Advanced Topics in Software Engineering
Detailed information about assessment 3 (40%)

Given Date: 12 May 2022
Submission Date: 8 June 2022 23:59
Submission should be done via Turnitin (pdf report submission)

The structure of the work

In the first assessment, you have conducted a research regarding a trending area in the Software Engineering field. In the second assessment, you were asked to use agile development and management methods to develop a software product/service using Scrum,Kanban or ScrumBan. In this last assessment, you will be asked to use a plan-driven development rather than agile, and provide solid documentation and planing regarding UML, and ground your work rto practice how you can approach to a plan driven project. The details regarding this are explained below.

This assessment is given to you on **12 May 2022** and you will need to submit your work by week 16 which means your submission deadline is **8 June 2022 23:59**. The submission should be done via TurnitIn on blackboard page of the course.

While the asessment submission only includes an **individual upload** for your project reported, you will be expected to attend to a scheduled demo with your peers regarding the software prototype you developed. On 9 June 2022, during lecture hours, you will be scheduled for this demo pitch – which means you and your team will show the developed software and the evidence regarding how the **plan driven approach** was applied by you and your team. **Everyone involved in the team should upload the same report**, **but you might be judged based on your knowledge and capability to answers questions during the demo**.

Remember that this is a team project (not a group project). In a team, people have different and these are usually precise responsibilities. While the workload can be divided by you, you as a team must be the sole owner of your project. The implementation of your software (i.e. the code) needs to be optimized, plagiarism-free, efficient and should not include any direct copy/paste from the Internet. Each team must deliver their own version of the project.

What is the purpose of this project?

The purpose of this project is to create the basis for the development of an **Online Theatre Ticket System**. The project is aimed to create requirements specification (i.e., a structured functional requirement list), system modelling and prototype for an online Theatre Ticket System which would be accessed from a mobile application and/or a web site you are responsible to build. Every team would work on this topic and you will need to wear the role of developers and form up a team (not a group) to select a plan driven software development model and investigate requirement analysis and system modelling for this project. Having done so, you are expected to implement a prototype of your plan-driven approach. While you are not expected to develop a fully functioning software product/service, you will be responsible to create design and documentation and an implementation prototype – that would set the ground for this project to be implemented in the future.

What is the problem we need to solve?

A new movie theatre (aka Cinema) company recently built a cinema with three different halls in Ealing (ironically, Ealing does not have a cinema!). They have 3 different theatres in the building, each one with a different size. Since this company is new in the business, they want to build a **web site** and a **mobile application** simultaneously which would allow their customers to buy tickets online and book their seats in a specific theatre hall either from the web site or from the mobile application and even add snacks to their order so this can be ready when they are at the movie theatre. The CEO of this company hired you and your team as software engineers to work in this project. Your responsibility is to create a detailed requirement analysis and system modelling for the planned **Online Theatre Ticket System** building the website and/or mobile application.

The scope of this project is to provide a robust, reliable, and user-friendly system to sell online tickets with an algorithm that would customize the most efficient usage of the seating in theatres. The visitors (also called guests) of the web site shall be able to view the list of the performances (in this case movie sessions) with dates and ticket prices, in addition to the seating plan of different theatre halls. While any user can view the performances, seating plan and other info such as process; only the users of the system shall be able to buy tickets. This means that those users who want to buy tickets online shall need to register to the website/mobile application first. Once this procedure is done, users shall be able to buy tickets and reserve the seats for their preferred performances.

As you know, in the UK, some people go to cinema on their own (individually) – which can be a very engaging experience. The CEO is aware of this, and they want all users to have (partial) freedom in selecting the single seats, in order avoid the single seats scattered all over the place. In other words, if single seats are scattered all over the rows, this will create circumstances that couples or group of peoples would not be able to sit side by side – and this not a desired outcome by the owner of the company.

For the reasons discussed above, you will need to design an algorithm for this at a conceptual level while modelling the system. This conceptual model shall be done via a flowchart, UML or a Data Flow diagram which would be part of your design documentation. The sitting plan for each theatre hall is different and these are provided below. Having completed the conceptual model, this will be applied in the implementation.

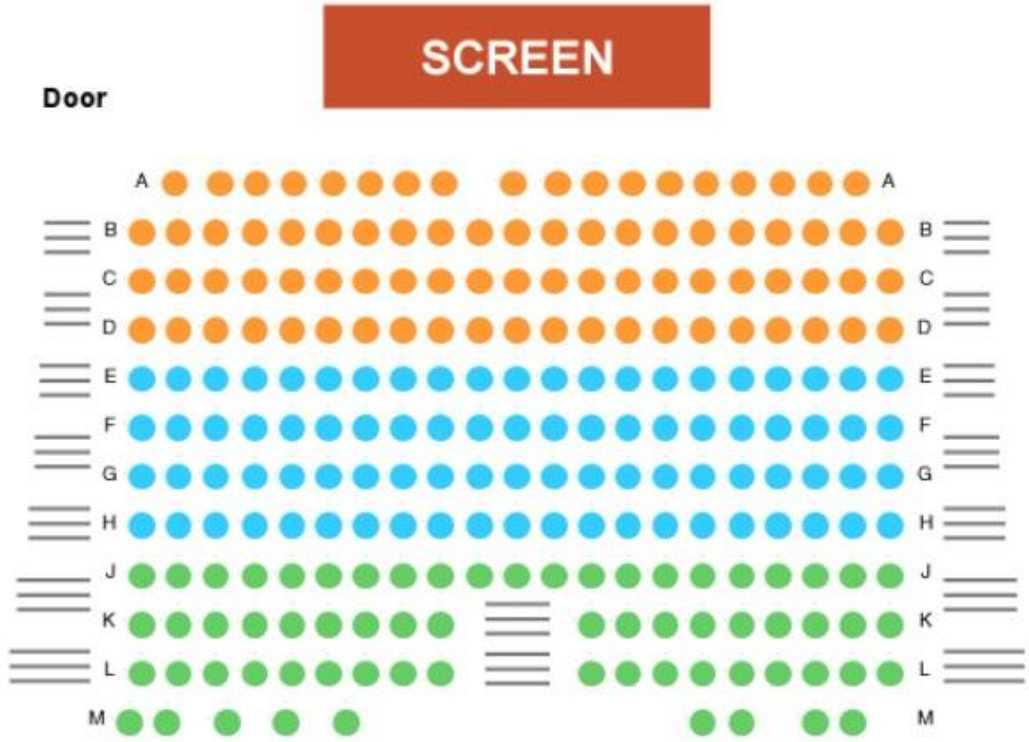
Once a performance (i.e., a movie) is selected, the system will show in which halls the movie is being showed (a movie can be showed in more than one theatre). The guest will be able to see the seating plan, the occupied seats, and even select a seat/multiple seats. However, in order to book the seats, the guest should be registered as a member. The registered members shall be able book their seats and they can cancel these as well. A cancellation procedure would release the booked seats so that other guests/members can book them. According to the policy of the movie theatre, a ticket cannot be cancelled when 24 hours or less left to the movie session. If the user attempts to cancel this, this should not be allowed as it is part of the agreement and the non-functional requirements.

The system shall be designed to meet the staff and management needs as well. The staff should be able to book for those customers who want to make a booking by phone or visit the counter. Finally, the management should add/delete and modify new and existing performances.

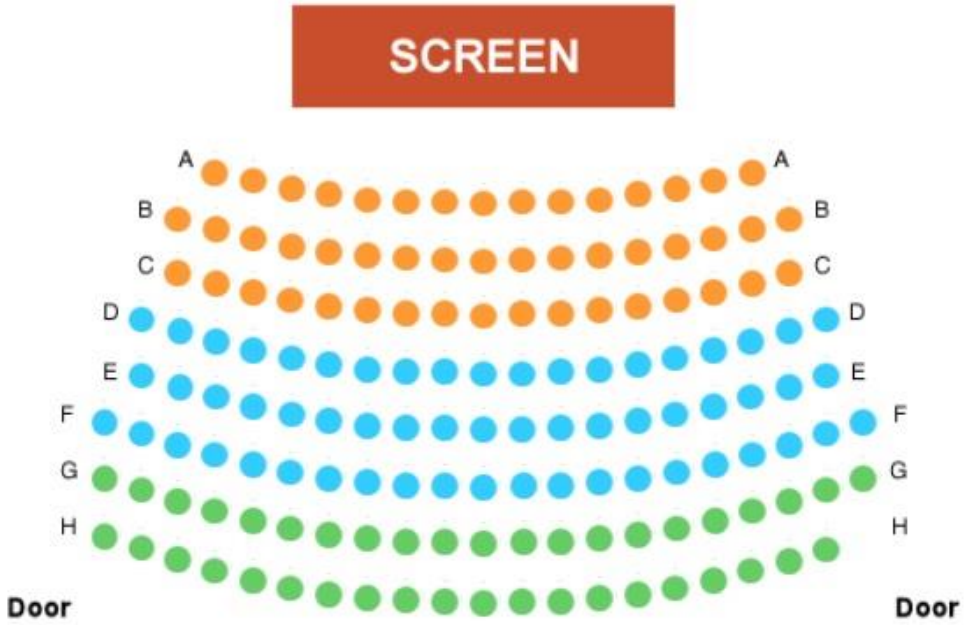
On the next page, you can view the seating plan of the theatre halls:

The Seating Plans for the 3 theatres:

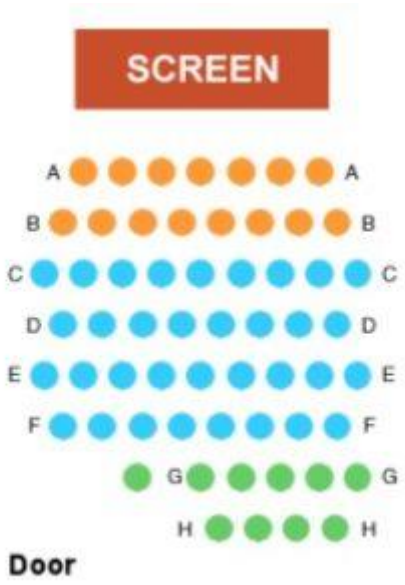
Theatre 1



Theatre 2:



Theatre 3:



How do we structure the work?

Similar to the previous assessments, this is a team project =. This means that you will need to form up different teams which would be consisted of 3 or 4 members – in rare cases this can be 5. Each member would have different responsibilities and these will be divided among the team members. As a team, it is your duty to divide the responsibilities among your team members. Each team would have a team leader and a secretary who would be responsible of the entire team to ensure that the tasks are carried out. While the team leader coordinates the activities; the secretary would take the *minutes of meetings* which should be submitted at the end of your report.

As a team, you would need to generate a report that would satisfy the scope of the project at a considerable level. In other words, you will be responsible to make a **requirement analysis** and **system modelling for the project**, and an implementation for this.

Please be aware that you cannot do this project individually as it must be the work of a team. Although you will need to work as part of a team, each team member should submit the same report via Turnitin.

What details the report should include?

In the Requirements Analysis section:

- At least 3 non-functional requirements.

- At least 10 functional requirements which should be connecting to non-functional requirements.

In the System modelling section:

- Level 0 DFD that should be levelled up to Level 1

- At least 1 Use Case Diagram – demonstrating how the system would work in the perception of visitors (guests), members, staff, and admin.

- At least 2 Activity Diagrams

- At least 4 Sequence Diagrams – demonstrating the behaviour of visitors, members, staff, and managers.

- Class Diagrams alongside with their relationship (no number limitation)

- An algorithm demonstrating a systematic modelling that would provide avoiding single scattered available seats in the theatre halls. As you are aware, most people do not come to theatres alone and it is imperative to prevent single scattered seats across the rows as this would prevent particular groups to book a number of seats at once. You will need to design an algorithm that would minimise the chance of single scattered seats in the theatre halls.

- you can use other UML diagrams if you want to do this, and you are encouraged to do so. The above list is the bare minimum you need to work on.

Implementation part

- Either a web page or a mobile application or both to demonstrate functionalities of the system – such as visitors booking a performance, selecting seats, being asked to login as a member and so forth.

- Implementation of your seating algorithm that presumably work in all three theatres.

Testing part

There should be testing procedures being applied to your application. A TDD approach is preferred here but it is not essential or the only approach you can use since big projects usually involve Q/As. A heuristic approach such as combination of TDD with smoke level testing or a full circle of TDD and BDD can be used as part of your testing procedures. Essentially, this is a section you will need to be creative, grounded. You will also need to defend the testing procedures in your demo/pitch.

What is the format of the report?

The assessment report should have the following titles:

1. Cover Page
2. Table of contents
3. Introduction
4. Work division – how the work is divided, who was responsible from what – secretary, and team leader and other roles should be mentioned.
5. Structure of the plan driven development (e.g. Waterfall. DSDM)
6. Requirements Engineering : Non-functional and functional requirements
7. System Modelling (in addition to UML, you can discuss software architecture in this section)
8. Implementation : algorithm developed, web page/mobile app or both
9. Testing procedures of algorithm andr web page/mobile
10. Conclusion/Final Words
11. References

How are you going to mark my/our work?

Please check the module information page for the grading criteria.

When and what do I need to submit? What should I pay attention when submitting?

You will only need to submit the report via Turnitin, your codes and implementation would not be submitted. However, this does not mean that your implementation is disregarded or not important. There will be a demo schedule, and you will need to demo your implementation similar to the second assessment. Although you are not expected to present a fully functioning software, you are expected to show an implementation of your approach, what kind of plan you have drawn, how did you achieve this, and how your algorithm works. The submission date and the demo date are all given above. There will be an announcement for demo times when you form your groups.

You do not need to prepare presentation slides; you only need to demo your work. You can however use your report to show the diagrams, and the planning process – or alternatively create a small power point presentation.

You might need to do research on Google scholar or simply on Google for reference purpose. Please do not shy away from citing the resources you have used and please reference these resources via using Harvard Style. If you do not know how to reference in Harvard style, please check the following link:

http://www.citethisforme.com/harvard-referencing

Bitter taste final notes…

As you might know from the school policy, there is **zero tolerance** to any **plagiarism** or **cheating**. **You must be the sole owner of your teamwork – both the implementation and the report**. If your work is taken from a resource, or if you attempt to steal someone else's work, this will be penalised heavily or not accepted at all. If you do your project parallel with other teams, both teams shall be penalized. You can use google to search for source code to find out how to perform specific tasks, but you are responsible to create your own solutions. Apologies if this came a bit harsh, but each team needs to submit original and their own work.

GOOD LUCK in your work