Bachelor Project

Czech Technical University in Prague

F3

Faculty of Electrical Engineering
Department of Computer Science

Design and implementation of e-commerce platform for home confectioners

Pavel Paklonski

Supervisor: Ing. Božena Mannová, Ph.D.
May 2021

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Paklonski** Jméno: **Pavel** Osobní číslo: **483795**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra počítačů**

Studijní program: **Otevřená informatika**

Specializace: **Software**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Návrh a implementace B2C e-commerce platformy pro prodej výrobků domácích výrobců cukrovinek

Název bakalářské práce anglicky:

Design and implementation of B2C e-commerce platform for confectioneries

Pokyny pro vypracování:

Navrhněte a implementujte aplikaci pro prodej výrobků domácích výrobců cukrovinek. Proveďte analýzu požadavků budoucích uživatelů aplikace a rozhodněte, jaké z nich jsou vhodné k implementaci. Definujte scénáře použití aplikace. Na základě získané informace navrhněte backend architekturu aplikace a její uživatelské rozhraní tak, aby motivovala budoucí uživatele k její použití. Zdůvodněte účelnost použití vybraných technologií. Při konstrukci aplikace dbejte na možnosti její dobré škálovatelnosti a rozšiřitelnosti. Implementujte prototyp aplikace pro platformu Android nebo iOS. Ověřte, jestli naimplementovaná aplikace odpovídá business požadavkům uživatelů. Vypracujte testovací strategie a vyhodnoťte funkčnost aplikace v definovaných scénářích.

Seznam doporučené literatury:

1. https://dspace.cvut.cz/discover?scope=10467%F3641&query=eshop&submit=&rpp=10
2. Pressmann R. S.: Software Enegineering,

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Božena Mannová, Ph.D., kabinet výuky informatiky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **04.03.2021** Termín odevzdání bakalářské práce: **21.05.2021**

Platnost zadání bakalářské práce: **19.02.2023**

Ing. Božena Mannová, Ph.D. podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	prof. Mgr. Petr Páta, Ph.D. podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

. Datum převzetí zadání

Podpis studenta

Acknowledgements

First of all, I really want to thank all the teachers who shared their knowledge with me during the 3 years of my bachelor's studies. It is a great honor for me and even though it was not easy at times, it was definitely worth it.

Also, many thanks to my supervisor, Ing. Božena Mannová, Ph.D., for useful advice and assistance throughout the writing of this thesis. May your positive mood remain with you for many years and fill your loved ones with joy.

And finally, to my parents, for their unconditional love, faith in me and the greatest support that gives me strength in difficult moments. It would be a great happiness for me to always see you smiling, healthy and loving each other.

Declaration

I hereby declare that this thesis represents my own work which has been done after registration for the Bachelor's degree at Czech Technical University, and has not been previously included in a thesis or dissertation submitted to this or any other institution for a degree, diploma or other qualifications.

Prague, May 21, 2021

Abstract

What is the main culmination of every celebration? Whether it is a birthday party with friends, or Christmas with family, weddings, or corporate events?

In the end there always comes a moment when a big beautiful cake is brought and smiles spread on people's faces. Almost everyone likes to be pampered with a sweet dessert.

There are also people for whom baking is their big hobby and would like to take the opportunity to sell their homemade sweet products and thus please others.

The purpose of this work is to create a e-commerce platform, thanks to which hundreds or thousands of home confectioners could find their future customers, make money, develop their business and make someone a little happier.

Keywords: confectioner, customer, web application, Java, Spring Framework, React, Material-UI

Supervisor: Ing. Božena Mannová, Ph.D.
Department of Computer Science
FEE CTU in Prague,
Karlovo náměstí 13,
12135 Praha 2

Abstrakt

Co je hlavním vyvrcholením každé oslavy? Ať už je to oslava narozenin s přáteli nebo Vánoc s rodinou, svatby nebo firemní akce?

Na její konci přijde chvíle, kdy se přinese krásný velký dort a na tvářích zúčastněných se rozzáří úsměvy. Téměř každý se rád rozmazluje sladkým dezertem.

Jsou také lidé, pro které je pečení jejich velkým koníčkem a rádi by využili možnost prodeje svých domácích sladkých výrobků a tím potěšili ostatní.

Účelem této práce je vytvořit e-commerce platformu, díky níž by stovky nebo tisíce domácích cukrářů mohli najít své budoucí zákazníky, vydělat peníze, rozvíjet své podnikání a udělat někoho o něco šťastnějším.

Klíčová slova: cukrář, zákazník, webová aplikace, Java, Spring Framework, React, Material-UI

Překlad názvu: Návrh a implementace e-commerce platformy pro domácí cukráře

Contents

Figures

Tables

Chapter 1

Introduction

A few months ago I wanted to try baking a Molehill cake, or, in Czech, Krtkův dort. This is an incredibly delicious German cake with chocolate crust, bananas, and whipped cream.

I went to the nearest supermarket and bought a ready-made set of flour, milk powder, and baking powder. For the cake, I also needed cream, eggs, and some butter. After mixing it all together, I expected to receive a masterpiece, but the story had a very different ending, and as a result, I messed the cake up. I decided to make amends and order the cake on one of the most popular food delivery platforms in the Czech Republic. That was particularly disappointing as the cake was not there. I started to search for homemade confectionery on Instagram, and at that moment, an idea came to my mind...

1.1 Motivation

My motivation is that some of my friends, as well as parents of friends, have a hobby in addition to their main job, which brings them a small income every month. They bake cakes for children's holidays and cupcakes for an upcoming corporate party, make salted caramel for Christmas and marshmallows for Valentine's Day, and much more.

Customers, in comparison with classic confectionery shops or retail chains, appreciate in home confectioners their very high-quality personal service, the ability to reach an agreement on the appearance, ingredients, or price of confectionery. Home confectioners, on the other hand, appreciate the flexibility and the ability to earn in the comfort of their own kitchen. Moreover, since many of them like to bake, they often devote more free time to it.

From conversations with them, I also noticed that confectioners often do not know how they can expand their audience of customers. At the same time, customers know only a few people who could help them obtain the desired sweets, and most often these people are their friends or relatives.

1.2 Purpose

The purpose of this bachelor's thesis is to create a common e-commerce platform for confectioners who make homemade confectionery for sale and customers who want to order the confectionery for personal consumption.

The content of this thesis is a description of the whole process that I went through from creating an idea and starting work on this bachelor's thesis to its publication and the final deployment of a web application.

More details,

in the second chapter of this thesis, I talk about the effect of sugar on the human body, analyze data on sugar consumption in the world and in particular in the European Union, conduct research on how the confectionery market has developed in the Czech Republic over the past 4 years,

in the third chapter of this thesis, I conduct a business analysis of a web application, determine its target audience, describe what roles the users of the web application will be divided into, and also collect their user stories for further implementation. Based on the stories, I also draw up use case diagrams for each role,

in the fourth chapter, I define what the architecture of the future web application will be, and also analyze what software technologies I can use for the implementation based on various opinions, surveys, as well as my own knowledge and skills that I acquired during my bachelor's studies,

in the fifth chapter, I talk about the development process of the backend and frontend, the difficulties that I encountered in doing so, and also attach screenshots of the current state of the web application,

in the sixth chapter, I talk about how the web application was tested, what tools were used, and what errors were identified.

and in the final chapter, I summarize what results in this bachelor's thesis I have achieved, as well as what improvements may be in the project in the future.

Chapter 2

Issue Description

2.1 Health Effects of Sugar Consumption

Since ancient times, people have loved sweets. Our primate ancestors evolved to seek out sweet foods for high-energy content to increase the chance of survival when food was scarce. However, today, when food is available in all qualities and characteristics, people still consume large amounts of sweets. Why is this happening?

The answer to this question lies in the principle of the human brain. When a human consumes sugar, the brain releases dopamine and serotonin. It is hormones that boost a human's mood, and then, stimulate the areas of the brain associated with reward [1]. For example, the same process occurs with drug use.

Various studies and scientific articles show different effects on the human body - on the brain, on the skin, on mental health, and the effects are both positive and negative.

Scott Edwards, in his 2016 study Sugar and the Brain for Harvard Medical School, writes that glucose, a form of sugar, is the primary source of energy for every cell in the body. Brain functions such as thinking, memory, and learning are closely linked to glucose levels and how efficiently the brain uses this fuel source. If there is not enough glucose in the brain, for example, neurotransmitters, the brain's chemical messengers, are not produced and communication between neurons breaks down.

However, he goes on to say that even though our brain needs glucose, too much glucose in the human body has a negative effect. As an example, he writes about a study carried out by a team of scientists at the University of Montreal and Boston College in 2009. It linked excess glucose consumption to memory and cognitive deficiencies [2].

In another brain imaging study, scientists found that increased sugar intake, which also leads to higher blood glucose levels, reduced the ability to process emotions in healthy young adults [3].

Sugar is found in our diets in a variety of ways:

- naturally built into the structure of foods such as fruits and vegetables;
- naturally present in milk and milk products;
- as 'added-sugar', which refers to sugars and syrups added to foods and drinks during processing and preparation;
- as 'free sugars', which refers both to added sugars, like sucrose or table sugar, and sugars naturally present in honey, syrups, and fruit juices. Most free sugars consumed are added to foods and drinks. Free sugars do not include sugar that is naturally built into the structure of foods or to sugars naturally present in milk and milk products [4].

In 2015, the World Health Organization released a guideline that defines an excessive intake of sugar as more than 10% of the daily calories per day for both children and adults [5].

2.2 Sugar Consumption in the World and the European Union

Despite the predominantly larger number of scientific articles that describe the negative effects of sugar on the human body, I decided to study how sugar consumption has changed recently, not only in the world but also in the European Union. For the research, I used data from the German business platform Statista, which specializes in market and customer data [6].

Below is a graph of the change in total sugar consumption worldwide from 2009 to 2020.

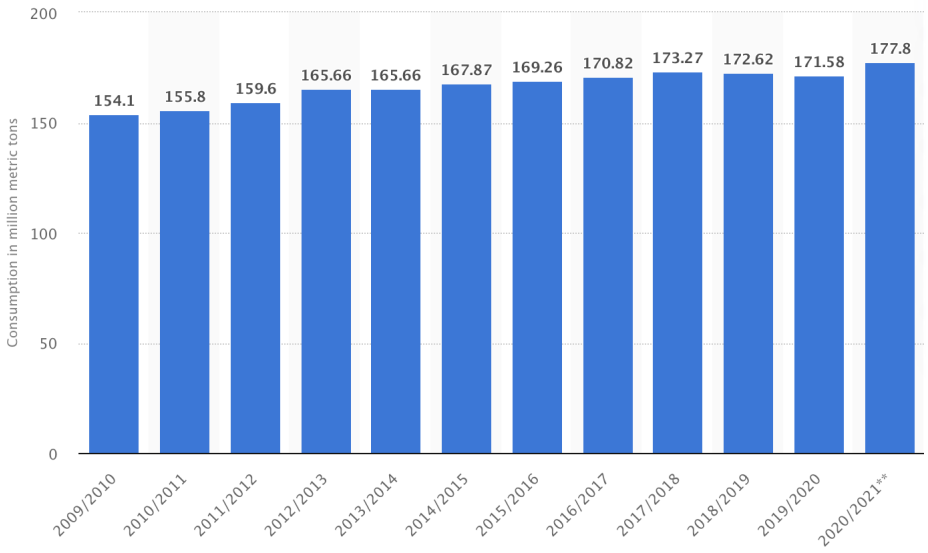

Figure 2.1: Total sugar consumption worldwide from 2009/2010 to 2020/2021 (in million metric tons) [7]

As follows from the graph above, the world growth in sugar consumption from 2009 to 2020 is from 154.1 to 171.58 million metric tons, which is 11.34% of the 2009 value. Further growth is also expected to reach 177.8 million metric tons in 2021. The researcher M. Shahbandeh associates such growth with the increase in world trade and the improvement of agricultural technologies, due to which the price of sugar on the world market is getting lower every year [7].

Next, I looked at which countries in the world consume the most sugar. The European Union came in second with 18.6 million metric tonnes in 2019/2020, behind only India with 27 million metric tonnes of sugar.

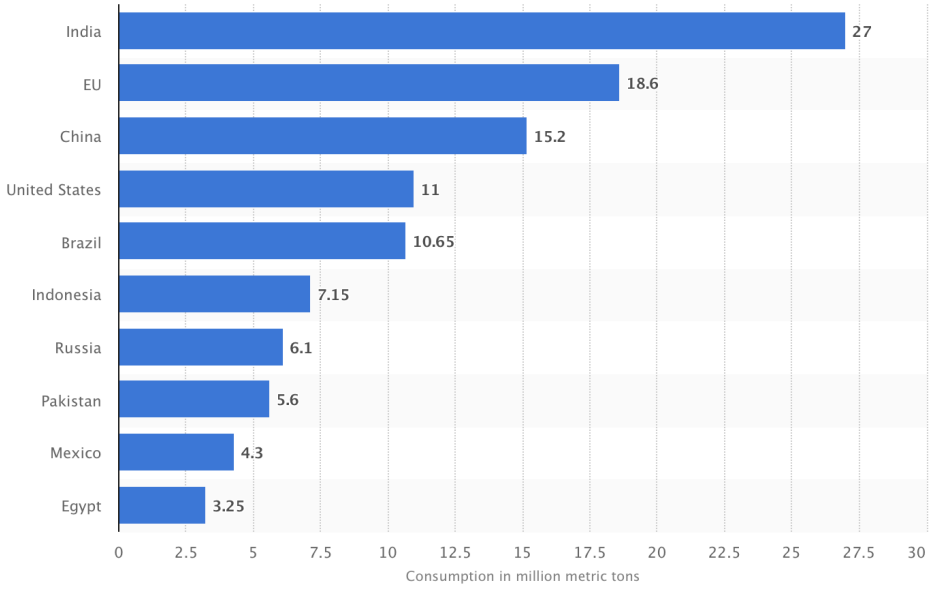

Figure 2.2: Sugar consumption worldwide in 2019/2020, by leading country (in million metric tons) [8]

According to European Statistical Office, 446,774,579 people were living in the European Union in 2019 [9]. Thus, due to the ratio of consumed sugar to the number of inhabitants of the countries that are at the top of the list, it can be stated that the consumption of sugar in the European Union is one of the largest in the world and amounts to 41.63 kg per person per year, or 114,06 g per person per day.

This data allows me to assume possible interest in the future web application from the residents of the European Union. However, is the interest in sugar consumption equal to the interest in people and organizations involved in the confectionery production? To answer this question, I studied the growth trends of the confectionery market in the Czech Republic.

2.3 Research of Confectionery Trade in the Czech Republic

To compile visual information about how the market for confectionery trade in the Czech Republic is developing, I used the official website of the Ministry of Industry and Trade. On the pages with statistical data on entrepreneurs, I found quarterly reports with information on the number of valid permits that were issued in various areas of business and divided by different regions of the Czech Republic. For the research, I needed data that includes information from quarter 1/2017 to 1/2021 in the Bakery and Confectionery section [10].

Because there are 14 regions in the Czech Republic and the table here would have turned out to be of impressive size, I decided to select 5 regions with the largest number of inhabitants and consider the remaining 9 regions in one column called Others. Population data for each region of the Czech Republic by December 31, 2019, were obtained from the website of the Czech Statistical Office.

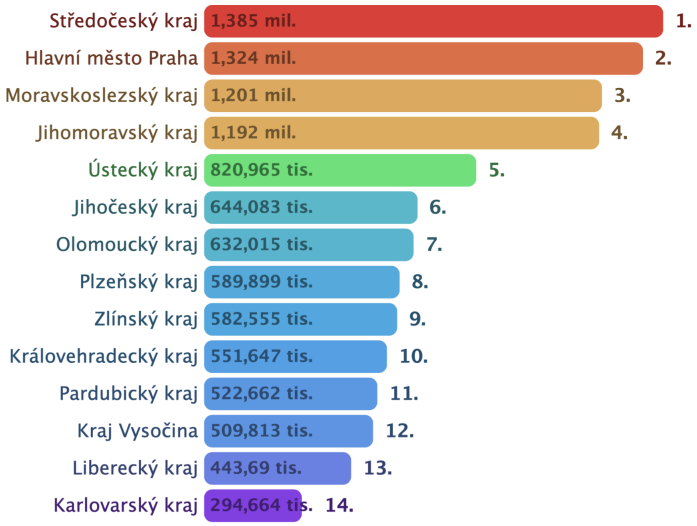

Figure 2.3: Comparison of regions of the Czech Republic by number of inhabitants [11]

As I could see from the diagram, these 5 regions are Central Bohemian, Prague, Moravian-Silesian, South Moravian, and Usti nad Labem.

Quarter	Regions						
	Central Bohemian	Prague	Moravian-Silesian	South Moravian	Usti nad Labem	Others	TOTAL
Q1 2017	1217	1387	956	1171	531	4513	9775
Q2 2017	1248	1418	979	1200	548	4646	10039
Q3 2017	1268	1432	993	1215	557	4703	10168
Q4 2017	1280	1454	1005	1227	565	4758	10289
Q1 2018	1310	1481	1021	1250	575	4815	10452
Q2 2018	1333	1521	1042	1278	584	4927	10685
Q3 2018	1346	1544	1056	1287	595	4976	10804
Q4 2018	1366	1573	1064	1306	602	5044	10955
Q1 2019	1401	1594	1069	1310	611	5138	11123
Q2 2019	1439	1633	1099	1338	617	5241	11367
Q3 2019	1463	1650	1106	1354	621	5304	11498
Q4 2019	1481	1661	1112	1374	631	5361	11620
Q1 2020	1504	1690	1119	1386	644	5435	11778
Q2 2020	1534	1732	1135	1411	665	5553	12030
Q3 2020	1552	1758	1144	1420	669	5633	12176
Q4 2020	1573	1768	1149	1441	668	5661	12260
Q1 2021	1595	1782	1156	1457	679	5726	12395

Figure 2.4: Number of valid business permits in the Bakery and Confectionery section from quarter 1/2017 to 1/2021 by regions of the Czech Republic

Based on the summary data, I drew a graph of the change in the number of valid permits over time.

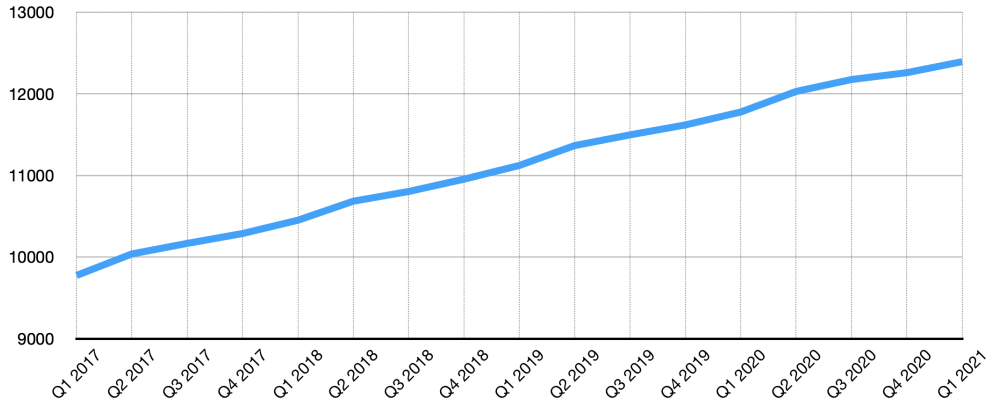

Figure 2.5: The change in the number of valid business permits

As the graph shows, the number of registered entrepreneurs in the Czech Republic in the Bakery and Confectionery business area increased from 9775 in the 1st quarter of 2017 to 12395 in the 1st quarter of 2021. The increase in 4 years has amounted to 2620 entrepreneurs, which is 26.8% compared to the index of the 1st quarter of 2017. This indicates a rapidly growing market for the production and sale of confectionery and the potential interest in the web application not only from home confectioners and their customers but also from private confectionery shops throughout the Czech Republic.

Chapter 3

Business Analysis

3.1 Overview

The purpose of the practical part of this bachelor's thesis is to build a base for future e-commerce platform that will allow other developers to improve the existing functionality, or in other words, to create an MVP.

A minimum viable product (MVP) is a development technique in which a new product or website is developed with sufficient features to satisfy early adopters. The final, complete set of features is only designed and developed after considering feedback from the product's initial users. This concept has been popularized by Eric Ries, a consultant and writer on startups [13].

To gather the requirements for the functionality of the e-commerce platform, I needed to conduct an initial analysis and determine the target audience, user roles, and their stories. Based on this data, I created use case diagram for the future web application.

3.2 Target Audience

Majority of the population consuming confectionery products globally falls in the adult age range, constituting from 16 to 60 years [14]. However, within it, there is also a category of people most actively using e-commerce platforms or services for ordering and delivering food. According to the Ecole hôtelière de Lausanne article, customers for delivery and takeaway services are mostly younger people up to 35 years old who could be students, managers, and full-time employees [15]. Together with them, a part of the target audience of the future web application is young families, possibly with children, as well as confectioners.

3.3 User Roles

3.3.1 Administrator

An administrator is the main person responsible for the content of the web application. Sometimes on various Internet resources, unwanted content may appear, incitement of inter-ethnic hatred, and insults. The function of the administrator is to remove such material, block users and other measures in accordance with the law, as well as which are agreed within the team responsible for the operation of the e-commerce platform. In addition, the administrator could add new administrators to the system if the project develops.

3.3.2 Customer

A customer is an individual who uses a web application to order or purchase confectionery. The customer can choose a specific confectioner and directly agree with him about the cooking time and ingredients of the confectionery products, about the appearance, etc, or choose from the ready-made offers of the confectioner's showcase, or leave an open request for the cooking of a confectionery, named as Wish, to which other confectioners can respond and offer their services. The customer will be able to leave a review to the confectioner so that other customers understand what kind of attitude and quality of work to expect.

3.3.3 Confectioner

A confectioner can be either a private person who bakes confectionery at home and wants to sell them to other individuals, or confectionery stores that want to expand their customer audience. Confectioners will have their own showcase for placing offers or completed orders. Also they will be able to add, hide and change the contents of their showcase, and respond to customer comments. A typical confectioner's offer will include a title, a detailed description, a photo, an estimated cooking time, and a price. Confectioners will categorize their offers to make it easier for the customer to find the right one. Confectioners will be able to respond to open requests from customers, named as Wish, in order to expand their client audience.

3.3.4 Anonymous User

An anonymous user is a user who has visited the home page of the application, but not logged into the system. If the transition to the pages of a web application happened by accident, an anonymous user should quickly understand what the web application is for. If the transition occurred intentionally, then the anonymous user should be able to register in the system for further full use of all the features of the web application, depending on the selected role.

3.4 User Stories

In the previous section, I determine that the users of the future e-commerce platform will be Administrator, Customer, Confectioner, and Anonymous User. To formulate the software features that will provide the required functionality to the client, I compiled user stories based on discussions with potential users. First, I define what a user story is.

A user story is an informal, general explanation of a software feature written from the perspective of the end-user or customer. The purpose of a user story is to articulate how a piece of work will deliver a particular value back to the customer [16].

User stories are expressed in a generally accepted structure - one sentence according to the template

«As a [persona], I [want to], [so that].»

that I also follow in the thesis.

3.4.1 Administrator

[US101] As an administrator, I want to manage the content of the web application, so that I can prevent unwanted user behavior.

[US102] As an administrator, I want to store the user's first and last name, so that I can tactfully address the user.

[US103] As an administrator, I want to store the user's email address, so that I can contact the user.

[US104] As an administrator, I want to see the user's status, so that I understand if the user is active, deactivated or stopped for temporarily any reason.

[US105] As an administrator, I want to store the user's username, so that the user can be uniquely identified.

[US106] As an administrator, I want to store the user's password, so that the user's information is only accessible to the verified one.

[US107] As an administrator, I want to store the type of user account, so that I can distinguish between the confectioner and the customer.

3.4.2 Confectioner

[US201] As a confectioner, I want to have a personal profile page, so that I can present myself to potential customers.

[US202] As a confectioner, I want to create new confectionery offers, so that I can attract new orders from customers.

[US203] As a confectioner, I want to see open customers' requests to cook confectionery, so that I can offer my services to them.

[US204] As a confectioner, I want to be able to set up discount codes or programs, so that I can attract new clients.

[US205] As a confectioner, I want to keep a history of my completed orders, so that I can use it for special purposes, for example, to improve the business strategy.

[US206] As a confectioner, I want to be able to communicate with the customer, so that I can immediately resolve issues that could arise.

[US207] As a confectioner, I want to be able to have the full information about my completed orders in print, so that I can pay taxes.

[US208] As a confectioner, I want to receive information about new comments on my profile page, so that I can quickly respond to them.

[US209] As a confectioner, I want to be able to see customer information, so that I can offer exclusive deals.

3.4.3 Customer

[US301] As a customer, I want to store my data, so that I can use it during checkout without re-entering the data.

[US302] As a customer, I want to store my address, so that I can use it to define where confectionery could be delivered.

[US303] As a customer, I want to filter the confectioners by their location, so that I can identify those who are close to me.

[US304] As a customer, I want to see a list of current confectioners' offers, so that I can choose the one that suits me best.

[US305] As a customer, I want to be able to pay for the confectionery in different ways, so that I can choose the most suitable one.

[US306] As a customer, I want to add a specific amount of confectionery into my order, so that I do not have to create a huge number of identical confectionery orders for the same confectionery unit.

[US307] As a customer, I want to be able to see the personal profiles of the confectioners, so that I can get full information about them.

[US308] As a customer, I want to read comments to read comments on confectioners, so that I can decide if I want to entrust my order to a specific confectioner.

[US309] As a customer, I want to filter the confectioners by their rating, so that I can identify the most customer-focused.

3.4.4 Anonymous User

[US401] As an anonymous user, I want to be able to log into my profile, so that I can use all features of the web application.

[US402] As an anonymous user, I want to be able to register, so that I can start using all features of the web application.

[US403] As an anonymous user, I want to be able to recover my password, so that I can enter my account if I have lost access to it.

[US404] As an anonymous user, I want to be able to view the offers of the confectioners, so that I can quickly find out whether this e-commerce platform is suitable for me or not.

3.5 Use Case Diagrams

To show the possible interactions of the user in the system depending on the role, I processed the user stories from Section 3.4. Next, I have depicted a use case diagram for each user role. Due to the fact that the diagram turned out to be large, I divided it into four parts for each role separately.

Note. Duplicate elements on different parts of the diagram are one unit!

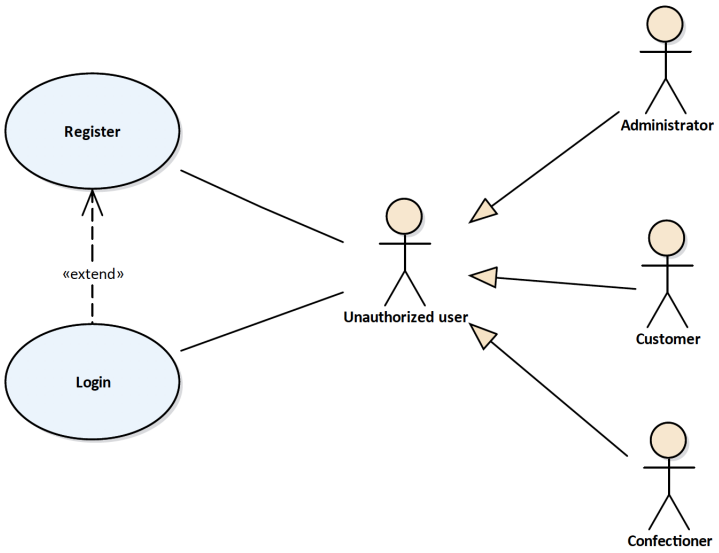

Figure 3.1: Use Case Diagram for Unauthorized, or Anonymous, User

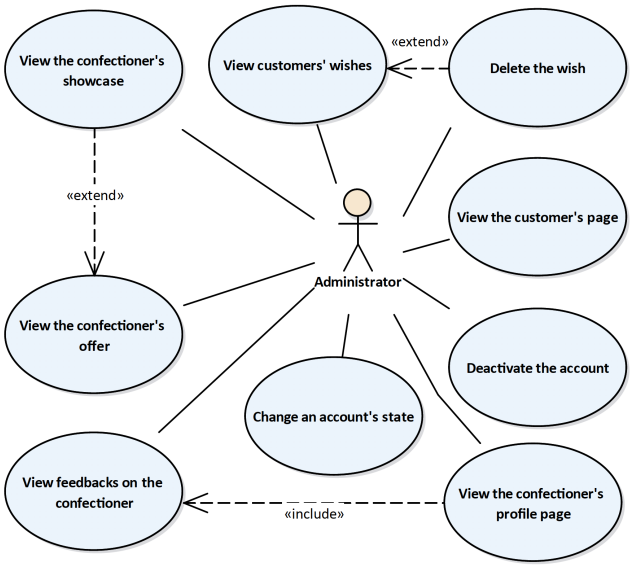

Figure 3.2: Use Case Diagram for Administrator

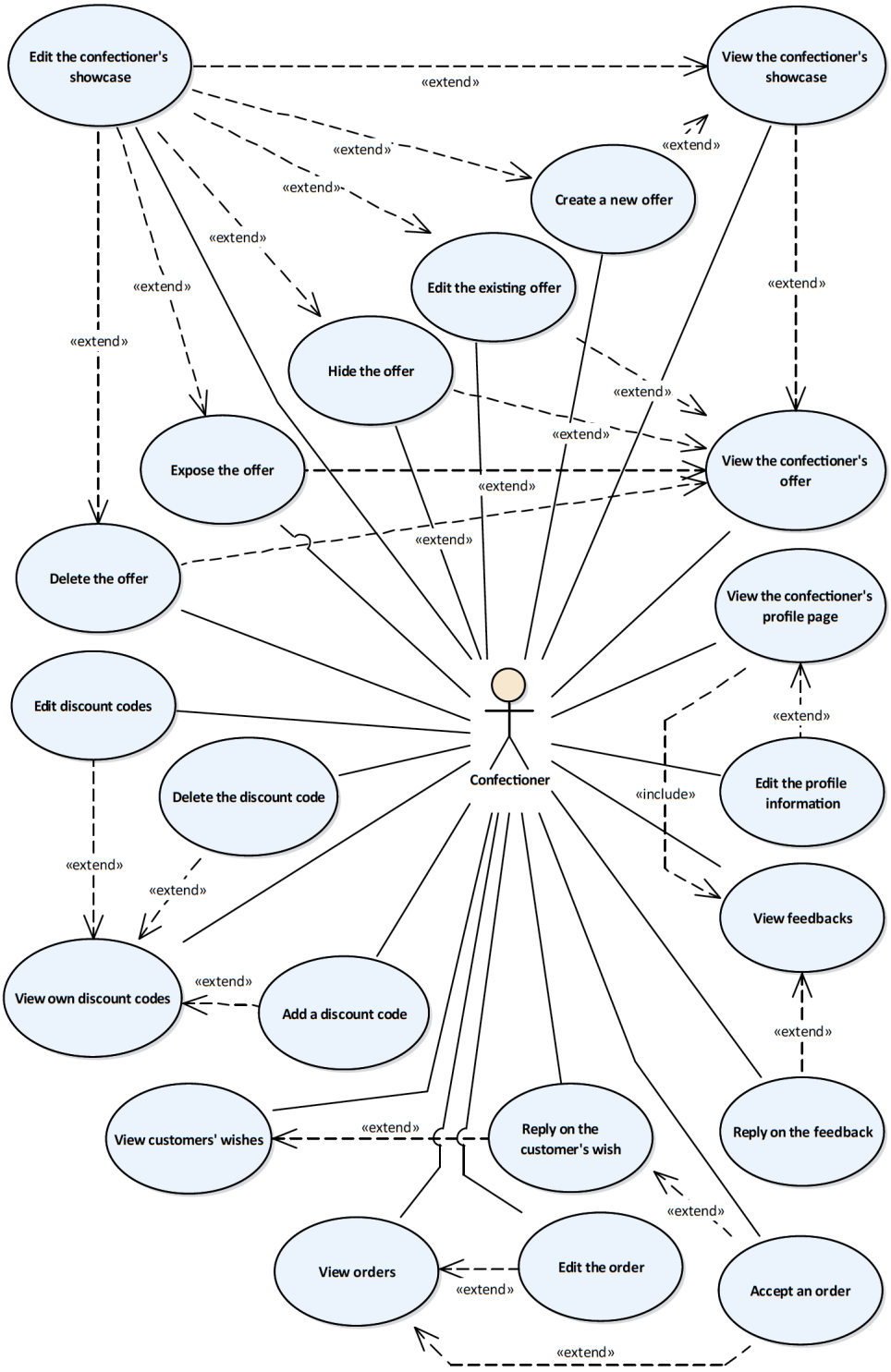

Figure 3.3: Use Case Diagram for Confectioner

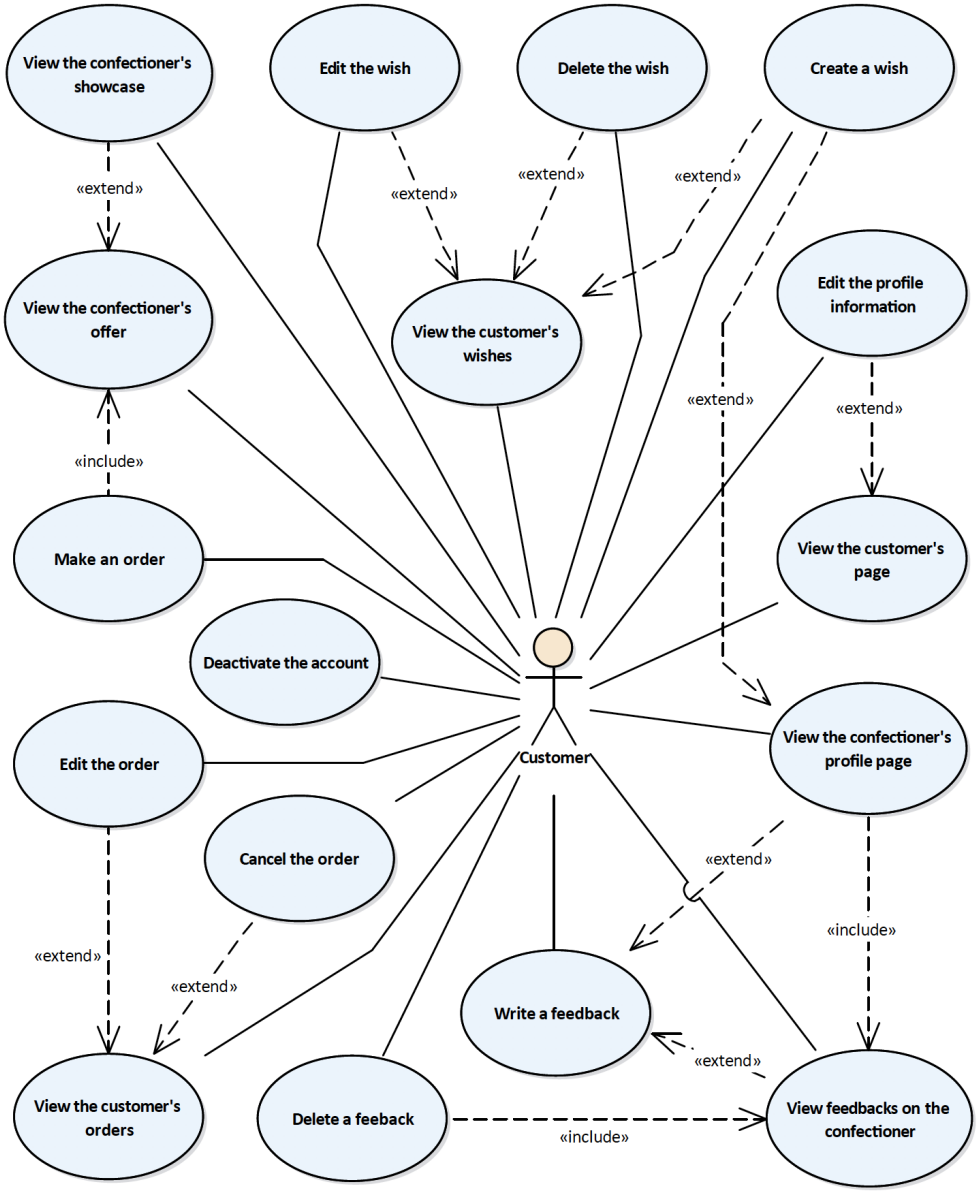

Figure 3.4: Use Case Diagram for Customer

Chapter 4

Software Analysis

4.1 Architecture

Any web application can be divided into two components: backend and frontend, or in other words, components corresponding to the server-client model.

The backend, or server-side of a web application, focuses on activities beyond the reach of the end-user. It connects the web to a database, manages user connections, interacts with other remote servers to maintain required functionality, and much more.

The frontend, or client-side of a web application, focuses on the graphical interface of a web application that the end-user interacts with directly in the browser, or in other words, it is all that the user sees.

For separating my web application into some logical parts, I chose a three-tier architecture [12]. In details, these three tiers are

- the presentation tier, or user interface;

  It is a tier, where the end-user will interact with my application. Its main purpose is to display information to and collect information from the user.

- the application tier, or business logic tier;

  Here is the heart of my application. In this tier, information collected in the presentation tier is processed. Also here, using business logic, the information received from the data tier is processed in order to transfer it to the presentation tier or to modify it to bring back to the data tier.

- the data tier, or sometimes called data access tier.

  Here is where the information processed by the application is stored and managed. This can be a relational database management system such as PostgreSQL, MySQL, or in a NoSQL Database server such as Cassandra, MongoDB.

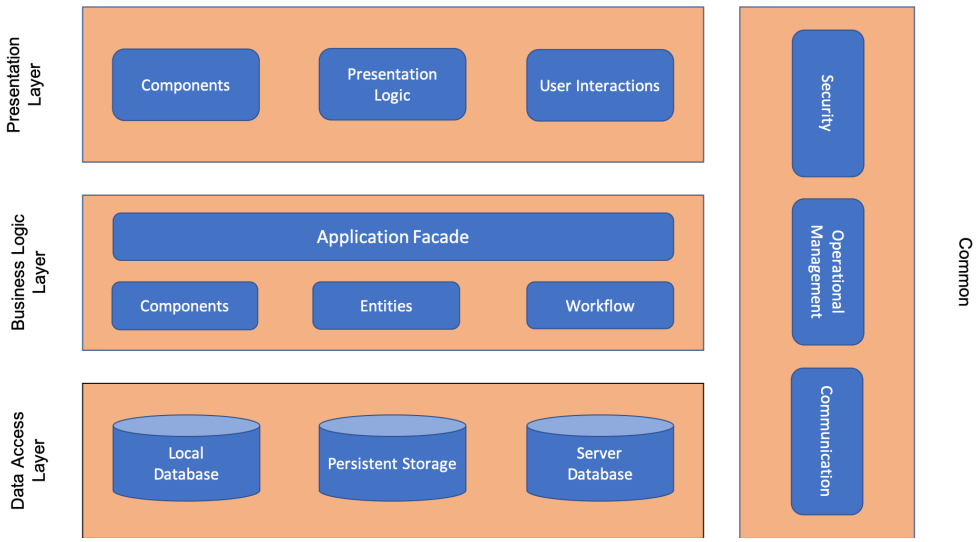

Figure 4.1: 3-tier architecture diagram

The benefits of using a three-tier architecture versus one- and two-tier architecture are that it allows

- to easily scale the web application because each tier is separate from the others;
- to be confident in the stability of the application since the failure of one of the tiers is unlikely to affect the performance of other tiers;
- to improve security because the presentation tier and the data tier cannot interact directly.

4.2 UML Class Diagram

I analyzed user stories from Section 3.4 and use case diagram from Section 3.5 in order to create a class diagram of the future e-commerce platform, and identified the necessary entities for further implementation.

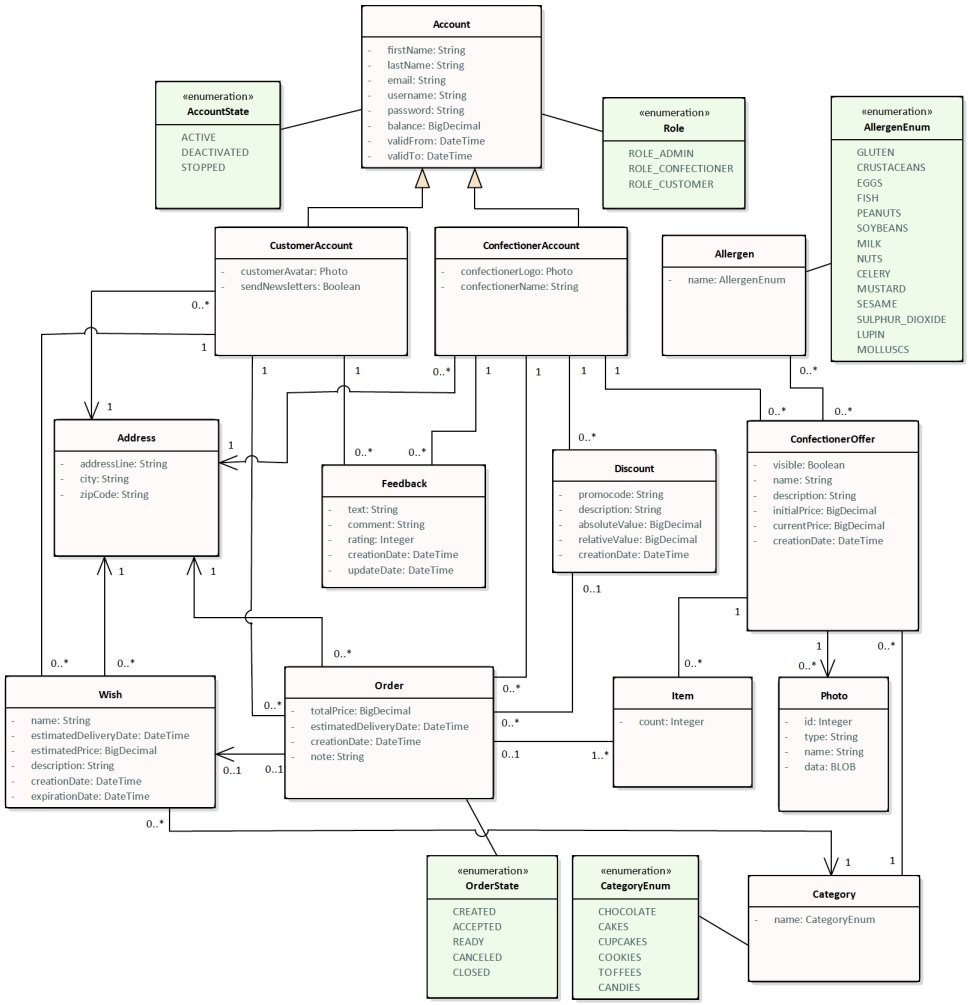

Figure 4.2: Class Diagram for the Application

In Section 3.3, four different user roles have been identified. There are Anonymous User, Administrator, Customer, and Confectioner.

Account

For all users, if they are not authorized, but registered in the system, at least two attributes must be stored in the database. There are *username* and *password*. They will also be used to authenticate and authorize the user in the system. In addition, for all users, regardless of their *Role*, it is necessary to store their *firstName* and *lastName*, *email*, *address*, *status*, and *type* of account in the database.

Address

Since the address includes three other attributes, such as *addressLine*, *city*, and *zipCode*, I decided to move them into a separate entity *Address* with a many-to-one relationship. As a result, the entity *Account* satisfies all of the Administrator user stories.

CustomerAccount, ConfectionerAccount

Further, two other entities, which represent Confectioner and Customer, are inherited from *Account*. *CustomerAccount* has an attribute *customerAvatar* to store the profile photo in addition to all the attributes from the parent *Account*. *ConfectionerAccount* stores the *confectionerLogo* and the *confectionerName* attributes to create the confectioner's professional showcase.

ConfectionerOffer

Confectioners can create an entity *ConfectionerOffer*, which is a representation of the confectioner's offer, and add it to their showcase to attract new customers. To keep the confectioner's offer informative, I added attributes *name*, *description*, *initialPrice*, *currentPrice*, and *creationDate*, into the entity.

Category

The confectioner's offer necessarily belongs to a certain *Category* for better user navigation through the web application. An enumeration *CategoryEnum* represents names of categories. In the future, they may be added or removed as the application develops.

Allergen

Also, the confectioner can add to the offer a list of allergens that the confectionery contains. The entity *Allergen* represents different kinds of allergens. In order for the customer to be able to filter offers by the presence of certain allergens, I added a many-to-many relationship between the *ConfectionerOffer* and *Allergen* entities.

Photo

The confectioner's offer will include several photos of the product. To store the files in the database, I created a *Photo* entity with the necessary technical attributes such as *id*, *name*, *type* and *data*, which is an array of binary data.

Discount

Confectioners, according to their business strategy, can come up with various promotional campaigns. For this, I created a *Discount* entity, which stores a *promocode* that the customer will use, a *description*, an *absoluteValue* if the discount is valid for a specific order amount, a *relativeValue* if the discount is expressed as a percentage of the order amount, and *creationDate*, so that the confectioner know when the promotional campaign was created.

Feedback

Customers can also rate the confectioner and write a review. For these purposes, I added the *Feedback* entity with five attributes to the class diagram. They store a *text* that is the customer's text of the review, a *comment* that is the confectioner's response to the review, *rating*, as well as the *creationDate* and *updateDate*, so that a user know when the review was written and the confectioner responded to it.

Wish

An important part of the functionality of the web application is the *Wish* entity. It represents the customer's request to bake certain confectionery products. In the request, the customer indicates the *estimatedDeliveryDate* of the product, the *estimatedPrice* that he is willing to pay for the product, a *description*, where the customer can write additional requirements, and a *name* of the request. The entity also has an *expirationDate* attribute. It represents the date after which the request will be deleted from the database if not canceled by the customer or transferred to the *Order* entity. Customers' *Wish*es are available to all confectioners registered in the system.

Order

Finally, an order that represents the *Order* entity can be created by the customer either directly from the confectioner's online showcase or as a confirmed *Wish*. The data stored inside the entity generally repeats the *Wish* entity, except that the *Order* can be associated with a *Discount*, and it is also possible to track the status of the order. Enumeration *OrderState* represents the state of the order, which the confectioner can change during processing.

4.3 Backend

The backend, or server-side of a web application, typically includes three major parts [17]:

- the server that receives requests;
- the app running on the server that listens for requests, retrieves information from the database, and sends a response;
- the database that is used to organize and persist data.

4.3.1 Database

The database is a system where the application stores and organizes all the data that is required for its functionality. When it came time to decide which type of database to use in my project, I picked between

- a SQL, or a relational, and
- a NoSQL, or a non-relational

data structures [18].

Comparison of SQL and NoSQL Databases

SQL databases are perfectly suited for storing and processing structured data, while NoSQL databases are the best solution for working with unstructured or semi-structured data. And since I structured the data of the web application into separate tables, as you can see from Section 4.2, the choice was mostly in favor of SQL database.

Also a database should be well-protected. SQL databases are more secure than NoSQL databases, which trade consistency and security for performance and scalability. To secure NoSQL database I would need to take additional steps.

But since I also have to pay attention to making the web application highly scalable, this is where the use of SQL databases imposes a limitation. NoSQL databases with their key-value stores are optimized for horizontal scaling, while SQL databases are optimized for vertical scaling. This means that if the web application needs to be expanded in the future, the scaling should be vertical, which can lead to some disadvantages.

Choosing a Database for the Project

Today the choice of SQL databases is very large. The most well-known and trusted solution is OracleDB. But since this is a very large platform for enterprise development, which requires a paid license, it does not suit me for a project. More cost-effective alternatives are MySQL and PostgreSQL.

MySQL is one of the most popular open-source SQL database managed by Oracle. The database works perfectly with structured data at the basic level. However, if the project will be scaled in the future, it will require additional paid support.

Unlike MySQL, PostgreSQL is a completely free database. It is is an object-relational, which means that all data is represented in the form of objects instead of columns and rows. PostgreSQL is an ideal solution for large systems, since it's scalable and designed to handle terabytes of data.

Since I already have experience with PostgreSQL, I decided to choose it for my project.

4.3.2 Application

Programming Language

During my bachelor's studies, I learned such programming languages as Java, C, Python.

All of them can be used to develop the server side of the application. However, due to the fact that I have the most experience with development in Java, I chose this one.

In 2020, according to the results of the Developer Survey on Stack Overflow [19], Java was voted the 5th most popular technology, eclipsed by JavaScript, HTML/CSS, SQL, and Python.

Java, like any other language, has its advantages and disadvantages [20]. For example, the advantages are

- Platform-independency (Write Once Run Anywhere);
- Object-oriented programming;
- Automatic memory management,

versus the disadvantages such as

- Poor performance;
- Verbose and complex code.

Despite its disadvantages, Java is one of the most commonly used languages in the software industry due to its platform independence, security, and support features.

Framework

To focus on the high-level functionality of the application, I had to choose a software framework. Here the choice for me personally was not so great, because the only framework I worked with when developing Java applications is Spring Framework.

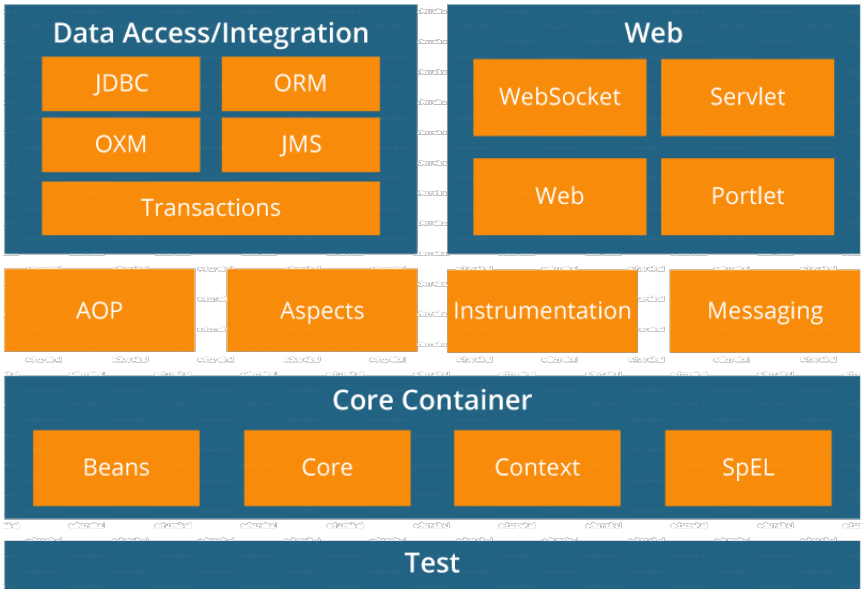

Figure 4.3: Spring Framework Architecture

However, it is exhaustive and covers a lot of features including security and configuration, which are easy to learn. Further, since it is the most popular web framework, I could find a lot of documentation and an active community on Stack Overflow.

Security

To prevent unauthorized access to data, backend access is protected by username and password. For authentication and authorization, I used the spring-boot-starter-security package from the Spring Framework and JSON Web Token (JWT). It is an open standard (RFC 7519) that defines a compact and self-contained way for securely transmitting information between parties as a JSON object.

The scenario for using a JWT in my web application is such that when the user successfully logs in using their credentials, a JSON Web Token will be returned. Whenever the user wants to access a resource, the user agent should send the JWT, typically in the Authorization header using the Bearer schema. The content of the header should look like the following:

Authorization: Bearer <token>

The server's routes will check for a valid JWT in the Authorization header, and if it is present, the user will be allowed to access protected resources [21].

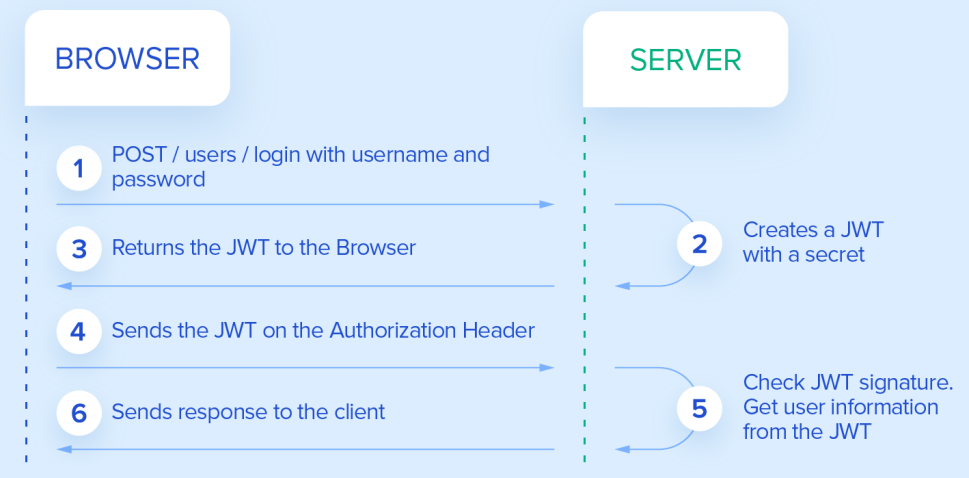

Figure 4.4: JSON Web Token working principle to authenticate the user

4.4 Frontend

The main frontend development tools are HTML, CSS and Javascript.

- HTML tells the browser what content is about, like headings, list, etc;
- CSS tells the browser how to display elements on the screen, such as colors of elements or their position;
- JavaScript is responsible for how the browser reacts to user actions.

Frontend and backend interact in two ways [22]:

- server-rendered apps;

  When a user types the URL of a specific post, the browser makes a HTTP request to the backend. The backend returns a HTTP response containing the HTML code interpreted by the browser. The only interaction between the two edges is indeed the HTTP flow.

- communication using AJAX.

  The JavaScript-powered frontend UI sends HTTP requests to the backend in order to read, create, update or delete data. The backend returns HTTP responses containing the necessary information for the UI or web application to change its state, display data to the user, etc.

For a faster development speed, and to avoid problems in the future with the scalability of the web application, the lack of boilerplate code, there are frameworks. On the Internet and from conversations with colleagues, I heard a huge number of opinions about which framework is better, for what purposes it serves, and so on.

One of the Internet resources, which specializes in software technologies, presents interesting statistics about the experience of using frameworks [23]:

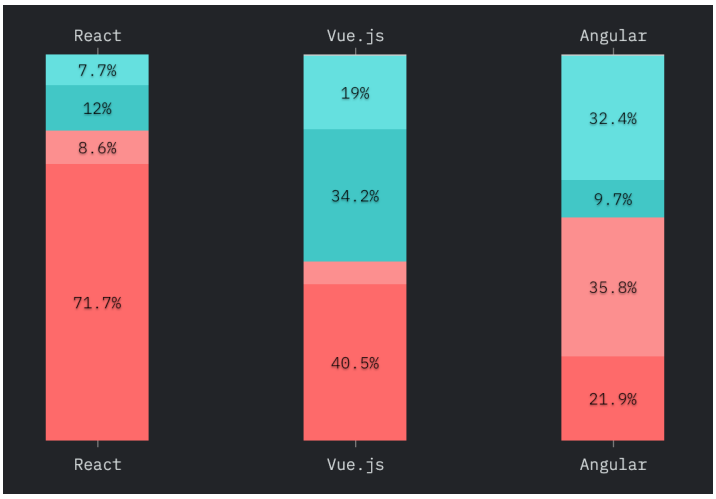

Figure 4.5: Overview of opinions on the most popular frontend technologies

The first column shows the survey data for the **React** framework:

- 71.7% of respondents answered that they have *used* React before, and *would* use it again;
- 8.6% that they have *used* React before, and would *not* use it again;
- 12% that they have *heard* of React, and *would* like to learn it;
- 7.7% that they have *heard* of React, and are *not* interested.

Fewer respondents want to use **Vue.js** again. The results of the survey about this framework can be seen in the second column and they are as follows:

- 40.5% of respondents answered that they have *used* Vue.js before, and *would* use it again;
- 8.6% that they have *used* Vue.js before, and would *not* use it again;
- 34.2% that they have *heard* of Vue.js, and *would* like to learn it;
- 19% that they have *heard* of Vue.js, and are *not* interested.

Finally, the third framework that I have also considered for use in frontend implementation is **Angular**. Its position in the survey does not look so strong because:

- 21.9% of respondents answered that they have *used* Angular before, and *would* use it again;
- 35.8% that they have *used* Angular before, and would *not* use it again;
- 9.7% that they have *heard* of Angular, and *would* like to learn it;
- 32.4% that they have *heard* of Angular, and are *not* interested.

Due to the fact that I had never built the frontend of web applications before starting work on my bachelor project, it was also important for me how quickly I could figure out the principles of creating a user interface.

Positive feedback about React framework following from the survey convinced me to explore its capabilities in more detail. It allows web developers to create components and reuse them in different parts of the application code, shows consistent performance using the virtual DOM.

However, the biggest advantage for me in terms of choosing the React framework was that it has a large community of developers on various Internet platforms. This made it easier to find solutions to difficulties encountered during the frontend development.

Chapter 5

Implementation

5.1 Backend

The first thing I did before starting the implementation of the backend, I generated the base of the future web application in the Spring Initializr [24]. This is a web service from VMware, Inc., which allows to configure Java application with Spring Framework as needed by the developer, and then upload a pre-configured project to a local computer to continue the development.

For the project, I chose Maven as a build automation tool, because I have experience with it. Maven makes very easy to configure the necessary dependencies only by copying settings from public Maven repositories and including it into pom.xml file. Also I set the project metadata and dependencies Spring Boot DevTools, Spring Web, Spring Security, Spring Data JPA, Lombok, etc.

Next, I created several packages for the future implementation code source, which I then filled with classes. Each of them is responsible for a specific area. From the beginning until the end of the implementation, the list of packages has remained unchanged:

- **config** contains configuration classes for the application, Java Persistence API and security;
- **exception** contains custom exception classes;
- **dto** contains objects that carry data between processes;
- **model** contains the representation of the database tables, which are the implementation of the data model from the UML Class Diagram from Section 4.2.
- **security** contains the implementation of the user authentication and authorization in the system according to the principle of JSON Web Token as I described in Subsection 4.3.2.

Then I split the application tier from Section 4.1 into three more tiers, which I represented in separate packages:

- **dao** contains objects to handle the details of the persistence mechanism;
- **service** contains objects to handle business requirements;
- **rest** contains controllers to handle HTTP requests to and responses from a server side of the application.

On localhost I created a PostgreSQL database which I connected to my application in Intellij IDEA. After that, I was possible to create database entities, or tables, in the model package. After I made sure that the tables were created, I created Data Transer Objects that partially duplicate the tables in the database. After that, I continued development in the dao package. Here I wrote the necessary queries for the database tables that JPA does not provide. I also added configuration files to the config package.

Further, for each entity simultaneously in service and rest packages, I wrote all the necessary logic for CRUD operations and tried to perform them through Postman. Here the first errors in implementation came to light, because the data in the JSON files did not always pass through the application to the database and back as it had to be.

After fixing the errors, I started implementing the user authorization and authentication. I spent a lot of time on it, because I needed to understand in detail how the JSON Web Token validation works in practice, how to implement filters, JWT provider, etc. Moreover, debugging was quite difficult due to the fact that the execution of the code, which is part of Spring Security, began already in the early stages.

As soon as the user authorization was successfully implemented, I began to horizontally expand my web application, that is, I began to add new functionality, new REST API endpoints for client-server interaction. As new functionality of the application was added, some errors were identified in the UML Class Diagram, due to which it has undergone changes in comparison with the initial version.

Thus, the server side of the web application was implemented and became available to the client side through controllers that include REST API endpoints for receiving requests and sending responses.

5.2 Frontend

The development of the client side of the web application was the most difficult stage for me in the process of working on my bachelor's thesis.

During the Software Engineering course, I learned how to create simple forms for CRUD operations without a complex user interface. However, in order for a web application to perform its functions and attract users to use it, these skills were not enough for me. I did some research of my own, the results of which I described in Chapter 4.4, and also consulted with colleagues who already had experience in frontend development. I was advised, in addition to the React framework, to consider using the Material-UI style of the graphical interface developed by Google [25]. It is a set of ready-made React components that web developers can customize for themselves, as well as supplement them with the necessary business logic.

Due to the fact that during the implementation process I faced many difficulties that a beginner usually faces, I decided to implement the minimum set of characteristics necessary for home confectioners, since they are the key users of the application. All the images that I used to create the user interface are from open internet sources [26][27].

Users of the web application are greeted by the main page, at the top of which there is a bar where they can select the category of confectionery they need. In the upper right corner there are navigation buttons to go to the Login and Registration pages.

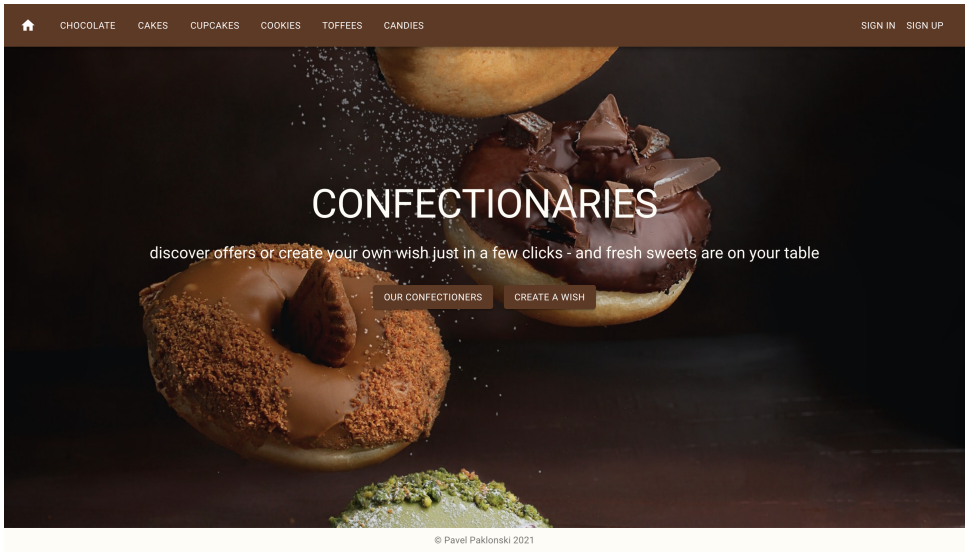

Figure 5.1: Home Page

The new user registration form consists of first name, last name, username, password and user role selection. It can be a Confectioner or a Customer.

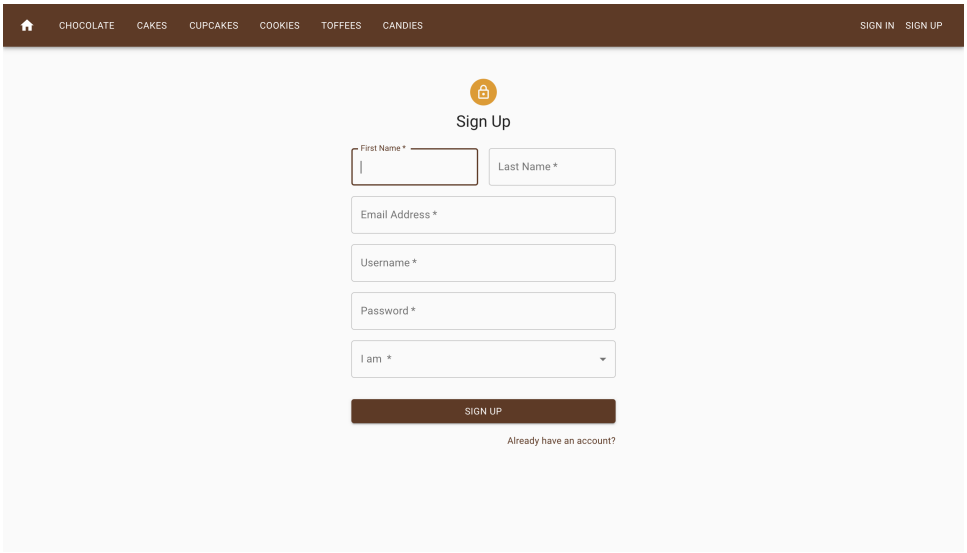

Figure 5.2: Registration Page

After logging in to the system, the confectioner has access to own dashboard. It can contain various notifications. So, for example, after registration, the web application still does not know the public name and the location of the home confectioner. Therefore, the confectioner sees a notification that the account will not be visible to customers until the confectioner fills in the required data.

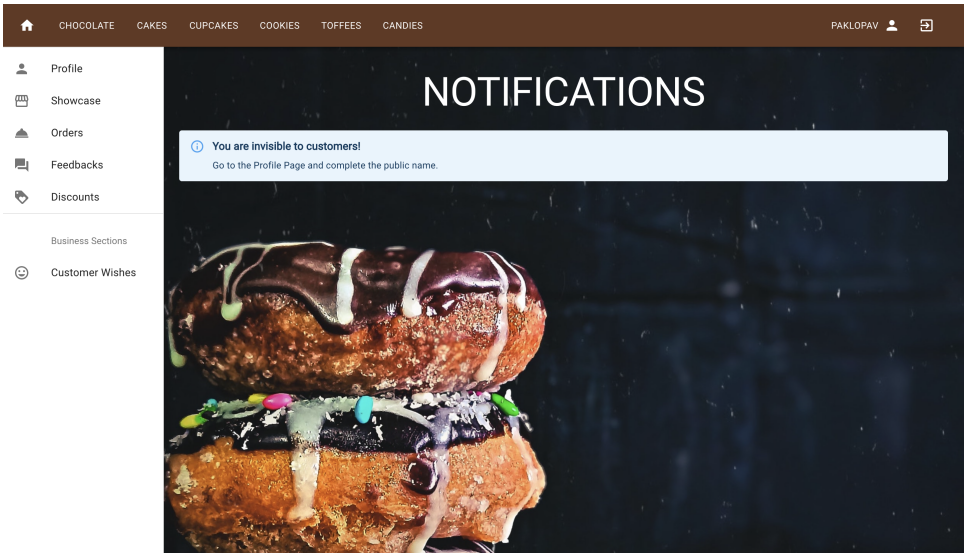

Figure 5.3: Confectioner Dashboard

In the Profile tab, the confectioner can update a personal data. Unfortunately, I was unable to implement the ability to upload an image by a user due to the fact that the server, despite all the permissions, did not allow the image to be uploaded to the database. The upload stalled at approx. 34% and the file in the database was incomplete. After a long search for solutions, for clarity, I used the saved images and decided to postpone the implementation of this feature for a future iteration of improving the project.

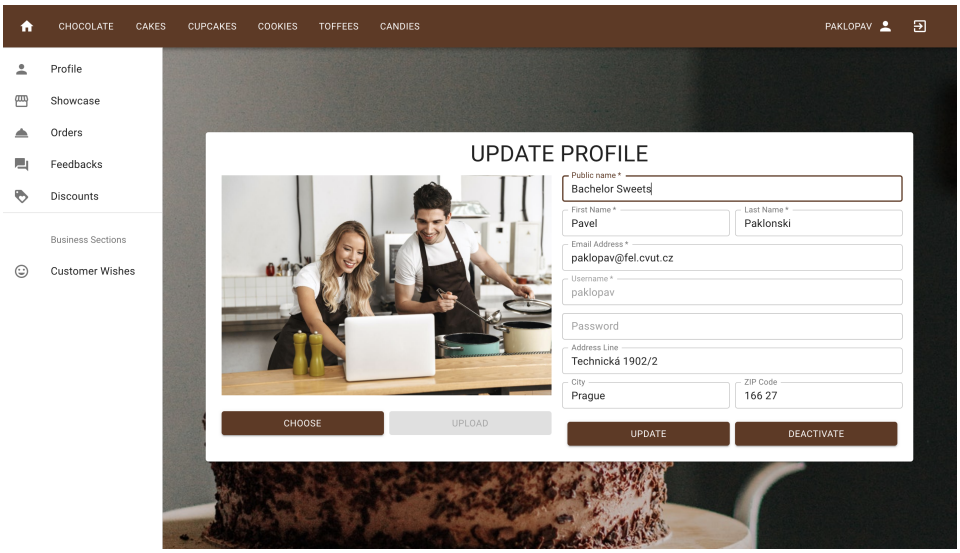

Figure 5.4: Update Profile Page

In the Showcase tab, the home confectioner can go to the page where offers can be managed. The form for creating a new offer consists of fields for choosing a photo, name, description, price, estimated cooking time, the allergens contained, and the category the confectionery falls under.

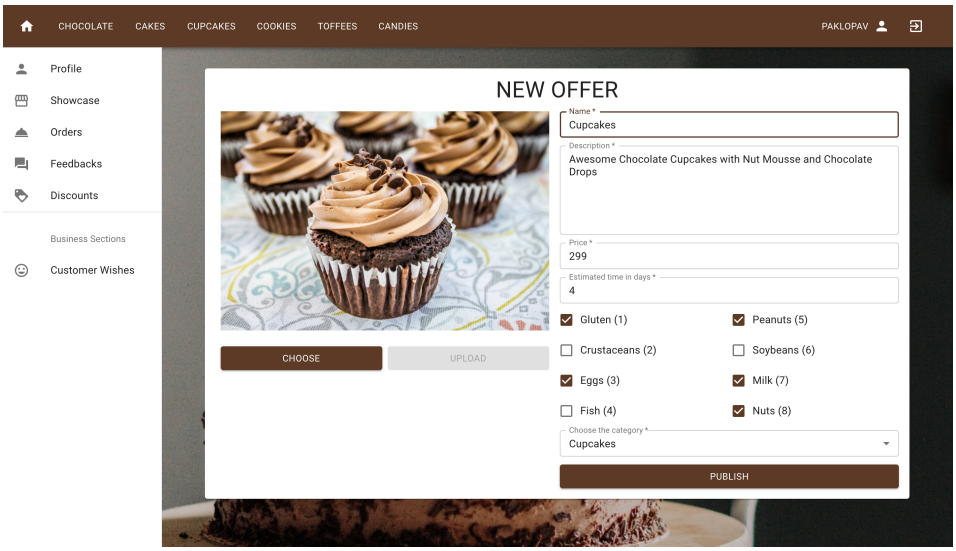

Figure 5.5: New Offer Page

After the confectionery was added by the confectioner into the web application, it appears in the confectioner's Showcase, from which it can also be removed.

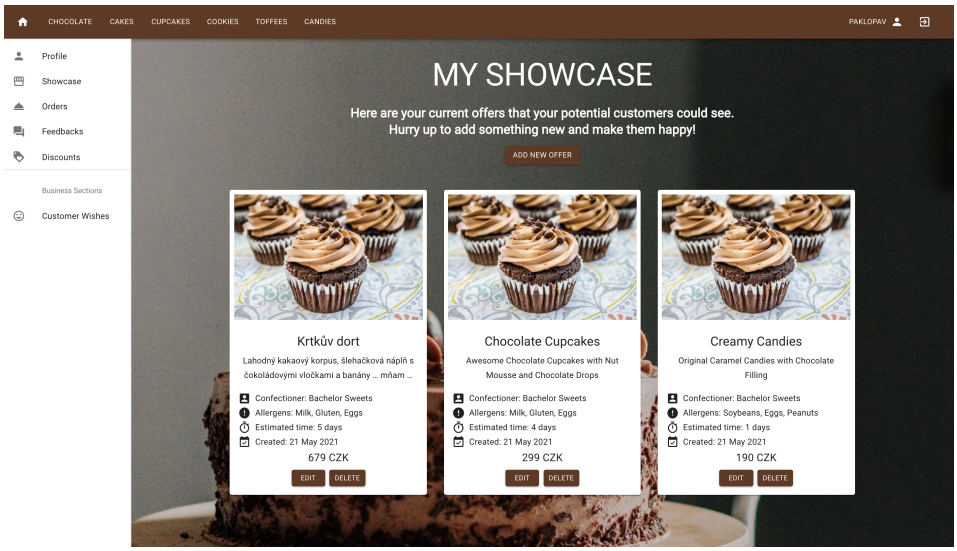

Figure 5.6: Showcase Page (zoom: 80%)

As I mentioned above, every confection falls under a specific category. On the screenshot, a page with offers of home confectioners in the Cupcakes category. For each offer, the customer can choose two options. The first one is to place an order for this confection. As a temporary solution, I came up with the fact that when a user wants to place an order, a dialog opens. There the customer enters the e-mail and the home confectioner contacts the customer to clarify the details. The second one is to discover other offers of the confectioner.

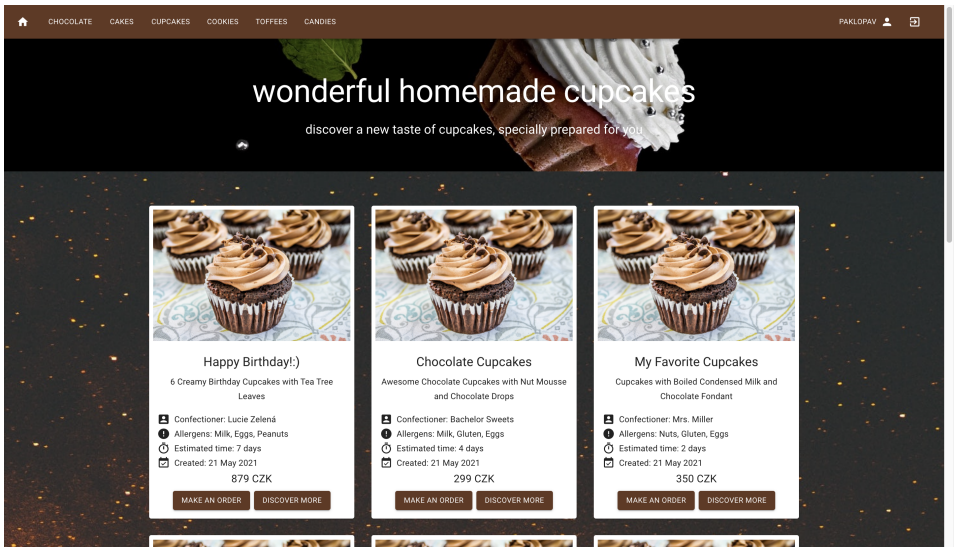

Figure 5.7: Cupcakes Page (zoom: 80%)

Here's a page with home confectioners who are registered with the web application. The customer can see where they are, find out the rating, as well as the number of published offers. There is also an e-mail, by which the customer can personally contact the confectioner.

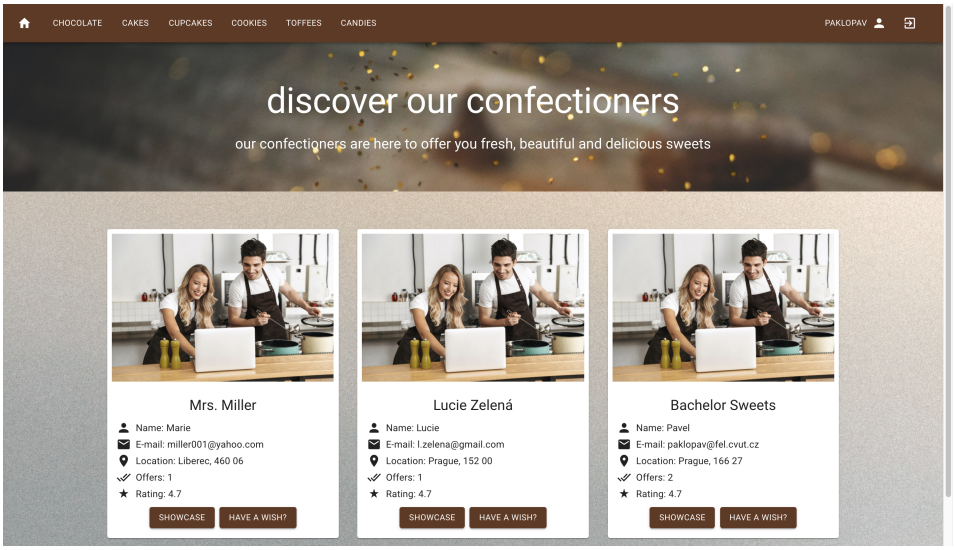

Figure 5.8: Confectioners Page (zoom: 90%)

Due to the fact that home confectioners are very creative people, my great focus has also been on the design of the web application. For the main theme, I chose dark colors with chocolate tints to make users of the web application associate with cocoa and products containing it.

Chapter 6

Testing

I divided the testing of the web application into two parts. The first is testing REST endpoints through Postman, the second is testing on end users through pre-written scenarios and collecting feedback on how the web application performs the tested features.

6.1 Postman

Postman is designed to check HTTP requests from the client to the server and receive a response from the backend. The request includes URL, method, header and body.

URL is a string to identify which endpoint the request should be sent to.

Method indicates the desired action to be performed on the identified resource. It can be POST, GET, PUT, and DELETE. Typically, the POST method is used to transfer data to the server, GET to receive, PUT to modify, and DELETE to delete data on the server side. I use each of them.

HTTP headers let the client and the server pass additional information with an HTTP request or response. In my case, for each request, except for login and registration, I use the Authorization header. There I pass a JSON Web Token that allows the server to understand whether the client that sent the request to receive certain data is eligible or not.

Finally, the body is the final part of the request. It contains directly the data for which the request is made.

To test the endpoints of the server side of the web application, I created several collections in Postman that contain a list of HTTP requests that I executed in order.

At this stage, I fixed errors with access rights to endpoints. For example, data that should have been available only to the certain confectioner was also available to other users, which is a breach of confidentiality.

Further, due to one-to-many, many-to-one, many-to-many relationships in the data model, the data that came in the request was incorrectly inserted into the database. That is, when a new row with data appeared in the parent entity, there was a link to the child entity, but there was null. This required a change in the mapping.

6.2 User Interface

To test the user interface, I initially wanted to involve a number of friends who are home confectioners, but live outside the Czech Republic. However, due to the fact that I experienced some difficulties with the deployment of client side of the web application, the testing was carried out on my friends, the 3rd-year students of the Faculty of Electrical Engineering of the Czech Technical University in Prague.

Also, due to the fact that they are software developers, they had the freedom in their behavior and could perform any actions with the web application. Thus, I was able to collect as many bugs as possible and now I have the opportunity to fix them for the next releases.

The test scenario looked like this:

1. Register in the web application
2. If registration was successful, try to register in the web application again using the same login and email
3. Log into the web application
4. Check that the confectioner account is invisible on public pages until the confectioner fills data in the Public Name field
5. Go to the Profile tab and add the missing data

   The confectioner should only become visible on public pages after completing the Public Name field.
6. Edit personal data, add the Public Name
7. Check that the confectioner has become visible on public pages and has no offers
8. Add a new offer
9. Check that the offer is displayed on the confectioner's showcase with all entered data
10. Check that the offer is displayed on the public pages of the corresponding category with all entered data
11. Edit the offer
12. Delete the offer
13. Deactivate the account
14. Check that the confectioner's account, as well as the offers of the confectioner, have been removed from public pages.

Testing was carried out by Artem Hurbych, the 3-rd year student FEE CTU in Prague, on May 20, 2021.

He writes, "During testing, I did not found significant issues. Unfortunately, some of the functionality is still unavailable. For example, it is not possible to do something as a customer. There is a lot of validation in the fields, and it prevents filling in incorrect data. However, sometimes it is enough options. For example, it is impossible to fill in the fractional part of the price or choose price currency. The website style and layout look good."

The tester pointed out that the web application has a good stability and styling, but he also drew attention to the insufficient number of features of the web application.

The reason is that I used much more time to understand the principles of the React framework than I originally planned. At the same time I am very glad that the tester paid attention to the stability of the web application, which allows me to talk about the minimal likelihood of the appearance of old errors during the further development of the project.

However, undoubtedly, all the shortcomings have been taken into account and will be corrected during further iterations of the web application development.

Chapter 7

Conclusion

The initial title of the bachelor's thesis was "Design and implementation of B2C e-commerce platform for confectioneries". In the process of working on the project, and after consulting with the supervisor, we decided to change the concept. From the B2C we decided to be closer to people with their hobbies that do not have a document that confirms their affiliation with the business. This work received a new title "Design and implementation of e-commerce platform for home confectioners".

The main aim that was defined before writing this bachelor's thesis was to create an e-commerce platform that would allow home confectioners to sell their homemade confectionery products to customers.

Market research was the first step in its creation. From it I learned that the level of sugar consumption in the world and in the European Union is increasing. This is really not the most positive trend, but sweets are such an integral part of our life that we cannot do anything about it.

The number of the confectionery shops in the Czech Republic is growing every quarter. This means that more and more active people appear in the country. They are ready to develop their business and will need various tools to promote it. And this platform can come to the rescue here.

The aim of the practical part of the thesis was to create a minimum viable product that fulfills some of the basic requirements of the earliest users. These are confectioners. Without them, the web application does not make sense. That is why, first of all, I have implemented a part of their personal account, from which they could manage the content of their profile. In order for them to receive orders, I implemented a dialog in which customers would send their contact details for further communication. I am very happy that I was able to try the frontend development. This allows me to see the software from the other side, since I had not had such experience before.

The next step in improving the web application could be the implementation of the customer profile, and interaction with the home confectioner without external tools. An internal chat, a payment system directly through the application, delivery and much more could help in this.

And most importantly, attracting new producers, customers, developers, so that every day the e-commerce platform for home confectioners becomes better and stronger.

Bibliography

[1] Joanna Stass, *WHAT HAPPENS TO YOUR BODY AN HOUR AFTER EATING SUGAR?*, [online]. [Cit. 17.2.2016] URL: https://www.independent.co.uk/life-style/food-and-drink/features/what-happens-your-body-hour-after-eating-sugar-a6879026.html

[2] Scott Edwards, *Sugar and the Brain*, [online]. [Cit. Spring 2016] URL: https://hms.harvard.edu/news-events/publications-archive/brain/sugar-brain

[3] Joel Fuhrman. *Negative Impact of Sugar on the Brain*, [online]. [Cit. 16.6.2020] URL: https://www.verywellmind.com/how-sugar-affects-the-brain-4065218

[4] World Cancer Research Fund International, *Curbing global sugar consumption*, pdf, 2015, [online]. URL: https://www.wcrf.org/wp-content/uploads/2021/01/Curbing-Global-Sugar-Consumption.pdf

[5] World Health Organization, *Guideline: sugars intake for adults and children*, [online]. [Cit. 4.3.2015] URL: https://www.who.int/publications/i/item/9789241549028

[6] Statista, Global No.1 Business Data Platform, [online]. URL: https://www.statista.com

[7] M. Shahbandeh, *Sugar consumption worldwide 2009/10-2020/21*, Statista, [online]. [Cit. 3.6.2020] URL: https://www.statista.com/statistics/249681/total-consumption-of-sugar-worldwide/

[8] M. Shahbandeh, *Global sugar consumption 2019/2020, by leading*, Statista, [online]. [Cit. 3.6.2020] URL: https://www.statista.com/statistics/496002/sugar-consumption-worldwide/

[9] European Statistical Office, [online]. URL: https://ec.europa.eu/eurostat/web/main/data/database

[10] Ministerstvo průmyslu a obchodu České republiky, *Počty živností dle oborů v jednotlivých krajích*, [online]. [Cit. 21.4.2021] URL: https://www.mpo.cz/cz/podnikani/zivnostenske-podnikani/statisticke-udaje-o-podnikatelich/pocty-zivnosti-dle-oboru-v-jednotlivych-krajich--222296/

[11] Český statistický úřad, *Porovnání krajů - pořadí krajů*, [online]. [Cit. 20. 1. 2021] URL: https://www.czso.cz/csu/czso/porovnani-kraju-poradi-kraju

[12] IBM Cloud Education, *Three-Tier Architecture*, [online]. [Cit. 28. 10. 2020] URL: https://www.ibm.com/cloud/learn/three-tier-architecture

[13] Richard Becker, *Minimum Viable Product (MVP)*, [online]. [Cit. 14.8.2020] URL: https://www.techopedia.com/definition/27809/minimum-viable-product-mvp

[14] Bhavana Thorat , Roshan Deshmukh, *Confectionery Market by Product Type*, [online]. [Cit. Oct 2020] URL: https://www.alliedmarketresearch.com/confectionery-market

[15] EHL Insights, *Key Trends in the Food Delivery Industry*, [online]. URL: https://hospitalityinsights.ehl.edu/food-delivery-industry-trends

[16] Max Rehkopf, *User Stories with Examples and Template*, [online]. URL: https://www.atlassian.com/agile/project-management/user-stories

[17] Codeacademy, *Back-end Architecture*, [online]. URL: https://www.codecademy.com/articles/back-end-architecture

[18] Liliia Harkushko, *Vital Things to Consider When Choosing a Database for Your App*, [online]. URL: https://yalantis.com/blog/how-to-choose-a-database/

[19] Stack Overflow, *Most Popular Technologies*, [online]. URL: https://insights.stackoverflow.com/survey/2020#most-popular-technologies

[20] Altexsoft, *The Good and the Bad of Java Programming*, [online]. [Cit. 9.8.2018] URL: https://www.altexsoft.com/blog/engineering/pros-and-cons-of-java-programming/

[21] *Introduction to JSON Web Tokens*, [online]. URL: https://jwt.io/introduction/

[22] Hugo Di Francesco, *In simple terms: code on the backend, frontend and how they interact*, [online]. [Cit. 31.3.2017] URL: https://codewithhugo.com/in-simple-terms-code-on-the-backend-frontend-and-how-they-interact/#beyond-frontend-and-backend

[23] Hiren Dhaduk, *Best Frontend Frameworks of 2021 for Web Development*, [online]. [Cit. 5.1.2021] URL: https://www.simform.com/best-frontend-frameworks/

[24] *Spring Initializr*, [online]. URL: https://start.spring.io/

[25] *Material UI*, [online]. URL: https://material-ui.com/

[26] *Pexels*, [online]. URL: https://www.pexels.com/

[27] *Unplash Source*, [online]. URL: https://unsplash.com/

[28] *ČVUT DSpace*, [online]. URL: https://dspace.cvut.cz/discover?scope=10467%F3641&query=eshop&submit=&rpp=10

[29] Pressmann R.S.: *Software Engineering*