**CS4125**

**Project Report**

**Food ordering system**

**Dean Kelly(19272898)**

**Jack Busher(19237499)**

**Quinn Painter(19234201)**

**Harneet Singh(18284728)**

**Table of Contents**

**Table of Contents 1**

**Business Scenario 2**

**Concept**

**Software Lifecycle Model**

Waterfall Model

V Model

Agile

**Project Plan**

Roles

Schedule

Experience

**Requirements**

**Use Cases**

Use Case diagrams

Use case descriptions

**Quality Attributes**

Usability

Efficiency

**GUI Prototypes**

**System Architecture**

MVC Diagram

**Analyst Design**

UML Diagram

Conceptual Class Diagram

Sequence Diagram

State Chart Diagram

**Business Scenario**

Our project is a food ordering system in which you do not have to interact with an employee for the majority of the time. This would drastically cut down on the time

customers spend ordering food. This then allows the restaurant to serve more orders per day, which results in more profit. This also reduces the number of employees

needed in the restaurant, as you can just focus on employees preparing the food, and the employees serving the food. We took inspiration from the menu screens at

McDonalds, which have been a huge success.

The program assigns each person with a unique identification number to diffrentiate between customers. This will allow us to apply various discounts and promotions to

each customer.

It is a screen which displays all the items available on the menu, such as food, drink, meals, deserts etc. There is lots of functionality contained within our project.

You will have the ability to choose the language you would like the menu to be in, you can add food to your basket, delete food from your basket. You can apply

discount codes while paying for your food. You can customise your order, such as removing onions from your burger, or adding extra bacon. The program will also have

a customer retention function, where it tracks how often you order from the store. Depending on how often you visit, you may be entitled to some prize such as a free

coffee, or a discount code etc.

There will also be log ins for an employee, and for a manager. The employee will be able to update the stock on the screen, and to update the price of an item. The

manager will be able to add items and delete items from the menu, and to apply promotions to a specific item on the menu.

**Concept**

This project entails the design and implementation of a contactless food ordering system that allows customers to order food and drink in a fast food chain without having to walk up to the till, tell the employee their order and then pay there. It allows them to order food directly from a screen in the store.

Customers will be able to avail of a plethora of features such as choosing which language to order in, customising your order, applying a voucher code. The screen will allow people to enter their log in details as to apply discounts. They may be able to redeem a free product or receive a free discount, based on how long its been since their last visit. This is a customer retention scheme which will encourage them to come back more often.

Customers are also able to receive certain discounts on special events or if there are special circumstances admins are able to apply a promotional discount.

Employees will be able to view order history and update stock.

Admins will be able to edit the menu and run promotions, as well as everything that an employee can do.

We were inspired to build a system like this because we all agreed that the machines that are used in the majority of McDonalds chains are user friendly and speed up the process of ordering your food.

By implementing a system like this we hope to be able to process more orders than a typical ordering system by the till. Due to the ability to take more orders we also think that having this system in place will increase the profitability of a company.

**Software Lifecycle Model**

We decided early on that we wanted to use an agile approach. This was after a discussion in which each team member gave their opinion on how we should go about executing this project. We still felt it was necessary to discuss each model in this report, as we wanted to give our justification to why we chose the agile approach.

**Waterfall Model**

This was discussed by our group and we considered using this model as our approach. We came to the conclusion that this wouldn’t be the best approach for us, as we believed that each of us would be working on different parts, and some parts may take longer than others. If we used this model we would be stuck at certain stages of the project. We also agreed that we weren’t going to get the entire project 100% correct on our first run. We were aware that there was going to be some changes to the project down the line.

Graphical user interface, application, website

Description automatically generated

**V-Model**

This was a strong contender for how we were going to approach the project. We all agreed that testing after each phase was a strong point of this approach. It would have allowed us to move through our project one phase at a time, with clear goals and a definitive way to verify our work and move on. It would have been a great way to track progress accurately, however we decided that the disadvantages outweighed the advantages. We believed that the risk of requirements changing later on in the project would negatively impact our ability to stick to this approach.

Text, chat or text message

Description automatically generated

**Agile**

We ultimately decided that proceeding using the Agile model would be the most beneficial for us. The agile approach would be the most realistic approach in our situation we thought. This was due to a few reasons. One of them was that we wanted to work in teams and we believed that the agile model would promote teamwork and enable it. As mentioned earlier, we believed that we would be making changes to our project along the way and agile seemed like the best way to enable these sort of unforeseen changes. The biggest reason why we chose agile was because we felt that none of us would be holding up the project from advancing any further if we were working on a specific item for a while. With the other approaches, there could have been bottlenecks while one person or a team tries to complete specific task, whereas with this approach we did not feel this problem would be as prominent.

We also discussed the disadvantages of this approach. We knew that the lack of an absolutely strict plan was a strength, it could also lead to being a weakness. We needed to hold each other accountable for what work needed to be done, and to not stray too far from the plan.

Diagram

Description automatically generated

**Project Plan**

**Roles**

|  |  |
| --- | --- |
| Project Manager | Jack |
| Document Manager | Dean |
| Business Analyst | Dean/Jack |
| Architect | Quinn/Jack |
| System Analyst | Group |
| Technical Leads | Quinn/Harneet |
| Programmers | Group |
| Tester | Jack/Harneet |
| Dev ops | Quinn |

**Schedule**

|  |  |  |
| --- | --- | --- |
| Task | Assigned | Due |
| Business description | Jack | W3 |
| Software lifecycle | Jack | W4 |
| Project Plan | Dean | W4 |
|  |  |  |
| Use case 1 | Jack | W4 |
| Use case 2 | Dean | W4 |
| Use case 3 | Jack | W5 |
| Quality attributes | Dean | W5 |
| GUI Prototypes | Quinn | W4 |
| MVC Diagram | Harneet | W5 |
| System architecture discussion | Harneet | W6 |
| Candidate objectives | Quinn | W6 |
| Conceptual class diagram | Quinn | W6 |
| Sequence diagram | Jack | W6 |
| State chart | Dean | W6 |
| ERD diagram | Quinn | W6 |
|  |  |  |

**Experience**

**Dean**

I have experience in python and flask. I was able to develop these skills throughout my placement at Ei Electronics.

**Jack**

My placement was at Regeneron. I was mainly working with old databases and converting them to Access databases

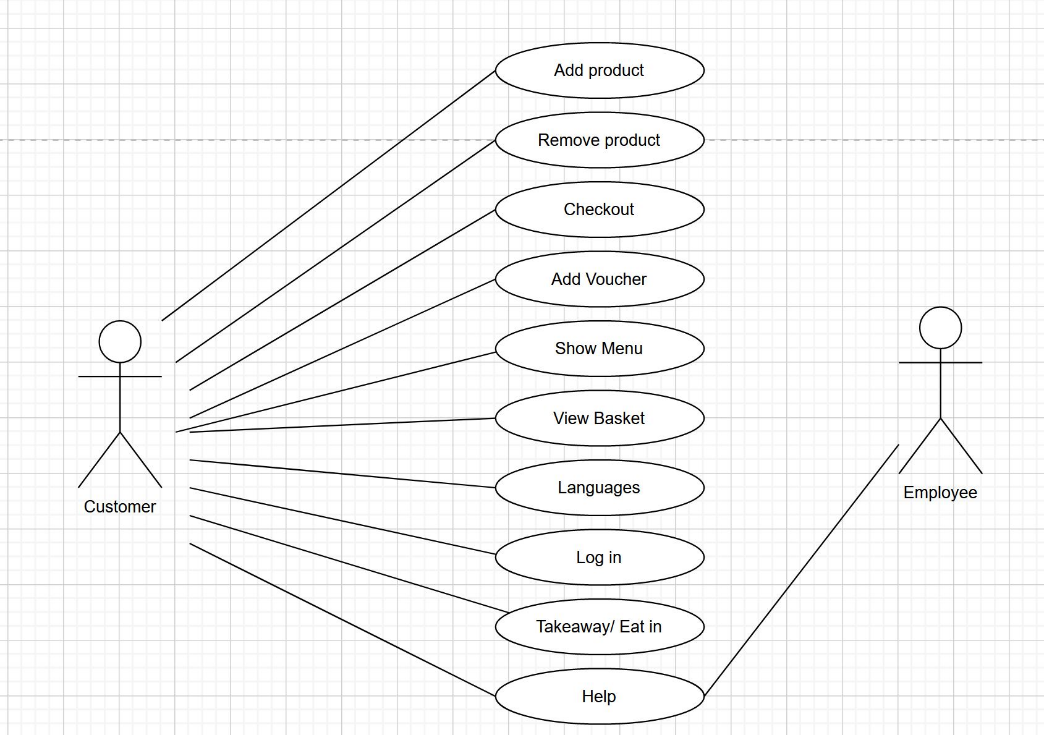
**Quinn**

My placement was at Jaguar Land Rover, doing Matlab and Python with QT.

**Harneet**

**Requirements**

**Use Cases**



|  |  |
| --- | --- |
| USE CASE 1 | ORDERING FOOD |
| Goal in context | The customer wants to view the menu, select the items they want, and pay for the items. |
| Scope & level |  |
| Preconditions | The customer has clicked the screen to activate it and the machine is working |
| Success end conditions | The customer is able to order the food they selected and successfully paid for the food. |
| Failed end conditions | The food isn’t added to the order correctly. |
| Primary, Secondary actors | Customer, employee, admin |
| Trigger |  |
| Description | 1. The customer activates the screen 2. The customer chooses the language 3. The customer chooses whether to enter their ID Number 4. they want to order in 5. Customer selects which items they would like to add to basket 6. Customer selects check out 7. Customer chooses whether to apply a discount code or not 8. The customer chooses take away or eat in 9. The customer pays |
| Extensions | The customer selects an item, but it is out of stock. |
| Variations | The customer chooses to customise their order |
| Related information | Help button |
| Priority | High |
| Performance | Seconds |
| Frequency | 200/day |
| Channel to actors | Not yet determined |
| Open issues |  |
| Due date | Week 8 |
| Superordinates |  |
| Subordinates | Customise order |
|  |  |

|  |  |
| --- | --- |
| USE CASE 2 | Apply a discount |
| Goal in context | Admin is applying a promotional discount/offer |
| Scope & level |  |
| Preconditions | Admin must use their employee login |
| Success end conditions | The system is updated and changes to the menu are applied |
| Failed end conditions | The promotion was never saved and the menu does not reflect the changes made |
| Primary, Secondary actors | Admin |
| Trigger | Admin clicks the apply discount button |
| Description | 1. Admin logs in 2. Clicks apply discount 3. Enters the type of discount (e.g 10% off, print voucher at end of purchase) 4. Enter time period the offer is valid for 5. Confirms the changes. 6. Log off |
| Extensions |  |
| Variations | 1. Discount percentage 2. Time period offer is valid for |
| Related information |  |
| Priority | High |
| Performance | Seconds |
| Frequency | 1 per week |
| Channel to actors | Not yet determined |
| Open issues |  |
| Due date | Week 7 |
| Superordinates |  |
| Subordinates |  |
|  |  |

**Quality Attributes**

**Usability**

This can be measured in terms of ease of use. Navigation should be simple. With a system such as ours, we wish to make it as straight forward as possible for our customers and staff to use.

The system must be:

* Easy to use for input preparation, operation, and interpretation of the output.
* Provide consistent user interface standards and conventions with our other frequently used systems.
* Easy for new or infrequent users to learn to use the system.

**Efficiency**

We wish to serve as may customers in a day as possible. We do this by delivering an easy to use ordering system which is as short a process as possible. We will provide only the necessary functionality to order, customise and pay. When taking this into

**GUI Prototypes**

**Text

Description automatically generated**

**A picture containing text

Description automatically generated**

**System Architecture**

**MVC Diagram**

**Analyst Design**

**Conceptual Class Diagram**

**Sequence Diagram**

Diagram, schematic

Description automatically generated

**Diagram

Description automatically generatedState Chart Diagram**