EEE102 C++ Programming and Software Engineering II

Group Project

Contribution to Overall Marks	25%
Submission Deadline	Monday, 29-May-2017, 23:55

How the work should be submitted?

SOFT COPY ONLY !

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 ICE.

What should be submitted?

• Complete program source code (a set of .cpp and .h files).

• An SDP report including the necessary analyses, design, test and debug.

• A simple user manual, describing its basic functionality and how to use to its user.

• The contribution form should be filled and signed by every members of the group. Breakdown of the roles and responsibilities of each group member during the duration of the project.

The report in Microsoft Word format **(.DOCX file)** or pdf format **(.pdf file)** and **C++ source code (with comments)**, for all questions should also be zipped into *a single file*. (It is a good practice to include comments in your code stating the aim of the program, what are the inputs, what are the outputs, which algorithm is used, who is the author and so on.)

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	40%
Specifications	10%
Overall spec.	2%
Customer spec.	3%
System spec.	5%
Analysis and Design	15%
Testing	10%
Bugs report	2%
User manual	3%

Coding	60%
Reasonable OOP design pattern	20%
Working code	20%
User interface	5%
Data handling	15%
Clarity and readability	10%
Robustness	10%
Code Robust	5%
Complete functionality (against system specs)	5%

General Guidelines

Select one out of the 5 projects.

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 **system specifications** (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.

The given specifications 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.

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.

The group does not have to implement all the customer requirements if it is not possible; however should aim to deliver at least three features mentioned in the customer specification. The system specification 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.

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.

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.

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: 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.

Project D: Restaurant Management System

Overall 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 E: 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.