

Cinema Movie Booking System

GRoup Project

Group A | Software Engineering | 21.04.23

# Introduction

[5%] Please provide a brief text description of the system that your team worked on. In your description, please clearly explain what the system is, who the intended users are, and how it is intended to be used. Additionally, please state clearly which part of the system your team was responsible for implementing. (maximum of 200 words)

System is to be used by employees of the ‘Keele Movie House’ whilst serving customers. Employees will be able view a list of films currently showing at the cinema including their showing dates and times.

Employees will then be able to select a seat for the customer using an infographic and a drop-down menu. Finally, they will be able to take the customer’s details and print of a PDF ticket for them and provide them with the bank details to pay for their ticket. At this point the Employee (User) has the option to restart the application and take another order.

Our team will be implementing the Graphical User Interface for the application, the backend functionality that reads a CSV containing the movies currently showing at the Cinema, the data structures used for storing a customers ongoing order and then order history for the cinema and the creation of a PDF to act as a movie ticket for the customer.

Excluded from implementation is a payments screen instead the Keele Movie House will take cash or bank transfers only.

2. [10%] Your team was encouraged to find and use tools or techniques to help with Scrum team management and UML modeling during your teamwork and software development. Please provide the following:

a. List up to 5 key tools or techniques identified by your team that were used for Scrum team management and UML modeling during your teamwork and software development, and briefly describe their uses in team software development.

# Heading 3

To replace the placeholder text on this page, you can just select it all and then start typing. But don’t do that just yet!

First check out a few tips to help you quickly format your report. You might be amazed at how easy it is.

* Need a heading? On the Home tab, in the Styles gallery, just click the heading style you want.
* Notice other styles in that gallery as well, such as for a quote, a numbered list or a bulleted list like this one.

For best results when selecting text to copy or edit, don’t include space to the left or right of the characters in your selection.

b. Choose one of the tools or techniques identified in part (a) and discuss how it contributed to your team's activities in team-working and/or software engineering. Include a description of its advantages and disadvantages, and your recommendation for whether it should be used in future team-based software development.

# Heading 3

To replace the placeholder text on this page, you can just select it all and then start typing. But don’t do that just yet!

First check out a few tips to help you quickly format your report. You might be amazed at how easy it is.

* Need a heading? On the Home tab, in the Styles gallery, just click the heading style you want.
* Notice other styles in that gallery as well, such as for a quote, a numbered list or a bulleted list like this one.

For best results when selecting text to copy or edit, don’t include space to the left or right of the characters in your selection.

3. [10%] Regarding your team's use of Scrum, provide concise explanations (maximum of 50 words each) on how you implemented or opted not to implement the following Scrum Framework elements. Please specify the elements considered with respect to explicit Scrum documentation or guidelines and describe how you adapted the Scrum Framework approach to meet the needs of your team and product.

a. Sprints

b. Scrum (Meetings)

c. Reviews

# Heading 3

To replace the placeholder text on this page, you can just select it all and then start typing. But don’t do that just yet!

First check out a few tips to help you quickly format your report. You might be amazed at how easy it is.

## PART 2: Conceptual Design and Validity

You might like the photo on the cover page as much as we do, but if it’s not ideal for your report, it’s easy to replace it with your own.

Just delete the placeholder picture. Then, on the Insert tab, click Picture to select one from your files.

1. [5%]

a. Utilise your team's product backlog to articulate the primary requirements of the software system you developed.

**i. Ensure that the requirements cover the entire system and not just the specific part you created.**

ii. Summarise the system using 10 key requirements (you can have many sub-requirements of cause), with most of them focusing on functionality.

b. Use a suitable numbering convention to enable cross-referencing within the report.

c. Use objective language that clarifies how it can be demonstrated that the requirement has been satisfied.

2. [10%]

a. Produce a UML class diagram at a conceptual level that summerises the structure of the entire system you worked on. Ensure that the diagram is consistent with other diagrams presented in the report.

b. If a UML tool is used, indicate the name of the tool, and explain how the tool has led to a more specific diagram than what is required at the conceptual level. Alternatively, if no UML tool is used, provide a brief.

(maximum of 100 words) description of how your diagram aligns with

the UML notations and the conceptual design level.

c. Use the numbering convention established in Part 2, Question 1 to describe, using 10 brief bullets or a simple table, how your conceptual class diagram satisfies each of the 10 key requirements identified in Part2, Question 1.

i. If a requirement is not fully satisfied, make sure to clearly state it and provide a brief explanation.

ii. E.g., “the end of game requirement, R1, is a behavioural requirement not a structural requirement; it is anticipated that the behaviour will be implemented via classes X and Y, which would provide the relevant attributes and methods to meet this requirement”.)

3. [15%]

a. Select one system-level behavior (such as a transaction or user operation) that you have implemented (either partially or completely).

Use the numbering convention established in Part 2, Question 1, to indicate how your selected behavior is connected to the requirements identified in Part 2, Question 1, using 10 brief bullets or a simple table.

i. If a requirement is not related to the chosen behavior, state this clearly (no justification is required, as it should be obvious).

b. Use a UML behavioral diagram such as a sequence or collaboration diagram, specify the object interactions for the chosen behavior.

c. Create an updated version of the UML class diagram that includes additional attributes, methods, classes, and/or relationships required to maintain consistency with the behavioral diagram.

i. \*\*Do not include implementation details at this point, such astypes, method bodies, and parameter types, as doing so will result in a loss of marks.\*\*