CPT106 C++ Programming and Software Engineering II

Group Project

Contribution to Overall Marks	30%
Release date	20 April 2023, 23:59
Submission Deadline	2 June 2023, 23:59

How the work should be submitted?

SOFT COPY ONLY !

1. All the files/codes (zipped into a single file) must be submitted from only one designated member (the group leader) of each group onto Learning Mall.
2. Students can organize a group with 4 students freely (must be 4 students, unless there are no more students left to join), and a group leader should be selected based on the discussion within the group.
3. The group leader should **organize group meeting** when necessary and submit report and source code files of the group project.
4. **A report** named with the **student ID number of the group leader** in **PDF format** should be submitted to Learning Mall with three sections as follows:
   a) A cover page with **a contribution table** with the breakdown of the roles and responsibilities of each group member during the duration of the project. The individual contribution should be based on your actual work and need to be discussed and agreed upon within the group with individual signatures in the table. Your contribution to the group will affect your final individual mark for the group project. The cover page looks like follows:

Group project name: xxxxxxxx

Group leader: xxxxxx

Group members:

Student ID	*Name*	*Responsibility*	*Contribution*	*Signature*
210001	*John*	*System analysis, coding*	*35%*	*student name*
210002	*Mike*	*Coding*	*25%*	*student name*
210003	*Marry*	*Report writing, testing*	*15%*	*student name*
210004	*Thomas*	*Coding, testing*	*25%*	*student name*

b) An report based on the Software Development Process, including the problem statement, analyses, design and testing.

c) A simple user manual, describing its basic functionality and how to use to its users.

5. **C++ source code files** of the whole project, the report and the user manual should be zipped into a **single zip or rar file**, named with **student ID number of the group leader** and submitted to Learning Mall.
6. Late Submission Policy: 5% of the total marks available for the assessment shall be deducted from the assessment mark for each working day after the submission date, up to a maximum of five working days.

Assessment Overview

This assessment requires the routine of code development using the software development process (SDP). The general marking scheme is shown as follows:

Documentation	***50%***
Specifications	**10%**
Overall spec.	2%
Customer spec.	3%
System spec.	5%
Analysis and Design	**20%**
Management of source code and lifecycle by software tools	**5%**
Testing	**10%**
Bugs report	**2%**
User manual	**3%**

Coding	***40%***
Reasonable OOP design pattern(weak coupling, strong cohesion)	**10%**
Working code	**10%**
User interface	5%
Data handling	5%
Clarity and readability	**10%**
Robustness	**10%**
Code Robust (e.g., checking invalid inputs)	5%
Complete functionality (against system specs)	5%

Module allocation and decoupling	***10%***

General Guidelines

Select one out of the 4 projects on page 4-7.

System design and coding

(1) The project descriptions are deliberately given in the form of simple *customer specifications*, which (as in the real world) are incomplete and often ambiguous, rather than a set of exact functional specifications. The group members should work methodically together (as the developers in a real-world software project) to:

- Analyse and formalize the **customer specifications on pages 4-7**(at this stage, the various design choices and the software features can be subject to the group's creativity). This should be different from customer supplication, which should clearly state the features of the delivered system.
- Apply objected-orientated design methodology **(encapsulation, inheritance and polymorphism, if needed)**, design and decompose the programmatic aspects of the problem using classes, and allocate constituent development tasks to each group member.
- Implement the product with frequent meetings to report progress and decisions to each other and re-evaluate the agreed courses of action.
- Implement test procedures, debug and correct the product.
- Finalize the deliverables.

(2) The given customer specifications on pages 4-7 are only basic, and most of the design choices should be decided in the group meetings. The systems described within the different projects have a variety of different features, and the disambiguation of the customer specifications can be based on the student's logic and real-life experience.

(3) Prioritization for the implementation of the different parts of the system is the group's decision; however, for the purposes of demonstration and testing, those parts that manage and populate the file stores with data such as customers' details, for instance, are necessary

(4) You should install the Github plugin component (third part component) in Visual Studio to share, manage and maintain your source code and documents with members in the group.

report

(1) The SDP report should clearly state all the features implemented and the hierarchical chart of your programme. It is better to have a few features fully working without run-time crashes than all features with many bugs without working properly or causing disrupting ripple-effects to other subsystems.

(2) In any case, any related decisions and compromises should be included in the report. The required testing procedures should simply take into account the input of each subsystem and what the output or operation of the software is supposed to do.

Assessment

Assessment will be based on whether the product offers reasonable functionality and features, its design quality, flexibility, software bugs, and other stated deliverables.

Project A: Taxi Company Management System

Overall description:

Your team is employed by a Taxi company for the management of taxies and drivers.

Customer specifications:

The system must be able to store the information of the vehicles and drivers of a Taxi Company. The administrator of this system should have higher authority to perform all the editing, while the driver user can only search for their own information and rewards and punishment records.

The system should be able to provide functionality as listed below:

- Browse, add, modify and delete the vehicle information and driver information;
- Make sure every vehicle is driving by one to two drivers;
- Register the payment information of every vehicle;
- Register rewards and punishments for every driver;
- Provides different authority to different type of users.

Project B: University Student Management System

Overall description:

Your team is employed by a university to implement a system for the management of students and courses.

Customer specifications:

The system must be able to store the information of the students and courses of a university. The administrator of this system should have higher authority to edit the course information and student information, but he/she cannot modify the student's selection of courses. On the other hand, the student user can search for their own information (personal information and past marks) and select new courses for next semester.

The system should be able to provide functionality as listed below:

- Browse, add, modify and delete the course information and student information;
- Search and display specific information as required;
- Allow user to select new courses which haven't been studied yet;
- While adding a new student information, make sure the student ID will not be repeated; similar rules apply to course name, course ID, etc.;
- Provides different authority to different type of users.

Project C: Car Parking System

Overall description:

Your team is employed by a parking lot for the management of parking vehicles.

Customer specifications:

The system must be able to store information of parking spots and customers. The parking lots should have multiple entry and exit points and multiple floors where customers can enter and exit and park their cars. Customers can collect a parking ticket from the entry points and can pay the parking fee at the exit points on their way out.

The system should be able to provide functionalities to the different users listed below:

- Allow administrator to browse, add, modify and delete spots and customer information;
- The system should not allow more vehicles than the maximum capacity of the parking lot. If the parking is full, the system should show a message to the customer;
- Each parking floor will have many parking spots, and the system should support multiple types of parking spots, such as compact, handicapped, motorcycle, etc.;
- Allow administrator and customer to search for all unoccupied parking spots for a specific type of vehicle, i.e., car, trunk, van, motorcycle etc.;
- Each parking floor should have a display board showing any free spot for each spot type;
- The system should support a per-hour parking fee model. For example, the first hour is free, and customers have to pay $3 for the remaining hours, maximally $50 per day.

Project D: Warehouse Management System

Overall description:

Your team is employed by a logistic company to implement a system for the management of warehouses.

Customer specifications:

The company has 2 large warehouses and 4 small warehouses. The large warehouse holds the space of 50 x 10 standard cargo containers, while the small warehouse holds the space of 40 x 10 small cargo containers. The warehouse keeper should be able to edit the warehouse information and customer (businesses which stores their goods in the warehouse) information, but he/she cannot modify the goods information. On the other hand, the customer user can search for their own stored goods, fees to pay and check in/out goods.

The system should be able to provide functionality as listed below:

- Browse, add, modify and delete the warehouse information, customer information and goods information;
- Search and display specific information as required;
- If possible, do not split a batch of goods into different warehouses;
- Calculate the storage fee according to the space the goods taken and the duration it stored;
- Provides different authority to different type of users.