University of Westminster
Department of Computer Science

7SENG004C Coursework 2 (2021/22)

Module leader	Mr. Saman Hettiarachchi
Unit	Coursework 2
Weighting:	50%
Qualifying mark	**35**%
Description	Object-Oriented Modelling and Implementation
Learning Outcomes Covered in this Assignment:	**LO1** be able to use object-oriented techniques to analyse the high-level architecture of software systems, and to document this architecture using suitable graphical notations. **LO2** demonstrate a critical understanding of the syntax, meaning and use of the range of notations provided by a typical object-oriented design method. **LO3** work as a team to produce, implement and critically evaluate designs of an application using a typical object-oriented language.
Handed Out:	13th December 2021
Due Date	24th January 2022 1.00 pm
Expected deliverables	Electronic document and presentation
Method of Submission:	online via Blackboard
Type of Feedback and Due Date:	written feedback and marks 10 working days (2 weeks) after the submission deadline. **All marks will remain provisional until formally agreed by an Assessment Board.**

Assessment regulations

Refer to section 4 of the "How you study" guide for undergraduate students for a clarification of how you are assessed, penalties and late submissions, what constitutes plagiarism etc.

Penalty for Late Submission

If you submit your coursework late but within 24 hours or one working day of the specified deadline, 10 marks will be deducted from the final mark, as a penalty for late submission, except for work which obtains a mark in the range 40 – 49%, in which case the mark will be capped at the pass mark (40%). If you submit your coursework more than 24 hours or more than one working day after the specified deadline you will be given a mark of zero for the work in question unless a claim of Mitigating Circumstances has been submitted and accepted as valid.

It is recognised that on occasion, illness or a personal crisis can mean that you fail to submit a piece of work on time. In such cases you must inform the Campus Office in writing on a mitigating circumstances form, giving the reason for your late or non-submission. You must provide relevant documentary evidence with the form. This information will be reported to the relevant Assessment Board that will decide whether the mark of zero shall stand. For more detailed information regarding University Assessment Regulations, please refer to the following website:http://www.westminster.ac.uk/study/current-students/resources/academic-regulations

Coursework Description

Introduction

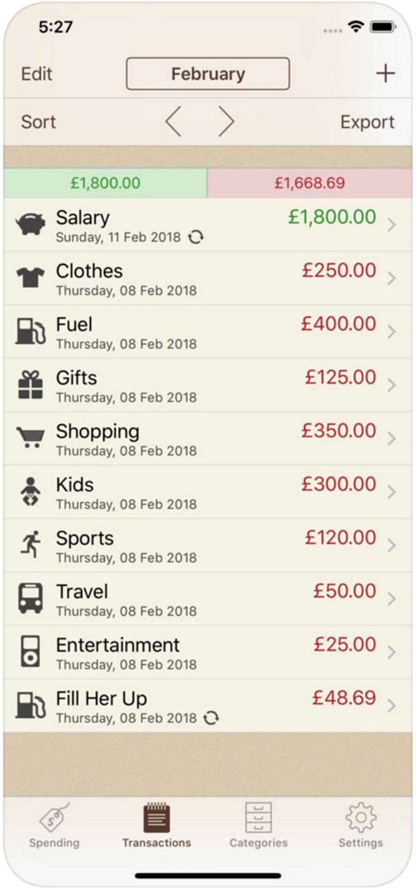
This is a group coursework. The aim of this coursework is

1. To give you experience of designing and developing an object-oriented application using UML
2. Give you an opportunity to use and demonstrate your knowledge of UML diagrams and syntax
3. To give you experience of working in a development team and of using C++ to implement an application

For this coursework, you will work in a group of 2 or 3 to develop an expense tracker and budgeting application. The purpose of the application is to help a user track what they spend their money on each month, and to compare their spending with a budget.

There are many such apps available – on the web, on smartphones or as standalone applications. One such can be found at https://itunes.apple.com/gb/app/spending-tracker/id548615579?mt=8. An example screenshot is given below.

In the example screenshot below of an iOS expense tracking application, expenses are given in red and are allocated to categories (such as Shopping or Kids), while income items are in green. Some items may be recurring.



£1,800.00	£1,668.69

Salary	Sunday, 11 Feb 2018	£1,800.00
Clothes	Thursday, 08 Feb 2018	£250.00
Fuel	Thursday, 08 Feb 2018	£400.00
Gifts	Thursday, 08 Feb 2018	£125.00
Shopping	Thursday, 08 Feb 2018	£350.00
Kids	Thursday, 08 Feb 2018	£300.00
Sports	Thursday, 08 Feb 2018	£120.00
Travel	Thursday, 08 Feb 2018	£50.00
Entertainment	Thursday, 08 Feb 2018	£25.00
Fill Her Up	Thursday, 08 Feb 2018	£48.69

The expense tracking and budgeting application designed and developed by your team should do the following:

1. Allow a user to see a list of recent transactions
2. Allow a user to enter a new transaction. At a minimum, this should include amount, whether it is an income or expense transaction and which category it falls under. A more advanced application will allow the user to enter a note and to specify whether it is recurring)
3. Allow the user to edit/delete transactions
4. Allow the user to see a list of categories. The application should come with some preset categories. An advanced application will allow the user to add new categories.
5. Allow the user to enter a budget, specifying amounts for each category.
6. Allow the user to track their progress against their budget by seeing how much they have spent in each category against the budget for that category, as well as overall spending against overall budget.

The user interface and menu should be console-based and use a simple character-based menu system (this is to simplify the code so that Visual C++ GUI programming is not needed).

Important note: you do not need to use a database for this coursework. Although, ordinarily, you would use a database or some other persistent store to keep data, this coursework only requires you to simulate use of a database – either through use of a dummy class, or flat files etc.

Working as a team

Although all members of the team are expected to work together on all aspects of the prpject (see below for deliverables), each team member should take one of the following key roles. Responsibility for a key role means responsibility for managing and co-ordinating effort by other team members on that task.

1. **Project manager – responsible for setting team objectives and milestones, overseeing and monitoring progress against objectives and milestones, overseeing team working and team meetings, resolving team problems etc.**
2. **Designer– responsible for detailed design (in the form of class diagrams etc., application of design patterns and heuristics**
3. **Developer – development of prototype from detailed design**

Note that a team member may take on more than one role, but not more than two.

The project manager is responsible for notifying the module leader of the composition of the group and any changes etc.

Deliverables

1. Group report. The group report should provide a detailed description of the project and the application. The structure of the report should include the following elements:
    a. design - which can include class and object diagrams
    b. implementation – which should encompass discussion of code and data structures, plus any reflection on changes made to design at implementation stage
    c. testing

    Each section should be written by the team member responsible for that phase of the project. In addition, each team member should write an section describing their role within the team, and how they contributed to the design and development of the application. (40% of coursework mark).

    The group report should also include a code listing as an appendix.

2. Team blog. Blog entries (using the Blackboard blog feature) should be used to enter minutes of team meetings. Minutes should specify clearly who attended each meeting, the detailed agenda, discussion of agenda items, actions arising from discussion of agenda items, who was delegated each action. Moreover, each meeting should cover progress against actions from previous meetings under the agenda item “Minutes of previous meeting”. (20% of coursework mark). There is an expectation that the blog should have at least 1 entry describing a meeting every 1-2 weeks during the semester. If a team member is unable to make a meeting, then they should send apologies via the project manager, and these should be recorded in the minutes. Entries should be faithful recordings of meetings and attendances. Meetings should normally be held on-site. There should be no online meetings.
3. Group presentation of the design and development of the application, including

demonstration. (40% of coursework mark).

Marking

Marks will be allocated to each component as follows:

1. Group report – 40%
2. Team blog – 20%
3. Group presentation – 40%

Individual team member marks will be calculated according to the following algorithm:

1. An overall mark will be calculated for the component. For the group report and presentation, this will be derived from marks for each section of the component.
2. The contribution of each team member to a component will be ranked on a scale from 0 (no contribution) to 5 (full contribution). For example, attending less than 50% of team meetings will result in a score 2 or less for the team blog, while not attending the presentation and not producing slides will result in a score of 0 for the presentation. On the other hand, producing good slides and presenting them will normally result in a rank of 4-5 for the presentation.
3. Individual team member marks for a component will be calculated as (0.5 * overall mark) + ((rank/5) * overall mark)