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

ASSIGNMENT

Building an OO Application

2024/2025 SEMESTER 1

SCHOOL OF COMPUTER SCIENCE & ENGINEERING
NANYANG TECHNOLOGICAL UNIVERSITY

1. OBJECTIVE

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:

Hospital Management System (HMS)

HMS is an application aimed at automating the management of hospital operations, including patient management, appointment scheduling, staff management, and billing. The system is expected to facilitate efficient management of hospital resources, enhance patient care, and streamline administrative processes.

User Roles and System Capabilities:

1. **All Users**:
    - Users must log in to the system using their unique hospital ID and a default password ("password").
    - The system should validate login credentials and provide role-specific access to the HMS.
    - Users can change their password after their initial login.
    - Users will have roles such as Patient, Doctor, Pharmacist or Administrator.

2. **Patient:**

    Information Access:

    - Patients can view their own **medical record**, which consists of:
        - Patient ID, Name, Date of Birth, Gender

    - Contact Information (e.g., phone number, email address)
    - Blood Type
    - Past Diagnoses and Treatments
  - Patients can update non-medical personal information such as email address and contact number.
  - Patients are **not allowed** to modify the past diagnoses, prescribed medications, treatments or blood type..

Appointment Management:

  - Patients can:
    - View available appointment slots with doctors.
    - **Schedule Appointments:** Choose a doctor, date, and available time slot to schedule an appointment.
    - **Reschedule Appointments:** Change an existing appointment to a new slot, ensuring no conflicts. Upon rescheduling, slot availability will be updated automatically.
    - **Cancel Appointments:** Cancel an existing appointment. Upon successful cancellation, slot availability will be updated automatically.
  - Patients can view the status of their scheduled appointments.
  - The status of the appointment will be updated according based on whether the doctor accepts of decline the appointment requests (eg: confirmed, canceled, completed)
  - Patients can also view their **Appointment Outcome Records** of past appointments.

3. **Doctor**:

Medical Record Management:

  - Doctors can view the medical records of patients under their care.
  - Doctors can update the medical records of patients by adding new diagnoses, prescriptions, and treatment plans.

Appointment Management:

  - Doctors can view their personal schedule and set their availability for appointments.
  - Doctors can accept or decline appointment requests.
  - Doctors can view the list of their upcoming appointments.
  - **Appointment Outcome Record**: After each completed appointment, the doctor will record the:
    - Date of Appointment
    - **Type of service** provided (e.g., consultation, X-ray, blood test etc).
    - Any **prescribed medications:**
      - medication name
      - status (default is *pending*)
    - Consultation notes

4. **Pharmacist**:

Prescription Management:

- Pharmacists can view the **Appointment Outcome Record** to fulfill medication prescriptions orders from doctors.
- Pharmacists can update the status of prescription in the Appointment Outcome Record (e.g., *pending* to *dispensed*).
- Pharmacists can monitor the inventory of medications, including tracking stock levels.
- Pharmacists can submit replenishment requests to administrators when stock levels are low.

5. **Administrator**:

Staff Management:

- Manage hospital staff (doctors and pharmacists) by adding, updating, or removing staff members.
- Display a list of staff filtered by role, gender, age, etc.

Appointment Management:

- Administrators can access real-time updates of scheduled appointments.
- Appointment details should include:
  - Patient ID
  - Doctor ID
  - Appointment status (e.g., confirmed, canceled, completed).
  - Date and time of appointment
  - Appointment Outcome Record (for completed appointments)

Inventory Management:

- Administrators can view and manage the inventory of medication including, adding, removing or updating stock levels.

- Administrators can update the low stock level alert line of each medicine.
- Administrators can approve replenishment requests from pharmacists. Once the request is approved, the stock level will be updated automatically.

System initialization:

- The initial staff list can be imported from a file.

- The initial patient list can be imported from a file.

- The initial inventory, including medicine name, initial stock, low stock level alert line can be imported from a file.

User Menus

Each user in the HMS will have a specific menu with options relevant to their role upon log in. Below are the menu options for each user role in the system:

Patient Menu:

- View Medical Record
- Update Personal Information
- View Available Appointment Slots
- Schedule an Appointment
- Reschedule an Appointment
- Cancel an Appointment
- View Scheduled Appointments
- View Past Appointment Outcome Records
- Logout

Doctor Menu:

- View Patient Medical Records
- Update Patient Medical Records
- View Personal Schedule
- Set Availability for Appointments
- Accept or Decline Appointment Requests
- View Upcoming Appointments
- Record Appointment Outcome
- Logout

Pharmacist Menu:

- View Appointment Outcome Record
- Update Prescription Status
- View Medication Inventory
- Submit Replenishment Request
- Logout

Administrator Menu:

- View and Manage Hospital Staff
- View Appointments details
- View and Manage Medication Inventory
- Approve Replenishment Requests
- Logout

Additional Instructions:

- **Tip:** You are encouraged to add more features and functionalities to your Hospital Management System. Use your creativity to enhance the user experience, improve system performance, or introduce innovative solutions to common healthcare challenges. Additional features will be considered during evaluation and can help demonstrate your understanding and application of Object-Oriented concepts.

- **Assumptions:**

- Document any assumptions made during the development process. Discuss how these assumptions influenced your design and what potential risks or challenges they may introduce.

- **Development Constraints:**
  - The application is to be developed as a **Command Line Interface (CLI)** application (non-Graphical User Interface).
  - Students are required to create their own data files.
  - **No database application** (e.g., MySQL, MS Access, etc.) is to be used.
  - **No JSON or XML** formats are 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) Highlight clearly any **additional features/functionalities implemented in the system.**

c) 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?

d) 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.

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

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

g) Member's work contribution and distribution breakdown. *If your group feels that marks should be given based on contribution, your group can fill up the WBS.xls(in the same folder as assignment doc) and include it in this report. All members MUST consent to the WBS contents.* ***You must also email the WBS.xls to the course-***

coordinator with ALL members in the loop

6. **DEMONSTRATION & Presentation (Deadline: Friday of week 14)**

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 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 (eg 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.

Design Consideration [**20 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 [**30 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 [**25 Marks**]

- Coverage of application essentials and functionalities, user friendliness, demoflow, 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** (eg FEP1, FSP1, etc) in **NTULearn**. The link is provided on the left panel "Assignment Submission".

   File name convention : <lab_grp>-grp<assignment_grp#>.<ext> Eg, FEP2- grp3.pdf

   [<ext> can be pdf, doc, zip,]

2. **DEADLINE** : **Week 14 Sunday, 11.59pm.**

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.visual-paradigm.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/java-serialization
- Windows Media Encoder (a suggestion) http://www.microsoft.com/expression/products/Enc oderPro_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.

1. Patient Actions

Test Case 1: View Medical Record

- Patients view their own medical record.
- Verify that the system displays the patient's medical record, including Patient ID, Name, Date of Birth, Gender, Contact Information, Blood Type, and Past Diagnoses and Treatments.

Test Case 2: Update Personal Information

- Patient updates their email address and contact number.
- Verify that patient's contact information is updated successfully, and the changes are reflected in the medical record.

Test Case 3: View Available Appointment Slots

- Patient views available appointment slots with doctors.
- Verify that the system displays a list of available appointment slots, showing doctors' names, dates, and times.

Test Case 4: Schedule an Appointment

- Patient schedules a new appointment with a doctor.
- Verify that the appointment is scheduled successfully with status **"confirmed"**. The selected time slot becomes unavailable to other patients. The system should prevent the patient from booking a time slot that is unavailable/already booked.

Test Case 5: Reschedule an Appointment

- Patient reschedules an existing appointment to a new slot.
- Verify that the appointment is rescheduled successfully. The previous time slot becomes available, and the new slot is reserved.

Test Case 6: Cancel an Appointment

- Patient cancels an existing appointment.
- Verify that the appointment is canceled successfully, and the time slot becomes available for others.

Test Case 7: View Scheduled Appointments

- Patient views their list of scheduled appointments.
- Verify that the system displays all upcoming appointments with details like doctor name, date, time, and status.

Test Case 8: View Past Appointment Outcome Records

- Patient views outcome records of past appointments.
- Verify that the system displays past appointment details, including services provided, prescribed medications, and consultation notes.

2. Doctor Actions

Test Case 9: View Patient Medical Records

- Doctor views medical records of patients under their care.
- Verify that the patient's medical record is displayed, including all relevant medical history.

Test Case 10: Update Patient Medical Records

- Doctor adds a new diagnosis and treatment plan to a patient's medical record.
- Verify that the medical record is updated successfully, reflecting the new information.

Test Case 11: View Personal Schedule

- Doctor views their personal appointment schedule.
- Verify that the system displays the doctor's upcoming appointments and availability slots.

Test Case 12: Set Availability for Appointments

- Doctor sets or updates their availability for patient appointments.
- Verify that the doctor's availability is updated, and patients can see the new slots when scheduling appointments.

Test Case 13: Accept or Decline Appointment Requests

- Doctor accepts or declines an appointment request from a patient.
- Verify that the appointment status changes to **"confirmed"** when accepted or **“cancelled”** when declined , and the patient is able to see the updated status of the appointment.

Test Case 14: View Upcoming Appointments

- Doctor views all upcoming confirmed appointments.
- Verify that the system displays a list of all upcoming appointments with patient details and appointment times.

Test Case 15: Record Appointment Outcome

- Doctor records the outcome of a completed appointment.
- Verify that the appointment outcome is recorded, and relevant updates are visible to the patient under “View Past Appointment Outcome Records”.

3. Pharmacist Actions

Test Case 16: View Appointment Outcome Record

- Pharmacist views appointment outcome records to process prescriptions.
- Verify that the system displays the appointment outcome details, including prescribed medications.

Test Case 17: Update Prescription Status

- Pharmacist updates the status of a prescription to "dispensed."
- Verify that the prescription status is updated, and the change is reflected in the patient's records.

Test Case 18: View Medication Inventory

- Pharmacist views the current medication inventory.
- Verify that the system displays a list of medications, including stock levels.

Test Case 19: Submit Replenishment Request

- Pharmacist submits a replenishment request for low-stock medications.
- Verify that the replenishment request is submitted successfully, pending approval from the administrator.

4. Administrator Actions

Test Case 20: View and Manage Hospital Staff

- Administrators can view the list of hospital staff and add, update or remove staff members.
- Verify that the displayed list of staff is updated with any changes.

Test Case 21: View Appointments Details

- Administrator views all appointments.
- Verify that the system displays a list of appointments including details like Patient ID, Doctor ID, status, and date/time.

Test Case 22: View and Manage Medication Inventory

- Administrator updates the stock level of a medication.
- Verify that the medication's stock level is updated in the inventory.

Test Case 23: Approve Replenishment Requests

- Administrator approves a replenishment request from a pharmacist.
- Verify that the request status changes to **"approved,"** and the medication inventory is updated accordingly.

5. Login System and Password Management

Test Case 25: First-Time Login and Password Change

- User logs in with default password and changes it.
- Verify that the password change is successful, and the user can log in with the new password.

Test Case 26: Login with Incorrect Credentials

- User attempts to log in with an incorrect password.
- Verify that the system displays an error message indicating invalid credentials, and login is denied.

APPENDIX B:

Declaration of Original Work for CE/CZ2002 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 beawarded for the assessed work. In addition, disciplinary actions may be taken.

Name	Course (CE2002 or CZ2002)	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 toexplain your solution.
- Detailed UML Class Diagram.
  - Further Notes, if needed
- Testing.
  - Test Cases and Results
- Reflection.
  - The difficulties encountered and the way to conquer, theknowledge 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 wellpresent 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:

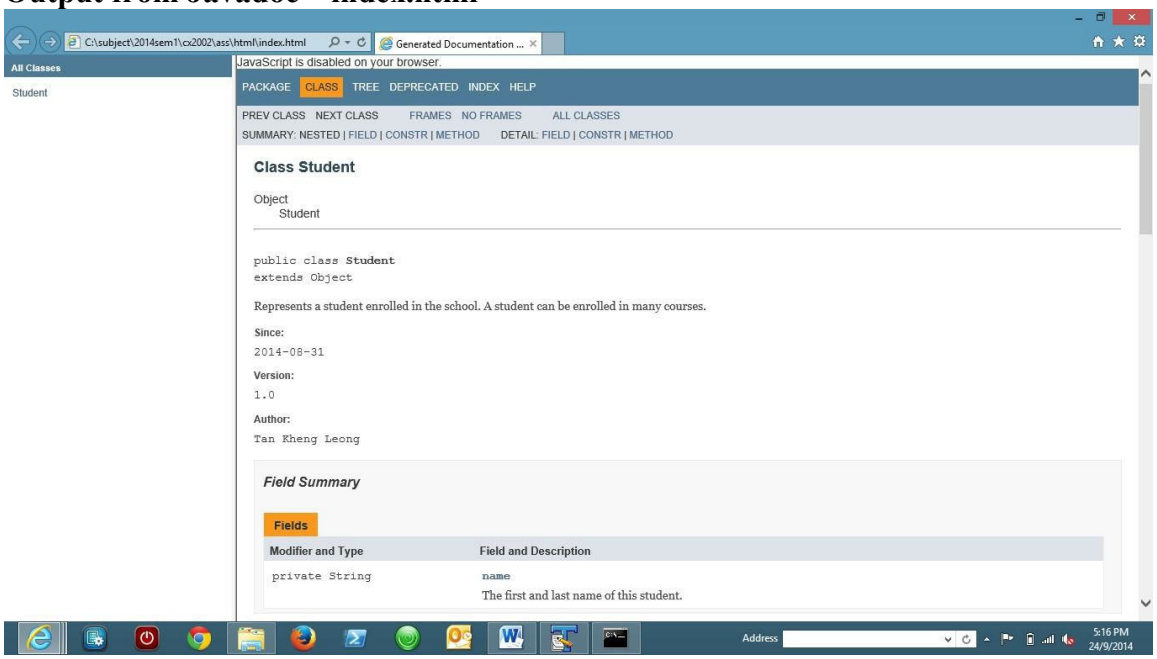
Detailed can be found at
http://www.oracle.com/technetwork/java/javase/documentation/index-137868.html

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