acacia Breeze
Message 1: Help
Reply   1: No
```

The enquiry John (applicant) sent has not been replied to yet. Daniel (HDBOfficer) can view all the enquiries of the project he is in charge of and reply, which will be reflected when the applicant checks his enquiries again afterwards. HDBManager can perform the same functions as well for all his projects.

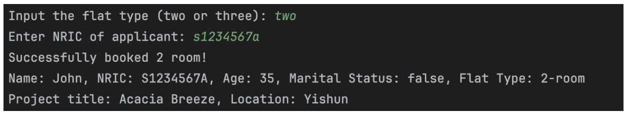
15. Flat Selection and Booking Management



```
Input the flat type (two or three): two
Enter NRIC of applicant: s1234567a
Unable to book flat because application status is not yet successful.
```



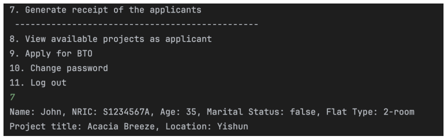
```
Applicant list:
John
Enter name of the applicant: john
Do you want to 1.approve or 2.reject them? Enter your choice: 1
Approval successful!
```



```
Input the flat type (two or three): two
Enter NRIC of applicant: s1234567a
Successfully booked 2 room!
Name: John, NRIC: S1234567A, Age: 35, Marital Status: false, Flat Type: 2-room
Project title: Acacia Breeze, Location: Yishun
```

HDBOfficer can book flat only when an application is successful; HDBManager needs to approve or reject applicant's submission first. After an application has been approved, HDBOfficer can book a flat according to the applicant's user group and print receipt.

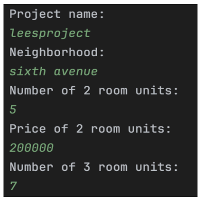
16. Receipt Generation for Flat Booking



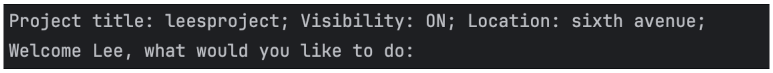
```
7. Generate receipt of the applicants
 ---------------------------------------------
8. View available projects as applicant
9. Apply for BTO
10. Change password
11. Log out
7
Name: John, NRIC: S1234567A, Age: 35, Marital Status: false, Flat Type: 2-room
Project title: Acacia Breeze, Location: Yishun
```

HDBOfficer can also choose to print the receipt for all the successful bookings manually.

17. Create, Edit and Delete BTO Project Listings

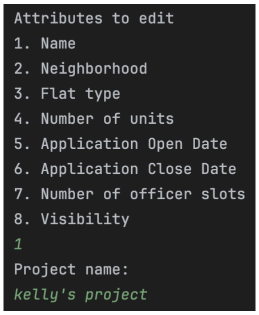


```
Project name:
leesproject
Neighborhood:
sixth avenue
Number of 2 room units:
5
Price of 2 room units:
200000
Number of 3 room units:
7
```

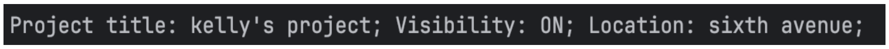


```
Project title: leesproject; Visibility: ON; Location: sixth avenue;
Welcome Lee, what would you like to do:
```

Lee (HDBManager) created a new project successfully.

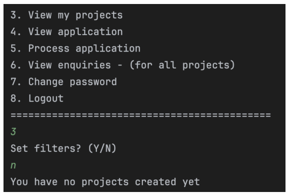


```
Attributes to edit
1. Name
2. Neighborhood
3. Flat type
4. Number of units
5. Application Open Date
6. Application Close Date
7. Number of officer slots
8. Visibility
1
Project name:
kelly's project
```

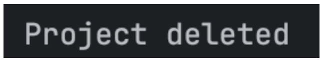


```
Project title: kelly's project; Visibility: ON; Location: sixth avenue;
```

Kelly (HDBManager) edited her project successfully.



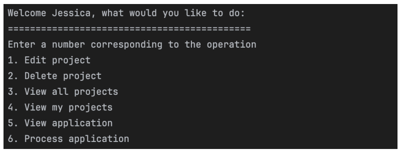
```
3. View my projects
4. View application
5. Process application
6. View enquiries - (for all projects)
7. Change password
8. Logout
==========================================
3
Set filters? (Y/N)
n
You have no projects created yet
```



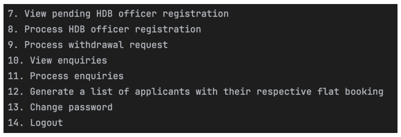
```
Project deleted
```

Lee (HDBManager) deleted his only project successfully, no project displayed when he viewed his own projects.

18. Single Project Management per Application Period



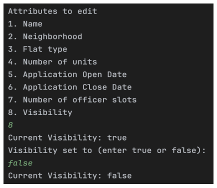
```
Welcome Jessica, what would you like to do:
==========================================
Enter a number corresponding to the operation
1. Edit project
2. Delete project
3. View all projects
4. View my projects
5. View application
6. Process application
```



```
7. View pending HDB officer registration
8. Process HDB officer registration
9. Process withdrawal request
10. View enquiries
11. Process enquiries
12. Generate a list of applicants with their respective flat booking
13. Change password
14. Logout
```

Jessica (HDB Manager) cannot create new projects when she has a current active project (option is unavailable in her dashboard)

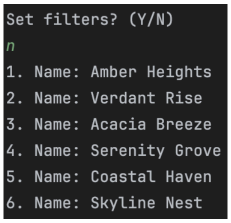
19. Toggle Project Visibility



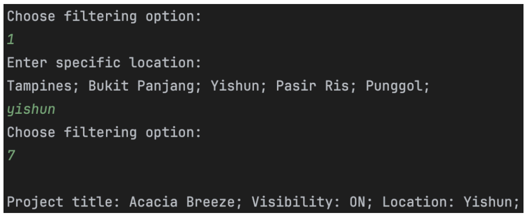
```
Attributes to edit
1. Name
2. Neighborhood
3. Flat type
4. Number of units
5. Application Open Date
6. Application Close Date
7. Number of officer slots
8. Visibility
8
Current Visibility: true
Visibility set to (enter true or false):
false
Current Visibility: false
```

Manager toggles his project's visibility, it changes project's attribute then applicant won't see it.

20. View All and Filtered Project Listings



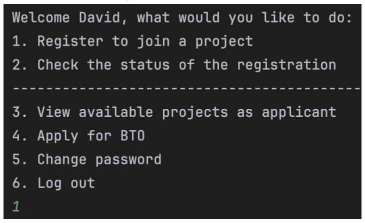
```
Set filters? (Y/N)
n
1. Name: Amber Heights
2. Name: Verdant Rise
3. Name: Acacia Breeze
4. Name: Serenity Grove
5. Name: Coastal Haven
6. Name: Skyline Nest
```



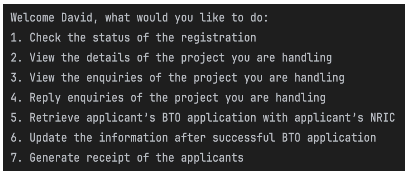
```
Choose filtering option:
1
Enter specific location:
Tampines; Bukit Panjang; Yishun; Pasir Ris; Punggol;
yishun
Choose filtering option:
7


Project title: Acacia Breeze; Visibility: ON; Location: Yishun;
```

HDBManager can choose to view all projects or his own projects, and set different filters on projects to view eg. Yishun projects only.

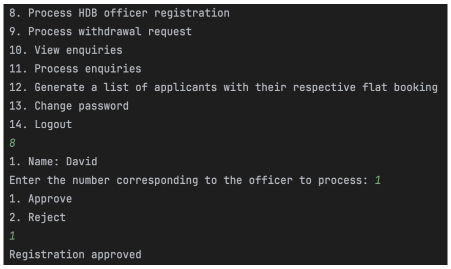
21. Manage HDB Officer registration



```
Welcome David, what would you like to do:
1. Register to join a project
2. Check the status of the registration
-----------------------------------------
3. View available projects as applicant
4. Apply for BTO
5. Change password
6. Log out
1
```



```
Welcome David, what would you like to do:
1. Check the status of the registration
2. View the details of the project you are handling
3. View the enquiries of the project you are handling
4. Reply enquiries of the project you are handling
5. Retrieve applicant's BTO application with applicant's NRIC
6. Update the information after successful BTO application
7. Generate receipt of the applicants
```

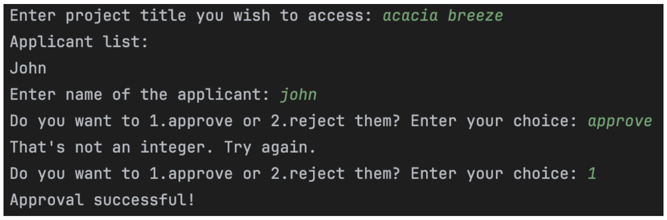


```
8. Process HDB officer registration
9. Process withdrawal request
10. View enquiries
11. Process enquiries
12. Generate a list of applicants with their respective flat booking
13. Change password
14. Logout
8
1. Name: David
Enter the number corresponding to the officer to process: 1
1. Approve
2. Reject
1
Registration approved
```

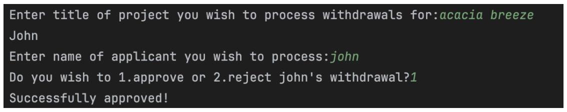
HDBManager can process officers' registrations of his project (approve or reject) , which will update relevant information automatically.

22. Approve or Reject BTO Applications and Withdrawals

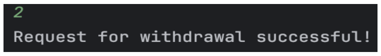
(1: Process applications) (2-3: Process withdrawals)



```
Enter project title you wish to access: acacia breeze
Applicant list:
John
Enter name of the applicant: john
Do you want to 1.approve or 2.reject them? Enter your choice: approve
That's not an integer. Try again.
Do you want to 1.approve or 2.reject them? Enter your choice: 1
Approval successful!
```



```
Enter title of project you wish to process withdrawals for:acacia breeze
John
Enter name of applicant you wish to process:john
Do you wish to 1.approve or 2.reject john's withdrawal?1
Successfully approved!
```

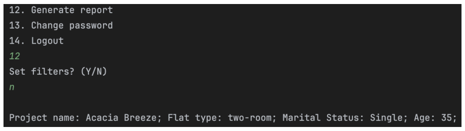


```
2
Request for withdrawal successful!
```

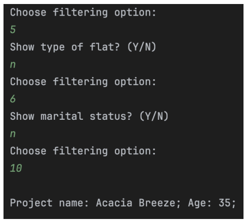
HDBManager can process applications and withdrawals of his project, which will update relevant information automatically.

<u>23. Generate and Filter Reports</u>

(1: unfiltered) (2: filtered)



```
12. Generate report
13. Change password
14. Logout
12
Set filters? (Y/N)
n

Project name: Acacia Breeze; Flat type: two-room; Marital Status: Single; Age: 35;
```



```
Choose filtering option:
5
Show type of flat? (Y/N)
n
Choose filtering option:
6
Show marital status? (Y/N)
n
Choose filtering option:
10

Project name: Acacia Breeze; Age: 35;
```

Reflections

Throughout this project, one of the main challenges we encountered was designing the class structure. We wanted to ensure that we applied Java's design principles in our code, such as Single Responsibility Principle (SRP) and Interface Segregation Principle (ISP). However, if we applied these principles to our entire project, we would end up creating too many singly-focused classes. Although this can improve our code reusability as such classes can be used for future projects, it would increase the overall complexity of our program. We would also face difficulty tracing program logic with the need to keep track of the functions implemented by each class. As such, we sacrificed such principles in some classes and focused on making our function logic well-defined and more generalised and reusable to avoid the need to modify our classes in future. We also focused on reducing dependency between our classes to avoid causing a butterfly effect if there is a need to modify any classes in the future to reduce the number of classes affected by the change.

Additionally, our team had differing opinions on how we wanted to structure the code and which type of classes (such as entity, boundary and control classes) to implement. Through long discussions and compromises, we were able to come to a consensus that was agreed by everyone. This experience allowed us to improve our teamwork and communication skills, and also helped us strengthen our understanding on the tasks of the group project, ensuring that we all had a clear idea on how to implement the software and were on the same page as each other.

This project also allowed us to gain hands-on experience with modern Java features and apply the design principles learned in lecture, which deepened our understanding on such concepts as we needed a good knowledge of these concepts in order to incorporate them accurately throughout our code.

Given more time, our group would like to add on more features in our system, such as file I/O support for saving Application reports and User details into excel sheets instead of just printing them using System.out.println in the terminal. This allows us to explore more aspects in Java coding and improve our hard coding skills in general.

Currently, HDBManager is also only able to create projects if the current project they are handling expires (and is moved into their list of projects). Given more time, we would like to implement the use of an automatic date checker to periodically check the current dates and the opening dates and closing dates of the project created and current active project of each HDBManager. This allows us to automatically shift the current project set as an active project to the list of expired projects in the Project Manager and replace it with an upcoming or current 'ON' project from the individual lists of projects created by the specific HDBManager. This allows automatic updating of the active project HDBManager is handling since HDBManager can only handle 1 active project at a time.

Link to the GitHub repository used in our project

https://github.com/cronky05/sc2002-project