

TITLE:

5Minuti Website

CANDIDATE NUMBER(E):

10089

10042

DATE:	COURSE CODE: IE303612	COURSE TITLE: Bacheloroppgave	RESTRICTION:
STUDY PROGRAM: Dataingeniør (004DA)	PAGES/APPENDIX: 68/3	LIBRARY. NO.:	

SUPERVISOR(S) :

Client: 5Minuti

Supervisor: [Girts Strazdins](#)

SAMMENDRAG:

I de siste tiår har nettsider spilt en stor rolle i vårt samfunn. nettsider har mange ulike roller. Nettsider kan tiltrekke og informere lesere om forskjellige ting. Dette er også tilfellet for vårt prosjekt. Våre klienters restaurant bestemte seg for at de også skulle lage en nettside med målet om å informere leseren og gi kundene en unik service, å la dem bestille fra en online Platform. Hoved målet med prosjektet var å lage en nettside for en pizza restaurant her i Ålesund. En nettside som ville bli brukt til å bestille mat og samtidig gi informasjon om restauranten. Nettsiden er designet for å gi kunder valget om å se og velge ulike produkter etter deres preferanse, samtidig som så kalte «admins» kan prosessere og se disse ordrene. Nettsiden vil også kunne prosessere ordre. Prosjektet var planlagt etter klientens behov. Prosjektet inkluderte planlegging, utvikling og til slutt en rapport. Rapporten oppsummerer planleggingen og inkluderes en rapport med detaljer av planlegging og utviklings fasene for prosjektet sammen med den samlede prosessen over prosjektet.

Denne oppgaven er en eksamensbesvarelse utført av studenter ved NTNU i Ålesund.

Postadresse
NRNU i Ålesund
N-6025 Ålesund
Norway

Besøksadresse
Larsgårdsvegen 2
Internett
www.hials.no

Telefon
70 16 12 00
Epostadresse
postmottak@hials.no

Telefax
70 16 13 00

Bankkonto
7694 05 00636
Foretaksregisteret
NO 971 572 140

Obligatorisk egenerklæring/gruppeerklæring

Den enkelte student er selv ansvarlig for å sette seg inn i hva som er lovlige hjelpemidler, retningslinjer for bruk av disse og regler om kildebruk. Erklæringen skal bevisstgjøre studentene på deres ansvar og hvilke konsekvenser fusk kan medføre. Manglende erklæring fritar ikke studentene fra sitt ansvar.

<i>Du/dere fyller ut erklæringen ved å klikke i ruten til høyre for den enkelte del 1-6:</i>		
1.	Jeg/vi erklærer herved at min/vår besvarelse er mitt/vårt eget arbeid, og at jeg/vi ikke har brukt andre kilder eller har mottatt annen hjelp enn det som er nevnt i besvarelsen.	<input checked="" type="checkbox"/>
2.	Jeg/vi erklærer videre at denne besvarelsen: <ul style="list-style-type: none">• ikke har vært brukt til annen eksamen ved annen avdeling/universitet/høgskole innenlands eller utenlands.• ikke refererer til andres arbeid uten at det er oppgitt.• ikke refererer til eget tidligere arbeid uten at det er oppgitt.• har alle referansene oppgitt i litteraturlisten.• ikke er en kopi, duplikat eller avskrift av andres arbeid eller besvarelse.	<input checked="" type="checkbox"/>
3.	Jeg/vi er kjent med at brudd på ovennevnte er å <u>betrakte som fusk</u> og kan medføre annullering av eksamen og utestengelse fra universiteter og høgskoler i Norge, jf. Universitets- og høgskoleloven §§4-7 og 4-8 og Forskrift om eksamen §§14 og 15.	<input checked="" type="checkbox"/>
4.	Jeg/vi er kjent med at alle innleverte oppgaver kan bli plagiatkontrollert i Ephorus, se Retningslinjer for elektronisk innlevering og publisering av studiepoenggivende studentoppgaver	<input checked="" type="checkbox"/>
5.	Jeg/vi er kjent med at høgskolen vil behandle alle saker hvor det forligger mistanke om fusk etter høgskolens studieforskrift §31	<input checked="" type="checkbox"/>
6.	Jeg/vi har satt oss inn i regler og retningslinjer i bruk av kilder og referanser på biblioteket sine nettsider	<input checked="" type="checkbox"/>

Publication Agreement

Studiepoeng: 20

Veileder: Girts Strazdins

Date:

Fullmakt til elektronisk publisering av oppgaven

Forfatter(ne) har opphavsrett til oppgaven. Det betyr blant annet enerett til å gjøre verket tilgjengelig for allmennheten ([Åndsverkloven §2](#)).

Alle oppgaver som fyller kriteriene vil bli registrert og publisert i Brage HiM med forfatter(ne)s godkjenning.

Opgaver som er unntatt offentlighet eller båndlagt vil ikke bli publisert.

Jeg/vi gir herved NTNU i Ålesund en vederlagsfri rett til å gjøre oppgaven tilgjengelig for elektronisk publisering:

ja nei

Er oppgaven båndlagt (konfidensiell)?

ja nei

(Båndleggingsavtale må fylles ut)

- Hvis ja:

Kan oppgaven publiseres når båndleggingsperioden er over?

ja nei

Er oppgaven unntatt offentlighet?

ja nei

(inneholder taushetsbelagt informasjon. [Jfr. Offl. §13/Fvl. §13](#))

PREFACE

We would like to extend our thanks, first and foremost, to Girts. He has been the supervisor throughout the project and the only other person that we have been able to ask for help during the semester. He has also done a very good job at providing help to each of the group members and has done a very good job making sure we were on track and productive every week during the project. He has had a meeting with us every other week for the project. Girts helped in the coding and planning processes.

We would also like to thank Cinque Minuti restaurant owners, Enrico, and Svetlana. We realize this was not the priority project for them and that the idea was a bit wobbly at first. However, they found us a project and we were able to plan and create it the best we could.

We would also like to thank our family and friends. Though out the project planning and writing of the report, they have come with some suggestions that helped us both design the frontend of the website as well as give ideas over what to write in the final report.

INNHold

SUMMARY	9
TERMINOLOGY	10
CONCEPTS	10
ABBREVIATIONS	11
1 INTRODUCTION	12
1.1 BACKGROUND	12
1.2 GOALS	12
1.3 MOTIVATION	12
1.3.1 <i>Restaurants goals</i>	13
1.3.2 <i>Group goals</i>	13
1.4 OBJECTIVE	13
1.5 CONTENTS	13
2 THEORETICAL BASIS	15
2.1 AGILE METHODS	15
2.1.1 <i>Scrum</i>	15
2.1.2 <i>Sprint Meetings</i>	15
2.1.3 <i>Story points</i>	15
2.1.4 <i>Planning poker</i>	16
2.1.5 <i>Backlog</i>	16
2.2 DATABASE THEORY	16
2.2.1 <i>Relational databases</i>	16
2.2.2 <i>Keys</i>	16
2.2.2.1 Primary Key	17
2.2.2.2 Foreign Key	17
2.2.3 <i>Database Relationships</i>	17
2.2.3.1 One-to-one	17
2.2.3.2 One-to-Many	17
2.2.3.3 Many-to-many	17
2.3 XMLHTTPREQUEST	18
2.4 FETCH	18
2.5 JSON WEB TOKEN (JWT)	18
2.6 AUTHENTICATION	19
2.7 REST	19
2.7.1 <i>Separation of client and server</i>	19
2.7.2 <i>HTTP-methods</i>	19
2.7.3 <i>Response codes</i>	19
2.7.4 <i>Paths</i>	20

3	MATERIALS AND METHODS	21
3.1	ORGANIZATION	21
3.1.1	<i>Project Team</i>	21
3.1.2	<i>Client</i>	21
3.1.3	<i>Supervisor</i>	22
3.2	MATERIALS USED DURING THE PROJECT	22
3.3	TECHNOLOGY	23
3.3.1	<i>Postman</i>	23
3.3.2	<i>Apache Netbeans</i>	23
3.3.3	<i>Github</i>	23
3.3.4	<i>Gitkraken</i>	23
3.3.5	<i>Diagrams.drawio</i>	23
3.3.6	<i>Gantt</i>	24
3.3.7	<i>Trello</i>	24
3.3.8	<i>W3schools</i>	24
3.3.9	<i>Zoom</i>	24
3.3.10	<i>Discord</i>	24
3.3.11	<i>Spring boot</i>	24
3.3.12	<i>Digital ocean</i>	25
3.3.13	<i>Nginx</i>	25
3.4	PROGRAMMING LANGUAGES	25
3.4.1	<i>HTML</i>	25
3.4.2	<i>CSS</i>	25
3.4.3	<i>JavaScript</i>	25
3.4.4	<i>Java</i>	26
3.4.5	<i>MySQL & SQL</i>	26
3.4.6	<i>jQuery</i>	26
4	RESULTS	27
4.1	PROJECT PHASES	27
4.1.1	<i>Project identifying</i>	27
4.1.1.1	<i>Evaluation</i>	27
4.1.1.2	<i>Defining project</i>	27
4.1.2	<i>Preplanning</i>	28
4.1.3	<i>Requirement analysis</i>	33
4.1.4	<i>Project implementation</i>	34
4.1.4.1	<i>Design</i>	35

4.1.4.2	Testing	36
4.1.4.3	Implementation of the project	36
4.2	SYSTEM ARCHITECTURE	37
4.3	BACKEND	37
4.3.1	<i>Controller</i>	38
4.3.2	<i>Repository</i>	38
4.3.3	<i>JWT and authentication</i>	39
4.3.4	<i>Database</i>	41
4.3.4.1	Tables	41
4.4	FRONTEND	43
4.4.1	<i>Administrator</i>	43
4.4.2	<i>Customer</i>	44
4.5	IMPROVEMENTS MADE DURING DEVELOPMENT?	45
4.6	THE WEBSITE	46
4.7	DEPLOYMENT	54
4.8	HOW DIFFERENT THEORETICAL BASIS WERE USED.	54
4.8.1	<i>Authentication</i>	55
4.9	HOW WE USED EACH TECHNOLOGY	55
4.9.1	<i>Postman</i>	55
4.9.2	<i>Github</i>	55
4.9.3	<i>GitKraken</i>	55
4.9.4	<i>Diagrams.drawio</i>	56
4.9.5	<i>Gantt</i>	56
4.9.6	<i>Trello</i>	56
4.9.7	<i>W3schools</i>	56
4.9.8	<i>Zoom</i>	57
4.9.9	<i>Discord</i>	57
4.9.10	<i>Spring Boot</i>	57
4.9.11	<i>Digital Ocean</i>	57
4.9.12	<i>Nginx</i>	57
4.10	HOW EACH PROGRAMMING LANGUAGE WAS USED	58
4.10.1	<i>Html</i>	58
4.10.2	<i>CSS</i>	58
4.10.3	<i>JavaScript</i>	58
4.10.4	<i>Java</i>	58
4.10.5	<i>MySQL</i>	59
4.10.6	<i>JQuery</i>	59

5 DISCUSSION	60
5.1 LIMITATIONS	60
5.2 PROBLEMS THAT AROSE UNDER DEVELOPMENT	61
5.3 WHAT CAN BE IMPROVED OR ADDED?	62
6 CONCLUSION	64
7 REFERENCES	65
ATTACHMENTS	68

SUMMARY

In the past decade, websites have played a major role in our society. Websites have many unique roles. Websites can attract and inform their readers about certain things. This is the case for our project. Our clients' restaurant decided they should also have a website created, with the purpose of informing their readers and providing their customers with a unique service, allowing them to order from an online platform.

The main goal with this project is to create a website for a pizza restaurant, here in Ålesund. A website that will be used as a food ordering service, will also provide information about the restaurant. The website is designed to allow customers the option to view and select products according to their preferred products, but also can be used by the so called "admins" to process and view these as orders. The website will also entail being used to process orders. The project was planned according to the clients' needs. The project included planning, developing and then a report. The report summarizes the planning and includes a report detailing the planning and development stages for the project along with the overall process over the project.

TERMINOLOGY

Concepts

Backend – the part of the project not visible by the user which stores the data.

Frontend – the GUI of a project.

Branches – unique set of code changes saved in a different location.

Sprints – set timeframe in which work will be done.

Responsive web design - a website that changes to look good on all devices.

Request – used for communication between server and client.

Boolean – data with two alternatives, true or false.

Method – procedure that is associated with an object.

Repository – Placement where data is stored.

Array – data stored in an organized list.

Abbreviations

SQL – Structured query language

HTML – Hyper Text Markup language

CSS – Cascading Style Sheet

ER-Diagram – Entity Relationship Diagram

APP - Application

ICT – Information and communications technology

API – Application Programming Interface

IDE – Integrated Development Environment

GUI – Graphical User Interface

REST-Representational state transfer

URL-Uniform Resource Locator

API –Application Programming Interface

JPA-Java Persistence API

HTTP-Hyper Text Transfer Protocol

PK- Primary Key

FK– Foreign Key

JWT-Java Web Token

1 INTRODUCTION

This section gives the reader a brief intro to the background basis for the project. It includes the goals of both the client and project group as well as lists the structure over what will be explained in the report. It explains the primary objectives of the project and details the content found within the report.

1.1 Background

Cinque minuti is a recently established café restaurant. The café is located in Ålesund. The café provides food such as pizza and pasta, but also, currently specialize in small cakes called “pasticcini”, in Italian. Since they are also able to make larger cakes, and want to make sure each cake gets sold, they came up with the idea for them to broaden their menu thru a website ordering service. Our project works as that ordering service. Considering, that the ideal situation with the project, is for customers to be able to preorder cakes using a mobile friendly website. This will allow the restaurant to know in advance what they need to prepare. This will essentially prevent the restaurant from making a large cake and displaying it, only to find out that it will not get sold.

1.2 Goals

Since the goal over the project has been different from the clients’ perspective and the students, we had to meet a “middle-ground”, fulfilling goals for both parties. This section explains the difference in goals between the restaurant owners and our own.

1.3 Motivation

Our project group was formed in November of 2020. The group consisted of three members, at the time. All the members had a general interest in a project that had something to do with creating and designing a website. Therefore, we applied for a project that would be based on web design. We also had an interest in a project where we would be able to utilize the same technology we had previously gone thru, in different classes since we started the bachelor’s program at NTNU.

As a group, we wanted to have a bachelors where we would also learn from. Our motivation also led to, wanting extra experience and broaden our knowledge with the programming languages required in making a website.

1.3.1 Restaurants goals

The goal of the restaurant is to be able to sell larger portion cakes without the need to make them and put them on display in the store. This way the cake can go bad before getting sold. By having a website ordering service, they will be able to sell only what is already ordered. In large, this will essentially reduce waste.

1.3.2 Group goals

The goal of the group is to make the website as consistent with the wishes of the client. It is also the main priority to be able to deliver a functional product that the restaurant can start working with. To make this website both mobile friendly and scale to bigger screens, is also a huge advantage towards the market for the clients. Learning from the project is also a major goal the group focuses on. This project will essentially give each group member “work” experience. We will also be getting experience working with real clients.

1.4 Objective

5minuti website revolves around making a website that functions mainly as cake catering service, with the possibility to expand its service to pizza and pasta eventually in later versions. The website will let customers order cakes as well as letting the 5minutti owners see and process those orders on a digital platform, being this website. The website will be made, so that it is mobile friendly, allowing users to order using phones. The user facing parts of the website will be responsive.

1.5 Contents

This report will further explain the problem and solution that we were faced with thru-out the project. The report will first analyze the theoretical foundation for the project. This will list all the theoretical concepts we utilized during the project. The theoretical basis will also explain how we used each aspect of their concepts on the website. After that, the project will talk

about the organization over the project group. The organization will define the tasks of the project group, clients and supervisor and what jobs each of us had. The project will then also detail the technology we used for the different areas of the project. The results will come towards the end. this will explain the preplanning process and thereafter go thru the implementation phases explaining each process that was completed and how we completed each task. After the results are explained, the report will also give details about things that can be implemented for further development. Thereby, giving the clients a product that has been planned further then the expectations have been. After that, any unfinished stages will also be defined in the discussions as well as determining things that could be improved.

2 THEORETICAL BASIS

The theoretical basis covers the collection of concepts used throughout the project. It briefly explains each concept and will be explained later in the report.

2.1 Agile Methods

Agile methods refer to processes that a software developer uses to ensure continuous delivery over the timeframe of the project. This subchapter explains methods that are relevant for our assignment. It also details the principals of scrum and techniques related to scrum.

2.1.1 Scrum

Scrum is an agile method where the work is focused on iterative goals. These iterative goals are usually set in a backlog of goals set by the product owner. In scrum you can use something called story points to give some measure of how difficult a task will be to solve. Each iteration in scrum is called a sprint and after each sprint you have meetings with the clients.

2.1.2 Sprint Meetings

At the start of every sprint, the project team has a meeting with the clients where progress is shown and information over the features and improvements that have been implemented in the previous sprint is shared. The project team then discuss these changes with the client and supervisor and then determines what the priority of upcoming development changes will include. In these meetings the goal is to find out what the client wants and that might change from one week to the next. Therefore, adapting to this constantly changing vision is what agile methods is all about.

2.1.3 Story points

Story points is an arbitrary value that gives the developers an idea of how much work will go into a given task. This value is usually only useful within the group that set it because it is based on the work of the team that set the value. This value becomes more useful as time goes on because the group will get an idea of how much work a given task is by looking at the points.

2.1.4 Planning poker

Planning poker is a way for a development team to estimate the difficulty of the different tasks that are in the backlog. You do this by having a set of cards with numbers on them from anywhere between 0-100 that you use to estimate the story points of a task. Then you have a meeting with the group where everyone gives their estimate for how many points a task should be worth and this way the group can discuss and adjust the perceived difficulty of a task together.

2.1.5 Backlog

The backlog is a list of the tasks that need to be done such as features that need to be implemented. As the client comes up with new ideas and features to be implemented these get added to the backlog.

2.2 Database theory

In this subchapter we will be talking about different database theory that is relevant for our project and explain terms and methods used in databases.

2.2.1 Relational databases

Relational databases are ways to organize data using tables. These tables are then linked to other tables that are using data that they have in common. These table links give you a better understanding of how your data relates to each other and how you can access one piece of data in relation to another piece of data.

2.2.2 Keys

In a database there are different types of keys that are used. This section will explain the concept of the different keys and what their purpose is.

2.2.2.1 Primary Key

The primary key is the key that defines the row in a table often called the id, this is usually a numerical value that is generated by the database and autoincremented every time a new row is made.

2.2.2.2 Foreign Key

Foreign keys are keys that link data together between two different tables, the foreign key act as a cross-reference between the tables because it points to the primary key in another table (Juliano Rabelo, 2021).

2.2.3 Database Relationships

In this subchapter database relationships will explain how the database uses these relationships. These relationships include, one-to-one, one-to-many, and many-to-many.

2.2.3.1 One-to-one

In a one-to-one relationship, a row in a table has only one corresponding row from another table. In some cases, this is done for security reasons, where the table is divided, and sensitive data is in its own table (Ian, 2016). An example of this can be an order system where the order points to a customer instead of having the customer directly in the order, with the assumption that a system where a new customer is added to the table for every order.

2.2.3.2 One-to-Many

One-to-many is generally the most used relationship type (Ian, 2016). In a one-to-many relationship you have one row in a table that has a reference to multiple rows in another table, but those tables will only have a reference a single reference back to that row. An example of this could also be an order system but in this example the customer has a user that can be used for multiple orders where each order would point to one user, but a user could point to multiple orders.

2.2.3.3 Many-to-many

In a Many-to-many relationship, you have multiple rows in a table that can reference multiple rows in another table this also means that those rows can have multiple references to the

original table as well(Ian, 2016). A many-to-many relationship can also be thought of as two one-to-many relationships where they use a link table or a junction table to link to each other. An example of this could be a relationship between products and costumers since a customer can buy multiple products and a product can be bought by multiple customers and the junction table in this case would be the order.

2.3 XMLHttpRequest

The XMLHttpRequest allows users to interact with servers using a URL to access data(MDN contributors, 2021). XMLHttpRequest is used as way to send asynchronous requests from JavaScript to the backend API.

XMLHttpRequest's were used to send data inputted in the frontend to be stored on the backend.

2.4 Fetch

Fetch is an API similar to XMLHttpRequest but has a more simplified approach which doesn't require handling redirects(Google Developers, 2019). Just like XMLHttpRequests, Fetch also is used, as a way, to send asynchronous requests from JavaScript to the backend API.

In the project fetch was used to retrieve data from the backend and then reading that data as json data. It then returns a function that creates an element in JavaScript for each menu item.

2.5 JSON Web Token (JWT)

JWT is a standardized way authenticate a user using tokens. The token is signed, using a secret key or a public/private key(Wikipedia, 2021). When a user successfully logs in by sending the login credentials to the server, they will receive a JWT back as a response, this token is then saved on the users end usually in some sort of local storage or a session. This token is then added in an authorization header in future requests that require authentication.

2.6 Authentication

Authentication is the process used to verify the user accessing the specific hidden data. A username and password are most used for authentication purposes because it identifies the person that is accessing the webpage(P.Christensson, 2018).

2.7 REST

REST is a type of architectural style that gives a way for a client to communicate with a server(Codecademy, u.å). With a REST API a client will send a request usually to a set URL/path for example “/product/list” this represents the resource the client wants to access.

2.7.1 Separation of client and server

A good thing about REST is that it helps separate the client and the server which increases a project coupling because the server and client become independent systems that are not necessarily reliant on each other. This means that code can easily be changed on one system without affecting the other. If the client knows what format the server is expecting these modules can communicate while being separate. This improves the scalability and makes the project more flexible.

2.7.2 HTTP-methods

In rest, the client communicates with the server using different types of http methods. These methods represent the action the client wants to do. These methods are GET, POST, PUT and DELETE. Where GET is used to retrieve a resource, POST is used to create a resource, PUT is used to update a resource and DELETE is used to delete a resource.

2.7.3 Response codes

A REST API uses response codes to give the client information about the status of the preformed request. This can be as simple as a 200 request to tell the user that the request was successfully preformed or something like a 401 that tells the client that it is missing the sufficient authorization to perform the request.

2.7.4 Paths

In REST paths are used by the client to tell the server what resource it is trying to interact with. This can be something like the path “/product/add”. Most paths are usually clear in their functions and they help the client know what is going on.

3 MATERIALS AND METHODS

This section explains the role in which each organization had in the team and the creating over this project. It details further the technology that was used in the planning and development stages. This section also talks about the programming languages that were used during the project.

3.1 Organization

The organization over the project has three parts, The Project group, the client, and the supervisor. This section explains each part and their roles for the project.

3.1.1 Project Team

The project team is made up of the members working on the specific project, sharing the same goals for the project. For the project to work efficiently, a team leader is assigned. The team leader's responsibility is to detail the tasks and hand them out accordingly. The responsibility of the team is to define the specific roles and objectives of the project in hand and execute the tasks, in order, to ensure that the project is successful in the expected time and that the right standards are met.

Our group consist currently of two students enrolled in NTNU in Ålesund, although we started with three. We are the ones that will plan and design the website for 5minuti restaurant. Since the team was rather small and only had two members, there was no actual leader defined. We worked towards the same goal and split the tasks primarily from frontend to backend. We worked in two-week sprints, meaning that we have a meeting with both the client and supervisor every other Monday. This leaves us ample time to make changes and to visually show the progression in smaller steps.

3.1.2 Client

The client's role for this project is very crucial. The clients are to define the criteria in which they would like to have a product. They will supply us with their needs for the project. The client will also have the final say in the visualization over the website so that it meets their needs. Client will approve or disapprove of project plans. They will be able to request

undergoing changes as the project progresses, and in the end they will either decide to use or not to use the final project.

The client for the 5minutti website project, are the ones that will end up with the project. They are the ones that created the idea of the project based on their needs and for the further success for their own restaurant. For them, this project was primarily focused on the idea of being able to preorder food, which was later narrowed down to only the sale of cake.

3.1.3 Supervisor

The supervisor for the project was assigned by NTNU to us. For this project, our supervisor was Girts Strazdins. The supervisor has the role of helping the team. If the group struggles or needs help he is the only one that is tied to the project with more experience than the group members itself. He is also there for the spring meetings and to check if we are on track for the completion of the project.

3.2 Materials Used during the project

Due to Covid 19, the work for this project was primarily done from home. Although we had some opportunities to work from school, we did not decide to do that. The materials used relied mainly on a personal computer along with internet connection along with other devices. Each device had to be supplied by each group member. A microphone and camera were also used during the meetings along with other preinstalled software programs. These programs include:

- Postman
- Netbeans
- Github
- GitKraken
- Diagrams.io
- Zoom
- Discord
- Spring Boot

3.3 Technology

This section lists the different software technology we used during the project. All the technology used in this project was decided as a group. The technology we used, have also been technology that we have used over the past years in different classes at our university. Therefore, we planned to use this technology according to experience and ease.

3.3.1 Postman

Postman is an API client that is used to send different types of HTTP requests. It is a simple way to test calls to the backend to check if it is working correctly. Postman allows the developer to create, share and test API with the use of HTTP requests.(Jacob Sharir, 2020)

3.3.2 Apache Netbeans

Apache Netbeans is an opensource IDE that allows developers to develop code with the use of different languages(Mark Stephens, 2016). Netbeans is the IDE we used for all the code in our project. It let us code in Java, Html, Css, and Javascript all on the same IDE.

3.3.3 Github

Github is a version control system that lets developers make undergoing changes to code and revert changed when necessary. This lets developers share versions of the code as it takes effect(KORBIN BROWN, 2019).

3.3.4 Gitkraken

Gitkraken is a user-friendly app that lets utilize the functions of git. Gitkraken works as a GUI to take care of project repositories(Hussnain Fareed, 2019). In these repositories, developers can control the branches of the project so that the work being done does not meet unnecessary conflicts. Gitkraken also lets the user see the history of the work that has been done.

3.3.5 Diagrams.drawio

Diagrams.drawio is a website that allows us to draw out the visual representation of the project. It uses a drag and drop feature that allows developers to easily make diagrams and charts(Computer Hope, 2020).

3.3.6 Gantt

Gantt chart is the project management tool used to determine the timeline of different assignments. The timeline over each assignment may vary due to the size and complexity. This will be reflected on the diagram. It also details the deadlines each assignment should have before it should be completed (Association for Project Management, u.å).

3.3.7 Trello

Trello a software that organizes projects into boards. With the use of Trello, the group can determine what and who is working on a specific task assignment. Trello also lets the group know how far into the process each assignment has reached (Trello, 2021).

3.3.8 W3schools

W3schools is a website that has tutorials for a wide variety of different programming languages. They provide detail instructions for examples over a wide variety of different programming languages.

3.3.9 Zoom

Zoom is an app that allows video communication. Its widely used for conferencing and other video chats because it does not require a registered account (William Antonelli, 2020).

3.3.10 Discord

Discord is an app used for communicating with each other. It allows the users to use voice chat, text chat, and to share screen or files. Discord allows the users to create own private communities or servers filled with their own channels (Devon Delfino and Grace Dean, 2021).

3.3.11 Spring boot

Spring boot is an open-source tool used by developers to build applications (Michiel Mulders, 2019). It is useful in that it reduces development time.

3.3.12 Digital ocean

Digital ocean is a cloud infrastructure that provides cloud services to help deploy applications in the cloud. Digital ocean lets a user rent a virtual machine where you can run different applications such as deploying a Mysql database or a static website(Salman Saleem, 2019).

3.3.13 Nginx

Nginx is an open-source web server that can be used on ubuntu to deploy a web server to run a website. Nginx is built to have low memory usage and has all the features you would expect from a web server(*What Is Nginx? A Basic Look at What It Is and How It Works*, 2021).

3.4 Programming languages

Throughout the project we utilized multiple different programming languages for the creation of this project. Below are the different programming languages we used throughout the project.

3.4.1 HTML

Html stands for Hypertext markup language(Howe, 2014). Html lets us define content by enclosing it in tags. All Html code is written in its own file ending with .html.

3.4.2 CSS

CSS stands for Cascading style sheets(Howe, 2014). CSS works hand to hand with Html in that it provides appearance to the content label in the html files, although, it is written in a different file location.

3.4.3 JavaScript

JavaScript is the language used to make the contents on the Html page dynamic and interactive (Hack Reactor, 2018).

3.4.4 Java

Java is a free to use programming language that us used everywhere, from cellphones to computers(Oracle, u.å). Java is based on the object-orientated programming model. It is widely used because of its reliability (Guru99, u.å).

3.4.5 MySQL & SQL

MySQL is a relational database model. SQL stands for structured query language. It is the domain specific language that is used in MySQL to interact with the database(Herawan, 2020). SQL uses queries to communicate with the database. In these queries, a user can, for example, ask for specific things such as asking for the database to give you a list of all the available books in a library system or you can do broad commands such as just telling it to show you an entire table.

3.4.6 jQuery

Jquery is a javascript library that requires less code, therefore making it simpler in terms of writing. It serves the same purpose as Javascript, in that it can create dynamicity on the webpage and that it also works in hand with Html and CSS(Christina Kopecky, 2020).

4 RESULTS

Overall, the project consisted of three parts, preplanning, development, and this report. The sections below will first explain how the preplanning went and the different stages of the project. The results will also show and explain the final outcome of the website.

This development phase of the project is split into two major parts, the frontend, and the backend. We chose to split the project in this way because it makes the project more flexible and reusable when you have the two parts separated in a way where they are not dependent on each other. This follows good software design principles by making our project more modular and makes it easier to scale the application.

4.1 Project Phases

The project consists of 3 main phases: project identifying, project defining, and project implementation. This section will explain the process between each of the project phases.

4.1.1 Project identifying

This stage of the project involves deciding if a project step should be undertaken based on the benefits, cost and risk involved. There are two parts to project identifying. The evaluation and the defining the project phase.

4.1.1.1 Evaluation

The evaluation part of identification in a project refers to the analyzation over the information identified in the project.

In this phase we collect and evaluate the necessary components of the project. We determined that since the focus is upon having clients being able to order thru the website, we would need to make that a possibility. The group also focused on that the project would be as user friendly as possible, so that customers would not need to order as much in the restaurant.

4.1.1.2 Defining project

The defining project part came at the start of the project. It details, how we first came about starting the project and determining what was necessary for the project. The clients knew a

“concept” of what they wanted but it was decided in this stage that it would become a website.

At first, we were not completely sure how our project would be, nor what we would be working on. We started off by “applying” to a project for ordering tables with the same client. We got accepted to that project, but we were not the first ones. Another group ended up getting that project since they applied first, and neither our group nor theirs wanted a collaborated work project. This meant that our group was not entitled to the first project. This was fine with our group since the clients had another idea for a project. Since the other idea was focused on building a website and other hardware was not going to be necessary, we ended up being on this project. After the clients explained the idea behind this website, we were able to come with suggestions and preplan the website towards their approval. They told us what they wanted to be able to do with the website and we created it that way. The project started with many uncertainties and a group member leaving, we ended up spending a bit more time planning than was anticipated. In the end, however, we worked to the best of our ability and was able to create something.

4.1.2 Preplanning

In the preplanning phase of the project, the group focuses on the thought process over the design and the functionality of the project. The group also has focus on how the backend will function and how the frontend will look.

Planning for the project started during the course we took at the start of the semester. We filled out a preliminary project plan. For the preliminary project, our project group focused on details pertaining to the project. We also planned for two-week sprint meetings throughout the semester.

Planning also entailed using a Gantt diagram, which helped us estimate how long we would be able to use for each task. Since we are limited to about four and a half months for the whole project, the time frame was estimated in accordance with that. Some tasks ended up taking longer and some shorter. An image of the Gantt diagram is as follows.

Preplanning for our group, consisted of sketching diagrams and agreeing with the clients about what each page was going to include. There were many changes underway, this led to a slightly extended time used on the planning phase. It was not always easy to communicate and explain the design ideas. There was a bit misunderstanding, so we recreated the planned slides on the diagram sketched a few times after getting feedback from the clients. The final image of the sketches is listed below.

In figure 4.1, we sketched the Home and contact pages. On this page, we also designed the idea for the navbar at the top. The navbar ended up just how we envisioned it during planning and was identical in the final solution to the website. In the preplanning, the about us page was declared as not needed. This was then removed from the sketch. After a few weeks, this page was then requested again for the final product, since the restaurant owners had content, they wanted to show there. The home page or index page is the first page that will be shown upon accessing the website, for both admins and customers. An image over the idea of home and contact page is listed below.

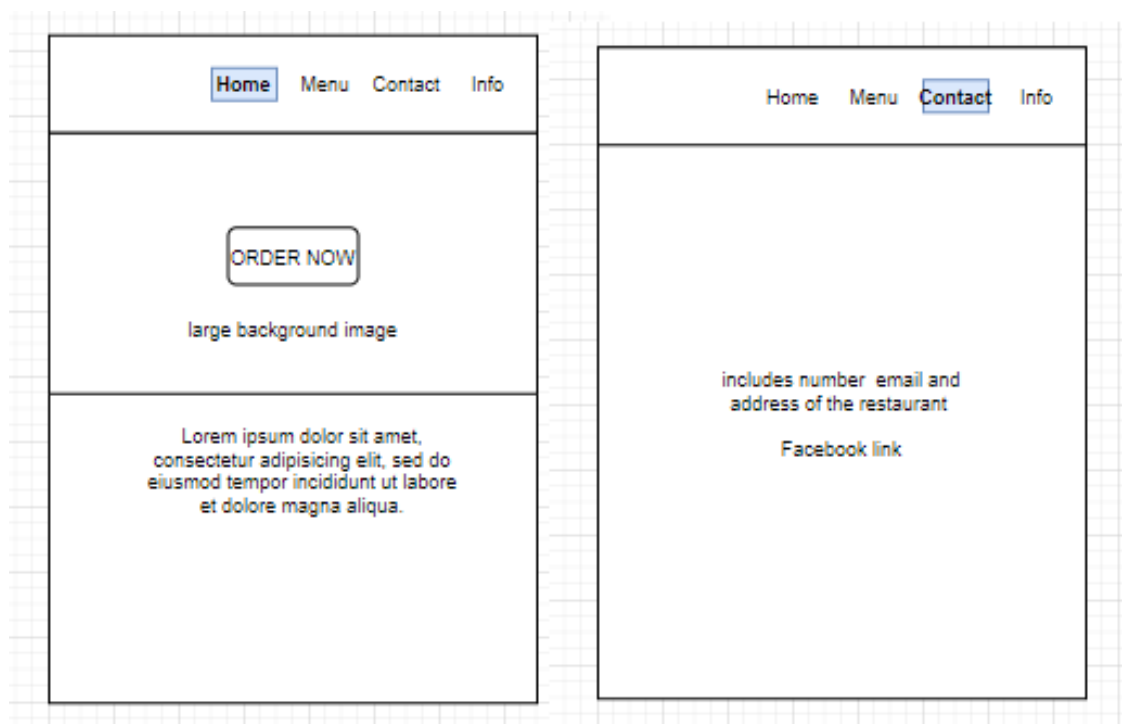


Figure 4.1. Home and Contact Screens

Figure 4.2 resembles the initial idea of how the menu would look. It would also redirect the user to another page, Figure 4, for the shopping cart and the order confirmation page. However, those were all implemented in the end on one single page. This allowed the website to be better suited for the customers using the system. It would also simplify the process of ordering. An image over the initial sketch is listed below.

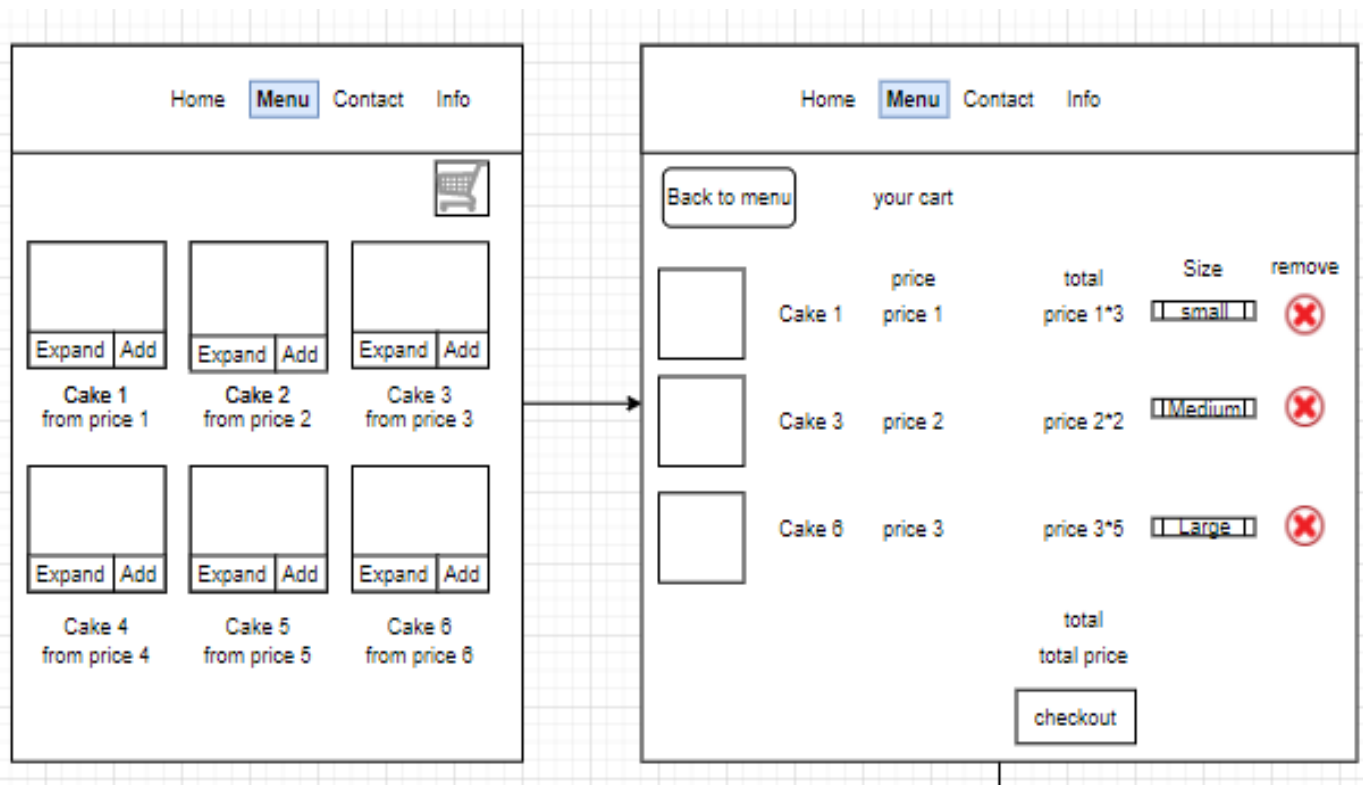


Figure 4.2. Menu and Shopping Cart screens.

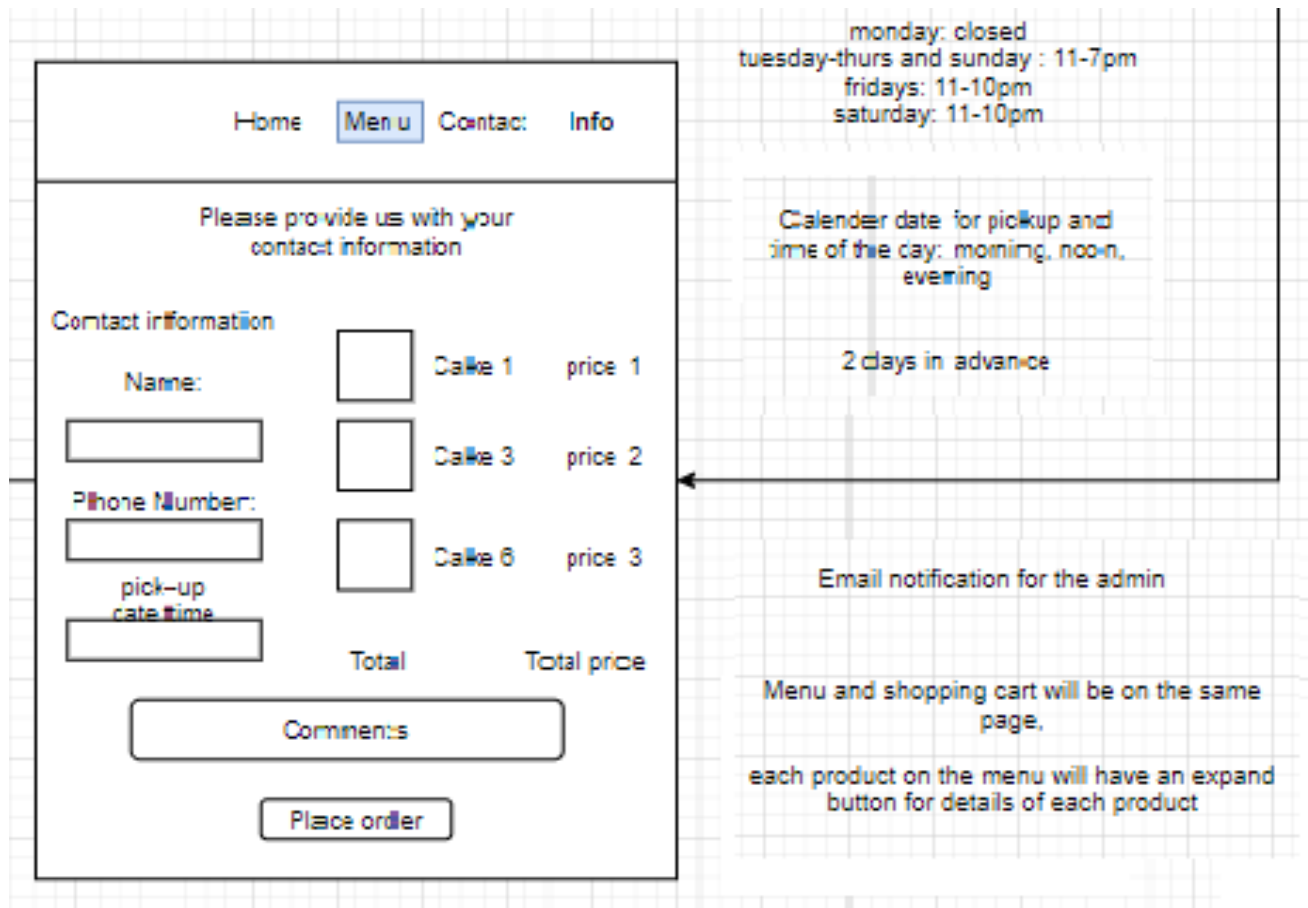


Figure 4.3. Order Confirmation screen.

Figure 4.4, 4.5 and 4.6 represents the admins side of the webpage. The login was visible at first for anyone using the website, however that was then removed. For the admins, to access the login function they would need to type the correct Url, directing them to the logging in screen. After they log in, they have access to three more options on the navbar. This essentially represents their capabilities within the website. That is to view orders, edit menu, and add a product. The order page was duplicated in the sketch to show how it would look with a menu item expanded and without the item expanded. It would also be used to represent the labels for what each product has under the expand function. On the edit menu page, the edit function was not really a reality from the start, but it was a good idea to have the page there in case of further development. The remove button would however be there.

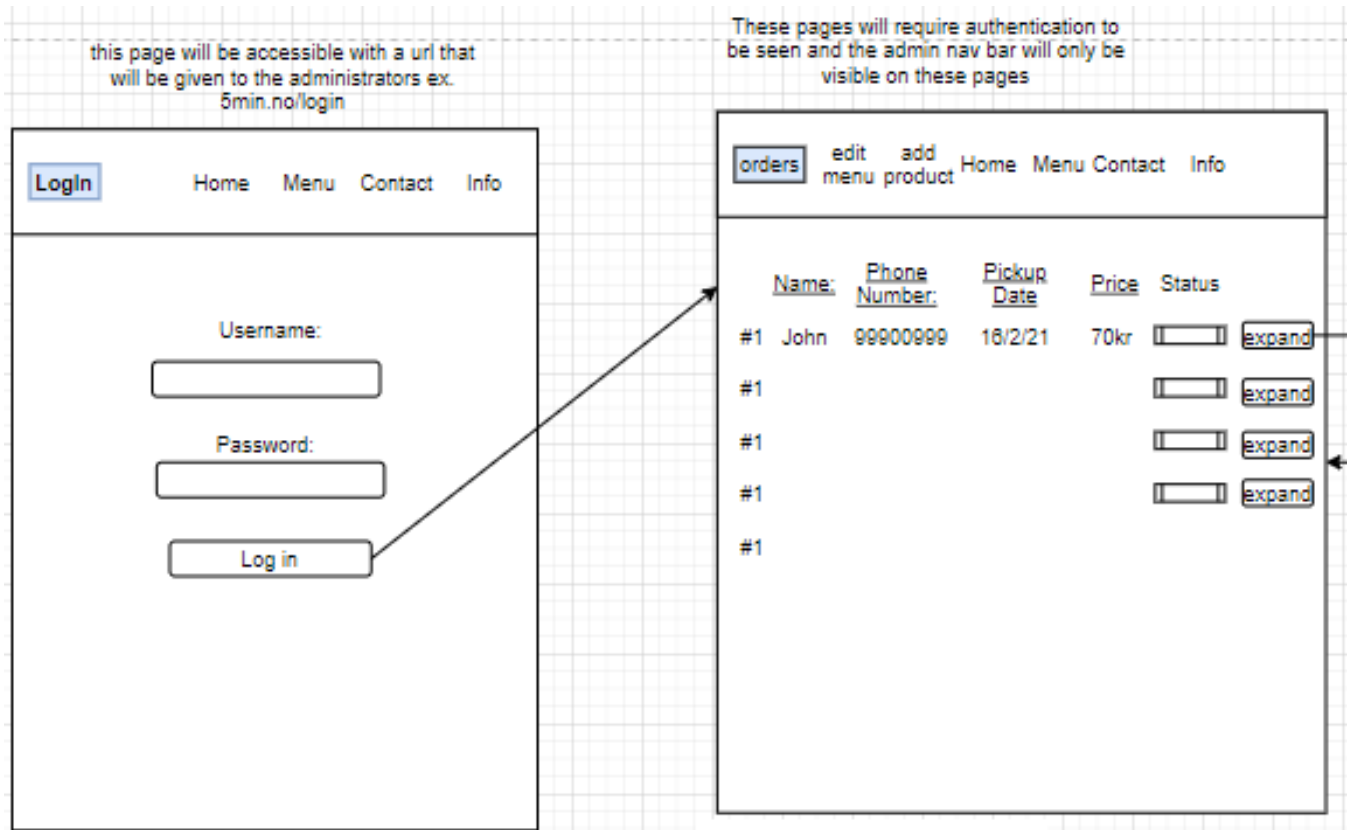


Figure 4.4. Login and Order screen.

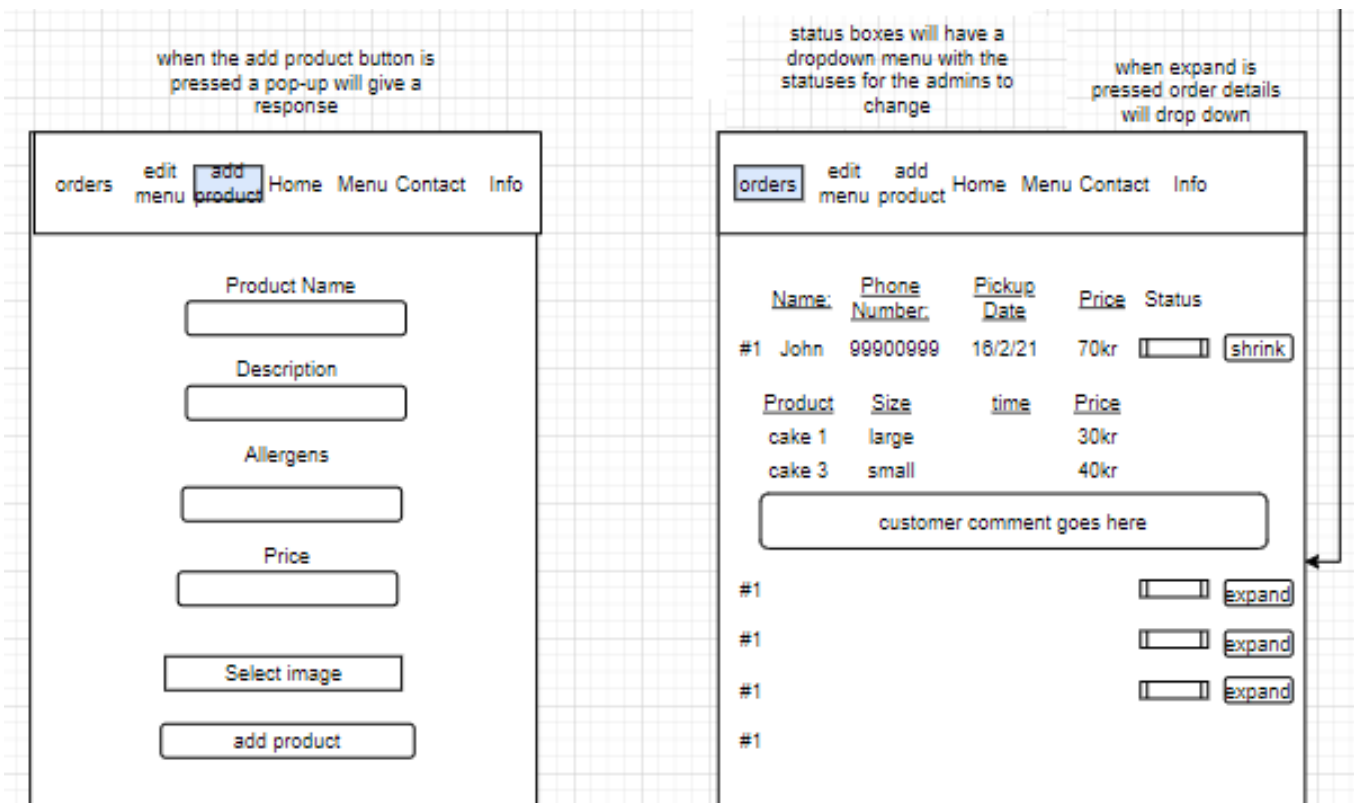


Figure 4.5. Add Product and Order(expanded) screen.

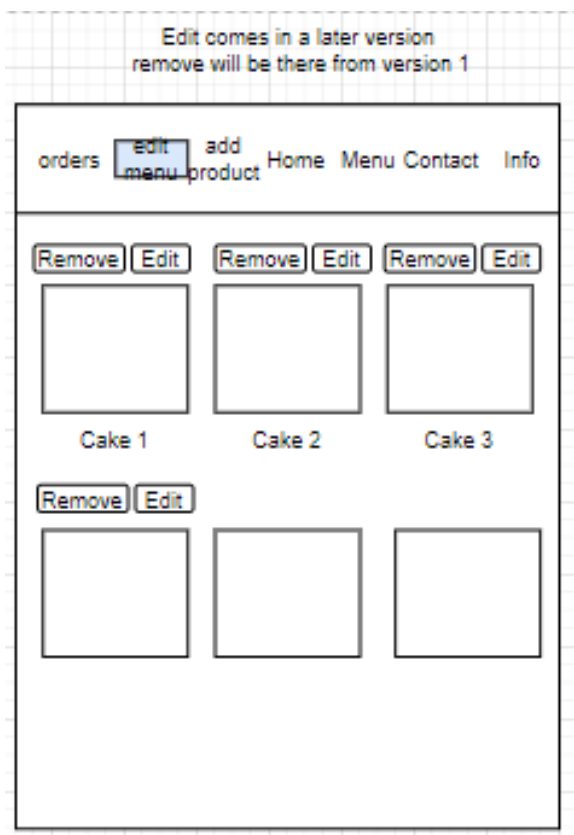


Figure 4.6. Edit Menu screen.

4.1.3 Requirement analysis

For the requirements analysis we determined what was the absolute need for the project in its first version. These were the “vital” components that made the website function. Some of the criteria for the website include:

1. A full “admin” page where the store owners will be able to see purchased orders, which include all the menu items and name and date of order and confirmation of payment. The admin will also, have the ability, to add a product.
2. Each product contains the name a description and 3 set prices for small, medium, and large.
3. Version 1 of the website will only include the possibility of removing an item, since this is more crucial than editing menu items, which can come in later versions.

4. A “home” page, providing customers with information about the restaurant. The clients will provide their personal information, in which will be listed on this page.
5. A “contact” page detailing contact information along with the address location of the restaurant. It also contains Instagram and Facebook links and an included google map location of the restaurant.
6. A “menu” page. This page will be used by the customers for placement of orders. On this page the customer will be able to place an order and read each menu item. They will be able to select each item they would like and see the total cost in the shopping cart, which is also embedded into this page. This page will allow customers to fill personal information for each order which will allow the restaurant owners to view on the admin page.
7. An “about us” page. This page will only provide users with a background knowledge about the restaurant. The restaurant will also provide the details of this page for their personal information.
8. It was also very important to plan for the database to function correctly. So, we have specific elements that had to be stored for the restaurant owners to process different orders accordingly. For all these options there also had to be a way to store the data that was being inputted by the customers using the webpage.
9. Some type of authentication since we knew there had to be a way to safely access the information on the database. The authentication would have to come before the data the admins see.
10. The websites primary focus will be on cake orders that require longer than a two-day preparation period.

4.1.4 Project implementation

Project implementation focused on implementing the, already thoroughly, thought out processed in the previous parts of the project. Since we already had diagrams drawn out for both the frontend and the backend, there was not many changes after we started the coding. We followed the plan accordingly and started designing each page according to the drawings of the website.

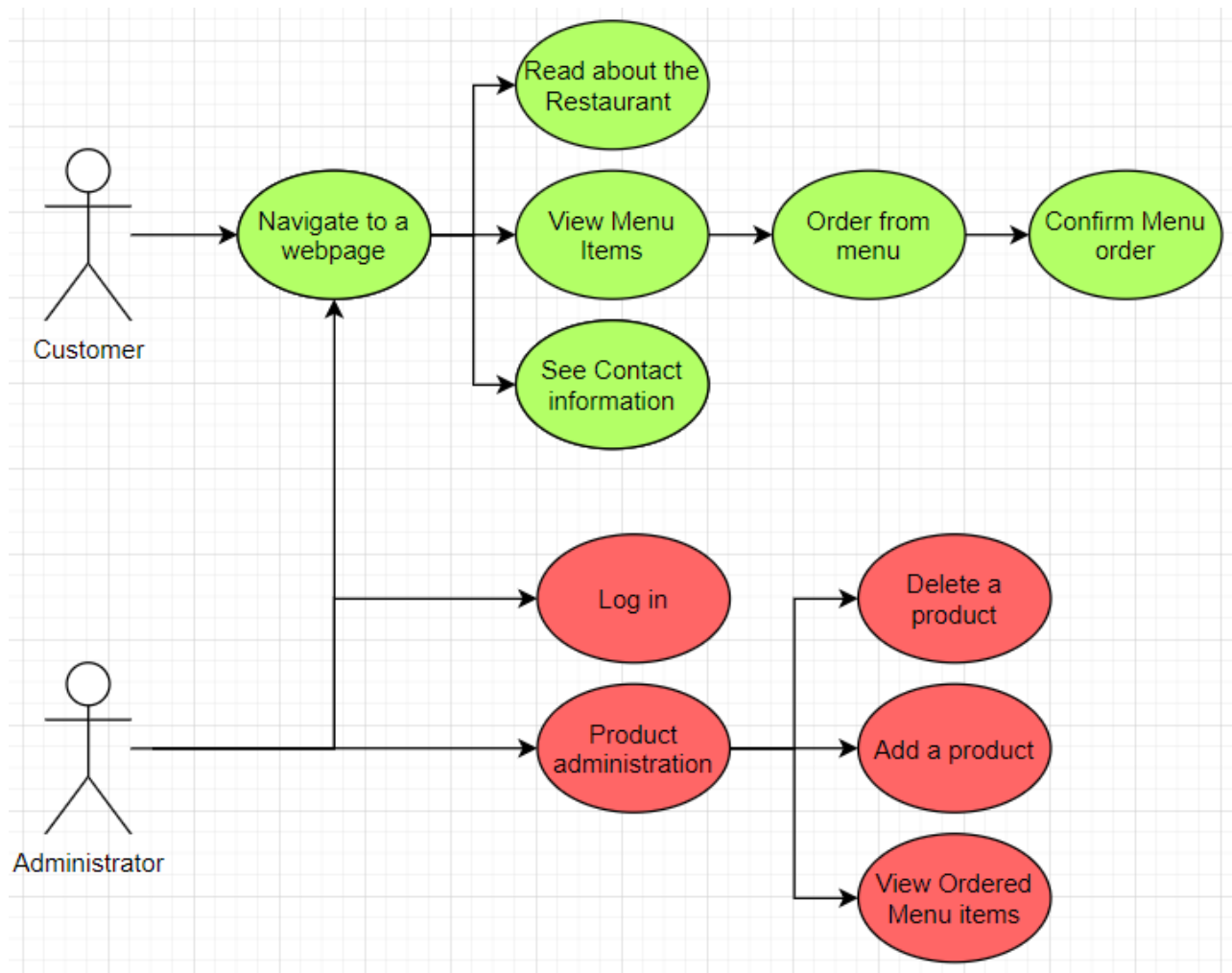


Figure 4.7. Customer and Administrator usage diagram.

4.1.4.1 Design

The design stage of the project focuses on the actual development over the preplanned criteria. The design phase was split into two sections, the frontend, and the backend. The frontend focuses on the interface in which the users will see and have access to while using the website, while the backend refers mainly to the database and applications that the server uses to access information for the user.

For the design we first focused on creating the backend, which is composed of some Java files along with a database written in SQL. Planning for this started with first-hand sketches of an ER-diagram and followed up with it by making changes. Once we decided on a rough draft of the database, we drew it again on the computer using spreadsheet database. We got our

supervisor to look at it and come with suggestions on it. We then developed the database with the use of MySQL, which we had some prior knowledge of, from an earlier course.

The second part of the design was focused on creating the frontend, after the database was planned. We started with the visualization of the frontend so that we could show the clients as soon as possible. With a visual representation over how the website looks, we were also able to agree upon different needs the restaurant will have. Although the client did not always understand the context of this, we were able to get plenty of feedback over what the client wanted.

4.1.4.2 Testing

This phase consists of getting help from others to try and test the product. This phase will help us eliminate the flaws created in the implementation stages of the project. During testing, we hope to find problems that we have not encountered yet while in the development phase.

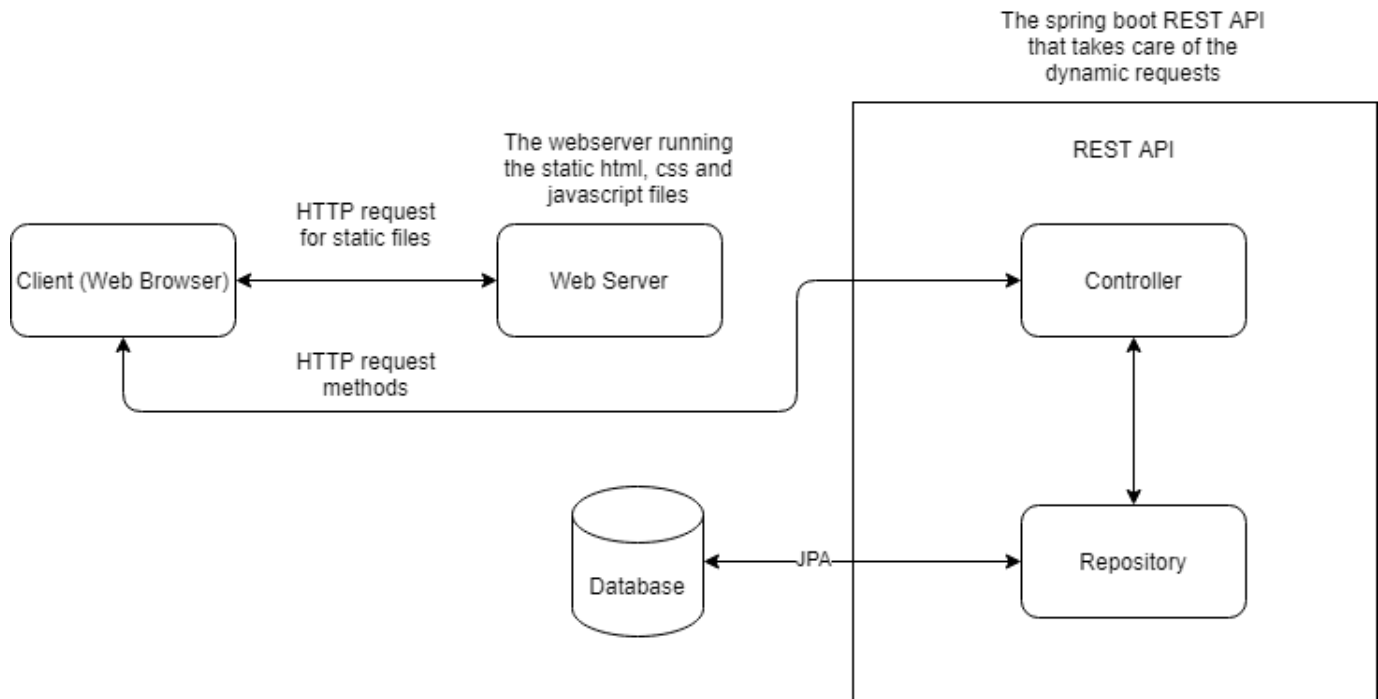
When the product is done testing and the final changes are made, the project will move to the implementation phase.

Testing is done by letting other “test” users gain access to the system and try it out. They will then give feedback based on what they liked, how user friendly it was, any bugs or other mistakes they encountered using this system, and what ways they think it could be improved. This feedback will be shared with the clients and the project team can then view the changes they would also like us to make. This would essentially allow us, the developer group, and the clients gain a better system with less flaws. After all these stages are done, the website will go into the implementation phase.

4.1.4.3 Implementation of the project

The implementation phase over the project happens after testing is done. This stage lets us put the website in use. At this stage, the project should be in tiptop shape and the restaurant can already take in orders. Should the clients choose to deploy this website, then they will be able to use it in this phase.

We will try to deploy the website for the client if they want that. They would have to buy their own domain and pay to keep the website online. They will then have access to use it in their restaurant however they may choose.



4.2 System Architecture

Figure 4.8 Diagram showing the architecture of the system.

Figure 4.8 shows the architecture of the system and you can see the frontend/backend split in our system. We chose to make this split because it makes the system more flexible and easier to work with. The system is set up so that the static pages such as HTML, CSS and JavaScript is served with a web server and all the dynamic requests are done through the REST API. Database interaction is also done through the REST API with repositories and JPA.

4.3 Backend

For the backend part of this project the code focuses on the server part of the project. This includes the java program functioning as a REST api and the MySQL database. All of the

backend runs in the back where the user can access through their frontend GUI. The backend takes care of the dynamic requests that are sent from the frontend. These requests usually involve adding data or receiving data from the database. The backend is a sort of gateway between the frontend and the database where the logic of how to respond to the frontends requests lies.

4.3.1 Controller

The controller receives requests from the client using paths like how it is described in [2.7.4](#). these requests get handled by the controller and it implements one of the many repositories to get or change the resource that is being requested. For each Entity in the system there is a repository that is created using JPA.

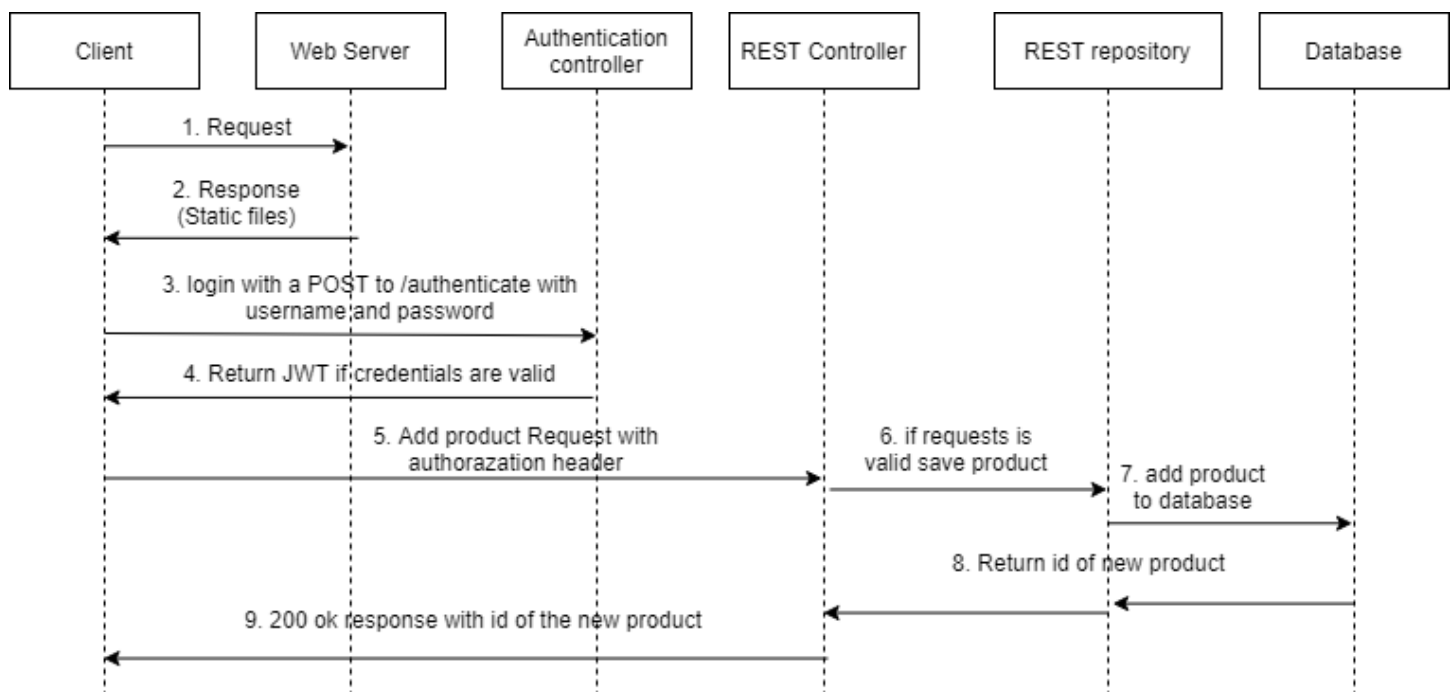


Figure 4.9 Example of a Request an admin might use.

4.3.2 Repository

The repositories are the way our system interacts with the MySQL database. There is one for every entity in the system as each entity represents a different table to be accessed. To interact with the database our repositories use JPA specifically they extend the JpaRepository class.

4.3.3 JWT and authentication

The backend uses JWT for its authentication and the authentication part of the backend has its own controller to handle authentication requests. Our solution uses the example from javainuse as our authentication system with some slight modifications. One of the modifications is to move the user details from the main code and into application.properties for added security. (Javainuse, u.å)

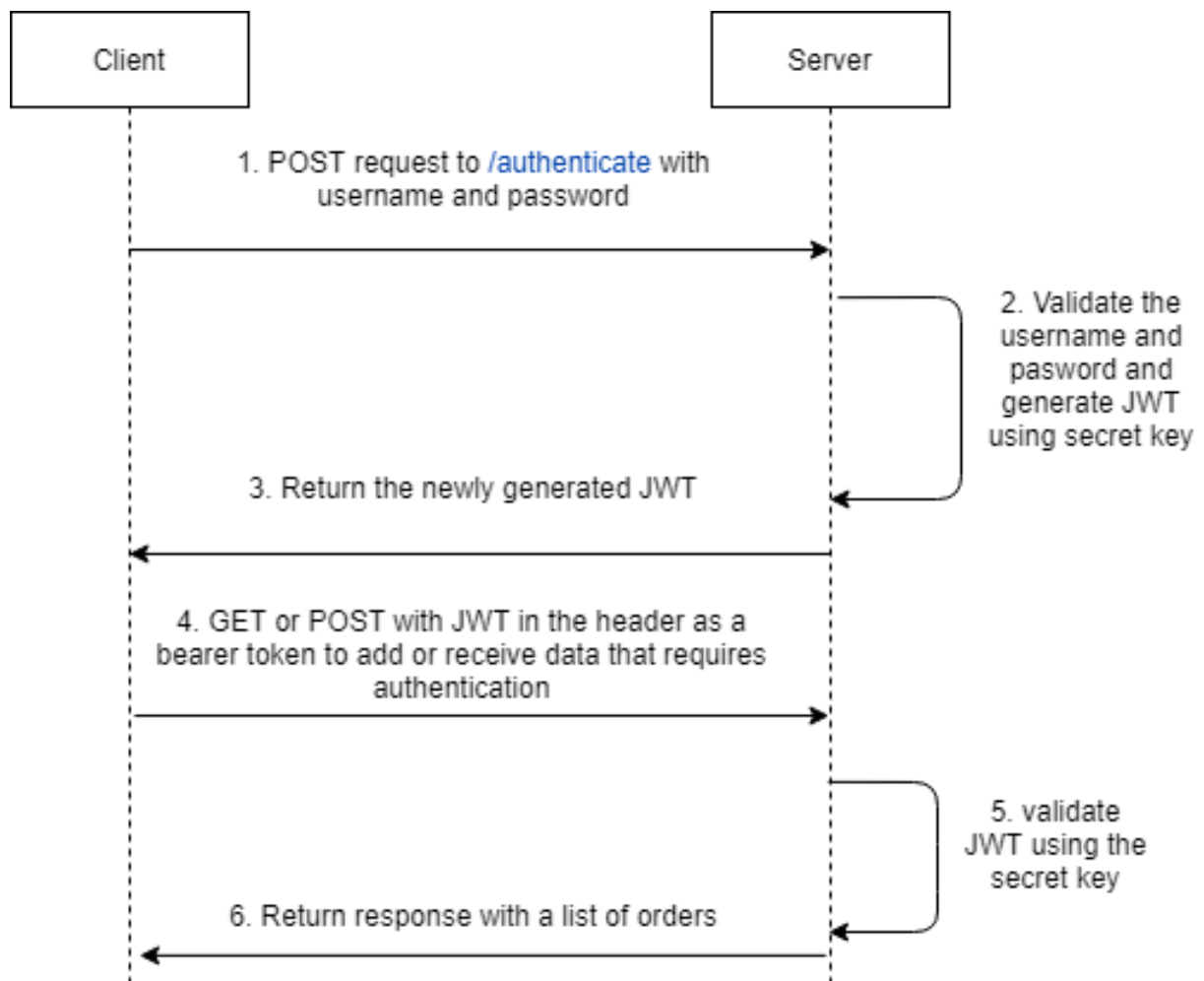


Figure 4.10 Example of an authentication request.

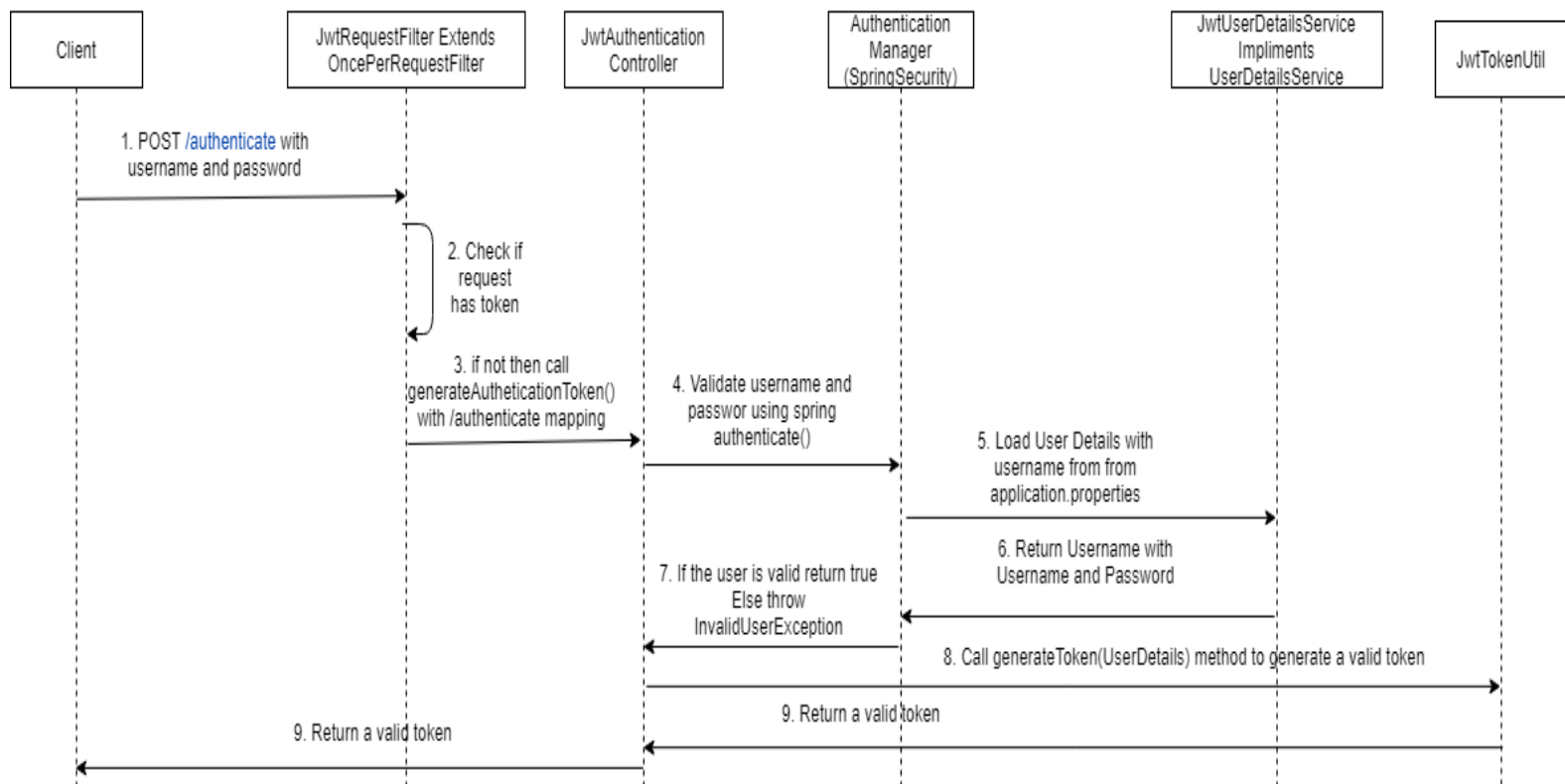


Figure 4.11 sequence diagram showing how a JWT gets generated.

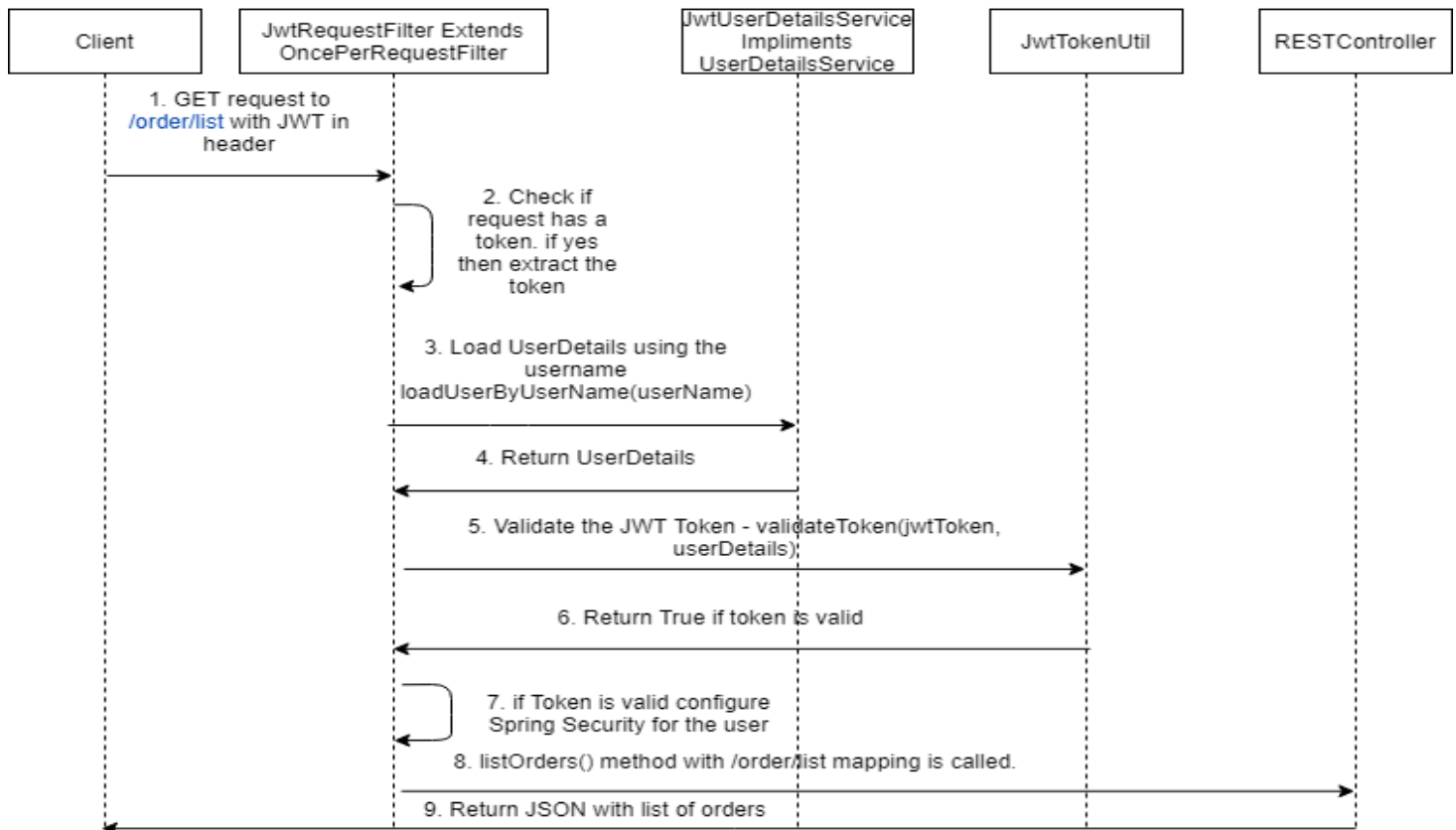


Figure 4.12 example of validating the token where the client asks for a list of the orders.

4.3.4 Database

The database is the collection of information that is stored electronically in tables. For the website, the main information stored is the customer order, their contact information and customer comments. These are the necessary fields the restaurant owners wanted to have access to so that they could contact customers on specific orders and have knowledge about comments the customer might have in addition to their order. These fields are split into diverse tables and organized thereafter.

4.3.4.1 Tables

The database consists of four different tables, the customer, products, orders, and order_detail tables. All which are found inside the database. A description of each table is as follows:

Customers table describes a customer with information about their name, phone number and email address. Figure

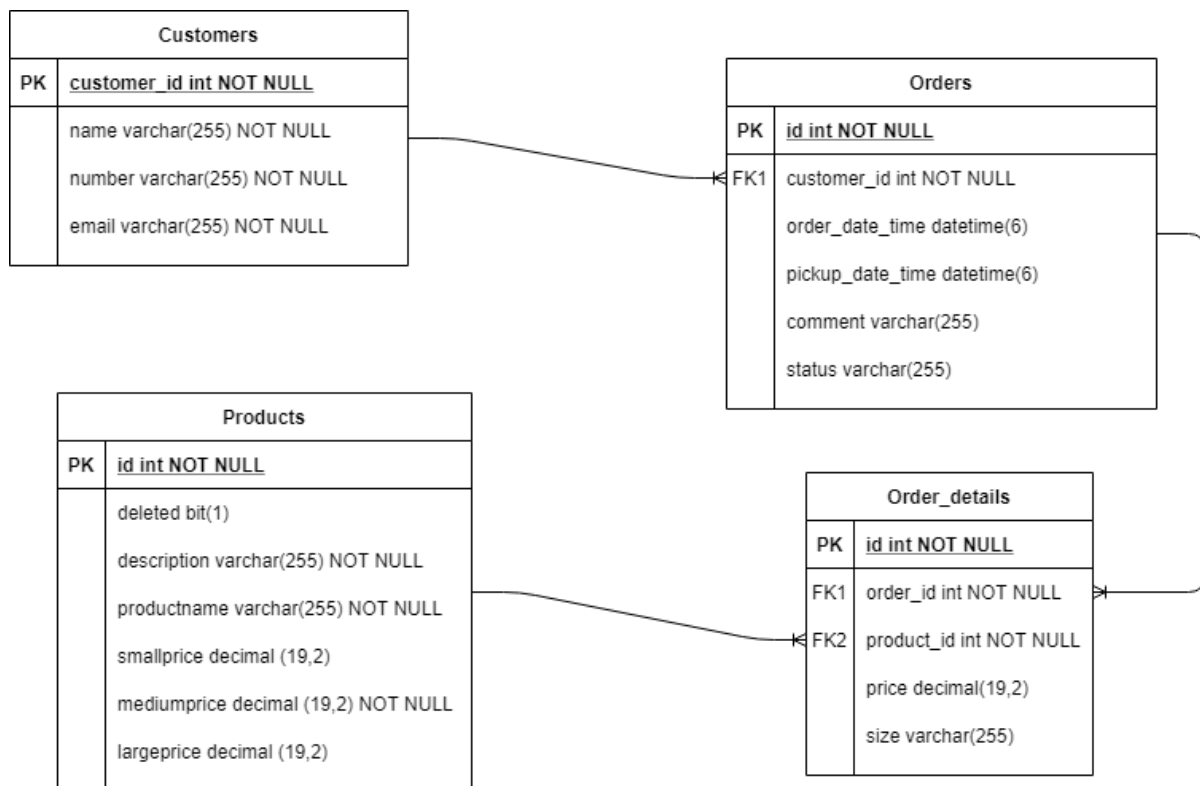


Figure 4.13. ER-Diagram over database.

Products describes a product with information about the name of the product a description and a price for each size small, medium, and large. Since the product does not actually get deleted from the database, the product also includes a Boolean that tells if the product is marked as deleted or not.

Orders contains information about an order with a datetime for when the order was made and a datetime for when the order is set to be picked up. The order also includes a comment the customer has added and a status of how the order is going.

Each order has a link to a customer from the customer table using a “foreign key”. This link has a many-to-one relationship with customer where one customer can have multiple orders and an order links back to a single customer. Currently our Backend system works with have one order per customer in the database, but the database is already set up to handle multiple

orders per customers should the backend be expanded to have registration of users in the future.

Order details table contains details for an order. Each order detail links back to the order it is from, with a Many-to-one relationship. This is how an order can link to many order details but also where an order detail, only links back to one order. Order details also link to a product, that it is giving information about, with another Many-to-one relationship. Where order details also only links to one product, but a product can be used in many order details. Order details also include the size of the product being bought and the price of the product at a specific size at the time of purchase. This extra information is useful to keep all the past orders correct, even if a products price changes in the future.

4.4 Frontend

The frontend part of this project includes the code for the user interface. The frontend refers to the part of the website that is visible to the users. The frontend part has one part for the administrators and another part for the customers, although you access the pages on the same website.

4.4.1 Administrator

The administrators page for the project defines the area where the client can manage products and see which products are being purchased. To start off with, the administrator page will not be visible for customers. The clients using the administrator page will first be directed towards a login screen. This is to ensure an extra security check to make sure not anyone can get access. The administrators have a predefined password that will lead them to the order page, upon the use of authentication as described in [\(2.6\)](#). They will then get three more options in the navigation bar that's located on the top of the screen. These options include the orders, the add product, and the edit product. In the orders option, the client will see each order as it is placed. These orders contain the information over all the purchased products and the total cost of each.

The add product option allows the client to add a new product to the menu. The add product page allows the client to define the products' own name, description, and image as well as determining three set prices for each product. As agreed, upon, for simplicity, the sizes are labeled small, medium, and large. As this product is added it will be displayed in the menu side for the customers to see. However, the clients will also be able to view this content thru either the menu page or the edit menu page. The last option on the administrators' side of the website, is the edit menu page. This page allows the client to only delete and view contents in the menu, although editing was planned for a later version.

Figure 4.14 shows the pages that only the administrators have access to on the website.

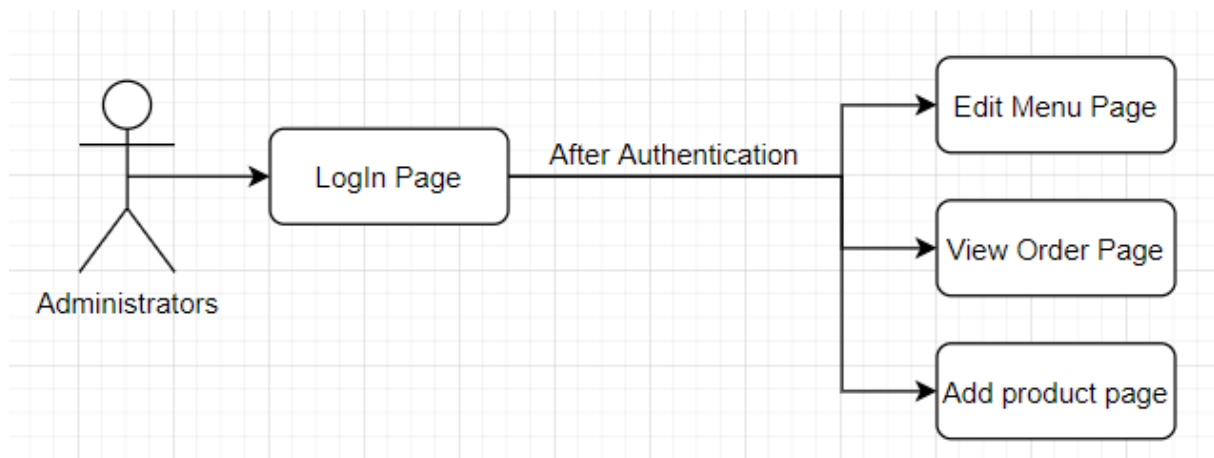


Figure 4.14. Administrator Access pages.

4.4.2 Customer

The “customers” side of the website includes the home page, the menu, the contact and the “about us” pages. The Home page gives a brief introduction to the restaurants' webpage. On this page, the user will find an image with text and a button that leads to the menu. The menu page contains all the menu items of the webpage. This is the backbone for the webpage.

Without it the purpose would be minimal. The menu page lets the customer order and view products supplied by the restaurant. The menu also includes a shopping cart which allows the customer to view their order and make changes before placing the actual order. After the purchase button is clicked, the customers will be prompted to fill in their contact information. This will let the administrators contact customers in case of any discrepancy. The contact

page shows the contact information of the restaurant. This includes the email and phone number. The contact page also details the location of the restaurant using google map, and it includes the link to the Facebook page and the Instagram page. The last page on this part of the webpage is the about us page. This page informs the user about the restaurant owners and the restaurant and includes a short video about the restaurant and some of their products.

Figure 4.15 resembles the websites flow of the overview for what the customer will be able to see when the website it first launched. Which explains that the website will be opened, and the main page will be visible. After the main page is visible, the user can access the menu the contact page or the about us page, using links on each of the pages. Further, the image shows that from the menu the user can access the shopping cart or be redirected towards the checkout/ order confirmation page.

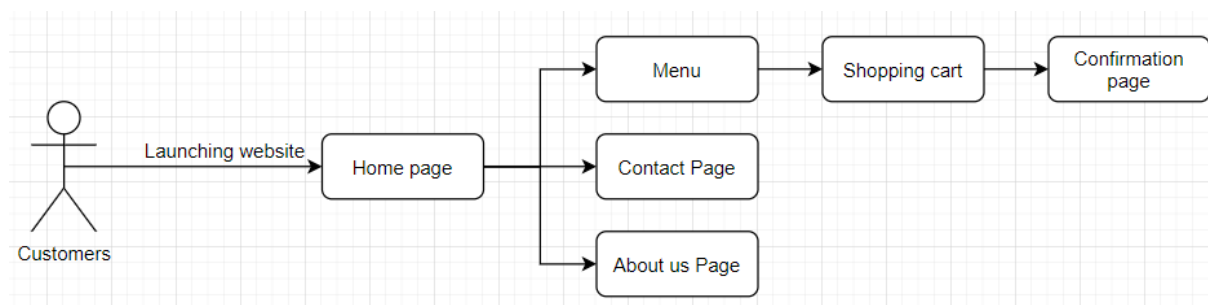


Figure 4.15. Customer access pages.

4.5 Improvements made during development?

While developing the frontend, we came across another few problems under the creation over the shopping cart. While planning for the shopping cart, we imagined it would be on its own Html page. Under development, we found out that it was better to put it on the menu page at the bottom anyways. Since this was done with the shopping cart, we also decided to do it with the user information below the shopping cart, since that as well was planned for another Html page. This would essentially reduce the total amount of pages by two. It would, however, also simplify the ordering process for each client that would be using this website.

4.6 The website

Each page has an included navbar at the top which allows the user to navigate thru the website. The navbar lets the customers navigate to the restaurant menu the contact page and the about us page. The home page will also, be on each of these pages, should the customers decide to return to the home screen.

When the website it initially launched. The user will be directed to the home screen. On the home screen, the users will find some written text and a link that directs the user to the menu page. This page is used mainly for informative purposes. The written text will be decided by the restaurant owners, as it is a representation of their restaurant. Figure 4.16 shows how the home page looks like.

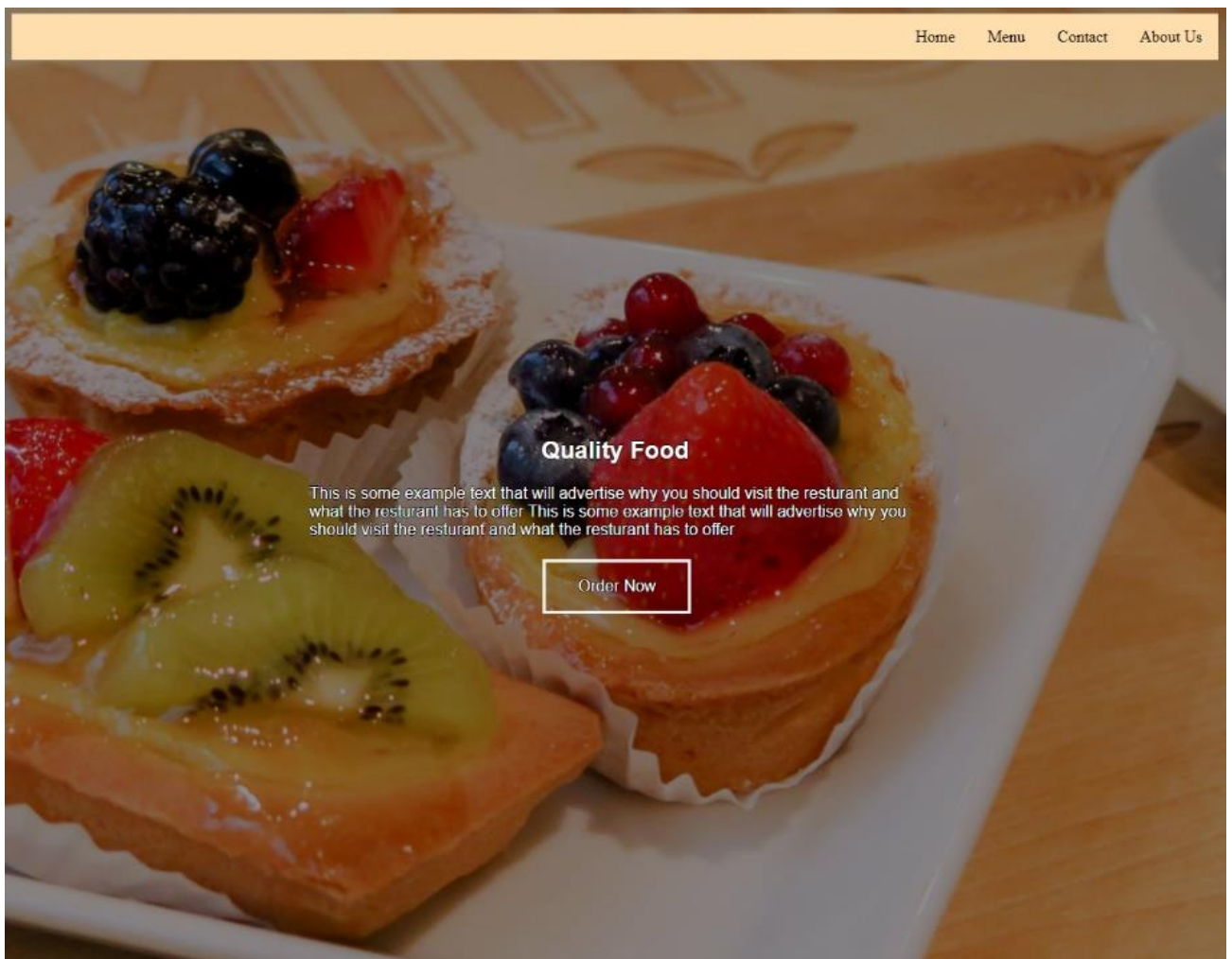


Figure 4.16. Home screen.

As the user navigates towards the menu, either by clicking the “order now” button in the middle of the home screen or menu thru the navbar selection. The screen will change to show the menu items. In the menu the website will display each menu item that the restaurant has previously chosen to input in one of their admin pages. This will display the product with all the inputted details for each. The details for the product can be found by clicking the info tab. This will then display a price for the product in small, medium, and large, along with the description of each menu item. The description of the cake is expected to also include the allergies of the cake. Each menu item also includes its own add to cart button. Clicking this button will place the selected menu item into the shopping cart.

At the top-right of the menu page, right under the navbar there is an additional button. This button has the image of a shopping cart and will direct the user towards the shopping cart functions for this page. Figure 4.17 shows an image of the menu below.

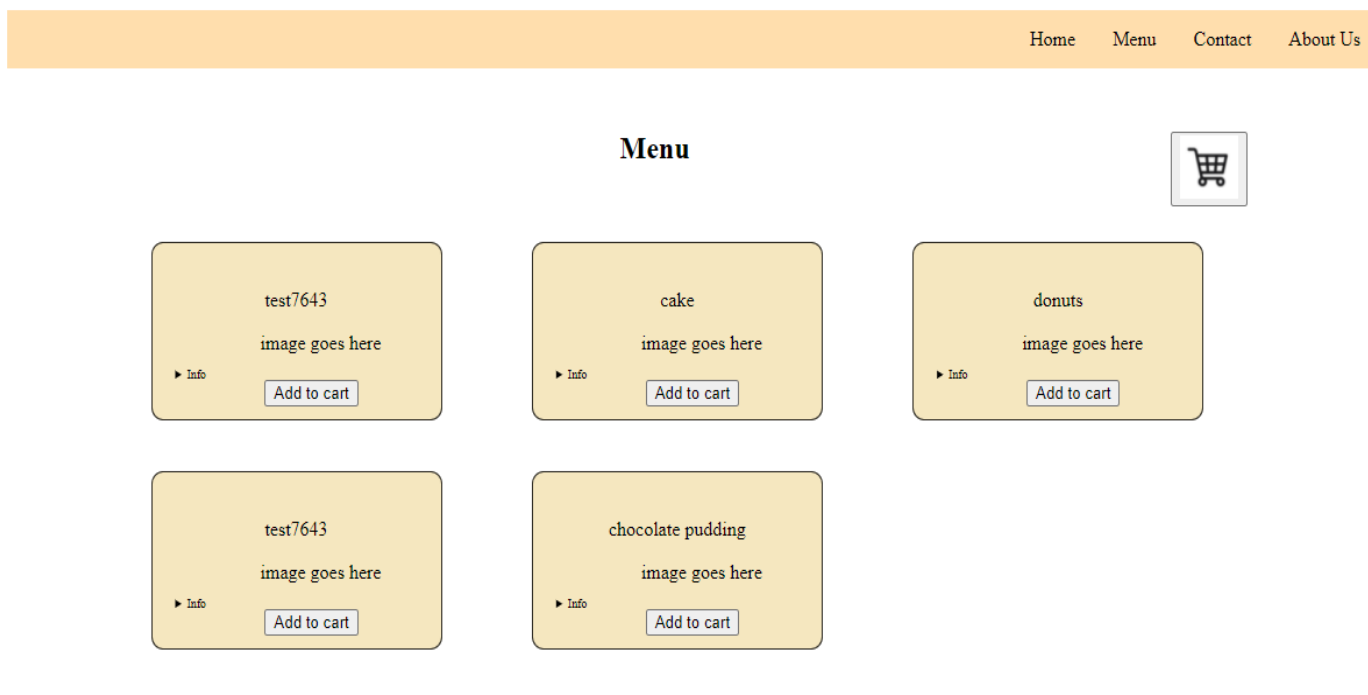


Figure 4.17. Menu

The shopping cart is located on the same document and can be accessed by clicking the navigation button. The page will then scroll down to where the shopping cart begins. Each added product from the menu will display into the shopping cart. For each item added, there are possibilities of selecting the price, where the total will also change depending on the selection. If the customer should change their mind, they are able to remove the item by clicking the button with the image of a trashcan.

An array was used in storing the data fields found in the shopping cart. It was then displayed under the heading in the shopping cart. Each array field included menu item name, size options of small, medium, and large, a price label and a trash can button. The array fields are not sent to the database unless the purchase is sent in. If the page is refreshed or changed the options will not be remembered otherwise.

For a product to be registered correctly the customer will fill out their details. The details will include first name, last name, phone number, and email address. The customers will also be able to select a pick-up-date along with providing comments with the orders, should this be necessary. When all the details are filled out the purchase button can be clicked, registering the order and sending it to the backend. It is then stored into the database where the restaurant owners will be able to see it on their own “admin” page for orders. Figure 4.18 is shown below. It shows the shopping cart with some example products added.

CART

Item	Size	Price	
test7643	Small	10 Kr	<input type="checkbox"/>
cake	Large	88 Kr	<input type="checkbox"/>
donuts	Medium	20 Kr	<input type="checkbox"/>

Total: 118 Kr

First name:

Last name:

Phone Number:

E-Mail Address:

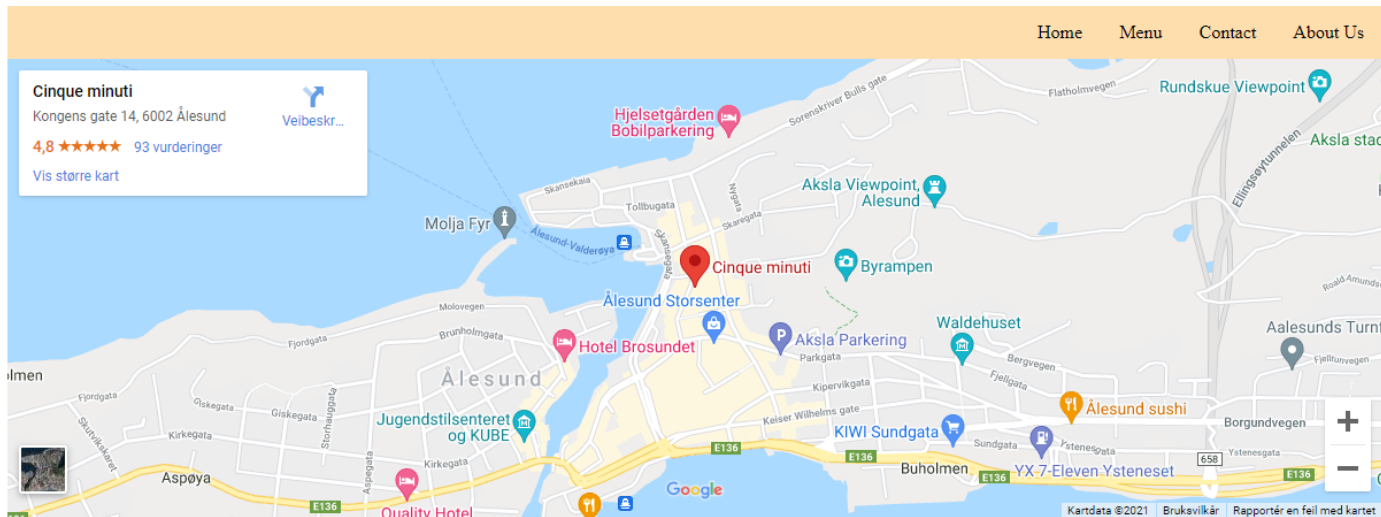
Choose preferred pickup date:
dd . mm . åååå

Additional Comments:

PURCHASE

Figure 4.18. Shopping cart

The navbar also includes a link to a page for the contact information and about us page of the restaurant. The contact page, shown in figure 4.19, provides the customers with a google map location of the restaurant along with the email and phone number. If they want more information of the restaurant there is a link included for the Facebook and Instagram of the restaurant.



Contact

Email: dragievasv@gmail.com

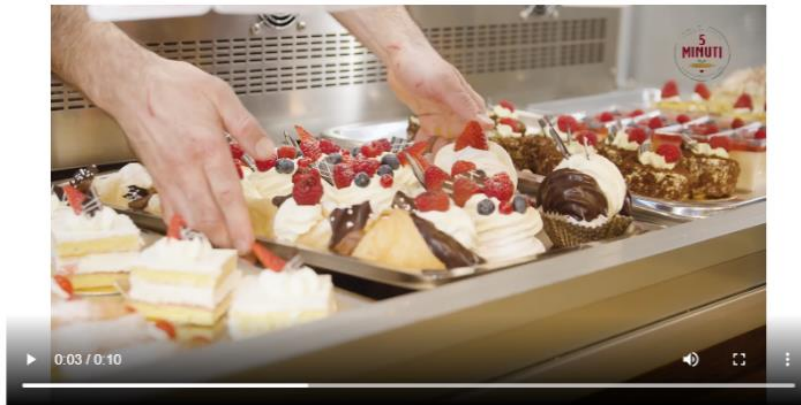
Phone: [+47 459 52 682](tel:+4745952682)

[Our Facebook page](#)

[Our Instagram page](#)

Figure 4.19. Contact

The about us page was added per request of the restaurant owners. They wanted to display a video that they had previously created along with some basic information about themselves. Currently, the group is still awaiting the information the clients want to post on this page but have the video added.



About Us

Figure 4.20. About us

Those are the only functions the customers will be able to see for this website. For security reasons the restaurant owners will have to enter /logIn.html as the last of the URL to access the log in screen.

The log in screen for this website is done very basic, only requesting a username and password. These are predefined according to what the restaurant owner wants them to be during deployment. Figure 4.21, shown below, shows the logging in screen, used by the administrators.



Administrator

Username:

Password:

Figure 4.21. Log in screen.

When log in is authenticated thru the backend and is successful the “admins” will then be directed towards the orders page. The navbar will then include three additional options for the admins. The orders page, the add product page and edit menu page.

On the order page, the restaurant owners will see each purchase order created by the customers. This will include the name, number, pick-up-date along with total price of order, for each customer. There will also be an info tab where they can see what was ordered and the size of the order along with any customer comments that was labeled on creation of the order. They will also have a way to select status of the item listed on the page. This will help in process of orders, and so that an order is not processed twice on accident.

Orders			Home			
Orders	Edit Menu	Add Product	Home	Menu	Contact	About Us
Current Orders						
Name	Phone Number	Pickup date	Price			
name	Phone number	Date	price	Info		
Stig	42623465	2021-04-25T14:54:34Z	undefined	Info		
John	555-66-777	2021-04-25T14:58:50Z	undefined	Info		
Tor Odinson	43567656	2021-04-25T15:02:57Z	undefined	Info		

Figure 4.22. Order page.

The add product page lets the admins define the name and description along with the price for each item in small, medium, and large for each menu item added. They will also be able to select an image for the products as well. Once all the input fields are defined, the “admins” can then push the add product button. This will display the product for all the customers along with displaying the product into the edit menu page. Below is an image over the add product page.

Orders Edit Menu Add Product Home Menu Contact About Us

Logged In

Product Name:

Description:

Price Small:

Price Medium:

Price Large:

Select image:
 Ingen fil valgt

Figure 4.23. Add product page.

As of version 1, the edit menu page, lets the administrators delete products, with hopes of editing menu items in later versions. Because of lack of time this was done this way. The admins will have to first delete an item and then add it again to make changes to that menu item. The edit menu page is identical to the menu page but differs in that it has a delete button instead of an add to shopping cart button, as this is not needed for the “admins”.

This page also gives an overview over the Menu, so that the admins do not have to go to the menu section every time they want to see a menu item. It also provides them with the inputted information for each item.

Edit Menu

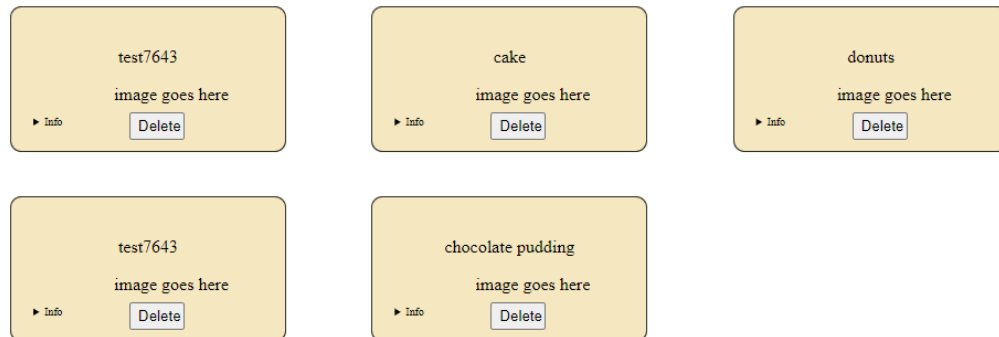


Figure 4.24. Edit menu page.

With respects to the project implementation, we succeeded in the task of being able to put out a product and succeeded in being able to view that product, all while being on a secured log in. We provided the clients with the possibility of ordering from the menu along with viewing content about the restaurant. Customers are also able to view selected items in the shopping cart before confirmation.

4.7 Deployment

The site was deployed on Digital ocean using a ubuntu virtual machine, on this ubuntu machine we installed nginx, MySQL and java to run the different parts of our project. For the backend part of the project, we ran it using the jar file created by springboot. For the static files, nginx was used to serve these files to any connecting clients. The database was deployed using mysql and was set up to work with the backend.

4.8 How different theoretical basis were used.

This section explains how the theoretical basis from chapter 2 were used. It explains the results of each theory and how it became a result.

4.8.1 Authentication

Authentication [\(2.6\)](#) was used in this project first by defining a single password and username the clients will use to log in. If they do not enter that exact password with the corresponding username then they will not be prompted to view the data, the customers have entered.

Customers' orders hold sensitive data, such as their names and phone numbers. For an additional way to secure the website, there will be no visible way to see where the login screen is. The client will rather have to type in the /login.html at the end of their URL to gain access to the login screen. With the right username and password, the client will then be "authenticated" and have access to the database thru their visual page.

4.9 How we used each technology

This section explains how each technology was used in the project. A link to the technology description is also provided in each subsection.

4.9.1 Postman

Postman [\(3.3.1\)](#) was used in terms of testing. This let us check for responses given primarily in the creating of the backend. We also used it to get the response from order list and menu list when we were retrieving them from the backend.

4.9.2 Github

Github [\(3.3.3\)](#) was used as a version control which allowed us to push changes and merge work done by both students. During the project, we had separate branches we would work on. This reduced the amount of merge conflicts we would normally expect to have. Version control for our group let us work on code and share it when tasks were finished or needed "another pair of eyes" for help. Without version control the project would have hit some major setbacks.

4.9.3 GitKraken

In this project, Gitkraken [\(3.3.4\)](#) was used, daily, as the interface where we would be able to see each other's branches and commits. It was also used during merging over work to the developer branch, after certain tasks were finished.

4.9.4 Diagrams.drawio

With the diagrams [\(3.3.5\)](#), our group drew the website and each page that was involved in it. It was used as a “rough draft” over the final project. It first detailed what links the clients would find on the navbar and the page that each link included. It allowed us to easily view and determine what was necessary when we showed these diagrams in all meetings with the clients, in the planning phase. After we had a visual representation over the whole webpage and agreed with the clients over how it would look, we were then able to start creating these Html pages based on the design of the page.

4.9.5 Gantt

Gantt diagrams [\(3.3.6\)](#), for our group, show a visual timeline over certain tasks in a project. This helps us by giving a visual representation over expected timeline each task has. It also helps keep track of what tasks needs to be done and when they are planned to be done. Gantt will also give a visual representation over any potential delays and further estimations based on those delays.

4.9.6 Trello

Trello [\(3.3.7\)](#) is the software we used for our Gantt diagram. We used trello, combined with the extension elegantt to make the diagram. As tasks were needed, our group would create them. They would then end up in new tasks in the start of the list. As an assignment was being worked on, we would move the assignment over so that the other person knows that assignment is already being worked on. Since we were only two in the group this process eventually got narrowed down and each person already knew which tasks were being done anyway. As a task was completed, we would move the task to the finished section.

4.9.7 W3schools

Our project benefited from the W3Schools [\(3.3.8\)](#) website because we were able to use this mainly for HTML, CSS, and JavaScript. W3schools.com provides us with many helpful tutorials. We were able to utilize some of the basic methods such as the grid example. This let

us define each grid and style it thru CSS. We also go ideas on how to implement the dropdown boxes and styles for buttons using this website.

4.9.8 Zoom

Zoom ([3.3.9](#)) was used for the sprint meetings every other week. Since all we needed was a link to join, this made it easy for the clients and supervisor to drag us all into a meeting. This allowed us to communicate thru voice and video with. It also allowed us to share the screen and show work being done on the website.

4.9.9 Discord

Our project group has used discord ([3.3.10](#)) since the group was created. We created a server called Bachelors 2021. This allowed us to create text channels where we link all shared work, as well as a talk channel, a voice channel, and a help channel. The help channel allows us as students directly contact our supervisor. This simplified things since we were all working from home. We would, also, not need to wait for emails as much with the discord app.

4.9.10 Spring Boot

Spring ([3.3.11](#)) boot was used for the backend to implement the REST API. This was implemented using a controller that implements repositories to interact with the database.

4.9.11 Digital Ocean

Digital ocean ([3.3.12](#)) was used for the deployment of the website and database. By renting a “droplet” with ubuntu running, we deployed the static web pages, the spring boot project and the MySQL database on this ubuntu machine.

4.9.12 Nginx

Nginx ([3.3.13](#)) was used with our digital ocean ubuntu server to deploy the static part of our website running the html, css and javascript files and serving them to users connecting to the server.

4.10 How each programming language was used

This section explains how we used each programming language specifically in the project. It includes a link to the description of the technology.

4.10.1 Html

In this project, Html [\(3.4.1\)](#) was used to structure the content of our website. The aboutUs.html, addProduct.html, contact.html, editMenu.html, logIn.html, menu.html and order.html are the seven different html pages in the project, separated in its own file location. Each page of the website has a unique structure to it.

4.10.2 CSS

CSS [\(3.4.2\)](#) was used in the actual design of the website. Using CSS, we were able to add color and style to each page. This is where we tried to mimic the appearance over what we drew out in the preplanning.

4.10.3 JavaScript

JavaScript [\(3.4.3\)](#) was used in this project for the dynamic elements created on the frontend page. With the use of JavaScript, our project becomes interactive on the user side. All menu items and order items for the administrator, are html tags in JavaScript file. The navbar is also generated by functions places in the onload of the body tags in html. JavaScript functions also use the fetch and XMLHttpRequest to get information thru the backend.

4.10.4 Java

Java [\(3.4.4\)](#) was used in the backend to communicate with the database. Java was also used in creating the rest controlled in addition to authentication, configuration, and entity of the project.

4.10.5 MySQL

MySQL ([3.4.5](#)) was used in the creation of the database. With the usage of MySQL, we were able to detail the tables in which we would be storing data in the database.

4.10.6 JQuery

JQuery ([3.4.6](#)) was used, for one function, in the start of the project. It was then, removed since it was no longer needed. Originally, it was used for the navigation bar found on every page. This way we would not have to make a copy-paste, the code for the navigation bar on each page. However, the navigation bar was then made thru JavaScript and the link and function from jQuery was thereafter, removed.

5 DISCUSSION

The discussion section explains some of the limitations and problems that we faced during the bachelor's project. It also explains what can be done if the project is further developed.

5.1 Limitations

The overall project faced some limitations, due to corona. Our group chose to do everything online, from home. While this was not a critical limitation due to the work being software developing, it also did not allow for any high level of social interaction between the group. We did not once use the facilities at school during the project, even though it was a conscious and very responsible choice, the project group made. All the meetings, with both client and supervisor was done online, except for one. We had one evening meeting at Cinque Minuti. This was a very successful meeting, since it was the first time, we felt everyone understood the meaning of our planning design. We were also offered some small cakes and pizza slices and got to interact with our clients by drawing out with marker the plan and how the website would work.

Another limitation was the sprint meetings. We had preplanned these sprint meetings which was approved by the supervisor, project group and client. After about 5-6 of these meetings the clients stopped showing up. This limited the feedback we would get towards the end of the project. It also prevented us from putting the design and the articles they write themselves on the main page and the information they would have liked to post on the about us page of the website.

Payment method was yet another limitation in the developing of the project. Due to time being scarce, this was not developed. This limits the functions of the website, however it was not in the original specification of the project. It will only process requests made by the customers and they would then have to pay at a later stage, physically in the restaurant.

5.2 Problems that arose under development

During the development, our project group faced a few problems. The problems that came underway were usually minor and mostly in the developing stages of the project. Most of these problems were fixed within a day or two. Some of the problems included:

- Preplanning the frontend
- Fixing issues with scaling
- Refactoring from rowMappers to JPA
- Using both the XMLHttpRequest and fetch

While the project group was planning the frontend, we had some delays trying to explain how the overall process would work. We had to go thru the backend and frontend, with the clients. For them not knowing a lot about IT, this process took a bit longer ~~then~~than expected. Since they had also just briefly given an idea of the project, we helped them analyze different criteria they would need and help them throughout the thought process. This ended up taking more time than planned as well but turned out to be a great learning process for us as developers. We were just glad they were engaged in the project as much as we were.

While developing the frontend, our project group decided to use a details and summary tag for the expanding functionality of the menu. The problem with this was that it expands everything on the same column as the button expanded. This essentially creates the problem that the information is not expanded as well. Instead, the box gets bigger with nothing to view. The other products must manually be clicked to expand. It was also used in admins page. Although the problem did not occur here because there is only one item per column.

During development it became increasingly clear that our existing solution with jdbc and rowMappers had too many limitations to work effectively with our solution and that refactoring to JPA would be simpler than trying to make it work our with our existing rowMappers. It was when we started working on an order with a list of order_details inside and the usage of a one-to-many relationship for this list that we realized JPA would be easier. With the help of our supervisor we managed to refactor the code in a few days.

Using the XMLHttpRequest and the fetch were used by two different group members. That is why we did not just use one of them. Problems that occurred using these happened when we tried to retrieve and send data back and forth from the backend to the frontend. All these problems were fixed, but it made the developing processes take a bit longer than originally planned. These problems essentially caused delays in the time plan. However, they had to be fixed for the functionality of the website. In the end, we learned a lot from using these API requests.

5.3 What can be improved or added?

At the start of the bachelors' project, the project group, the clients and the supervisor spent about three sprint meetings, planning the overall project. A lot of ideas were given to fully understand the scope of the project. Some of these ideas, were really good ideas but they also drastically increased the scope of the overall project, and there were uncertainties over the outcome of these ideas. This section details some of the ideas that were considered for the project. These ideas would be a good addition to the website if the website would be further developed.

During the preplanning phase of the project, the restaurant owners mentioned a chat function with customers. This was declined by the supervisor along with the group since it would be too big of a project to do in the time frame we had and the complexity of what they already wanted was high. Since we had already lost a team member early this semester, we would have to simplify the website. The outcome of this function, though, would have been neat and can be added at a later stage. It would also be a good addition for the restaurant in that it lets customers get rather quick response to different concerns. The restaurant would also have to put more time into using the website and check it more frequently in case of message popups.

For later versions of the website, if the website would be continued, the edit function should and would be created. This was thought out and planned by the group members but not implemented because uncertainties over the time it would take to make. Editing products was also deemed unnecessary as well for the website to function, in the end. The editing would allow the "admins" to alter the information they have previously put on each item, instead of

deleting it and remaking it just to alter the name or price of a product. This would simplify adding products to the website.

The design over the website such as colors and font types could be improved. Under development, we wished we had more feedback from the clients for the design and layout over the website. The color scheme was best imaged towards the colors they use in the restaurant already. There were also talked about maybe having some sort of outer boarder to each product item. This would just change the border design of the menu items.

Another thing that could be implemented, is a payment method. However, there were many uncertainties behind the payment aspect of the project and was therefore not implemented as of version 1. We had a general thought over using Vipps for the payment, since it does not require it to be built into the website. The client would then have to update each order after payment confirmation thru vipps. There would have to also be some sort of payment feedback to the customer. This is also just an idea of how to do it without having it built into the website. There are, however, many other solutions to this. This would essentially give the restaurant security in that it guarantees payment for an order before it is started on.

A notification for the administrators when they receive an order, would also be of huge benefit for them. Currently, the administrators have to regularly check the website for updates on orders. If they would be notified by noise or a popup when the orders come in, they would spend less effort in checking the website.

6 CONCLUSION

The goal of the project has been to develop a website. This website has had focus over the clients' needs and the purpose in which they will use it. It fulfilled the purpose of the plan at the start of the semester for a digital platform, in which their customers can order food.

Although it was designed to the purpose of selling cake, the clients can choose to add other types of food and will be displayed in the menu based on the criteria they set for each product. It was developed in a way where they can come with additional content in later versions.

In conclusion, the project has been a way to develop ourselves as engineers. We have faced problems and solved them accordingly. Our project group has been able to reuse a lot of technology that we have previously learned about, in school, and utilized them to a useful extent in the project. Our group has learned to face work related obstacles and get through them. We have gotten to, first-hand, experience the way developers work together with client and their needs above our own. This has all been a tremendous work experience, that we both have developed ourselves from.

The project overall, turned out to be a success. There is, however, room for improvements, should it be developed further. Ideas that are not implemented can be further developed. It will be a good addition to the restaurant, in that it does its designed job, to let customers preorder larger cake.

7 REFERENCES

- Association for Project Management. (u.å). *What is a Gantt chart?* Retrieved 22.04 from <https://www.apm.org.uk/resources/find-a-resource/gantt-chart/>
- Christina Kopecky. (2020, Oct 15). *What is jQuery? Learn the most popular JavaScript library.* Educative. Retrieved 02.05 from <https://www.educative.io/blog/what-is-jquery-javascript-library>
- Codecademy. (u.å). *What is REST?* Retrieved 28.04 from <https://www.codecademy.com/articles/what-is-rest>
- Computer Hope. (2020, 02.06). *Draw.io.* Retrieved 28.04 from <https://www.computerhope.com/jargon/d/drawio.htm>
- Devon Delfino and Grace Dean. (2021, Mar 24). *What is Discord? A guide to the popular group-chatting app.* Retrieved 04.05 from <https://www.businessinsider.com/what-is-discord?r=US&IR=T>
- Google Developers. (2019, 01.05). *Working with the Fetch API.* Retrieved 21.04 from [https://developers.google.com/web/ilt/pwa/working-with-the-fetch-api#:~:text=The%20Fetch%20API%20is%20a,Origin%20Resource%20Sharing%20\(CORS\).](https://developers.google.com/web/ilt/pwa/working-with-the-fetch-api#:~:text=The%20Fetch%20API%20is%20a,Origin%20Resource%20Sharing%20(CORS).)
- Guru99. (u.å). *What is Java? Definition, Meaning & Features of Java Platforms.* Retrieved 22.04 from <https://www.guru99.com/java-platform.html>
- Hack Reactor. (2018, October 18). *What is JavaScript Used For?* Retrieved 22.04 from <https://www.hackreactor.com/blog/what-is-javascript-used-for>
- Herawan. (2020, March 4). *What is MySQL: MySQL Explained For Beginners.* Hostinger Tutorials. Retrieved 22.04 from <https://www.hostinger.com/tutorials/what-is-mysql>
- Howe, S. (2014). *Learn to code HTML & CSS : develop & style websites.* New Riders.
- Hussnain Fareed. (2019, Oct 11). *What is GitKraken? Why Students Should Use GitKraken?* Retrieved 27.04 from <https://hussnainfareed.medium.com/what-is-gitkraken-why-students-should-use-gitkraken->

[cebc1f36c187#:~:text=What%20is%20GitKraken%20about%3F,work%2C%20commits%20and%20so%20on.](#)

Ian. (2016, May 21). *The 3 Types of Relationships in Database Design*. Retrieved 27.04 from <https://database.guide/the-3-types-of-relationships-in-database-design/>

Jacob Sharir. (2020, Mar 24). *How to use Postman for API Testing Automation*. BlazeMeter. Retrieved 27.04 from <https://www.blazemeter.com/blog/how-use-postman-manage-and-execute-your-apis>

Javainuse. (u.å). *Spring Boot Security + JWT Hello World Example*. Retrieved 29.04 from <https://www.javainuse.com/spring/boot-jwt>

Juliano Rabelo. (2021, August 14). *Foreign Key*. Techopedia. Retrieved 27.04 from <https://www.techopedia.com/definition/7272/foreign-key#:~:text=A%20foreign%20key%20is%20a,establishing%20a%20link%20between%20them>

KORBIN BROWN. (2019, Nov 13). *What Is GitHub, and What Is It Used For? How-To Geek*. Retrieved 22.04 from <https://www.howtogeek.com/180167/htg-explains-what-is-github-and-what-do-geeks-use-it-for/>

Mark Stephens. (2016, September 29). *Is moving NetBeans to Apache good or bad news?* Retrieved 27.04 from <https://blog.idrsolutions.com/2016/09/is-moving-netbeans-to-apache-good-or-bad-news/>

MDN contributors. (2021, Feb. 24). *XMLHttpRequest*. MDN Web Docs. Retrieved 22.04 from <https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest>

Michiel Mulders. (2019, September 16). *What is Spring Boot?* Retrieved 03.05 from <https://stackify.com/what-is-spring-boot/>

Oracle. (u.å). *What is Java technology and why do I need it?* Retrieved 22.04 from https://java.com/en/download/help/whatis_java.html

P.Christensson. (2018, July 13). *Authentication*. Retrieved 23.04 from <https://techterms.com/definition/authentication#:~:text=In%20computing%2C%20authentication%20is%20the,actually%20you%20accessing%20the%20website.>

Salman Saleem. (2019, March 27). *What Is DigitalOcean and Why Should You Select It for Your Web Hosting?* Retrieved 18.05 from <https://www.cloudways.com/blog/what-is-digital-ocean/>

Trello. (2021, February 5). *What is Trello?* Retrieved 22.04 from

<https://help.trello.com/article/708-what-is-trello>

What Is Nginx? A Basic Look at What It Is and How It Works. (2021, March 11). Kinsta,.

Retrieved 18.05 from <https://kinsta.com/knowledgebase/what-is-nginx/>

Wikipedia. (2021, April 22). *JSON Web Token.* Retrieved 27.04 from

https://en.wikipedia.org/wiki/JSON_Web_Token

William Antonelli. (2020, Nov 18). *What is Zoom? A comprehensive guide to the wildly*

popular video-chatting service for computers and smartphones. Retrieved 28.04 from

<https://www.businessinsider.com/what-is-zoom-guide?r=US&IR=T>

ATTACHMENTS

As informed by the supervisor, the prerequisite project report, the work log and the source code are the only attachments in this report. The source code along with all the project files are included in a .zip folder. To access the Administration side of the website, make sure the Url ends in /HTML/logIn.html, and use the username: javainuse along with the corresponding password: password.

Attachment 1	Prerequisite project report
Attachment 2	Work log
Attachment 3	Source code