**Portfolio**

Introduction:

My previous acquired knowledge regarding the subject of full stack app building is with C# and windows forms, as well as my experience with PHP and CSS.

The learning style I prefer is demand based because it allows me to choose my own path while still having some sort of structure that provides me with a sort of check point system to make sure I’m going on the right path.

I took upon myself to choose C# for the reasons listed in one of my research papers comparing asp net to other available options.

This portfolio aims to show how I successfully achieved the required learning outcomes needed.

Project Description:

My project is simply based off of me wanting to build something that meets the requirements and quality standards of a commercial level software product, that is also why I have chosen to have a stake holder with an aspiring business in jewellery to be.

Learning Outcomes IP:

LO 01: Web-Application:

What did you do to complete the task?

My Web Application is built using the React javascript framework and it is linked to the C# Asp net rest API,the web application communicates with the controllers of the API to retrieve the information (GET functions) or send information needed using the POST functions.

The Web Application homepage:

Graphical user interface

Description automatically generated

The Web Applications Register Page:

Graphical user interface, text, application, email

Description automatically generated

The WebApplications Login page:

Graphical user interface, text, application

Description automatically generated

The WebApplications Products Page:

Graphical user interface, application

Description automatically generated

The web applications Product page:

Graphical user interface

Description automatically generated

Example of error handling:

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

The use of a mediator to minify the direct interaction between different objects.

Graphical user interface

Description automatically generated with medium confidence

I have ensured the usage of persistence of data (by usage of ORM) through the usage of the entity framework which is essentially an ORM framework. Below is an example of the entity framework composition.

Graphical user interface, application

Description automatically generated

I have also used Migrations to update the database schema while preserving existing data in the database:

Text, letter

Description automatically generated

The use of asynchronous functions is clear in the controllers as an example the products controller following functions:

Text

Description automatically generated

Also for the reference for any software developer that would want to work on my project in the further future, below are the uml class diagrams for every part of my project:

The Repository class diagram

A dbentityconfig class where the mapping inherits from it, the crud repo is a abstract class where all the repositories inherit from it.

Graphical user interface, diagram

Description automatically generated

The API class diagram:

* You have a base controller that handles the bus, mediator and notifications which are inherited by the other controllers, order controller specifically requires authentication so it inherits from the authenticated controller.

Graphical user interface, diagram, schematic

Description automatically generated

The Domain class:

* The Commands for the products,users and orders are handled by the command handler.
* The class is an abstract class that is inherited by the rest of the entity classes.
* The commands are extracted from an abstract class called message and are inherite by the specific commands for every part.
* Event is also inherited by the events that correspond to the commands

Graphical user interface

Description automatically generated

A picture containing diagram

Description automatically generated

Graphical user interface

Description automatically generated

What have you learned?

* I have learned how to Implement a C# API from scratch.
* I have learned how to create a mediator.
* I have learned how to make an SQL server interact with an API.
* I have learned how to use entity framework.
* I have learned how to create async functions for controllers
* I have learned how to communicate React javascript with a C# API
* I have learned the basics of React
* I have learned how to implement other APIs into my front end

LO 02: Software Quality:

What did you do to complete the task?

I integrated code analysis using code QL and SonarCLoud on github and used both the visual studio IDE code analysis and the webstorm code inspector to examine my code and check the pitfalls of my code and allow me to further improve its overall quality.

Below is an image from webstorm showing my project code analysis and the warnings based on each category:

Graphical user interface, application, Word

Description automatically generated

Representation of CodeQL and SonarCloud in github actions and the issues they pick up:

The sonar cloud dashboard that allows me to view the code quality, every time I push into GitHub sonar cloud analyses my code:

Graphical user interface, application

Description automatically generated

Issues detected by CodeQL and SonarCloud in github:

Graphical user interface, text, application, email

Description automatically generated

GitHub successful run and build through the actions of sonar cloud and code QL indicating continuous Monitoring and analysis after very push through GitHub

Graphical user interface, application

Description automatically generated

Created an issue for the Security issue and fixed by choosing more robust encryption method:

Before Showing SonarCloud and Codeql picking up on same issue with 7 security alerts:

Graphical user interface, text, application

Description automatically generated

After:

Graphical user interface, application

Description automatically generated

Background pattern

Description automatically generated with medium confidence

My code QL yml file link: <https://github.com/Hishamabboud/Ecom/blob/main/.github/workflows/codeql-analysis.yml>

My Sonar Cloud yml file: <https://github.com/Hishamabboud/Ecom/blob/main/.github/workflows/sonarcloud.yml>

**Testing:**

For testing I chose to go for integration testing with postman for the reasons that I have explained in my document “Why Integration testing”.

The following link takes you to the tests file of postman in git <https://github.com/Hishamabboud/Ecom/blob/main/Tests/apitesting.postman_collection.json>

The following image is the result of the postman tests run by github actions using postman.

Table

Description automatically generated

My tests involved the most important end points for the user side while using real life situations in the testing to get rid of any bugs that might occur from normal usage, tests such as create a product, create a user, login, change a product, change a user’s details where all successfully implemented into the postman test collection as can be seen in the actual collection file in Git.

The link to the reason of choosing the integration testing method: <https://github.com/Hishamabboud/Ecom/tree/main/Documents>.

LO 03: CI/CD:

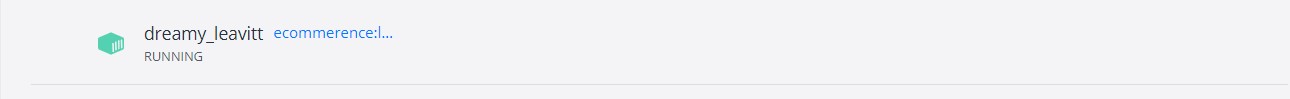
Docker:

As shown below in the containers tab in visual studio and after building the App through docker:

Graphical user interface, text, application, email

Description automatically generated

As shown below in the docker desktop app showing the running containers:



Below if proof of build through continuous integration through GITHUB CI and docker creating a CI/CD pipeline with the docker container, thus decreasing build time and optimize workflow.

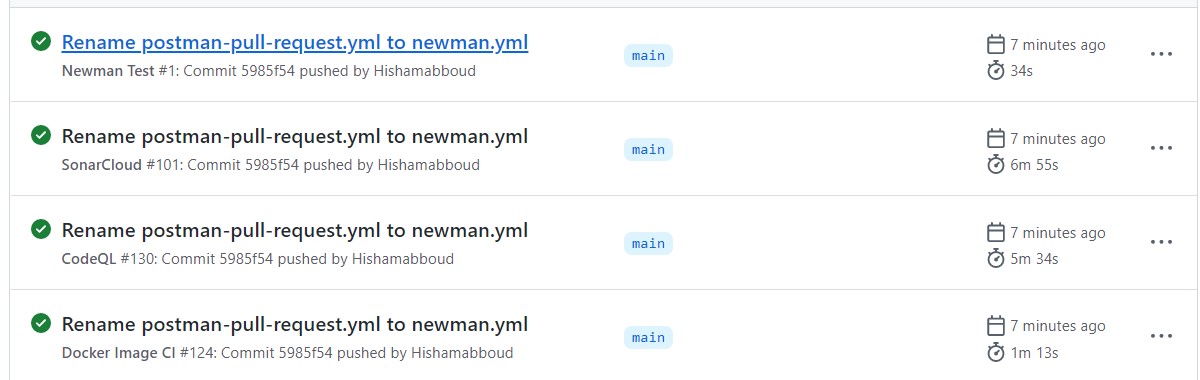
Text

Description automatically generated

Further proof while pushing changes to github and code analysis coupled with postman testing on push:

My DockerImage yml file: <https://github.com/Hishamabboud/Ecom/blob/main/.github/workflows/docker-image.yml>

The below Actions show continuous code analysis and the newman testing (postman) as I pushed into the repository as well as docker pushing and building:



The below image shows the jobs the newman action went through and they ran successfully when I pushed into the repo, continuously checking and testing the important parts of my project.

Shape

Description automatically generated with medium confidence

The workflow file of the newman test can be found here: <https://github.com/Hishamabboud/Ecom/actions/runs/2511861836/workflow>

I deploy to docker hub through github actions and the result can be seen in the below screenshot taken from github actions:

My workflow file for dockerhub building and pushing:

**https://github.com/Hishamabboud/Ecom/actions/runs/2524575491/workflow**

Text

Description automatically generated

LO 05: Professional:

What did you do to complete the task?

During the semester we set up ways in our group to schedule our tasks and organise them, the way I have seen it done in my own company where I work. These tasks where separated by a program called shortcut which allowed us to use agile sprints to create a short, timed schedule that provides the stakeholder with continuous updates and for group members to provide updates on their progress on each task.

Through this in my group project and personal project through my own stakeholder I have proved the professional learning object as can be seen with some of the shortcut tasks

Graphical user interface, text, application, chat or text message

Description automatically generated Graphical user interface, text, application, chat or text message

Description automatically generated Graphical user interface, text, application, chat or text message

Description automatically generated

I have also communicated with my colleagues by creating a whatsapp group that allows easier and more urgent contact between the different team members, replying to any help they might need and asking for help as well from them.

Graphical user interface, text, application, chat or text message

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated

I also used the concept of user stories in my own personal project, using the git issues tab:

Graphical user interface, text, application, email, Teams

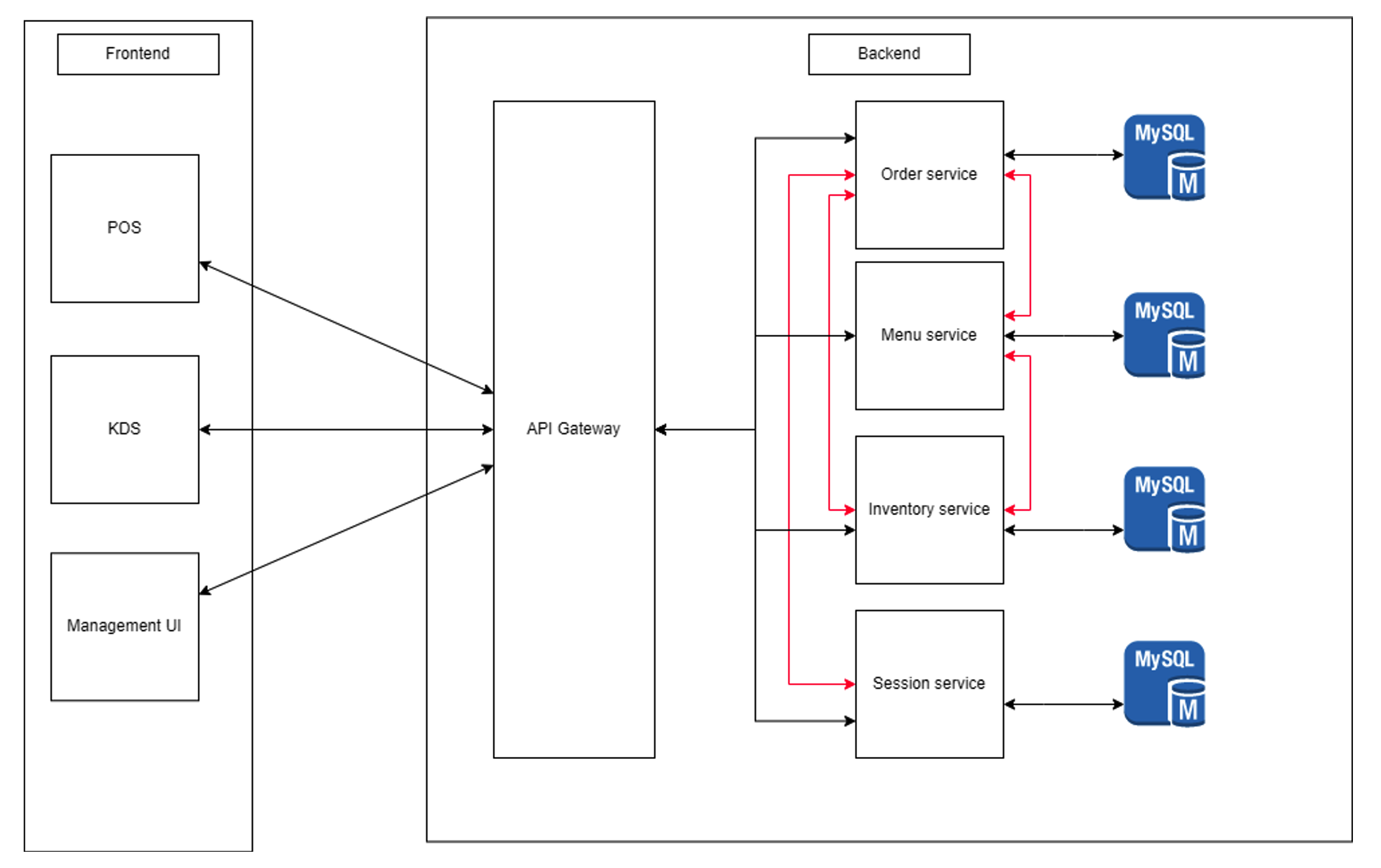
Description automatically generated

Learning Outcomes GP:

LO 01: Web-Application:

In the group project we built a full stack web application, with a react JavaScript framework and a C# Asp net back end. The back end handles the crud functionalities in a method that the REST API does meanwhile the front end handles the POS, KDS and IMS.

Architecture diagram:



In the group project we also made the frontend act in such a way the components in the actual user interface are in fact just react components.

We use Entity framework to create an ORM application backend, It is dockerized and deployed on a digital ocean hub.

LO 03: Agile Method:

Definition:

Agile is an approach to project management and specifically software development. It is pivotal in an easier and more efficient deliveries to the customer instead of a one delivery at the end of the deadline made for the project.

The point of agile is to deliver smaller consumable increments to the customer in batches every sprint.

What we did:

For this project we used shortcut as It includes the ability to use the stories, sprints, epics. We had a backlog as well that was closely followed and monitored.

We had five sprints, each of them where a few weeks, these sprints allowed us to continuously deliver to the customers while receiving feedback and making the right adjustments.

The sprints had a sprint review where the customers got to provide their feedback and the group got to reflect on what could have been done better while providing feedback to each other.

Agile also has a definition called scrum roles that are assigned every sprint at the start of it, and where exchanged between the group members every sprint.

On project school days we had a stand-up, where we discussed what will be done in the day and then a stand down where we discussed what we finished at the end of the day.

SCRUM:

**What is scrum?**

Scrum is a framework that helps teams and organizations that helps teams generate solutions in a more effective way and delivering them with the highest possible value.

It is used for project development, and through it one can plan sprints so that modifications and improvements can be added onto every next sprint.

It organises the group by scheduling stand ups that are essentially for planning the individual day and what tasks lay ahead and stand downs that allows the group members to declare the finished tasks and evaluate that overall productivity in the day.

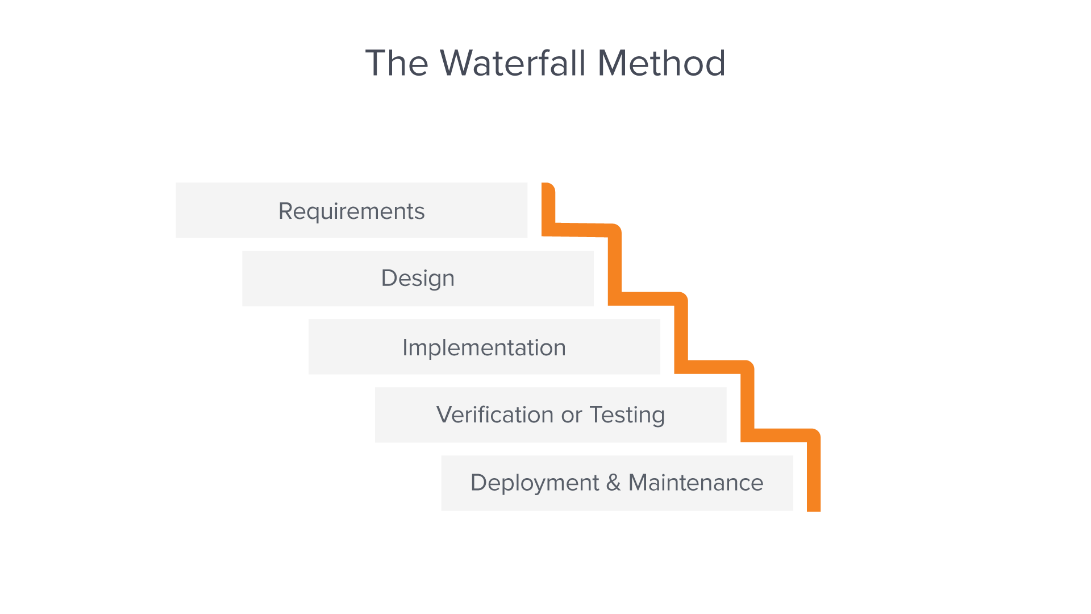
The sprints also allow the product owner to completely be sure that the product meets their expectations and follows their requirements since after every sprint feedback and an evaluation is given for the deliverables, it also makes it easier for the group as they know the demands and time frame that they are placed in thus removing any surprises.

WaterFall:

**What is Waterfall?**

As the name suggests it is the continuous and chronological process that works on fixed dates, requirements and outcomes, with this method communication between different integration teams is nullified, status reports also do not exist where team members work independently which is the opposite of the agile method.

With waterfall you have multiple stages where one does not finish before the other, below is a diagram depicting the stages:



Requirements:

Requirements, unlike agile are presented upfront and known from the start, it is done through a single written document and is used to describe the costs, assumptions, risks dependencies, success metrics and timelines for completion.

Design:

The data models, layouts and scenarios are all laid out in this stage of the process, firstly we create a high level design where the scope and purpose of the project is defined the this design has to describe also the traffic flow of each component and the integration points. Once that is done we can transform into a ready physical design.

Implementation:

The shortest process of waterfall is defined here in the implementation, here the programmers have to code the actual application and based on the requirements previously specified as well as some testing and implementation.

Verification or testing:

The one of the more important parts of waterfall is testing which constitutes the verification of an error free product, and that all requirements have been met by this application, the defects are assigned to a team that has to take care of them and release updates.

Deployment and maintenance:

The last phase of water fall is the process of deploying it to the market and customers, once any defects are found here the maintenance team comes into responsibility and has to take care of these errors and update the app.

LO 05: Cultural differences and ethics:

Introduction:

The main objective from this semester is software development and group management skills, however we had some other learning objectives as well and one of them is cultural differences and ethics.

This learning objective was met the second the group was started as we in the group had many different nationalities, being the most international group, we could implement that into our project and our method of work.

We also considered the working culture in the Netherlands vs our own work culture from back where we came from.

How we faced the cultural differences:

Due to our differences in culture, we were aware that for example we had to ask about if tipping was involved in the application or how important it was to make sure everything was scheduled during working hours such as the meetings,

Meanwhile scheduling in my culture is not all that important, which represents a cultural difference.

I come from an Arab background with some European culture as well as I studied in a British school and have been living in the Netherlands for 2.5 years, this taught me a lot about the importance of learning how to interact with different cultures and how to respect them and make them gel together.

Cultural differences cover everything from work, school, social and family culture. This makes complying with these differences in a respectful manner a difficult task but doable.

Communicating with the group mates about help regarding the project allowed me to interact with them on a personal level, which allowed me to be introduced into their culture especially with my Dutch groupmates after I tried to speak Dutch to them.

In Terms Of Ethics

My personal project was driven by the love to find solutions for problems, as my project is made for my girlfriend and while having a personal relationship with her I could have overpriced my product and went through shortcuts with my application side as she does not know the required standards on the software level.

The required ethics a software engineer must go through are those that restrict him from taking advantage of his knowledge superiority to create a lower standard product and charge more for it, or to take advantage of getting paid by the hour and having a bug-ridden application which take time to get fixed.

With the group project it was a bit different as they are driven by monetary reasons, which is natural as they are working for a business after all which is different than our purpose of software development in this moment of our career as it is mainly for studying.

Addictive design

The term “good UX design” is something every developer loves to hear, however the problem is that sometimes people love the UX too much.

Apps often use user manipulation, monetization methods, user data collection for cooperate use, the important thing here is that the users must know what they are doing and how they are being monitored (their data being monitored) which defines transparency.

For example, Duolingo users are definitely not aware of all the data usage when they get ads is not transparent by the company to the extent that it needs to be.

Another example is tiktok, where users especially the younger ones, I am quite sure are largely and happily oblivious to the methods and outcomes of their addictions.

Questionable personal data ownership

The AI based Software can profile users and predict behaviours to a dangerous extent, the biggest ethical dilemma for developers is the data one, where the business generates revenue and incentivizes developers and business managers and which forces them to go through the exploitation (ethically) of the data they receive and store.

This is regulated in the European union however in other countries, the US for example it is not taken seriously as we have seen with mark Zugerberg at the congress.

In my project the user has the ability to create data and then replace it or delete it thus complying with the ethics of data, above that it is abundantly clear how this data is being used and with the absence of ads that further proves that none of the data is exported to other companies or manipulated in a negative way.

Weak security

The security of an application is only tested after the end of the project, this raises security concerns as the developers are usually done with improving the product after submitting it to the customer.

The team is the main asset in this case that needs to be educated on the dangers of bad cyber security, as any hacker can cause real damage to people’s lives through any loophole they find in a software that might include bank information or any other personal messages or images.

**Hofstede's cultural dimensions theory:**

According to Wikipedia and their definition Hofstede's cultural dimensions theory is a framework for cross-cultural communication, developed by Geert Hofstede. It shows the effects of a society's culture on the values of its members, and how these values relate to behaviour, using a structure derived from factor analysis.

It is used to understand the differences between the different cultures across the world, it is one that measures power distance, the collectivism vs individualism, uncertainty avoidance index, femininity vs masculinity, short term vs long term orientation and restraint vs indulgence.

Power Distance Index

The power distance index is one which measures the tolerance in the inequality and power from the viewpoint of the followers also known as the lower level of society.

If the power index is high then it indicates that the common culture accepts inequity and power differences, encourages bureaucracy, and shows high respect for rank and authority.

If the power index is low then it indicates that it is a culture that encourages organizational structures that are flat and feature decentralized decision-making responsibility, participative style of management, and place emphasis on power distribution.

Masculinity vs. Femininity

The masculinity vs. femininity dimension is also referred to as “tough vs. tender,” and considers the preference of society for achievement, attitude towards sexuality equality, behaviour.

If the masculinity is high then the culture comes with a distinct gender roles, assertive, and concentrated on material achievements and wealth-building.

If the femininity is high then the culture comes with the following characteristics fluid gender roles, modest, nurturing, and concerned with the quality of life.

Long-Term Orientation vs. Short-Term Orientation

If the orientation is long term in the culture then it shows the ability to delay short term gains that could result in negative long term gains, then the culture would prefer and has the view to go to the long term positive gains, with persistence and persevance.

If the orientation is short term in the culture then it shows the focus on the short future which involve focusing on short term satisfaction rather than the future. Short-term orientation emphasizes quick results and respect for tradition

Indulgence vs. Restraint

This represents the tendency for the culture and society to fulfil their desires and impulses, and if they have any control over them or the opposite.

Indulgence indicates that a society allows relatively free gratification related to enjoying life and having fun.

Restraint indicates that a society suppresses gratification of needs and regulates it through social norms.

Source: <https://www.techtarget.com/searchsoftwarequality/tip/5-examples-of-ethical-issues-in-software-development>

Source: https://en.wikipedia.org/wiki/Hofstede%27s\_cultural\_dimensions\_theory#:~:text=Hofstede's%20cultural%20dimensions%20theory%20is,structure%20derived%20from%20factor%20analysis.

LO 06: Requirements and Design:

Introduction:

Requirements are a very important part of our project as we set out to meet them so that we can keep the product owners happy and satisfied with their investment, this is done by applying user acceptance studies and you evaluate the quality of the design you have made.

How We did it:

The main point of agile is having a satisfied product owner by contuinously asking for feedback when you delvier the product so that makes it easier for us as a group of deliver to meet their demands and conditions by using some methods of communication to be 100% sure of their needs.

These requirements and needs change every sprint according to the product owners’ feedback and comments, we asked the product owner about the sessions and we got this response

Graphical user interface, text, application, email

Description automatically generated

For the UX design we had some frameworks that we based our designs on for example this was the screenshot of the first concept we had of the dishes page:

Graphical user interface, application

Description automatically generated

And the overview page:

Table

Description automatically generated

These frameworks represented a solid base to work off of with the approval of the product owner and some feedback regarding the design that we implemented later on in the project and which also proved our reaction to the user response and experience.

LO 07: Business processes:

Introduction:

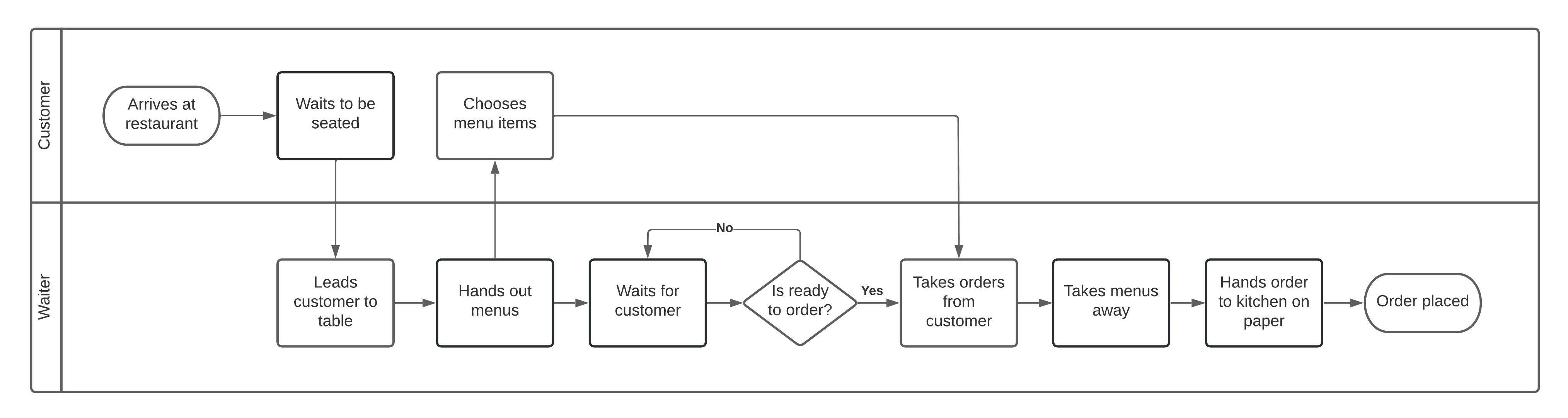
The essentials of the business process were not monetary as the project did not involve any payments or any kind of monetary transactions however, they were more orientated to the solution that the product must provide.

The business process defines our goal in this project as how we are setting out to automate something or improve on something using our software skills.

What did we do ?

Our goal was to automate the process from ordering food to your table to paying for it on your way out without any issues, this was specified by the product owner as they wanted us to revamp their already existing out dated system.

We created a diagram of the flow of the interactions between the waiter and the customer:



Our system sets out to create an independence and to reduce reliance on the employees of the restaurant, were waiting times are reduced and the waiters can focus on the human needed tasks instead of getting bother by every small detail.

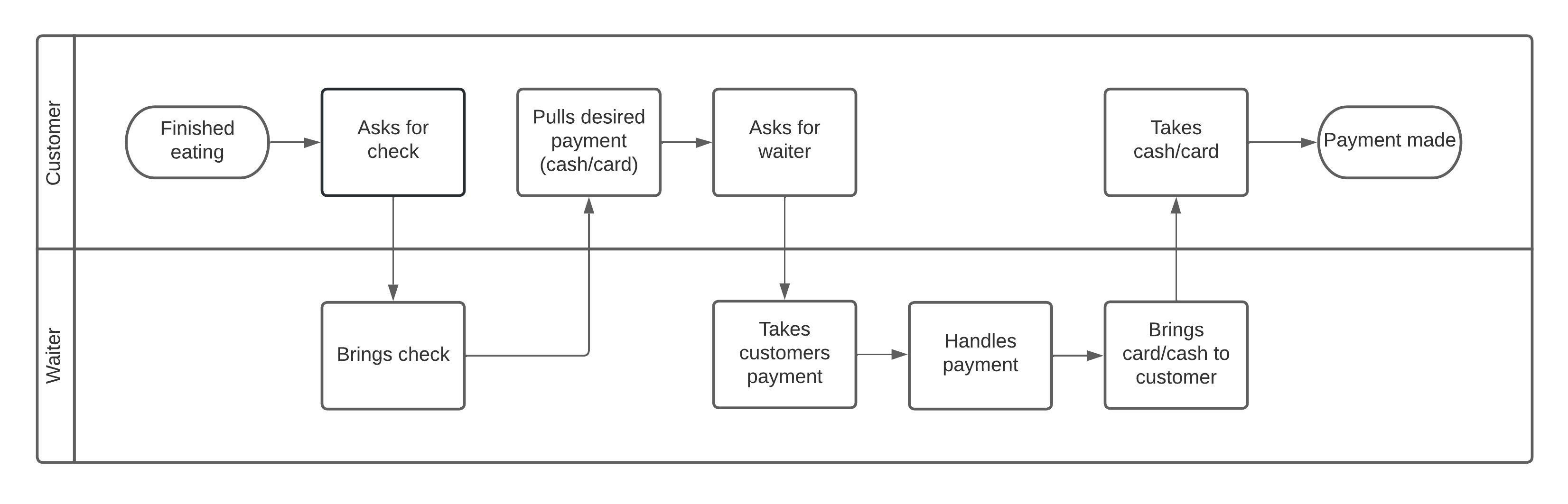
Here is the ordering system flow:

Diagram

Description automatically generated

This automates the process of going through the different stages of ordering and makes it much more cost effective and reduces the amount of effort exerted by the waiter and waiting time by the customers at the table thus reducing costs for the restaurant and raising customer satisfaction.

We also had an idea for the payment idea where the payment doesn’t take long and the interaction of the waiter, this is useful for the pressure of tipping for example or for the unprofessional customers that might want to haggle with the waiter thus wasting the time of the restaurants employees.



LO 08: Professional:

Introduction:

Professional skills are essential especially in a work environment where we must comply with certain requirements schedule and communication wise, this also goes for work division where equal and fair work division must be respected and for everyone to be equally focused on the tasks at hand.

What did I do?

I worked on the asp net backend of the project as I had worked on it before hand in the previous semesters I had while others worked on the project’s front react.

Working with my experienced group mates only helped in advancing my knowledge in this sector and my experience in solving problems more efficiently where me and my other group mates that worked on the ASP net part of the project communicated thoroughly on discord, whatsapp and teams.

For example this is a conversation between me and one of my groupmates where he provides me with references on how to solve some certain tasks easier then the method I was taking:

Text

Description automatically generated

Our communication as a group as a whole was still present and not just restricted to the front and back end respectively, as we had a main voice and chat channel where we would communicate on a daily basis through our stand ups and stand downs.

All my researches that are present in the documents folder of my Github repository constitute a clear representation of how a DOT framework documentation should be, such as the main and sub questions that were raised and the use of literally study in my JWT document as one example or my spring boot vs asp net document as another example.

The meetings with the product owners and the emails sent to them also represent my attitude as a professional that I have also learnt from my own personal internship as well this semester.

For the below researches I used the Library research method while following the swat analysis study as can be seen in my ASP net vs SpringBoot by comparing the two frameworks by listing the most pivotal aspects to me and seeing which one fits me better.

I also used literature study as I looked through articles in the JWT authentication to find the diagrams, to find the necessary steps needed to install it and preview its mechanism.

I also used community research to write my document on asp net vs spring boot as my fellow peers were using SpringBoot to build their APIs I could find the advantages and disadvantages of them.

The following are links to my IP research documents:

<https://github.com/Hishamabboud/Ecom/blob/main/Documents/ASP%20NET%20VS%20SPRINGBOOT.docx>

<https://github.com/Hishamabboud/Ecom/blob/main/Documents/Why%20Integration%20Testing.docx>

<https://github.com/Hishamabboud/Ecom/blob/main/Documents/JWT%20Authentication.docx>

**Conclusion and reflection**

My experience in this semester was a good one as I had great groupmates who were very good at what they did, and it was a better experience because they were from different backgrounds which helped me learn more and more about their own culture and experiences.

The group project was also very interesting and proved to be a challenge, but we faced that challenge well and were able to finish tasks at a very high pace.

Mean while in my own personal project I found that some of the CI/CD learning outcomes were not easy as they were a new concept however I know that I met the requirements all around and was able to achieve the level need to pass this semester especially my backend which in my opinion is exceptional taking into consideration the independence of all the data and the compliance with the persistence of data reaching a commercial level and standard.

I had some trouble at first because I also had an internship at Delta Electronics where I was working with a software developer manager from Switzerland and provided me with a great experience and opened a whole sea of opportunities.

I feel also that I have met the requirements from my own personal product owner for the time being while still providing the ability to expand the project further in the future to create a publishable application to the public and market.