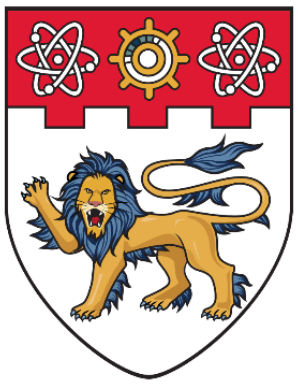


NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

SC/CE/CZ2002: Object-Oriented Design & Programming

ASSIGNMENT

Building an OO Application

2024/2025 SEMESTER 2

COLLEGE OF COMPUTING AND DATA SCIENCE (CCDS)

NANYANG TECHNOLOGICAL UNIVERSITY

1. OBJECTIVES

The main objective of this assignment is

- to apply the Object-Oriented (OO) concepts you have learnt in the course,
- to model, design and develop an OO application.
- to gain familiarity with using Java as an object-oriented programming language.
- to work collaboratively as a group to achieve a common goal.

2. LABORATORY

Assigned SCSE lab.

3. EQUIPMENT

Hardware:	PC (or Laptop)
Software:	Your preferred Java IDE or simply notepad and Java Development ToolKits (JDK)

4. THE ASSIGNMENT

The assignment for your group will be to design and develop a:

Build-To-Order (BTO) Management System

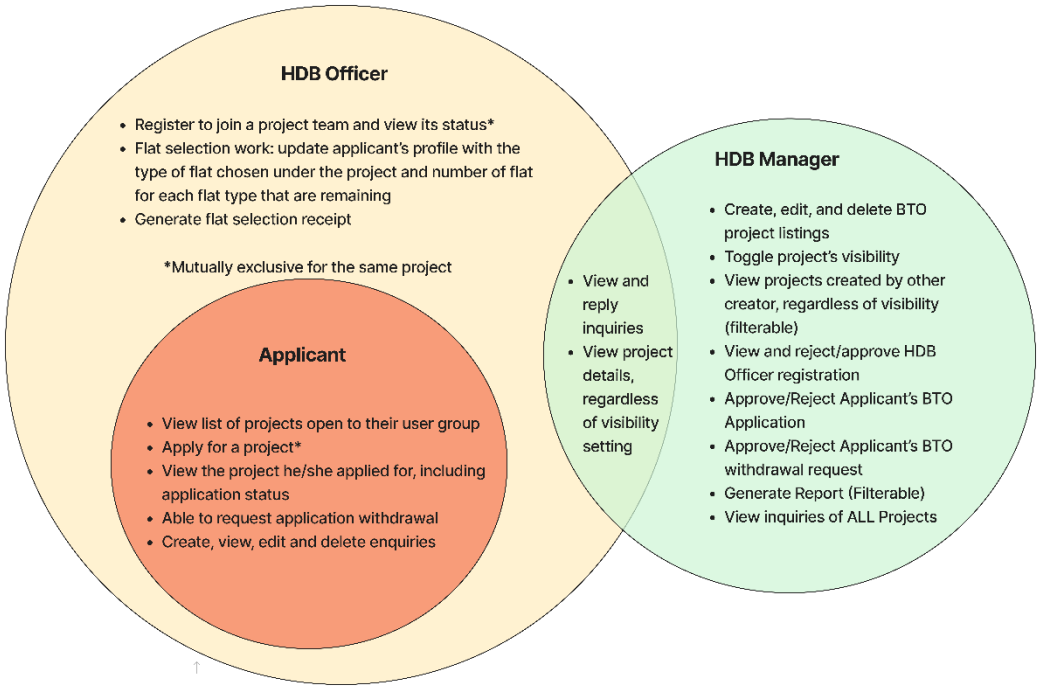
BTO Management System is a system for applicants and HDB staffs to view, apply and manage for BTO projects.

The following are information about the system:

Overview of the System:

- The system will act as a centralized hub for all applicants and HDB staffs
- All users will need to login to this hub using their Singpass account.
    - User ID will be their NRIC, that starts with S or T, followed by 7-digit number and ends with another letter
    - Assume all users use the default password, which is password.
    - A user can change password in the system.
    - Additional Information about the user: Age and Marital Status
- A user list can be initiated through a file uploaded into the system at initialization.

User's capabilities:



1. Applicant
    - Can only view the list of projects that are open to their user group (Single or Married) and if their visibility has been toggled "on".
    - Able to apply for a project – cannot apply for multiple projects.
        - Singles, 35 years old and above, can ONLY apply for 2-Room
        - Married, 21 years old and above, can apply for any flat types (2-Room or 3-Room)
    - Able to view the project he/she applied for, even after visibility is turned off, and the application status
        - Pending: Entry status upon application – No conclusive decision made about the outcome of the application
        - Successful: Outcome of the application is successful, hence invited to make a flat booking with the HDB Officer
        - Unsuccessful: Outcome of the application is unsuccessful, hence cannot make a flat booking for this application. Applicant may apply for another project.
        - Booked: Secured a unit after a successful application and completed a flat booking with the HDB Officer.
    - If Application status is "Successful", Applicant can book one flat through the HDB Officer (Only HDB Officer can help to book a flat) - cannot book more than one flat, within a project or across different project

- Allowed to request withdrawal for their BTO application before/after flat booking
- Able to submit enquiries, a string, regarding the projects.
- Able to view, edit, and delete their enquiries.

2. HDB Officer
   - HDB Officer possess all applicant's capabilities.
   - Able to register to join a project if the following criteria are meant:
     - No intention to apply for the project as an Applicant (Cannot apply for the project as an Applicant before and after becoming an HDB Officer of the project)
     - Not a HDB Officer (registration not approved) for another project within an application period (from application opening date, inclusive, to application closing date, inclusive)
   - Able to see the status of their registration to be a HDB Officer for a project
   - Registration to be a HDB Officer of the project is subject to approval from the HDB Manager in charge of the project. Once approved, their profile will reflect the project he/she is a HDB Officer for.
   - Able to apply for other projects in which he/she is not handling – Once applied for a BTO project, he/she cannot register to handle the same project
   - Able to view the details of the project he/she is handling regardless of the visibility setting.
   - Not allowed to edit the project details.
   - Able to view and reply to enquiries regarding the project he/she is handling
     - With Applicant's successful BTO application, HDB Officer's flat selection responsibilities:
       - Update the number of flat for each flat type that are remaining
       - Retrieve applicant's BTO application with applicant's NRIC
       - Update applicant's project status, from "successful" to "booked".
       - Update applicant's profile with the type of flat (2-Room or 3-Room) chosen under a project
   - Able to generate receipt of the applicants with their respective flat booking details – Applicant's Name, NRIC, age, marital status, flat type booked and its project details.

3. HDB Manager
    - Able to create, edit, and delete BTO project listings.
    - A BTO project information entered by the HDB Manager will include information like:
        - Project Name
        - Neighborhood (e.g. Yishun, Boon Lay, etc.)
        - Types of Flat – Assume there are only 2-Room and 3-Room
        - The number of units for the respective types of flat
        - Application opening date
        - Application closing date
        - HDB Manager in charge (automatically tied to the HDB Manager who created the listing)
        - Available HDB Officer Slots (max 10)
    - Can only be handling one project within an application period (from application opening date, inclusive, to application closing date, inclusive)
    - Able to toggle the visibility of the project to "on" or "off". This will be reflected in the project list that will be visible to applicants.
    - Able to view all created projects, including projects created by other HDB Manager, regardless of visibility setting.
    - Able to filter and view the list of projects that they have created only.
    - Able to view pending and approved HDB Officer registration.
    - Able to approve or reject HDB Officer's registration as the HDB Manager in-charge of the project – update project's remaining HDB Officer slots
    - Able to approve or reject Applicant's BTO application – approval is limited to the supply of the flats (number of units for the respective flat types)
    - Able to approve or reject Applicant's request to withdraw the application.
    - Able to generate a report of the list of applicants with their respective flat booking – flat type, project name, age, marital status
        - There should be filters to generate a list based on various categories (e.g. report of married applicants' choice of flat type)
    - Cannot apply for any BTO project as an Applicant.
    - Able to view enquiries of ALL projects.
    - Able to view and reply to enquiries regarding the project he/she is handling.

<u>Miscellaneous:</u>

- All users can use filters to view project (location, flat types, etc.) Assume that default is by alphabetical order. User filter settings are saved when they switch menu pages.

The system is to be developed as a **Command Line Interface (CLI) application (Non-Graphical User Interface).**

The sample data file for user list is given in excel in assignment folder. You can

- **Use them directly,**
- **Or copy the content to text file if you plan to read from text file,**
- **Or make your own data files.**

But *No database application (e.g. MySQL, MS Access, etc) is to be used.*
No JSON or XML is to be used.

5. THE REPORT

Your report will include the following:

a) A detailed UML **Class** Diagram for the application (exported as an image)
   - Show clearly the class relationship, notation
   - Notes to explain, if necessary
   - Annotate your UML diagram to highlight where specific OO principles (e.g., encapsulation, polymorphism) are applied.

b) A detailed UML **Sequence** Diagram (exported as an image)
   - Showing the flow of the HDB Officer's role in applying for a BTO and register to handle a project.
   - The diagram should show clearly all participating objects involved with sufficient detailed flow and relevant interaction fragments.

c) Highlight clearly any **additional features/functionalities implemented in the system.**

d) Based on the concepts learned in the lecture, **write-up** on your **design considerations** and use of OO concepts in your current design, extensibility and maintainability of your design. Discuss any trade-offs you made in your design and reflect on how the design patterns you used contribute to the overall system design. Were there alternative patterns you considered? Why did you choose the ones you did?

e) Reflection: The difficulties encountered and the way to conquer, the knowledge learnt from this course, further improvement suggestions. Strong demonstration of learning points and insights of good design and implementation practices, based on experience gained from doing the assignment.

f) Include the link to your GitHub repository used for this project, containing all the relevant files and code.

g) A duly signed **Declaration of Original Work** form (Appendix B)

h) By default, every group member receives an equal mark. However, if your group feels that contributions vary and you wish to assign marks based on individual input, please complete the WBS.xls (located in the same folder as the assignment document) and include it in your report. All members must agree to the contents of the WBS. Additionally, you must email the completed WBS.xls to the course coordinator, ensuring that all members are copied on the email.

6. DEMOSTRATION & PRESENTATION (Deadline: Week 14 Friday, 11:59pm)

Your group is required to present your work to your TA to demonstrate the working of the application – presenting ALL the required functionalities of the application. It is advised that you planned your demonstration in a story-boarding flow to facilitate understanding of your application. Please introduce your members and group number at the start of presentation, all the group members must take turn to present. The presenter should show his/her face while presenting.

In the production, you may include:

a) Explaining essential and relevant information about the application
b) Run-through and elaborate on essential part/s of your implementation/coding

- **The presentation duration must not exceed 15 minutes in total.**
- **The font size used must be large enough to be readable and viewable.**
- **The demo of the application is to be done in real-time and NOT pre-run display.**

- **The presentation can be conducted either through online meeting or physically.**

**You will create your own test cases and data to test your application thoroughly.

7. THE DELIVERABLE (Deadline: One day before your scheduled oral presentation with your TA)

Your group submission should include the following:

a. The report (separate diagram file if diagram is unclear in report)
b. All implementation codes and java documentation (Javadoc).
c. Other relevant files (e.g. data files, setup instruction, etc.)

8. ASSESSMENT WEIGHTAGE

UML Class Diagram [**25 Marks**]

- Show mainly the Entity classes, the essential Control and Boundary classes, and enumeration type (if there is).
- Clarity, Correctness and Completeness of details and relationship.

UML Sequence Diagram [**20 Marks**]

- Show only the sequence Diagram mentioned in 5(b)
- Clarity, Correctness and Completeness of flow and object interactions details (Boundary-Control-Entities)

Design Consideration [**15 Marks**]

- Usage of OO concepts and principle - correctness and appropriateness
- Explanation of design choices and how it fits the project requirements
- Coupling and cohesion of classes

Implementation Code [**20 Marks**]

- Diagram to Code correctness, readability, Java naming convention, exception handling, completeness of Java Doc and overall quality.
- Creativity of the additional features/functionality added to the system.
- A Java API HTML documentation of ALL your defined classes using Javadoc must be submitted. The use of javadoc feature is documented in Appendix D.

Demonstration and report [**20 Marks**]

- Coverage of application essentials and functionalities, user friendliness, demo flow, innovation.
- Report structure and reflection
- Highlight clearly any **additional features** implemented in the system.

9. SUBMISSION

This is a **group assignment**, and one submission from each group.
Report format guidelines are provided in the Appendix C below.

1. Soft copy of your deliverables to be **uploaded** to your individual SC/CE/CZ2002 **LAB site** (e.g. FEP1, FSP1, etc.) in **NTULearn**. The link is provided on the left panel "Assignment Submission". **File name convention** : <lab_grp>-grp<assignment_grp#>.<ext> E.g., FEP2- grp3.pdf [<ext.> can be pdf, doc, zip,]
2. **DEADLINE**: One day before your scheduled oral presentation with your TA.

Important:
Note that THREE (3) marks will be deducted for the delay submission of each calendar day. Lateness is based on the date the captured in NTULearn or subsequent resubmissions (whichever is later). So check your work before submitting.

10. REFERENCES & TOOLS

- UML Diagrams tool - Visual Paradigm
  http://www.visual-paradigm.com/
  http://www.visualparadigm.com/support/documents/vpuserguide/94/2576/7190_drawingclass.html
- NTULearn Cx2002 main course site content
- NTULearn Cx2002 course site content on “File Input/Output”
- Object Serialization tutorial
  http://www.javabeginner.com/uncategorized/javaserialization
- Windows Media Encoder (a suggestion)
  http://www.microsoft.com/expression/products/EncoderPro_Overview.aspx
  [You can also try with the video recording feature for gaming in Windows 10 – press ‘Windows key + G’]

APPENDIX A:

You may refer to the list of sample test cases below as a guide for your testing and demo video. Depending on your design and user-friendliness of your data entries process, there may be multiple steps taken.

The test cases provided are intended as examples. You are encouraged to develop your own test cases to thoroughly validate your system.

No.	Test Cases	Expected Behavior	Failure Indicators
1	Valid User Login	User should be able to access their dashboard based on their roles	User cannot log in or receives incorrect error messages
2	Invalid NRIC Format	User receives a notification about incorrect NRIC format	User is allowed to log in with an invalid NRIC
3	Incorrect Password	System should deny access and alert the user to incorrect password	User logs in successfully with a wrong password
4	Password Change Functionality	System updates password, prompt re-login and allows login with new credentials	System does not update the password or denies access with the new password
5	Project Visibility Based on User Group and Toggle	Projects are visible to users based on their age, marital status and the visibility setting	Users see projects not relevant to their group or when visibility is off
6	Project Application	Users can only apply for projects relevant to their group or when visibility is off	Users can apply for projects not relevant to their group or when visibility is off
7	Viewing Application Status after Visibility Toggle Off	Applicants continue to have access to their application details regardless of project visibility.	Application details become inaccessible once visibility is off.
8	Single Flat Booking per Successful Application	System allows booking one flat and restricts further bookings	Applicant is able to book more than one flat
9	Applicant's enquiries management	Enquiries can be successfully submitted, displayed, modified, and removed.	Enquiries cannot be submitted, edited, or deleted; or do not display correctly.

10	HDB Officer Registration Eligibility	System allows registration only under compliant conditions	System allows registration while the officer is an applicant or registered for another project in the same period
11	HDB Officer Registration Status	Officers can view pending or approved status updates on their profiles.	Status updates are not visible or incorrect
12	Project Detail Access for HDB Officer	Officers can always access full project details, even when visibility is turned off.	Project details are inaccessible when visibility is toggled off
13	Restriction on Editing Project Details	Edit functionality is disabled or absent for HDB Officers.	Officers are able to make changes to project details
14	Response to Project Enquiries	Officers & Managers can access and respond to enquiries efficiently.	Officers & Managers cannot see enquiries, or their responses are not recorded.
15	Flat Selection and Booking Management	Officers retrieve the correct application, update flat availability accurately, and correctly log booking details in the applicant's profile.	Incorrect retrieval or updates, or failure to reflect booking details accurately.
16	Receipt Generation for Flat Booking	Accurate and complete receipts are generated for each successful booking	Receipts are incomplete, inaccurate, or fail to generate
17	Create, Edit, and Delete BTO Project Listings	Managers should be able to add new projects, modify existing project details, and remove projects from the system	Inability to create, edit, or delete projects or errors during these operations.
18	Single Project Management per Application Period	System prevents assignment of more than one project to a manager within the same application dates.	Manager is able to handle multiple projects simultaneously during the same period.
19	Toggle Project Visibility	Changes in visibility should be reflected	Visibility settings do not update or do not

		accurately in the project list visible to applicants	affect the project listing as expected
20	View All and Filtered Project Listings	Managers should see all projects and be able to apply filters to narrow down to their own projects.	Inability to view all projects or incorrect filtering results
21	Manage HDB Officer Registrations	Managers handle officer registrations effectively, with system updates reflecting changes accurately.	Mismanagement of registrations or slot counts do not update properly.
22	Approve or Reject BTO Applications and Withdrawals	Approvals and rejections are processed correctly, with system updates to reflect the decision	Incorrect or failed processing of applications or withdrawals
23	Generate and Filter Reports	Accurate report generation with options to filter by various categories.	Reports are inaccurate, incomplete, or filtering does not work as expected.

APPENDIX B:

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 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

Important notes:

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

APPENDIX C:

Report requirement

1. Format:

For the main content, please use Times New Roman 12 pt font size and 1.5 line spacing. You may choose to use other fonts (e.g., Courier New) for code segments. Please use the following report structure:

- Cover page: Declaration of original work (Appendix B)
- Design Considerations
    - Approach taken, Principles used, Assumptions made, etc.
    - ***Optional***: You can show the important code segment (e.g., a method or a few lines of code) and necessary illustrations to explain your solution.
- Detailed UML Class Diagram.
    - Further Notes, if needed
- Detailed UML Sequence Diagram of stated function.
    - Further Notes, if needed
- Testing.
    - Test Cases and Results
- Reflection.
    - The difficulties encountered and the way to conquer, the knowledge learnt from this course, further improvement suggestion.

2. Length:

The report should be at most 12 pages from cover to cover including diagrams/Testing results/references/appendix, if there is any. If you could well present your work in fewer than 12 pages, you are encouraged to do so.

DO NOT include source code in the report but instead store the source code in a folder. You are to ensure that the diagrams are readable and clear to the reader. [You can save the diagrams as image files and include in a folder]

APPENDIX D:

Creating Javadoc:

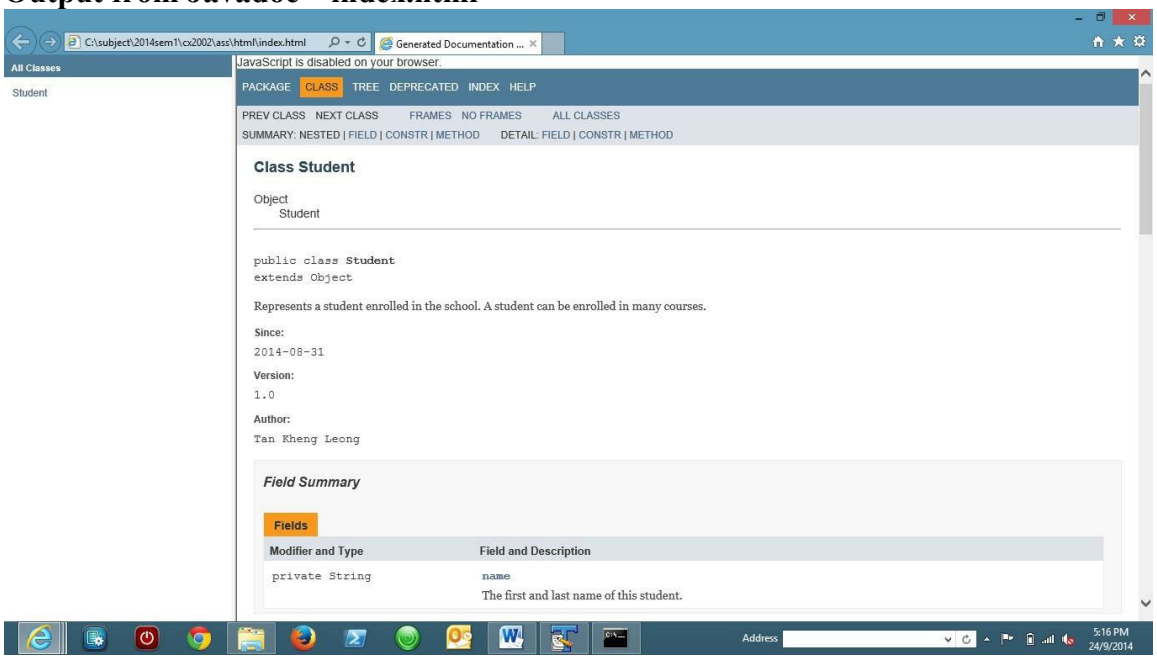
Detailed can be found at
http://www.oracle.com/technetwork/java/javase/documentation/index-137868.html

Using Javadoc in Eclipse : Youtube : http://www.youtube.com/watch?v=Hx-8BD_Osdw

Below is a short example :

```
/**
    Represents a student enrolled in the
 school. A student can be enrolled in many
 courses. @author Tan Kheng Leong
    @version 1.0
    @since 2014-08-31
  */
public class Student {

    /**
        The first and last name of this student.
      */
    private String name;

    /**
        The age of this student.
      */
    private int age;

    /**
        Creates a new Student with the given name.
        The name should include both first and
        last name.
        @param name This Student's name.
        @param age This Student's age.
      */
      public Student(String name, int age)
   { this.name = name;
        this.age = age;
    }

    /**
        Gets the first and last name of this Student.
        @return this Student's name.
      */
      public String
   getName() { return name;
    }

    /**
        Changes the name of this Student.
        This may involve a lengthy legal process.
        @param newName This Student's new name.
        Should include both first
        and last name.
      */
      public void setName(String
   newName) { name = newName;
      }
}
```

Output from Javadoc – index.html



For those familiar with using command prompt :

Steps to general API doc :

(1) Locate the installed path of JDK (java development kit)
- In Windows, it should be in C:\Program Files\Java\jdk<version>\

(2) Open command prompt

(3) Go to your src directory using cd

(4) At prompt …..src> <path to jdk>\bin\javadoc" -d ./html -author -private -noqualifier all -version <packagename1> <packagename2> <….>

Eg .

C:\subject\2014sem1\cx2002\src>"C:\Program Files (x86)\Java\jdk1.8.0_05\bin\javadoc" -d ./html -author
-private -noqualifier all -version edu.ntu.sce.cx2002 edu.ntu.sce.cx2003

Statement	**Purpose**
C:\subject\2014sem1\cx2002\src>	Path to your src root
"C:\Program Files (x86)\Java\jdk1.8.0_05\bin\javadoc"	Path to your jdk javadoc.exe [using double quote if path has space in between, eg Program Files]
-d ./html	-d : specific folder to store html doc Eg ./html means current directory create a html folderto store
-author	Include @author in doc, if provided
-private	Include all methods and fields
--noqualifier all	Omitted all full package name. Eg show **String** instead of **java.lang.String**
-version	Include @version in doc, if provided
edu.ntu.sce.cx2002 edu.ntu.sce.cx2003	Different package names