

CPT106 C++ Programming and Software Engineering II

Group Project

Contribution to Overall Marks	35%
Release date	Wednesday, 5-May-2021, 23:59
Submission Deadline	Sunday, 30-May-2021, 23:59

How the work should be submitted?

SOFT COPY ONLY !

1. All of 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, 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 form of **name, student ID and signature of each group member** as well as **a contribution form** with the breakdown of the roles and responsibilities of each group member during the duration of the project.
 - b) A SDP report including the necessary analyses, design, test and debug.
 - 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 should be zipped into a single file, named with **student ID number of the group leader** and submitted to Learning Mall.
6. **The report and zipped source code file** should **be submitted separately** to two different submission boxes on Learning Mall.
7. 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.	<u>2%</u>
Customer spec.	<u>3%</u>
System spec.	<u>5%</u>
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	<u>5%</u>
Data handling	<u>5%</u>
Clarity and readability	10%
Robustness	10%
Code Robust (e.g., checking invalid inputs)	<u>5%</u>
Complete functionality (against system specs)	<u>5%</u>
Module allocation and decoupling	10%

General Guidelines

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

- (1) The project descriptions are deliberately given in a 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 formalise the **customer specifications in page 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, 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.
 - Finalise the deliverables.
- (2) The given customer specifications in page 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 to the student's logic and real life experience.
- (3) Assessment will be based on whether the product offers reasonable functionality and features (for the group size and allocated time), its design quality, flexibility, software bugs, and other stated deliverables.
- (4) 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.
- (5) Prioritisation 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.
- (6) 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.
- (7) You're advised to use GitHub to manage and maintain your codes.

Project A: Restaurant Management SystemOverall description:

Your team is employed by a restaurant to implement a system for the management of restaurant.

Customer specifications:

The system must be able to store the information of the raw material, dishes and customers. It can calculate the total fee for a customer according to the dishes he/she ordered. Moreover, the system can also calculate the gross profit based on the dishes prices and raw material prices.

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

- Browse, add, modify and delete the raw material information, menu information and customer information;
 - Manager has the highest authority limit to perform all the task;
 - Chef user can search and browse the raw material information and edit the menus;
 - While adding a new dish to the menu, make sure the dish ID and name will not be repeated; when customer is ordering dishes, make sure that's not repeated either;
 - Customer user can browse the menu, order dishes and check-out;
 - Provides different authority to different type of users.
-

Project B: Taxi Company Management System**Overall description:**

Your team is employed by a Taxi company for the management of taxis 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 C: Airways Management System

Overall description:

Your team is employed by an Airways company for the implementation of a computer system responsible for a large part of the operation of the company.

Customer specifications:

The system must be able to store the information of planes, flights and passengers of the company during the years of its operation. There are two types of planes P62 & P124 with rectangular arrangements of 6×2 & 12×4 seats, respectively. The sources and destinations of the currently available flights of the company (for simplicity, assume only direct flights) are allocated from a set of city airports which can be potentially extended. The different passengers can be allocated to specific flights and seats.

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

- An administrator who can include new flights, prices, dates, airports and perhaps new planes for each of the two available types.
- A ticket agent who can flexibly search for a specific flight inquired by a customer. When the customer reserves or books a ticket, then the required details must be stored. Such information includes flight id, payment details, expiration date of reservation, route, allocated seat, viewing facilities of the seating plan, etc. Facilities to amend this information must be provided.
- A manager who can retrieve statistics about the company's operation, such as the number of planes for each type, total passengers per flight, total revenue, etc.
- A customer can only book one ticket for himself/herself during the same duration.
- The ticket agent can calculate how many tickets he/she sold during a particular duration.

Project D: Bus Management System**Overall description:**

Your team is employed by a bus company for the implementation of a computer system responsible for a large part of the operation of the company.

Customer specifications:

The system must be able to store the information of buses, drivers and passengers. The sources and destinations of the currently available buses of the company (for simplicity, assume only direct routes) are allocated from a set of bus stations in a city which can be potentially extended.

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

- Allow administrator to browse, add, modify and delete the bus information and driver information;
- Make sure every bus is driven by one or two drivers;
- Allow passenger to search information of buses for a specified source and/or destination and display related information;
- Allow passenger to book seat for a specific bus, but the system should not allow more passengers than the maximum capacity of the bus;
- Register records of traffic violations for drivers, one with more than 3 records should be forbidden from driving.