FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Assignment of master's thesis

Title:	Design of Information System for Confectionery
Student:	Bc. Jan Fara
Supervisor:	Ing. Petra Pavlíčková, Ph.D.
Study program:	Informatics
Branch / specialization:	Web and Software Engineering, specialization Information Systems and Management
Department:	Department of Software Engineering
Validity:	until the end of summer semester 2021/2022

Instructions

The goal of the diploma thesis is to develop the information system for the confectionery.

1. Create a business model of the system.
2. Make a timeline and risk analysis.
3. Create a financial plan including a return of investment.
4. Analyse functional and non-functional requirements.
5. Design an architecture and user interface of the system.
6. Implement and test the designed system.
7. Evaluate and design a further system development

Electronically approved by Ing. David Buchtela, Ph.D. on 3 February 2021 in Prague.

Master Thesis

Czech
Technical
University
in Prague

F8

Faculty of Information Technology
Department of Software Engineering

Design of Information System for Confectionery

Bc. Jan Fara

Supervisor: Ing. Petra Pavlíčková, Ph.D.
May 2021

Acknowledgements

I take this opportunity to express gratitude to my supervisor Ing. Petra Pavlíčková, Ph.D. for her advice, motivation and the time spent helping me with the thesis.

Eventually, I would like to thank my family and all friends for the unceasing encouragement and support.

Last but not least, I would like to thank my mom for her help with English grammar.

Declaration

Abstract

This diploma thesis deals with the design and implementation of the Information System for the real customer. The main emphasis is put on the analysis of the business plan of the whole project. On the basis of the customer requirements, the design of the System is introduced. Afterwards, the implementation, testing, and deployment of the System are described. From the thesis, the method emerged that the author can apply to begin with the development of the Information Systems for the next end-users with the help of the gained experience and reference.

Keywords: Business Plan, Customer focus, Information System, Enterprise Application

Supervisor: Ing. Petra Pavlíčková, Ph.D.

Abstrakt

Tato diplomová práce se zabývá návrhem a implementací informačního systému pro reálného zákazníka. Hlavní důraz je kladen na rozbor business plánu celého projektu. Na základě požadavků zákazníka je představen návrh systému. Následně je popsána implementace, testování a nasazení navrženého systému. Z práce vyplynul pro autora i postup, jak na základě získaných zkušeností a reference začít s vývojem informačních systémů pro další koncové zákazníky.

Klíčová slova: Podnikatelský plán, Priorita zákazníka, Informační systém, Podniková aplikace

Překlad názvu: Návrh Informačního systému pro cukrárnu

Contents

Figures

Tables

Chapter 1

Introduction

With the expanding information technologies, the electronic devices, such as, a phone, laptop or tablet are integral parts of our everyday lives. The arrival of the internet changed the attitude towards information and a new communication channel for companies towards their potential customers has arisen.

It is common that the information about services that are offered can be found on the company websites. The issue is not just about the information but also various goods can be bought via the internet. Online shopping is very comfortable for people as they can view products or services without leaving their homes. They can compare products and by a few clicks they can order the goods to be delivered either directly to their homes or to the nearest collection points.

Not only can companies provide the information and offer goods via the internet, but also, thanks to the Information Systems, it is possible to automate managing the business, for instance, the evidence and administration of orders, the available goods overview, and task management. The more responsibility the company delegates to the Information System, the less employees must be employed and the smaller the expenses are.

Many people prefer browsing the required services from their homes and that is why companies without websites lose their potential customers. The absence of websites presents an significant competitive disadvantage. Not only does not the company reach all potential customers, but also it loses the possibility to automate, and therefore running the business is more demanding and more expensive.

Some big companies offer a box solution that can be used for creating a simple E-shop. This way is cheap and easily applied, however, it need not be adaptable for every kind of business, above all, for bigger companies that need to create their own custom e-shop. A custom e-shop, as well as an Information System, are very expensive, therefore, not many companies can afford it. For smaller entrepreneurs it generally can be a problem, and the author of this thesis tries to look into this issue and find entrepreneurs who cannot afford the Information System or have not been interested in this issue. The author wants to help these entrepreneurs with the issues and implement and deploy an Information System to the production.

Chapter 2

Goals of the Thesis

The primary goal of this thesis is to improve Confectionery's business by designing and adding an information system for an online ordering and order management. This System should help the Confectionery's employees not to waste time by processing orders, and also to find new Confectionery's customers by the option of providing online ordering.

A part of the thesis is also to plan on how to get profit thanks to this reference and experience of the System development. The similar System can be developed for another business, and gradually, bigger Systems can be developed and other developers can be employed.

2.1 Structure of the Thesis

At the beginning of this thesis in Chapter 3, practices, which are used for creating business plan, and technologies for the System development, are introduced. The procedure of the business plan is introduced in Chapter 4. In the following Chapter 5.1, there is the analysis of the Information System. The goal of Chapter 6 is to design the System's architecture including selecting a suitable platform, creating a database model, and describing how the Information System was implemented. In Chapter 7, of the thesis a further System development is evaluated and designed.

Chapter 3
Theoretical Background

In this Chapter the technologies and the methods that are used in the design of the Information System for the Confectionery are described.

3.1 Business plan

A business plan is a written document that describes in detail how a business defines its objectives and how it is to go about achieving its goals. Every company should have a business plan. Periodical review and update of the plan is ideal to see if goals have been met or have changed and evolved. [3]

A Business plan consists of several steps and there is no exact structure of this document. There usually has to be planned something like:

- What has to be done to make money.
- Who is a customer and why he wants our product.
- How long the entire process will take.
- What risks exist, how to prevent them and how to solve them.
- How much has to be invested.
- How long the investment will take.

To summaries it, this is a nice overview of the entire business process. Thanks to this, it can be also figured out what a weaknesses of the plan are and how to eliminate them.

3.2 Business model

As a startup, it is not easy to find a customer for a software development. Because of this, it can be useful to use the Business Model by the customer. A Business Model is a framework that shows how a company will make money. It can be used to capture fundamental assumptions and decisions about the opportunity in one place, setting the direction for success. [4] Using this, the customer can see how his business will be improved, what is a source of the income, and what threats exist.

3.2.1 Lean Canvas

Lean Canvas is a 1-page business plan template that helps to deconstruct the idea into its key assumptions. [5] Lean Canvas was created by Ash Maurya from Canvas, and beside Canvas is more general and useful for the startups. Lean Canvas format is shown in Figure 3.1. Lean Canvas consists of eleven parts and each one is defined to answer a question:

- **Problem** – What issue does a customer have?
- **Existing alternatives** – What alternatives can a customer use to solve his issue?
- **Solution** – How can the the customer's issue be solved?
- **Key metric** – What metrics are used to monitor performance?
- **Cost Structure** – What are the most important costs inherent in our business model?
- **Unique value proposition** – What value do we deliver to the customer?
- **Unfair Advantage** – What we actually have, and cannot be easily copied or bought?
- **Channels** – Through which channels do our customer's segments want to be reached?
- **Customer segment** – Who are our most important customers?
- **Early adopters** – How can be the ideal customers characterized?
- **Revenue Streams** – What is our source of revenue?

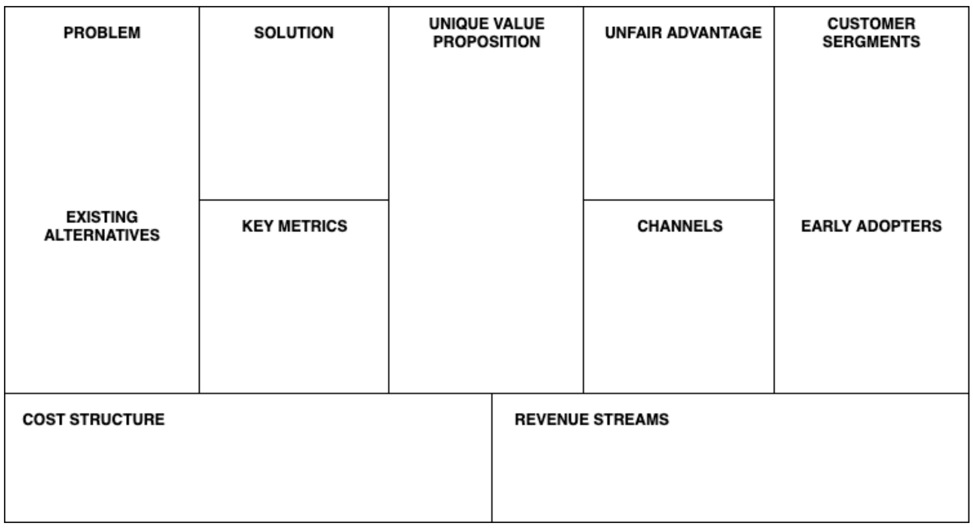


Figure 3.1: Lean Canvas

3.3 Timetable

Each customer would like to have his software, which was ordered, as soon as possible. A supplier needs to set up a deadline, when the software will be deployed to the production or delivered to the customer. But this deadline should be realistic and as accurate as possible. There could be a high fee that the supplier would have to pay if he was late with the product delivery. To get the most accurate date for the product delivery, there should be used a kind of timetable. The timetable helps to plan each step of software development, such as:

- analysis,
- design,
- implementation,
- testing,
- deployment,
- maintenance.

3.3.1 Gantt Diagram

Gantt Diagram, commonly used in a project management, is a useful way of showing activities displayed against time. The list of the activities is on the left side of the chart and the time scale is along the top side. The activities are represented by bars and each bar has a position and length which reflects the start date, duration, and the end date of the activity. [6] Example of the Gantt Diagram is shown in Figure 3.2.

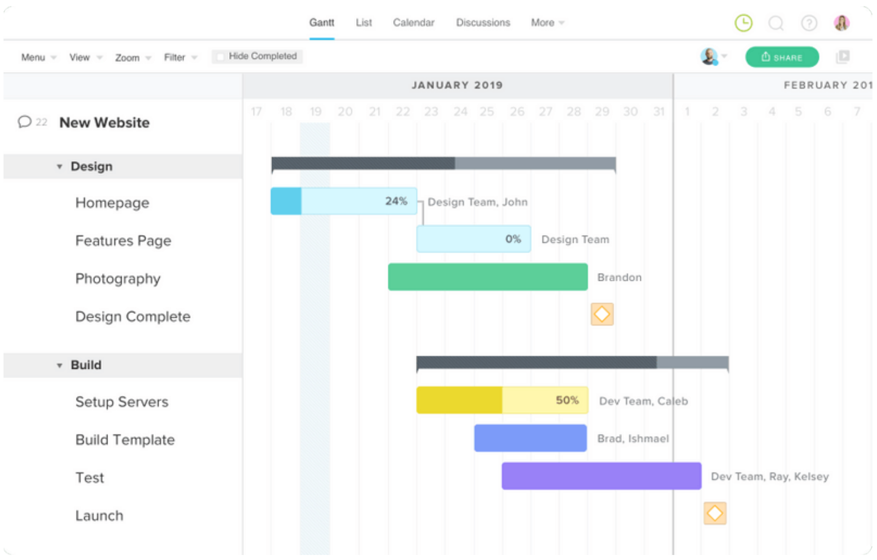


Figure 3.2: Gantt Diagram example [1]

3.4 Analysis

Software analysis include all activities which help the transformation of requirement specification of the System into implementation. The requirement specifications specify all functional and non-functional expectations from the System and are described in the following sections. [7]

This is a main part which must be clarified between the development team and the stakeholders. Based on analysis, price of the software product and timetable are evaluated. To create software analysis many tools can be used, such as: [7]

- Data Flow Diagram,
- Structure Charts,
- HIPO Diagram,
- Structured English,
- Entity-Relationship Model.

3.4.1 Functional requirements

Functional requirements describe the behaviour, functions and features of the System which are required by System users. These requirements must be implemented to enable users to accomplish their tasks. Without these requirements, the System cannot be deployed to the production, and it is very important to make them clear both for the development team and the stakeholders.

Requirements are usually written in a text, but there are many common formats and documents, such as: [8]

- Use Cases,
- User stories,
- Prototypes,
- Models and diagrams.

3.4.2 Non-functional requirements

Non-functional requirements, also known as quality attributes, describe how a System must behave and establish constraints of its functionality. Typical non-functional requirements are: [8]

- **Security** – Defines different levels of authorization and authentication across different users roles. Describes who can see, create, change, or delete resources.

- **Reliability** – Defines how likely it is for the software to work without failure for a given period of time. To measure software reliability, the percentage of operations that are completed correctly can be counted or the average period of time the System runs before failing can be tracked.

- **Performance** – Describes the responsiveness of the System to various user interactions with it. Performance is usually defined as how fast a user will receive a response on a specific load in a specific case.

- **Availability** – Define the period that the System's functionality and services are available for using with all operations. It's important to define how the impact of maintenance can be minimized.

- **Scalability** – Describes how the System must grow without negative influence on its performance. This means serving more users, processing more data, and doing more requests.

Typically something like good looking web, fast application, save application or modern application are wrongly defined non-functional requirements.

3.5 Use Case

A Use Case is a written description of how users will interact with that software System to achieve a goal. Each Use Case is represented as a list of steps, beginning with a user's goal and ending when that goal is fulfilled. [9]

Use Cases help explain how the System should behave and in the process, they also help to figure out what could go wrong. It provides a list of goals and it can be used to establish the cost and complexity of the System. [9]

Use Case Diagram can be used to simplify and visualize a Use Cases. It is a dynamic or behavior diagram in UML (Unified Modeling Language) and models the functionality of a System using actors and Use Cases. [10] There is a simple notation with followings elements:

- **Use Case** – Represents the System's functions.

- **Actors** – Users of a System. It can be also another System or hardware device.

- **Relationships** – Illustrate relationships between an Actor and a Use Case. There are two options of this relationship: *«include»* and *«extends»*. *«extends»* is used when a Use Case adds steps to another Use Case and *«include»* is used to extract Use Case that are duplicated in multiple Use Cases. [11]

The elements are also shown in Figure 3.3.

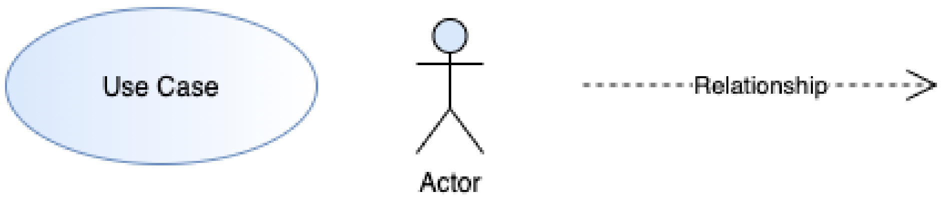


Figure 3.3: Use Case Diagram – elements

3.6 Wireframe

Wireframe is a two-dimensional skeletal outline of an application. Wireframe provides a simple overview of the page structure, layout, user flow and functionality. Wireframe should be as simple as possible, there is no emphasis for images, content, texts, colors and styling. [12]

There are many applications which can be used for Wireframe creation, for example: [13]

- Justinmind,
- Mockplus,
- Balsamiq,
- Axure,
- others.

Not only can applications for Wireframe creation be used, but also Wireframe can be created by using a sheet of paper and a pencil. That is why Wireframe is also called Paper Prototype.

3.7 Architectural model

„*The software architecture of a system is the set of structures needed to reason about the system, which comprises software elements, relations among them, and properties of both.*“ [14]

Every System has an architecture, it is a basic part in designing a System. There are many ways to see the architecture of the System. Architecture of the System can be described from the top perspective, such as:

- Which programming language is used for application development.
- Which database is used as the data storage.
- Which platform is used for application deployment.

On the other hand, the architecture of the System can be described by a single service in the complex System, such as:

- How security is developed.
- Which library is used for the data handling.
- How a state management is designed.

This is why we cannot describe architecture of each part of the System, there should be a line of an abstraction.

There are many ways how the System architecture can be defined, usually using: [15]

- UML,
- 4+1 view model,
- ADL (Architecture Description Language).

3.8 Hi-fi prototype

Hi-fi (High-fidelity) prototype is a draft version of the product that allows to explore ideas and show the intention behind a feature or the overall design concept to users before investing time and money into the development. [16] The Hi-fi prototypes cover the user interface of the product in terms of visuals and aesthetics, and also the user experience aspects in terms of interactions, user flow and behaviour. [17]

Hi-fi prototype tries to prevent expensive changes after System is developed. Prototyping allows to gather feedback from users while it is still being planned and designed and do cheaper changes of a product early in the development process. [16]

Hi-fi prototypes contain user scenarios with real data as are texts, pictures, but there is no emphasis to cover all functionality of the System. Data must not be stored in the database and only the functionality that needs to be tested is implemented.

3.9 Technologies

The main technologies used for the Server and Client development, chosen Database, and technologies used for Server deployment are described in this section.

3.9.1 Kotlin

Kotlin is a cross-platform, statically typed, general-purpose programming language with type inference. Kotlin runs on the JVM (Java Virtual Machine), it can fully interoperate with Java and can be compiled to the JavaScript. It is a modern language and, for example, compared to the Java it has several benefits: [18]

- **Conciseness**
  - POJO (Plain old Java object) with instant `getters`, `setters`, `equals()`, `hashcode()`, `copy()`.
  - Streams with lambda expression.
  - One line function without brackets.
- **Safety**
  - Nullable types – No `NullPointerExceptions`.
  - Type is automatically cast by compiler if match while checking.

3.9.2 Spring Boot

Spring Boot is an extension of Spring framework and eliminates lots of configuration which was necessary in Spring framework. Spring Boot is an open source framework for enterprise application development. This framework is a nice way how to customise, develop, and run huge enterprise applications. For more information about Spring Boot framework see [19].

3.9.3 React

React is an open source JavaScript library for building user interface. The goal of this framework is to be fast, simple and scalable. One of the React advantages is an ability to change data over time without reloading pages, and thanks to this capability it is fast. React is primarily used for Client web application development, but it can also be used for full-stack mobile application development called React Native. [20]

Redux

Redux is a JavaScript open source library for the state management. This library is mostly for React, but it can also be used for Angular. Angular is the next option alongside React to develop the Client side of the application. The goal of this library is to hold the global state through the whole application. The application state is stored in the tree structure as a JSON (JavaScript Object Notation). [21]

3.9.4 JSON

JSON (JavaScript Object Notation) is a data-interchange format. This format is very simple to read. JSON format consists of three plain types:

- **object** – `{...}`,
- **variable with value** – `"variable": "value"`,
- **array** – `[...]`.

By composing these types it is possible to represent complex data.

3.9.5 HTTP

HTTP (Hypertext Transfer Protocol) is the internet protocol on the application layer that allows fetching resources. It is used to exchange data between the Client and the Server. HTTP has several methods, such as, `GET, POST, PUT, PATCH, OPTION, DELETE` and they are more specifically described in Table 3.1. The methods can have properties, such as:

- **Body** – Data in request.
- **Safe** – Method is called Safe if it does not alter the state of the Server.
- **Indempotent** – Method is called Indempotent if an identical request can be made once or several times in a row with the same effect. [22]

HTTP methods always return status code that describes what has happened. That means that the methods return the information about the request successfulness, these return codes are described in Table 3.2.

Method	**Note**	**Body**	**Safe**	**Idempotent**
GET	only request for data of the resource	No	Yes	Yes
POST	for creating new resource	Yes	No	No
PUT	replace existing resource by new one	Yes	No	Yes
PATCH	change data of existing resource	Yes	No	No
DELETE	delete existing resource	Yes	No	Yes
OPTIONS	get information about permission	No	Yes	Yes

Table 3.1: HTPP methods

3.9.6 REST

REST (Representational State Transfer) is an architectural style for providing standards among systems on the web. REST is usually used with an HTTP combination and is called RESTful. Benefit of RESTful is to be familiar to anyone accustomed to using HTTP. REST is data oriented that means only data are transferred through the network, for example, in JSON format. Systems which follow REST paradigm should be stateless. That means there is no state on the Server and the Client and the Server must understand each request without any state or previous requests. [23, 24]

Response status code	Note
1xx	Informational responses
2xx	Successful responses
3xx	Redirects
4xx	Client errors
5xx	Server errors

Table 3.2: HTPP response status codes

3.9.7 ETag

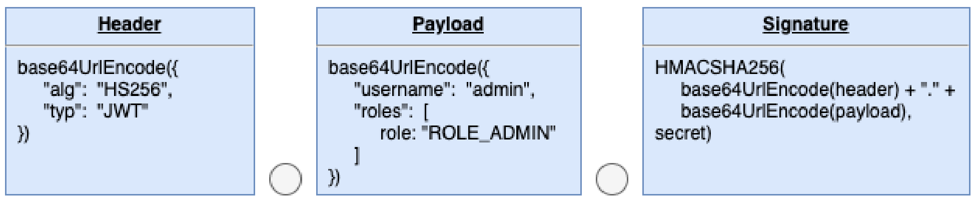
Etag serves to reduce the load on communication between a Server and a Client. If a user creates a request and receives data from the Server, and then creates the same request, then the same data are sent again through the network. ETag manages that the Server sends parameter with hashed data with each response `ETag` in the HTTP header. The Client with each request for the same endpoint sends `If-None-Match` parameter with hash which was previously received as ETag. If the Server matches these hashes, then it sends `HTTP/1.1 304 Not Modified` response with `ETag` header parameter and the same hash. If the browser receives this response, it takes data from the browser memory which have been received in the past. Otherwise the data are sent again. [25]

3.9.8 JWT

JWT (JSON Web Token) is used for an authentication and authorization of the Client on the Server. JWT contains following three parts:

- **Header** – Contains information about encryption algorithm and type of signature.
- **Payload** – Contains data about the user and his roles, it also contains metadata about itself.
- **Signature** – Combination of encoded Header and encoded Payload, that is encrypted using an encryption algorithm.

The example is also shown in Figure 3.4.



```
base64UrlEncode({
    "alg": "HS256",
    "typ": "JWT"
})
```

```
base64UrlEncode({
    "username": "admin",
    "roles": [
        role: "ROLE_ADMIN"
    ]
})
```

```
HMACSHA256(
    base64UrlEncode(header) + "." +
    base64UrlEncode(payload),
secret)
```

Figure 3.4: JWT – structure

This token is created by the Server when the user signs up, and then the access token is sent as a response. The Client stores JWT in the memory and sends this access token with each request to the Server in `Authorization` header. After that, the Server validates the token and if validation is successful, the Server decodes the user with its roles from the payload part of this token, and checks for the user authorization. If the user has got required roles, the Server returns the secured resource to the Client. [26] The communication between the Client and the Server is shown in Figure 3.5.

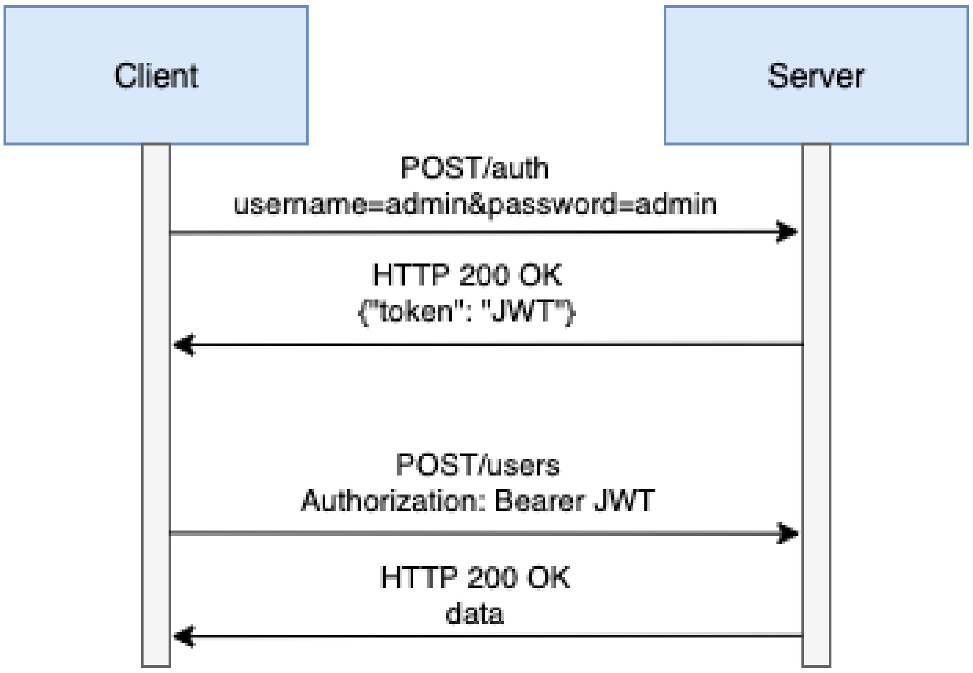


Figure 3.5: JWT – communication between the Server and the Client

3.9.9 Spring Security

Spring Security is a customizable framework which provides an authentication and authorization for Java application. Spring Security supports the authentication integration with many technologies, such as: [27]

- OpenID authentication,
- Kerberos,
- LDAP (Lightweight Directory Access Protocol),
- Automatic *remember-me* authentication,
- others.

Thanks to this framework it is easy to prevent many hacker attacks, like: [28]

- session fixation,
- clickjacking,
- cross site request forgery.

3.9.10 Spring Data JPA

Spring Data JPA is a library which makes it easy to implement JPA based repositories. This library significantly improves the implementation of data access layers. Using Spring data JPA, it is easy to persist data through the Server and the Database. [29, 30]

3.9.11 Cloud

Cloud solution for an application development means that the application runs in the servers that are located in data centers all over the world and are accessed over the Internet. Cloud characteristics are: [31]

- **On-demand self-service** – Resources are provided at a request.
- **Broad network access** – Users have an access to their capabilities from anywhere.
- **Resource pooling** – Computing sources are shared among many customers.
- **Rapid elasticity** – Infrastructure might change according to the needs.
- **Measured Service** – Automatically measures or monitors the provision of services.

High benefit of the Cloud is that users have almost infinity Server performance thanks to the scalability. The scalability enables that there are always as many resources as it is currently needed. This is a *pay-as-you-go* model because we pay only for resources which are currently needed. The comparison of the Cloud solution and the traditional hardware model of hardware resources usage is mentioned in Figure 3.6 and 3.7, these Figures are from the internal FIT CTU sources from lecture MI-WEB20.16 by doc. Ing. Tomáš Vitvar, Ph.D.. The Cloud providers are, for example:

- Google,
- Amazon,
- Microsoft,
- Oracle.

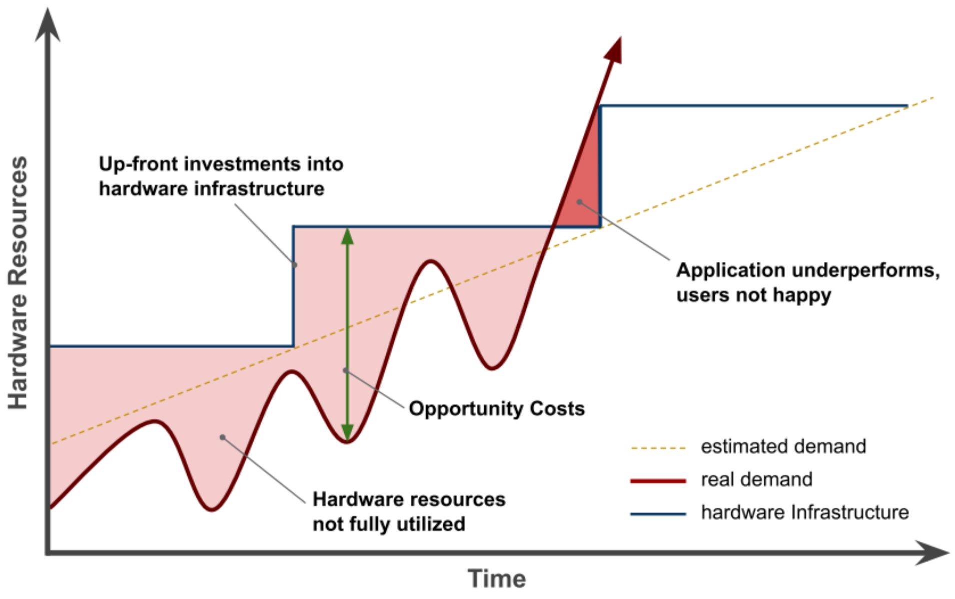


Figure 3.6: Traditional hardware model

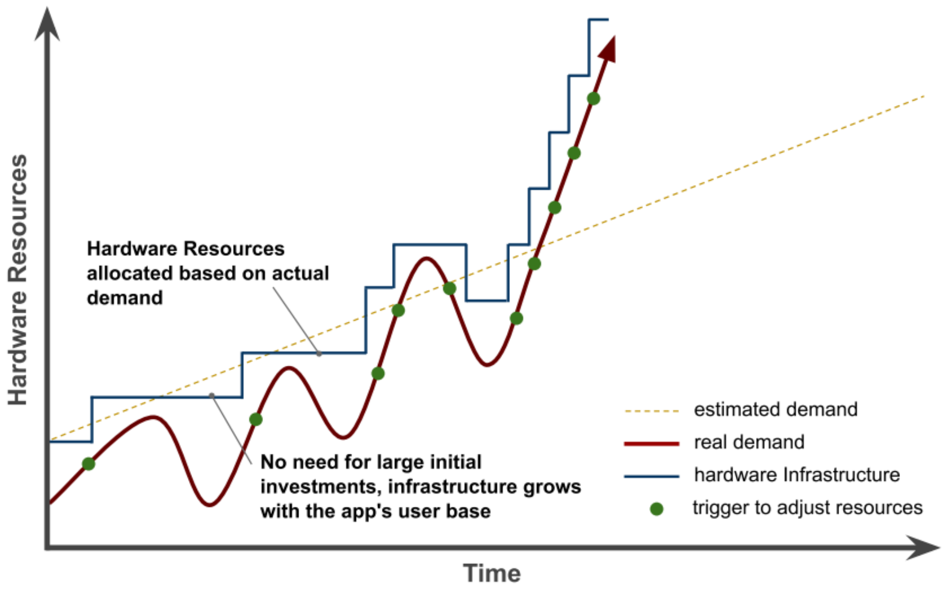


Figure 3.7: Cloud solution

3.9.12 Docker

Docker is an open source project written in go and developed by Dotcloud. [32] It makes easier to create, deploy, and run applications by using containers. Containers allow to package up an application with all dependencies, and deploy it as one package. By doing this, there is no problem to run the application on any system using Docker. [33]

3.9.13 Kubernetes

Kubernetes is an open source system for automating deployment, scaling, and management of containerized applications. [34] Kubernetes configuration on the Cloud, such as:

- the number of containers,
- environments variables,
- port which the application is exposed on,
- secrets

can by easily managed by configure of `.yaml` files. It is a good way how to ensure that there is no downtime at the container application. Scaling is not only good for the performance but also for an application stability. For example, on application deployment the containers are gradually switched to the new version but the application is never unavailable. If a container fails for any reason, Kubernetes restores it to the healthy state, therefore it is called self-healing. [35] In Kubernetes Cluster many Nodes (Virtual Machines) can run and in each Node contains one or many Pods. The Pod is the smallest deploy unit and can contain a few Docker Containers.

3.9.14 PostgreSQL

PostgreSQL is one of the most popular objective relation database systems. The popularity of this database in 2019 at stackowerflow survey is shown in Figure 3.8. [2] It is free and open source, but there is also a supported option. This database is usable on all platforms as Microsoft Windows, macOS and almost all Linux and Unix distributions. [36]

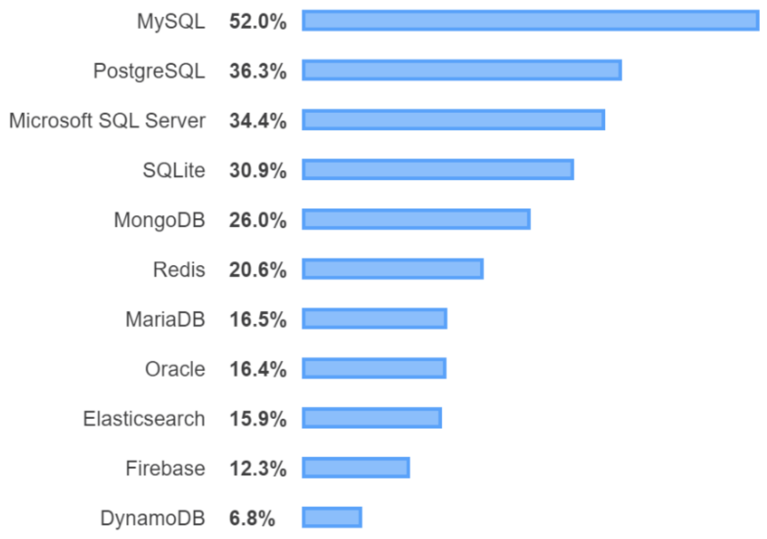


Figure 3.8: Database servers – stackowerflow survey [2]

Chapter 4
Business plan

At the beginning of this Chapter, in Section 4.1, there is the idea how the author of this thesis chose the customer and what the customer is like. In Section 4.2 the Business Model for the customer is designed. In Section 4.3 the timetable for this thesis is designed. Risks that can influence this thesis are described in Section 4.4. In the end, in Section 4.5 and 4.6 the costs and the revenues are analysed.

4.1 Idea

The aim of this thesis is to find a real customer and on the basic of his requirements develop and implement an Information System that can enhance the customer's business.

The Author of this thesis presented the Business Model, which is described in the following section, to a Confectionery in Prague since they had not had any System for ordering management and any website to present their products. This small Confectionery is managed by a person who employs about four employee. Each employee, in his working hours, must serve customers in the shop and also manage orders, which are created by customers by phone. Some customers have a lot of specific wishes and it takes lots of time to agree with the customer. If the employee accepts an order, they write the information about the order to a paper notebook, which takes some time.

By implementing the System the employee will have more time for serving customers and a part of the orders will be accepted and registered by the System. The form for custom cakes and wedding cakes is designed to avoid orders that the Confectionery cannot meet. Thanks to the online product offer it is assumed that there will be more orders and the business turnover will increase.

After finishing this project the author would like to develop a similar System for other entrepreneurs. If this System is successfully completed, there will be a reference of the System development and also approximate estimate of the price. Moreover, the amount of the time that was spent on the similar System development will be known. The new Systems can be based on the same technologies with similar architecture. Thanks to that and with previous experience, the creating of the similar System will be much faster.

4.2 Business model

This Business model is initiated by the customer and the goal of this model is to clear the customer how his turnover will be improved by using this System.

Nowadays it is common to extend business and provide service online on the website. Most of people find service they need on the websites. Without a website it is a noticeable disadvantage compared to their competition. A company without any website can be found by a new customer only when they walk by or by references.

This is not the only problem which the Confectionery has. The employees of the Confectionery waste lots of time on phone calls giving information to customers, which could be found on the website as well as placing orders. Using this System, these problems will be solved. Confectionery customers can find the shop on the website as well as the information about the service and they will be able to create online orders.

The customer could buy his own custom Information System either from a company or freelancer. The companies which implement custom Information Systems are very expensive because lots of experts take part in the development, such as, an analyst, an architect, and the whole team of programmers and testers. Choosing a freelancer is a better choice, but the price from the experienced freelancer might be also high. The box solution of an Information System is undoubtedly cheaper but cannot be always modified according to the customer's requests, both the user interface and the functional requirements.

The System designed by the author also provides a unique form, which enables to customize cakes by the customer's wish and a gallery, where customers can be inspired by photographs of cakes.

Solving all these issues increases the number of customers, which means the higher turnover. That is the goal of all business models. The business model is shown in Figure 4.1.

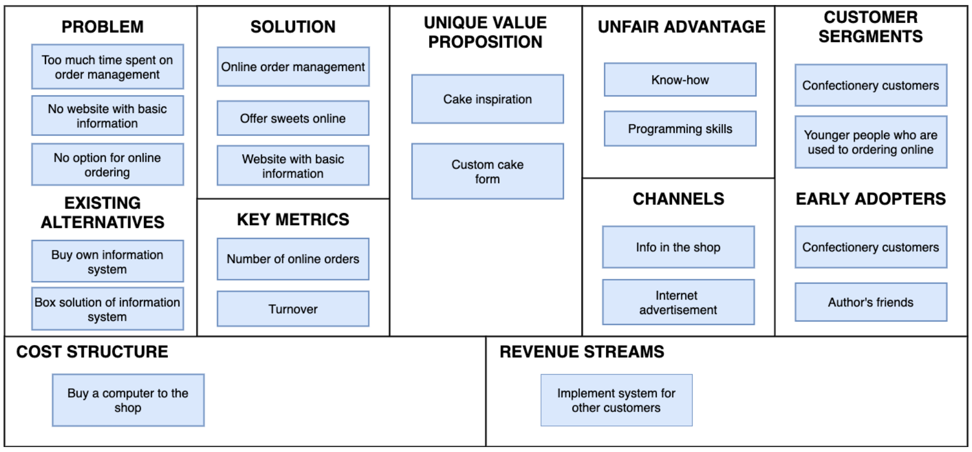


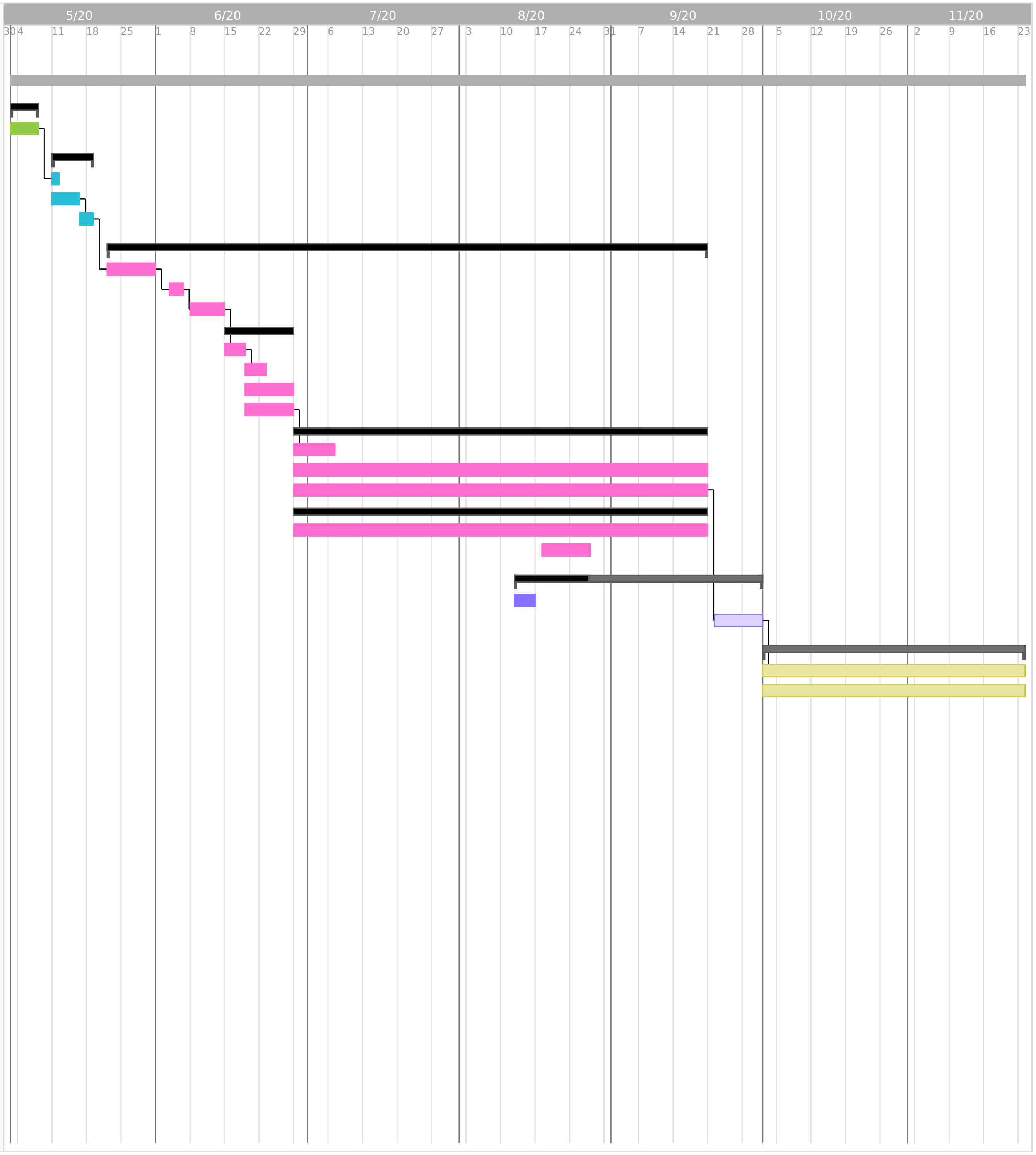
Figure 4.1: Business Mode for the customer

4.3 Timetable

A timetable is created as Gantt Chart by online tool available at `https://www.teamgantt.com/`. The entire process starts at 2020.05.01 and ends 2020.30.09. without any support and maintenance. Gantt Chart is shown in Figure 4.3.

The business model was created and successfully presented to the customer. The customer agreed with the terms and at the next meeting function and non-function requirements were discussed. On the basis of this, Use Cases, Scenarios, and Wireframes were created. This paper prototype was then presented to the Customer and his feedback was processed. After this, implementation process started. The System contains automation tests that test services and controllers, which means that the System was tested during the entire development. After Hi-fi prototype was deployed the System was tested by real users. The System was ready for the deployment to the production, support, and maintenance, but the customer declined deployment because of COVID-19 situation, and therefore these processes are not successfully completed.

Confectionery	**0h**	**74%**
Business plans	**0h**	**100%**
Business model	0	100%
Analysis	**0h**	**100%**
Non-functional requirements	0	100%
Functional requirements	0	100%
Use Case and Scenarios	0	100%
Realization	**0h**	**100%**
Wireframes	0	100%
Feedback	0	100%
Processing of requests	0	100%
Architecture	**0h**	**100%**
Tech stack	0	100%
Database	0	100%
Server	0	100%
Client	0	100%
Implementation	**0h**	**100%**
Database	0	100%
Server	0	100%
Client	0	100%
Testing	**0h**	**100%**
Automation testing	0	100%
Usability testing	0	100%
Deployment	**0h**	**30%**
Hi-fi prototyp	0	100%
Production version	0	0%
Maintenance Services	**0h**	**0%**
Maintenance	0	0%
Support	0	0%



4.4 Risk Analysis

Each business can be interrupted or complicated by a risk. The risks which can disrupt this business plan are described in this section in Tables 4.1, 4.2, 4.3, 4.4, 4.5, 4.6. Each risk has the following parameters:

- **id** – A unique identifier which is used in the Risk Matrix.
- **Description** – Description of the risk.
- **Risk mitigation** – How to prevent this the risk.
- **Solution** – What author will do to solve the problem if the risk arises.
- **Impact of risk** – The negative consequences of the risk.
- **The probability of occurrence** – Assumed probability of the risk occurrence.
- **Impact assessment** – Rate of the risk, the value is from the set $\{1, 2, 3, 4\}$, so the higher number it is, the worse risk impact is.
- **Type of risk** – The category in which the risk is classified.

The risks are integrated into the Risk Matrix 4.7, which represents classification of each risk.

Risk id	1
Description	The lack of interest in sweets. People want to eat only healthy food.
Risk mitigation	Highlight the information on how healthy the cakes in the System are.
Solution	Deal with the customer to bake cakes with minimal amount of sugar.
Impact of risk	The lack of interest in a similar System. Hard to find a new applicant for the next System.
The probability of occurrence	2 %
Impact assessment	1
Type of risk	Human

Table 4.1: Risk Analysis – risk 1

Risk id	2
Description	Easy applicable service provided by a big company which will allow the small businesses to create an E-shop.
Risk mitigation	The System has a unique form for custom cakes. Adapt services of information Systems according to the customer's requirements.
Solution	Find out which extra services are provided by a big company and implement it to the System.
Impact of risk	Customer loss.
The probability of occurrence	15 %
Impact assessment	4
Type of risk	Business

Table 4.2: Risk Analysis – risk 2

Risk id	3
Description	Closure of business.
Risk mitigation	Motivate the customer by the business model.
Solution	In case of the business sale to another owner, negotiate with a new customer to keep the System running.
Impact of risk	The probability of the loss of the customer.
The probability of occurrence	5 %
Impact assessment	3
Type of risk	Business

Table 4.3: Risk Analysis – risk 3

Risk id	4
Description	Customer does not want to keep using the System.
Risk mitigation	Present facts of the business model.
Solution	Find another customer.
Impact of risk	The probability of the loss of the customer.
The probability of occurrence	5 %
Impact assessment	2
Type of risk	Business

Table 4.4: Risk Analysis – risk 4

Risk id	5
Description	Problems with using the System.
Risk mitigation	System testing.
Solution	Fix the bugs of the System as soon as possible.
Impact of risk	The customer does not want to keep using the System.
The probability of occurrence	10 %
Impact assessment	1
Type of risk	Technical

Table 4.5: Risk Analysis – risk 5

Risk id	6
Description	Worsening of the current state of the pandemic. The customer does not want to deploy the System.
Risk mitigation	Motivate the customer by the business model.
Solution	Wait for the situation to calm down.
Impact of risk	Delay against the timetable.
The probability of occurrence	20 %
Impact assessment	3
Type of risk	Business

Table 4.6: Risk Analysis – risk 6

Probable 10+ %	5	6		2
Possible <5; 10) %			3, 4	
Improbable <0; 5) %	1			
Likelihood / Impact	1 Acceptable	2 Tolerable	3 Unacceptable	4 Intolerable

Table 4.7: Risk Analysis – Risk Matrix

4.5 Financial plan

In this section the approximate estimate of the costs is described. The costs are comprised by labor costs and hardware software operation. Labor costs are required for the whole project development. The hardware software operation are necessary for running Hi-fi prototype and product version of the System.

4.5.1 Labor costs

The most expensive part of the project are labor costs. The project is considered to be an investment, and except the testers, only the author of this thesis worked on it. In the following Table 4.8 there are estimate man-days for each part of the timetable listed, which is shown in Figure 4.3. Estimated man-day rate of testers is 3.500 CZK and for the author is 5.000 CZK.

The biggest part of labor costs is made up of the System development by the author of this thesis. The second part of this price is made up of the System testing. It is planned to employ two testers for seven man-days. The price calculation of labor costs is shown in Table 4.9. In total, the price is estimated **526.000 CZK**.

There is also a price for the System support and maintenance which author of this thesis has to perform. It is assumed that the support and maintenance take 3 man-days a month. With author man-day rate it costs **15.000 CZK per 1 moth**.

Process	**MD**
Analysis	7
Wireframes	7
Architecture	10
Implementation	60
Testing	7
Deployment	10
Total	101

Table 4.8: Financial plan – MD for processes

Process	**Quantity**	**MD**	**Rate [CZK]]**	**Total [CZK]**
Tester	2	7	3.500	49.000
Author	1	94	5.000	470.000

Table 4.9: Labor costs – price estimation

4.5.2 Hardware software operation

The System was developed in the local machine and therefore there is no fee for the System development. Hi-fi prototype ran on VPS (Virtual Private Server) which the author of this thesis has already subscribed and therefore there is also no fee. The product version will run on the Google Cloud with compute engine, which is shown in Table 4.10. Google Cloud environment with this compute engine is calculated by Google calculator to **700 CZK per 1 month**. With a growing number of users more Server instances should run and there will be more requirements for compute engine of Google Cloud. Therefore, it is necessary to compute with better performance of compute engine, which means higher price for hardware software operation. For example, the estimate of more powerful compute engine is shown in Table 4.11 and it costs **4,200 CZK per 1 month**.

Parameter	Value
Number of instances	1
vCPUs	1
RAM	4 GB
Total hours per month	730

Table 4.10: Google Cloud – compute engine

Parameter	Value
Number of instances	3
vCPUs	2
RAM	8 GB
Total hours per month	730

Table 4.11: Google Cloud – bigger compute engine

4.6 Return of Investment

The plan how to get profit from this investment is to get reference and experience of the whole process of this project, such as, communication with the customer, creating analysis, designing user interface, development, and deployment. The goal is also to calculate the price which can be used while calculating the price of the similar System. After finishing this project, the plan is to find other customers and implement next Information Systems. With the increasing number of projects, less time will be needed to develop the System since more experience will the author have. That means the bigger number of projects is, the lower the costs of projects will be. It is estimated that in five years the costs of the project will decrease by 10 % with each subsequent project. Every year two projects could be realized. There are two models of costs and revenue estimates – pessimistic and optimistic. For each model it is also calculated NPV (Net Present Value), which is the value of all future cash flows over the entire time of an investment discounted to the present. [37] The formula for NPV is:

$$NPV = \sum_{t=1}^{n} \frac{R_t}{(1+i)})^t$$

where

- t – Number of time period.
- R_t – Net cash flow during a single period t.
- i – Discount rate.

to calculate NPV i was set to average inflation over the last year, which is 0.32 %.

4.6.1 Pessimistic model

The pessimistic model presupposes that to find next clients will not be so easy and Systems cost is expected to be low and cannot cover project investments for a while. It is planned to start at 200,000 CZK as a project price and with an increasing number of projects slowly increase the project cost. After three years it is planned to make the investment by employing a new developer to complete the projects faster and to be able to develop bigger Systems. This model is shown in Table 4.12 where MS means Maintenance Services. The final costs and revenue for each project are also shown in Figure 4.2. Total costs for five years is 7,211,115 CZK and total revenue for five years is 11,060,000 CZK, that means there is **4,648,885 CZK** net profit after five years. Net profit and NPV for each year is shown in Table 4.13.

4.6.2 Optimistic model

The optimistic model presupposes that there will be no problem to find next clients and System's cost should be profitable. It is planned to start at 400,000 CZK as a project price and with an increasing number of projects slowly increase the project cost. After two years it is planned to make an investment and employ a new developer with the same reason as in the Pessimistic Model. After four years it is also planned to employ a new developer and develop bigger Systems. This model is shown in Table 4.14 where MS is Maintenance Services. The final costs and revenue for each year are also shown in Figure 4.3. Total costs for five years is 7,211,115 CZK and total revenue for five years is 11,060,000 CZK, that means there is **16,666,580 CZK** net profit after five years. Net profit and NPV for each year of this model is shown in Table 4.15 and a comparison between pessimistic NPV and optimistic NPV is shown in Figure 4.4.

#	Year	Project costs [CZK]	MS costs [CZK]	Cloud costs [CZK]	Project revenue [CZK]	MS revenue [CZK]	Total costs [CZK]	Total revenue [CZK]
1	0	519,000	25,000	8,400	0	0	552,400	0
2	1	467,100	50,000	16,800	200,000	20,000	533,900	220,000
3	1	420,390	75,000	25,200	300,000	50,000	520,590	350,000
4	2	378,351	100,000	33,600	400,000	90,000	511,951	490,000
5	2	340,516	125,000	42,000	600,000	150,000	507,516	750,000
6	3	306,464	150,000	50,400	600,000	210,000	506,864	810,000
7	3	275,818	175,000	58,800	700,000	280,000	509,618	980,000
8	4	551,636	200,000	67,200	700,000	350,000	818,836	1,050,000
9	4	496,472	250,000	117,600	1,200,000	470,000	864,072	1,670,000
10	5	446,825	300,000	168,000	2,000,000	670,000	914,825	2,670,000
11	5	402,143	350,000	218,400	2,000,000	870,000	970,543	2,870,000

Table 4.12: Costs and revenue – pessimistic

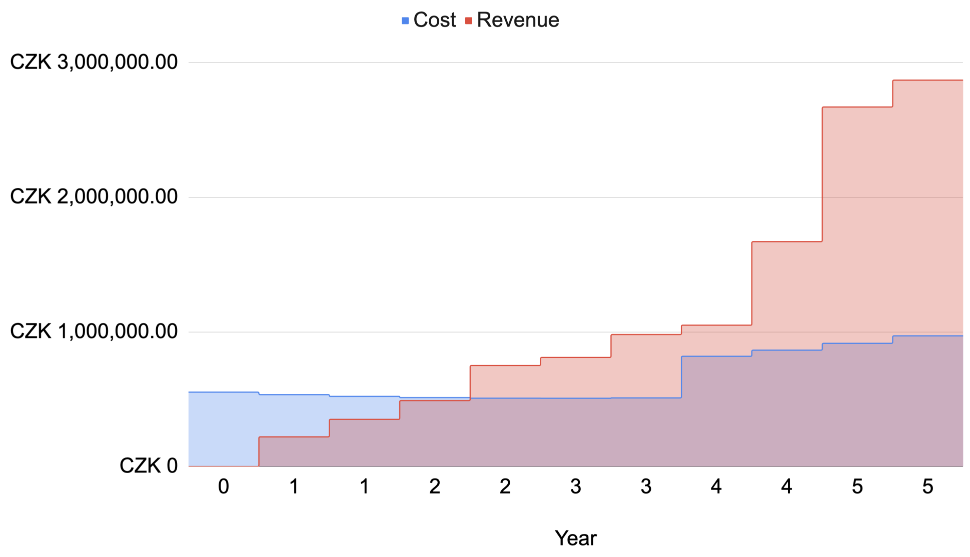


Figure 4.2: Final costs and revenue – pessimistic

Year	Costs [CZK]	Revenue [CZK]	Net profit [CZK]	NPV [CZK]
0	552,400	0	-552,400	-552,400
1	1,606,890	570,000	-1,036,890	-1,557,138
2	2,626,357	1,810,000	-816,357	-2,323,653
3	3,642,839	3,600,000	-42,839	-2,362,629
4	5,325,747	6,320,000	994,253	-1,486,076
5	7,211,115	11,860,000	4,648,885	2,485,384

Table 4.13: Net profit and NPV – pessimistic

#	Year	Project costs [CZK]	MS costs [CZK]	Cloud costs [CZK]	Project revenue [CZK]	MS revenue [CZK]	Total costs [CZK]	Total revenue [CZK]
1	0	519,000	25,000	8,400	0	0	552,400	0
2	1	467,100	50,000	16,800	400,000	40,000	533,900	440,000
3	1	420,390	75,000	25,200	500,000	90,000	520,590	590,000
4	2	378,351	100,000	33,600	600,000	150,000	511,951	750,000
5	2	340,516	125,000	42,000	600,000	210,000	507,516	810,000
6	3	681,032	150,000	50,400	800,000	290,000	881,432	1,090,000
7	3	612,929	200,000	100,800	1,200,000	410,000	913,729	1,610,000
8	4	551,636	250,000	151,200	3,000,000	710,000	952,836	3,710,000
9	4	496,472	350,000	201,600	3,000,000	1,010,000	1,048,072	4,010,000
10	5	992944	450,000	252,000	5,000,000	1,510,000	1,694,944	6,510,000
11	5	893,650	550,000	302,400	5,000,000	2,010,000	1,746,050	7,010,000

Table 4.14: Costs and revenue – optimistic

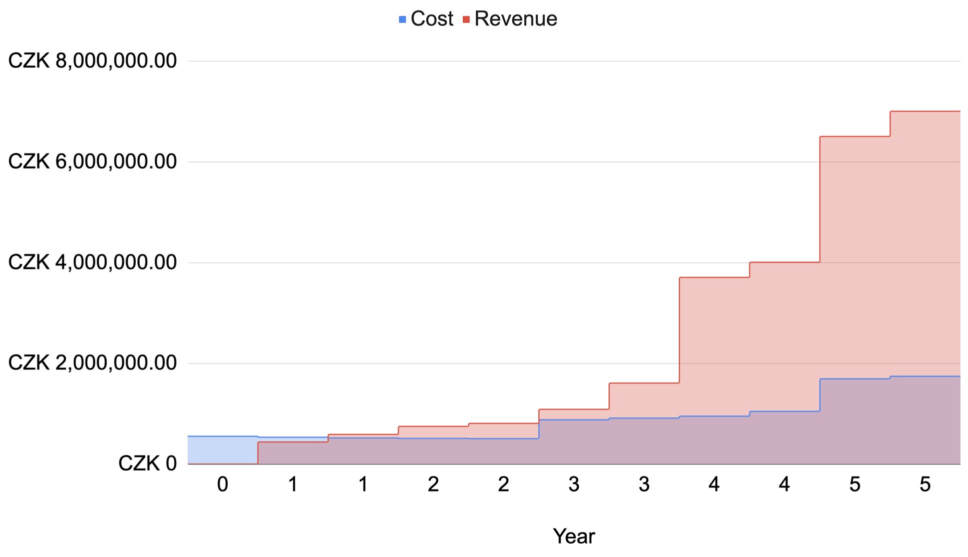


Figure 4.3: Final costs and revenue – optimistic

Year	Costs [CZK]	Revenue [CZK]	Net profit [CZK]	NPV [CZK]
0	552,400	0	-552,400	-552,400
1	1,606,890	1,030,000	-576,890	-1,111,401
2	2,626,357	2,590,000	-36,357	-1,145,539
3	4,421,518	3,600,000	868,482	-355,367
4	6,422,426	13,010,000	6,587,574	5,452,366
5	9,863,420	26,530,000	16,666,580	19,690,334

Table 4.15: Net profit and NPV – optimistic

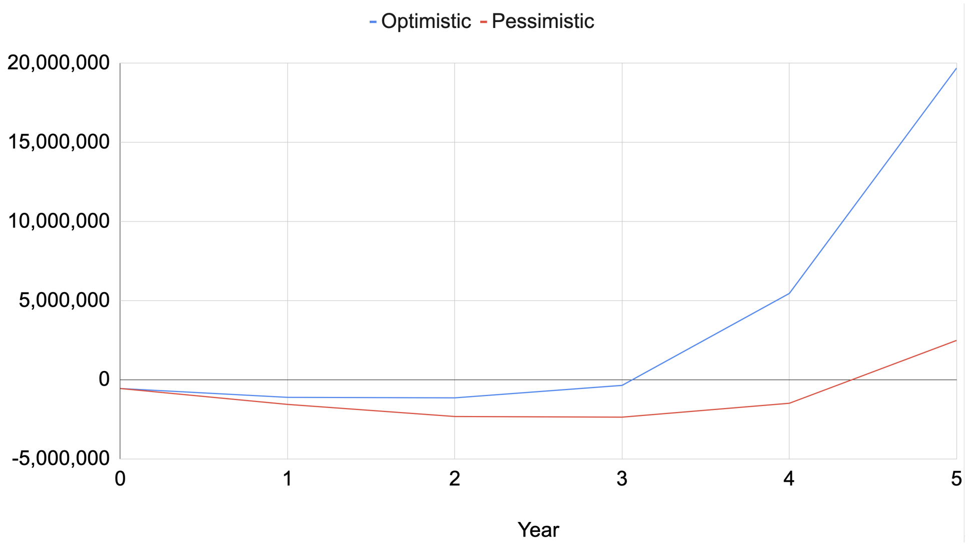


Figure 4.4: NPV – Pessimistic model vs. Optimistic model

Chapter 5

Analysis

In this chapter functional and non-functional Requirements of the System defined by the customer are described. There are also Use Cases and Scenarios which describe interaction between the user and the System. At the end there is shown how functional Requirements are covered by the Use Cases in Table 5.1.

5.1 Functional Requirements

5.1.1 Customer

- **FR1 View products** – The System allows users to view the products.
- **FR2 View gallery** – The System allows users to view the gallery.
- **FR3 Create custom product** – The System allows users to create a custom product.
- **FR4 Cart management** – The System allows users to manage content of their cart.
- **FR5 Create order** – The System allows users to order the products.
- **FR6 Sign in** – The System allows users to sign in.

5.1.2 Employee

- **FR7 Product management** – The System allows signed in users to manage the products.
- **FR8 Gallery management** – The System allows signed in users to manage the gallery.
- **FR9 Account management** – The System allows signed in users to manage their account.
- **FR10 Users management** – The System allows signed in users to manage registered users.

- **FR11 Orders management** – The System allows signed in users to manage the orders.

5.2 Non-functional requirements

Non-functional requirements defines following specifications:

- **Performance** – 95 % of the System requests should respond to 3 seconds and all requests should respond to the maximum 5 seconds.
- **Scalability** – The System should be scalable to 1000 users within the next two years.
- **Availability** – The System should be available from 06:00 a.m. to 01:00 a.m. from Monday to Sunday. And from 01:00 a.m. to 06:00 a.m. System should be available for System maintenance purposes.
- **Security** – The System should remain resilient in the face of attacks. The system must be available and behave reliably even under DOS attacks.

5.3 Use Cases and Scenarios

There are two actors, a customer and an administrator. A user is a customer of the Confectionery and the administrator is an employee of the Confectionery. The user becomes the administrator when he successfully signs in to the System. Use Case Diagram for the customer is shown in Figure 5.1 and for the administrator in Figure 5.2.

5.3.1 Customer

UC1 View product list

- **Use Case** – On the *Product page*, user expects a product list that Confectionery offers.
- **Scenario** – If the user visits *Product page* then view product list and each product have:
  - photo of the product,
  - product name,
  - description,
  - price,
  - *Add to cart button.*

UC2 View product detail

- **Use Case** – After clicking on any product at the *Product page* the user expects more detail of the product.
- **Scenario** – On the *Product page* after clicking on any product the user will view:
  - photo of the product,
  - product name,
  - description,
  - price,
  - *Add to cart button*,
  - allergens,
  - number of days for which the product is created,
  - info if product is vegan.

UC3 Add a product to the cart

- **Use Case** – On the *Product page* the user expects that each product has *Add to cart button* which adds the product to the cart.
- **Scenario** – At *Product page*, after the user clicks on *Add to cart button* that product which was clicked on will be added to the cart. The user can also see how many of these products he has in the cart.

UC4 Remove product from the cart

- **Use Case** – On the *Cart page* the user expects that each product has a button which removes product from the cart.
- **Scenario** – On the *Cart page* after the user clicks on the *Remove button* a confirmation window will appear and if the user confirms the action, the product will be removed from the cart. If the user declines this confirmation window, nothing will be deleted from the cart.

UC5 Change the number of items in the cart

- **Use Case** – On the *Cart page* and *Product page* the user expects that each product has an option how to increase or decrease the number of items in the basket.
- **Scenario** – On the *Cart page* or *Product page* after the user clicks on the *+ button*, the number of items in the cart will be increased. If the user clicks on the *- button*, the number of the items in the cart will be decreased.

UC6 Create custom product

- **Use Case** – On the *Custom cake page* the user excepts, that he can create a custom cake and add it to the cart.
- **Scenario** – On the *Custom cake page* the user will see a form with the following parameters:
  - Corpus * – Select box with available corpus list options.
  - Creme * – Select box with available creme list options.
  - Internal filling – Select box with available internal filling creme list.
  - Decoration * – Select box with available decoration list options.
  - Allergens – Multiple choice select box with all allergens.
  - Number of portion * – Number input which describes how many people (pieces) the cake will be for.
  - Cake inscription – A text on the top of the cake.
  - Description – Additional information about the cake.
  - Photo – *Upload button* for cake inspiration photo.

  Where the parameter is marked by * it is mandatory to be filled in the form. After the user fills the mandatory parameters of this form and confirms the form, the custom cake will be added to the cart.

UC7 View gallery list

- **Use Case** – On the *Gallery page*, the user expects a gallery list of products.
- **Scenario** – If the user visits *Gallery page* then he will view a gallery list and he can filter it by the following parameters:
  - product,
  - custom product,
  - wedding cake,
  - other.

UC8 View gallery detail

- **Use Case** – After clicking on the gallery at *Gallery page* the user expects the detail of the photograph.
- **Scenario** – On the *Gallery page* after the user clicks on the photograph the user will view the detail of the photograph. Below the photograph the user can view a button and by clicking on it he can create a similar custom product.

UC9 Create order

- **Use Case** – After the purchase finishes, the user expects:
  - date select option,
  - filling in contact details,
  - order summary information,
  - *Create order button.*
- **Scenario** – There are four steps which the user has to do if he would like to create an order:
  - Product selection and date selection.
  - Adding contact information, such as: first name, last name, email and phone number. All these parameters are mandatory.
  - Checking all the information in the order.
  - *Create order button* to create order.

UC10 View order info

- **Use Case** – After the order is created the user expects an option for checking the state of the order.
- **Scenario** – Having created an order the user gets an order number and automatically receives an email with all the information about the order. Also on *My order page* the user can check the state of the order. If the user visits *My order page* and inserts the order number, then he will view which products are ordered and the state of the order.

UC11 View cart

- **Use Case** – If the user visits *Cart page*, then he expects to be able to view all products which were added to the cart.
- **Scenario** – On the *Cart page*, the user can check all products in the cart that he wants to buy. Also he can view the following parameters for each product in the cart:
  - photograph,
  - name,
  - number of items which he wants to order,
  - price for one item,
  - price for all items,
  - *Remove button.*

5.3.2 Employee

UC12 Sign in

- **Use Case** – The administrator expects, that he can sign in the System administration.
- **Scenario** – On the *Administration page* the administrator can sign in by filling the form with his username and password.

UC13 Sign out

- **Use Case** – The administrator expects, that he can sign out of the System administration.
- **Scenario** – If the administrator clicks on the *Logout* icon in an application bar, the administrator is logged out of the System administration.

UC14 Create a new product

- **Use Case** – On the *Product page* the administrator expects a button with the form to add a new product.
- **Scenario** – If the administrator clicks on the *Add product button* he is redirected to the form where the following parameters to be filled are:
  - Photo * – *Upload button* for cake photo.
  - Name * – Text input for product name.
  - Description * – Text area for note about product.
  - Price * – Number input for price of the product.
  - Vegan * – Checkbox if the product is vegan.
  - Allergens – Multiple choice select box with all allergens.
  - Production days * – Number input for days that are required to create a product.

  Where the parameter is marked by * it is mandatory to be filled in the form.

UC15 Update product

- **Use Case** – On the *Product page* the administrator expects that he can update the product.
- **Scenario** – After the administrator views the product detail, there is an option to update each parameter of the product. If the administrator updates a parameter and clicks on *Save button*, the product will be updated.

UC16 Delete product

- **Use Case** – On the *Product page* the administrator expects that he can delete the product.
- **Scenario** – If the administrator views the product detail, there is an option to delete the product. If the administrator clicks on the *Delete button*, the confirmation window will appear and if the user confirms the action, the product is deleted. If the administrator declines the confirmation window, the product will not be deleted.

UC17 Add picture to the gallery

- **Use Case** – The administrator expects that on the *Gallery page* he can add a new photo to the gallery.
- **Scenario** – On *Gallery page*, after the administrator clicks on the *Add picture button*, the form for uploading the photograph and with select category options:
  - product,
  - custom product,
  - wedding cake,
  - others

  will appear.

UC18 Delete picture from gallery

- **Use Case** – On the *Gallery page* the administrator expects that he can delete the photograph from the gallery.
- **Scenario** – If the administrator views the photograph detail, there is an option to delete the photograph. If the administrator clicks on the *Delete button*, the confirmation window will appear and if the administrator confirms the action, the photograph is deleted. If the administrator declines the confirmation window the photograph will not be deleted.

UC19 View all administrators

- **Use Case** – The administrator excepts, that the System provides viewing all administrators of the System.
- **Scenario** – If the administrator visits *Settings page* there is a list of all administrators of the System. For each administrator there is a username and *Delete button*.

UC20 Create new administrator

- **Use Case** – The administrator excepts, that he can add a new administrator of the System.
- **Scenario** – If the administrator visits *Settings page*, there is a form where he can fill administrator's username and password. There is also *Create button* and if the administrator clicks on this button, a new administrator will be created.

UC21 Delete administrator

- **Use Case** – The administrator excepts, that he can delete an administrator of the System.
- **Scenario** – If the administrator visits *Settings page*, next to the username in the list there is *Delete button*. If the administrator clicks on the button, the administrator whose button was clicked on is deleted. The administrator cannot delete his own account.

UC22 Change password

- **Use Case** – The administrator excepts, that he can change the account password.
- **Scenario** – If the administrator visits *Settings page*, there is a form with two password inputs. If the administrator fills both inputs with the same password and clicks on the *Save button*, a new password is set up.

UC23 View orders

- **Use Case** – On the *Orders page* the administrator expects, that he can view the list of all orders.
- **Scenario** – If the administrator visits *Orders page* then he views a table with the following columns:
  - order number,
  - creating order date,
  - name of the customer,
  - customer's phone number,
  - number of items,
  - pick up date,
  - order state.

UC24 View order detail

- **Use Case** – After clicking on any order at *Orders page* the administrator expects more details of the order.
- **Scenario** – On the *Orders page* after clicking on the specific order the administrator views a detail of the order with following information:
  - order number,
  - creating order date,
  - name of the customer,
  - customer's phone number,
  - customer's email address,
  - number of items,
  - pick up date,
  - order state,
  - list of all items,
  - *Decline order button*,
  - *Confirm order button*.

UC25 Manage order

- **Use Case** – On the *Order detail page* the administrator expects, that he can edit order.
- **Scenario** – After the administrator clicks on the *Edit button* in order detail view, then he can edit the order by:
  - adding new product,
  - remove product,
  - change number of items of the products,
  - add new custom product,
  - change customer's contact information,
  - change pick up date,
  - decline order,
  - confirm order,
  - save order.

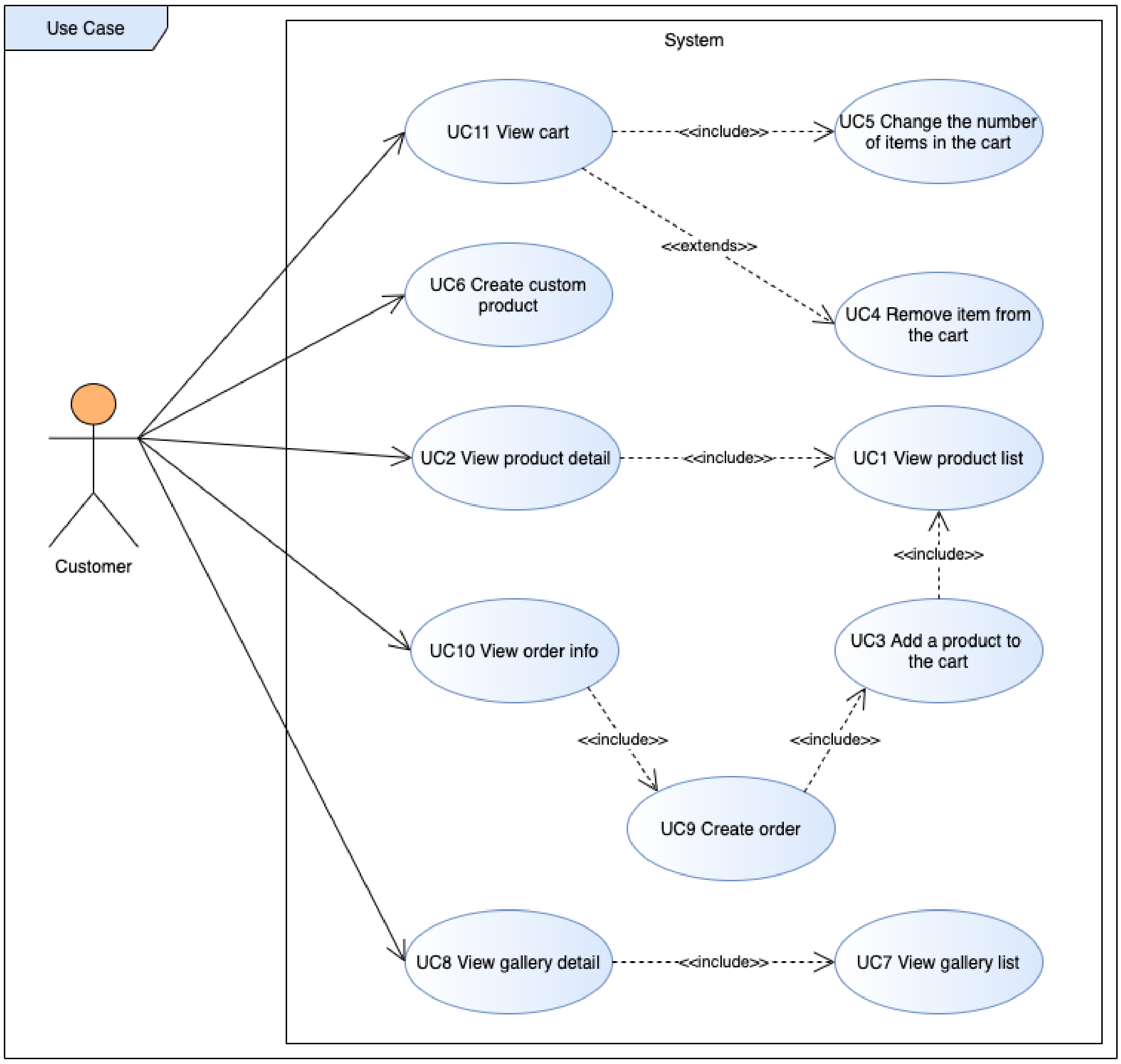


Figure 5.1: Use Case Diagram – Customer

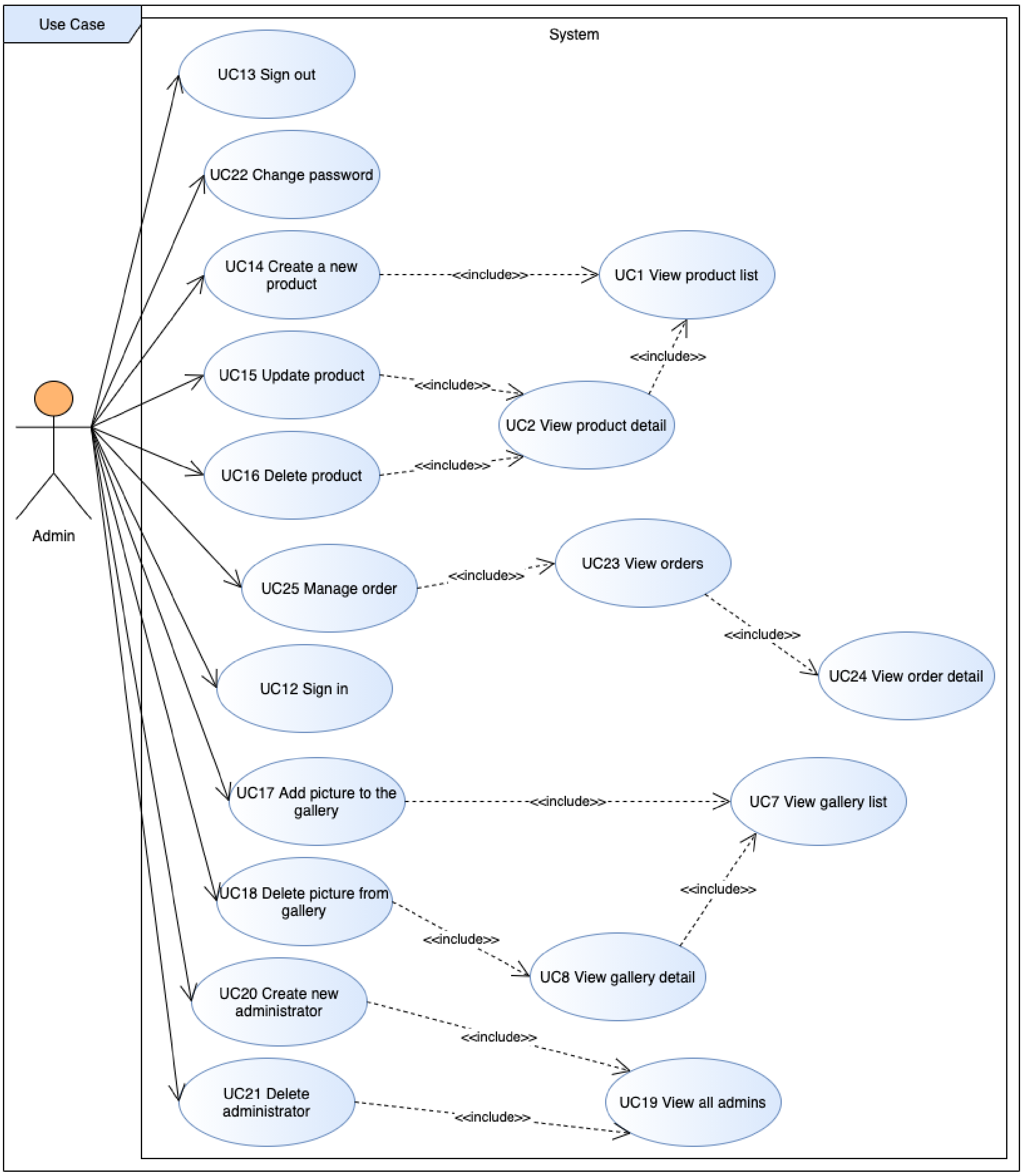


Figure 5.2: Use Cases Diagram – Administrator

5.4 Use Case coverage

In Table 5.1 it is shown that all functional requirements are covered by Use Cases. That means for each System's functional requirement there is one or more Use Cases which meet this requirement.

	FR1	FR2	FR3	FR4	FR5	FR6	FR7	FR8	FR9	FR10	FR11
UC1	✓										
UC2	✓										
UC3				✓							
UC4				✓							
UC5				✓							
UC6			✓								
UC7		✓									
UC8		✓									
UC9					✓						
UC10					✓						
UC11				✓							
UC12						✓					
UC13						✓					
UC14							✓				
UC15							✓				
UC16							✓				
UC17								✓			
UC18								✓			
UC19										✓	
UC20										✓	
UC21										✓	
UC22									✓		
UC23											✓
UC24											✓
UC25											✓

Table 5.1: Functional Requirements coverage

Chapter 6

Architecture

The steps of the author's process of developing this System are described in this Chapter. In Section 6.1 there is a description of the choice of the appropriate platform. Then the design of the User Interface in the form of Wireframes is described in Section 6.2. In Section 6.3 the architecture of the whole System is designed and in Section 6.4 there is a description of the testing of the System. In the end, in Section 6.5 there is a description of the System deployment.

6.1 Platform

The goal of this System is to help the Confectionery increase their revenue by providing their customers to create online orders and their employees provide order management. It is assumed that most System users will be the customers who will create their orders. The suitable platform for this problematic is a web application. Both customers and employees can easily run an application on all devices, such as, PC, laptop, tablet, mobile. To run the web application it is necessary to use only a browser and there is nothing to have to be installed.

6.2 Wireframes

The whole user interface was designed as Wireframe in balsamiq available at `https://balsamiq.com`. Using this tool it was easy to design the user interface and modify it according to the customer requirements. Modifying user interface can last a few man-days in implementation, but to modify Wireframe lasts only a few minutes or hours.

The System provides Confectionery customers to view products, gallery, create custom cakes and create an order. There are also services for Confectionery employees to manage all System content, check and manage orders. The main Wireframes of the user interface will be shown in the following sections, and the rest of the Wireframes are shown in the Appendix B.

6.2.1 Basic User Interface

The System does not have many web pages, there have to be pages for the following content:

- Homepage with basic information about the Confectionery.
- Product list and product detail.
- Page where customer can create their custom cake.
- Gallery of products.
- Order list and order management for the administrator.

All these links for these web pages are in the horizontal menu on the top of the web page. Below the menu there is a content of the application which is mentioned in the previous list. On the bottom of the web page there is a footer with links to the terms and conditions, contact, order tracking and login to the administration.

6.2.2 Homepage

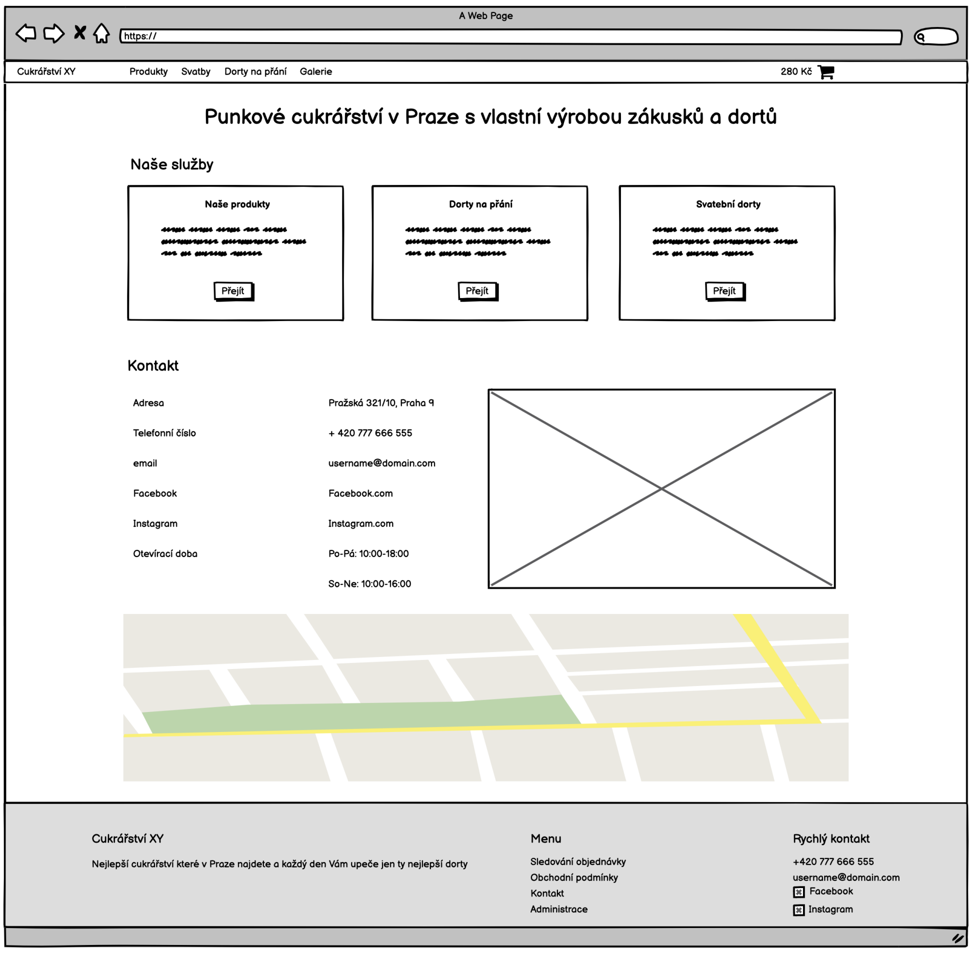
At the homepage there are all the basic information which the user needs when visiting the website. There are services which the Confectionery provides and a link to their web pages shortly described. Next, there is a Confectionery's contact which includes a phone number, an email, a link to social media and opening hours. There is also a map with the location of the Confectionery with a link to Google Maps and a picture of the sweets to attract the Confectionery customers. This layout is shown in Figure 6.1.

6.2.3 Products

At the product page there is a list of products, each product has a a photograph, name, description and price. Customers can view details of the products if they click on them. A bigger photograph of the product, product name, price of the product, product description, information about allergens, if product is vegan and the number of the dates that are required to create the cake can be seen on the product detail. Product list page is shown in Figure 6.2 and product detail is shown in Figure 6.3.

6.2.4 Custom cakes

There is also a page, where the Confectionery customers can create cake by themselves, for example, a birthday cake or wedding cake. On this page there are several select boxes for each cake's parameter where the Confectionery customer can choose one of available variants. The Confectionery customers can also describe their idea about the cake in the text area and attach a photograph with the similar cake for inspiration. This form is shown in Figure 6.4.


A Web Page

https://

Cukrářství XY · Produkty · Svatby · Dorty na přání · Galerie · 280 Kč

Punkové cukrářství v Praze s vlastní výrobou zákusků a dortů

Naše služby

Naše produkty — Přejít

Dorty na přání — Přejít

Svatební dorty — Přejít

Kontakt

Adresa	Pražská 321/10, Praha 9
Telefonní číslo	+ 420 777 666 555
email	username@domain.com
Facebook	Facebook.com
Instagram	Instagram.com
Otevírací doba	Po-Pá: 10:00-18:00
	So-Ne: 10:00-16:00

Cukrářství XY

Nejlepší cukrářství které v Praze najdete a každý den Vám upeče jen ty nejlepší dorty

Menu

Sledování objednávky

Obchodní podmínky

Kontakt

Administrace

Rychlý kontakt

+420 777 666 555

username@domain.com

Facebook

Instagram

Figure 6.1: Wireframe – Homepage

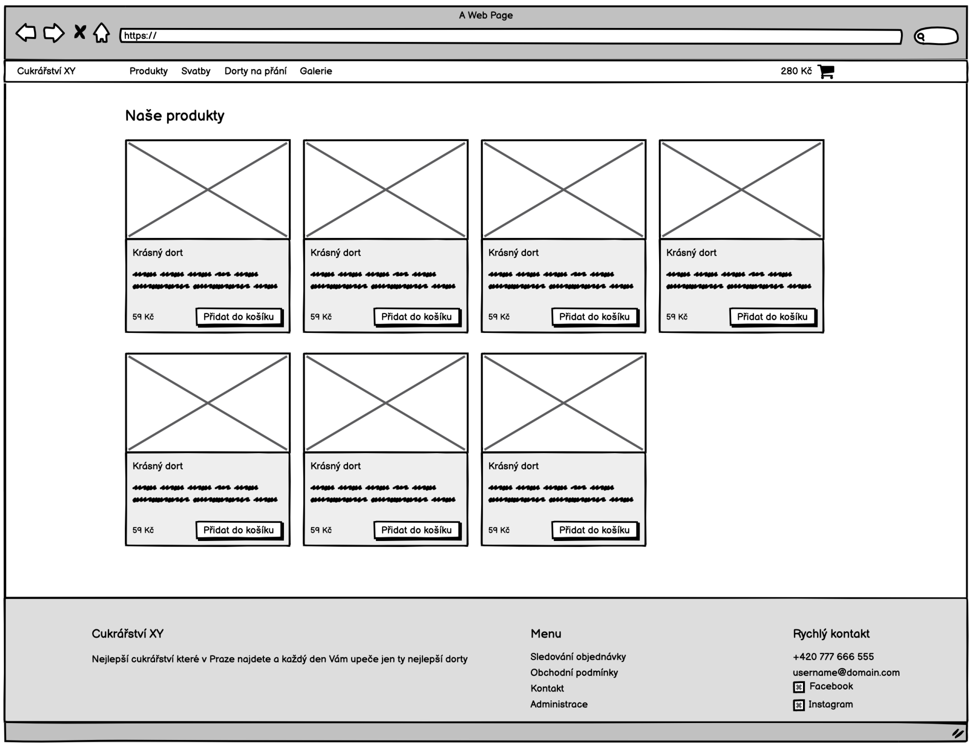


Figure 6.2: Wireframe – Product list

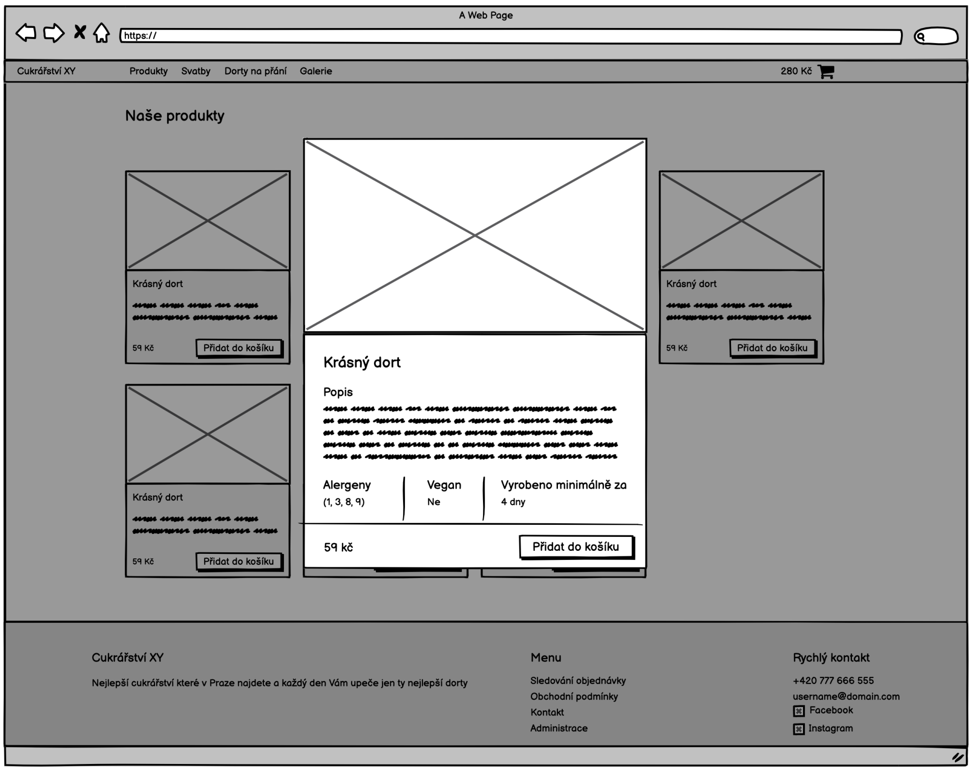


Figure 6.3: Wireframe – Product detail

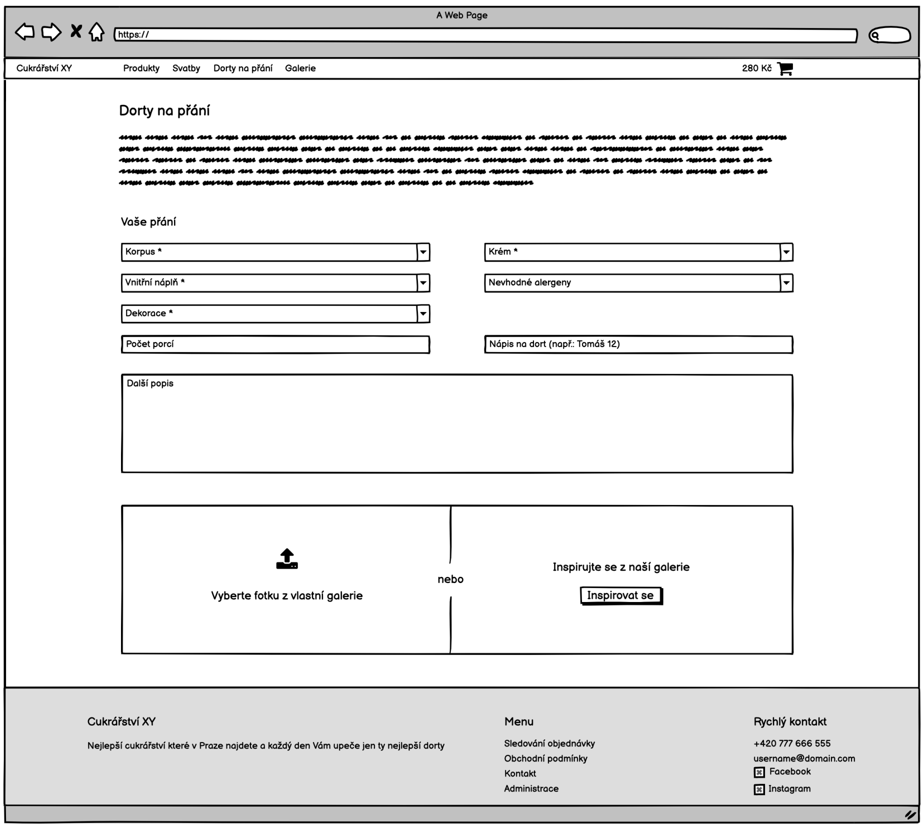


Figure 6.4: Wireframe – Creating a custom cake

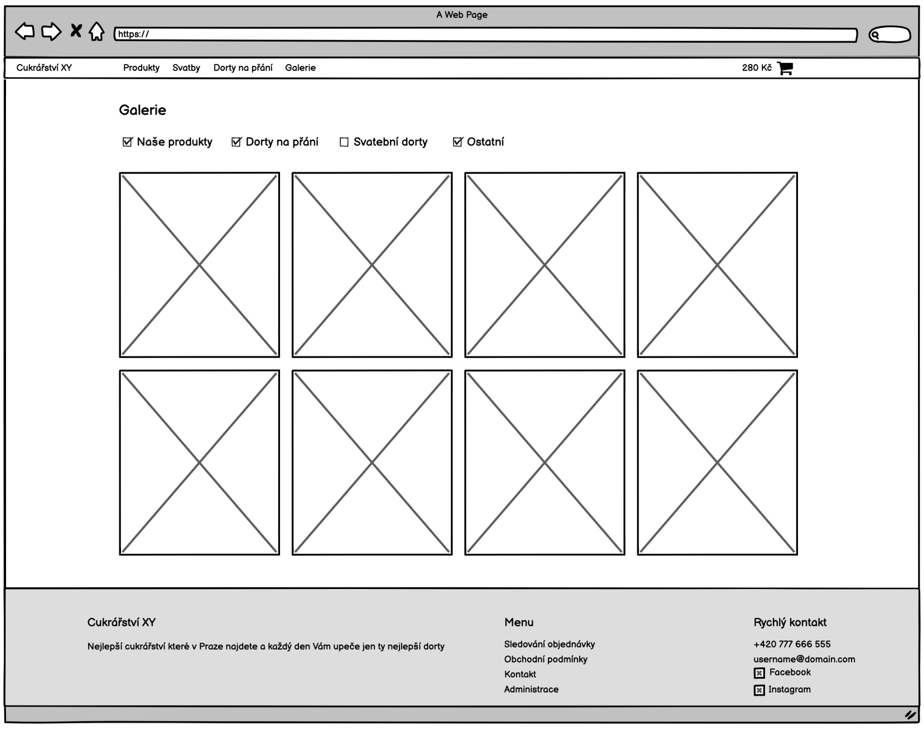


Figure 6.5: Wireframe – Gallery

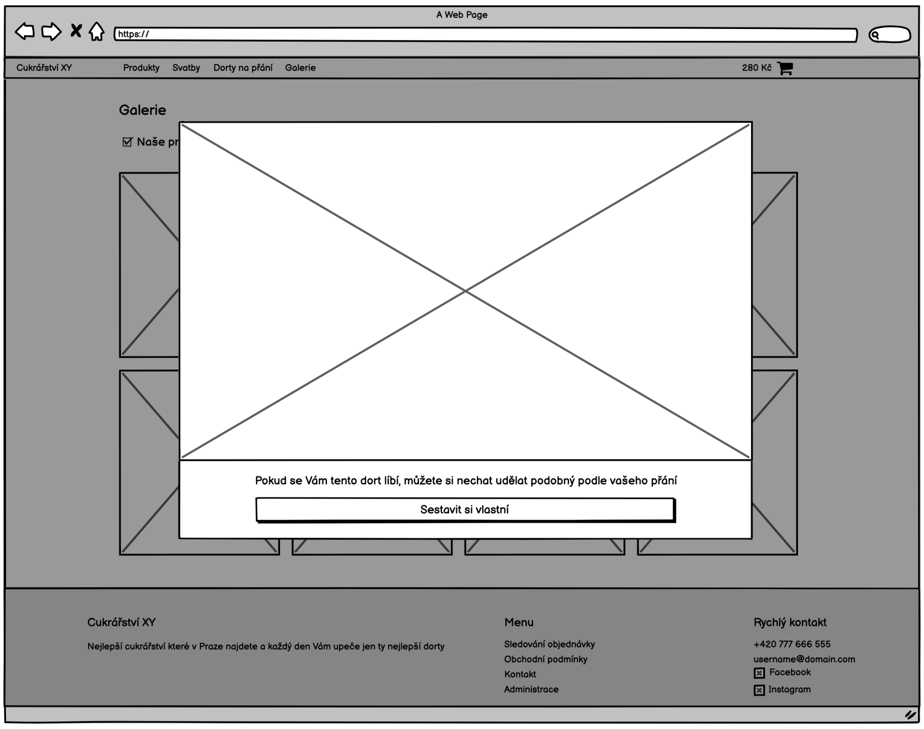


Figure 6.6: Wireframe – Gallery's picture detail

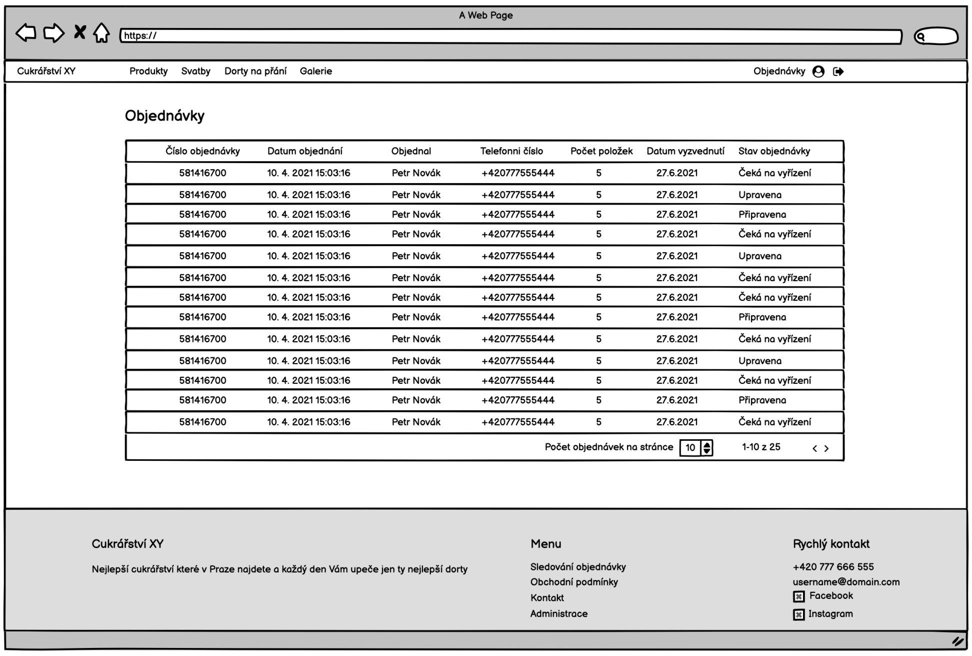
6.2.5 Gallery

Sweets are on the gallery page which is shown in Figure 6.5. The Confectionery customer can click on the photograph and view bigger photograph of the product, this Wireframe is shown in Figure 6.6. If the Confectionery customer likes a cake, he can click on the button below the picture to be redirect to the Custom cake page with the photo from the gallery and he will be able to order a similar cake.

6.2.6 Orders

The administrator needs to view all orders and therefore there is an Order page shown in Figure 6.7 where the administrator is automatically redirected after successful signing in. On this page there is a table of orders and administrator can easily filter and sort it by order's parameters. If the administrator clicks on the order, then he can manage it:

- accept the order,
- cancel the order,
- update Confectionery customer's contact information,
- change the number of items in the order,
- add a new product or a custom product.



Objednávky

Číslo objednávky	Datum objednání	Objednal	Telefonni číslo	Počet položek	Datum vyzvednutí	Stav objednávky
581416700	10. 4. 2021 15:03:16	Petr Novák	+420777555444	5	27.6.2021	Čeká na vyřízení
581416700	10. 4. 2021 15:03:16	Petr Novák	+420777555444	5	27.6.2021	Upravena
581416700	10. 4. 2021 15:03:16	Petr Novák	+420777555444	5	27.6.2021	Připravena
581416700	10. 4. 2021 15:03:16	Petr Novák	+420777555444	5	27.6.2021	Čeká na vyřízení
581416700	10. 4. 2021 15:03:16	Petr Novák	+420777555444	5	27.6.2021	Upravena
581416700	10. 4. 2021 15:03:16	Petr Novák	+420777555444	5	27.6.2021	Čeká na vyřízení
581416700	10. 4. 2021 15:03:16	Petr Novák	+420777555444	5	27.6.2021	Čeká na vyřízení
581416700	10. 4. 2021 15:03:16	Petr Novák	+420777555444	5	27.6.2021	Připravena
581416700	10. 4. 2021 15:03:16	Petr Novák	+420777555444	5	27.6.2021	Čeká na vyřízení
581416700	10. 4. 2021 15:03:16	Petr Novák	+420777555444	5	27.6.2021	Upravena
581416700	10. 4. 2021 15:03:16	Petr Novák	+420777555444	5	27.6.2021	Čeká na vyřízení
581416700	10. 4. 2021 15:03:16	Petr Novák	+420777555444	5	27.6.2021	Připravena
581416700	10. 4. 2021 15:03:16	Petr Novák	+420777555444	5	27.6.2021	Čeká na vyřízení

Počet objednávek na stránce 10 1-10 z 25

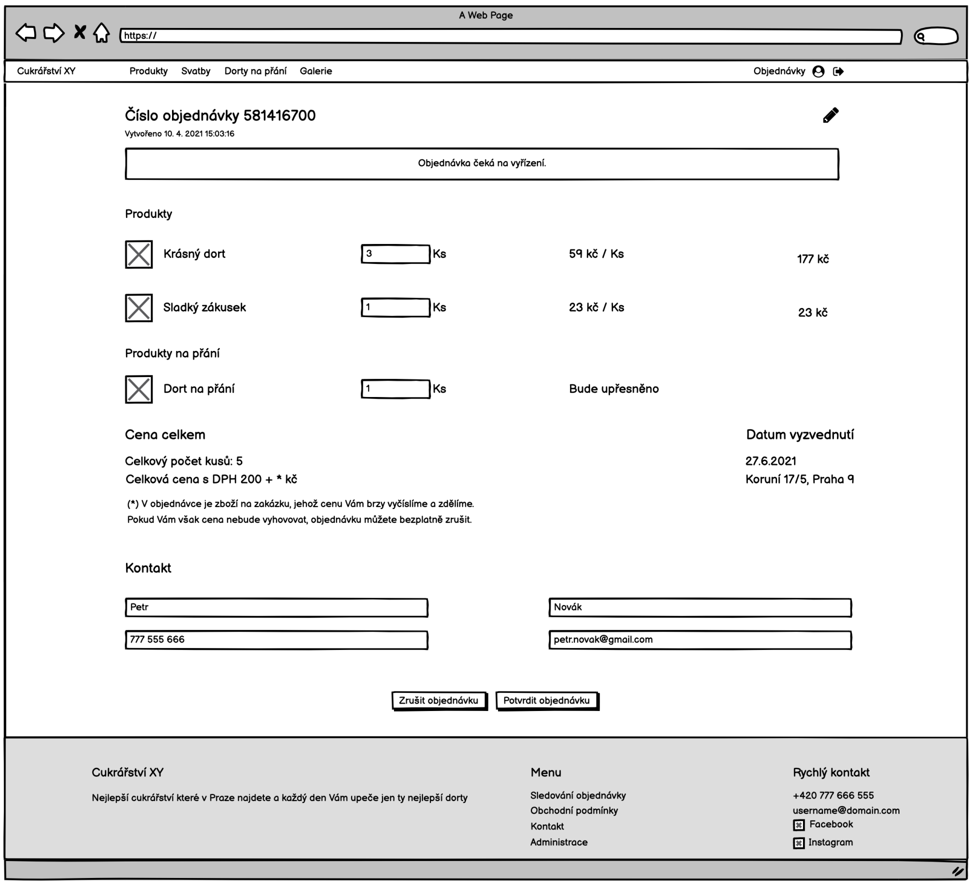
Cukrářství XY

Nejlepší cukrářství které v Praze najdete a každý den Vám upeče jen ty nejlepší dorty

Rychlý kontakt

+420 777 666 555
username@domain.com

Figure 6.7: Wireframe – Order list



Číslo objednávky 581416700

Vytvořeno 10. 4. 2021 15:03:16

Objednávka čeká na vyřízení.

Produkty

Krásný dort	3 Ks	59 kč / Ks	177 kč
Sladký zákusek	1 Ks	23 kč / Ks	23 kč

Produkty na přání

Dort na přání	1 Ks	Bude upřesněno	

Cena celkem

Celkový počet kusů: 5
Celková cena s DPH 200 + * kč

(*) V objednávce je zboží na zakázku, jehož cenu Vám brzy vyčíslíme a zdělíme.
Pokud Vám však cena nebude vyhovovat, objednávku můžete bezplatně zrušit.

Datum vyzvednutí

27.6.2021
Koruní 17/5, Praha 9

Kontakt

Petr	Novák
777 555 666	petr.novak@gmail.com

Cukrářství XY

Nejlepší cukrářství které v Praze najdete a každý den Vám upeče jen ty nejlepší dorty

Rychlý kontakt

+420 777 666 555
username@domain.com

Figure 6.8: Wireframe – Administrator – detail of a new order

6.2.7 Database model

This section describes what data are stored in the database. The goal of the System is to provide the list of Confectionery products and gallery and enable customers to create an order online on the website. An order contains products and custom products. Products are created by the administrators and custom products are created by Confectionery customers.

Each product and custom product contain a photograph and the allergen list. Moreover, custom products contain information of ingredients which are selected from the menu of available ingredients. This menu is managed by the administrator of the System.

Customers are enabled to create an order. Each order contains a list of products and a list of custom products.

There is also a gallery for present Confectionery photographs, each photograph has a category from the gallery category list.

The database model is shown in Figure 6.9.

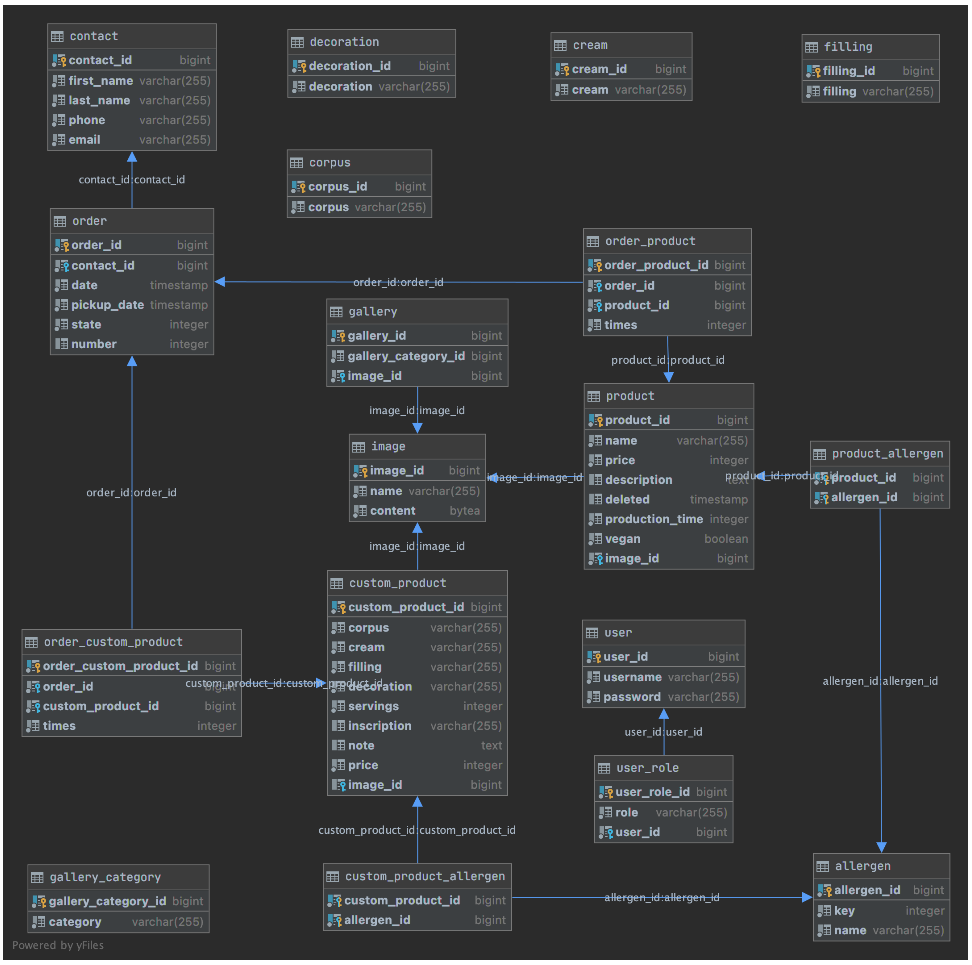


Figure 6.9: Database model

6.3 Architectural model

The Information System is divided to following three parts:

- Server,
- Client,
- Database.

All the data which need to be persisted are stored in the database. As database PostgreSQL database system has been selected.

The Client contains the entire user interface and responds to the user's activities. The Client is written in JavaScript using React and Redux library.

The goal of the Server is to respond to the Client requests. The Server also communicates with the Database because it is necessary to fetch or store data to the Database for most of the requests. The Server is programmed in Spring Boot framework and written in the Kotlin language.

The Client and the Server communicate with each other through the network using http protocol and REST.

6.3.1 Client

The Client has been developed in React library using Redux state management library. The client architecture is divided into four parts, which are described in the following sections and shown in Figure 6.10.

The Client consists of many components. Each component can be nested to the another component. Within the component or nested components is the state of the data managed by React Hooks. Redux library is used if the state of the data need to be shared among independent components or data need to be fetched from the Server.

Store

All data which have to be shared among more components or have to be fetched from the Server are persisted in the store. This data are stored as JSON in the tree structure. For a nice view on the stored data Redux DevTools Chrom extension was used. This extension makes debuging and developing much easier.

Reducer

Reducer manages data in the Store and is called from the Actions with JSON data and the action. To keep code clean, Reducer is divided to many Reducers according to the type of the data. For example, there is `UserReducer` to manage data regarding users and `ProductReducer` to manage data regarding products. Each Reducer is implemented as a switch which decides by the action which it receives.

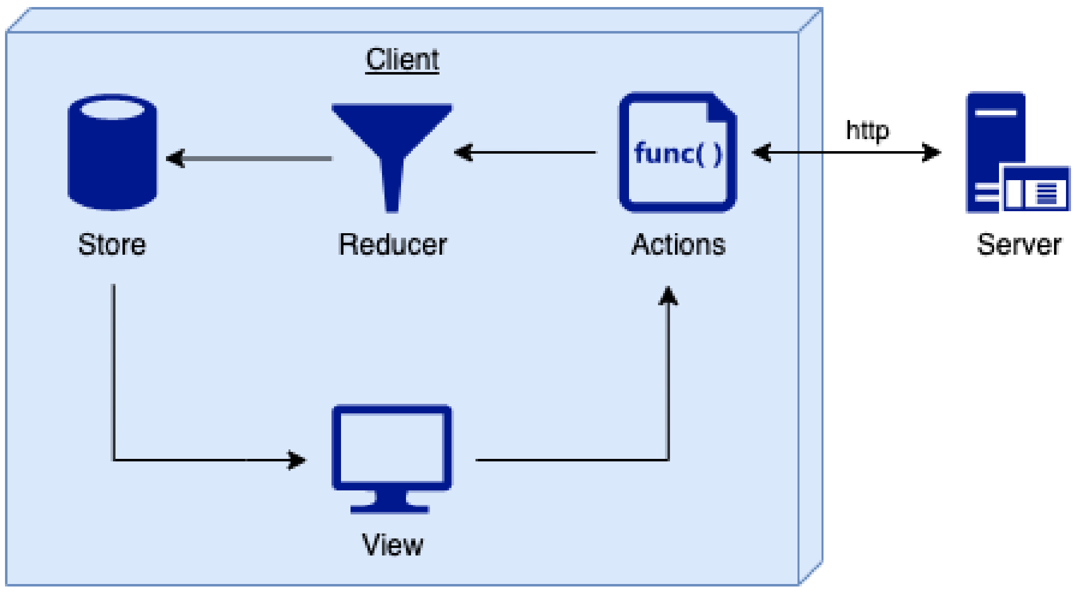


Figure 6.10: Client – architecture

Actions

Actions are methods that can fetch data from the Server and transmit them to the Reducer with specific action. Actions methods are called from the View and can include parameters, for example, id of the product which needs to be fetched from the Server.

View

A View renders content of the Client including data from the Store and reacts to the user's actions. If a situation when data need to be stored or fetched from the Server occurs, View calls function from the Actions to do that.

6.3.2 Server

The Server is responsible for business logic and communication with database and Client. Server and Client communicate through the REST interface which Server provides. Unlike the Client, the Server does not run in the users computer. It runs in some virtual server and there is only one or a few instances of the Server.

The Server is developed in Kotlin language using Spring Boot framework. The architecture of the Server is described in the following sections and also shown in Figure 6.11.

Entity

The data in the database are mapped to the Entity classes. That means that each non-binding table in the database is represented by single Entity classes. Each Entity class has `@Entity` annotation and contains id. Columns in the database correspond to class variables and each row in the database represents one instance of the class.

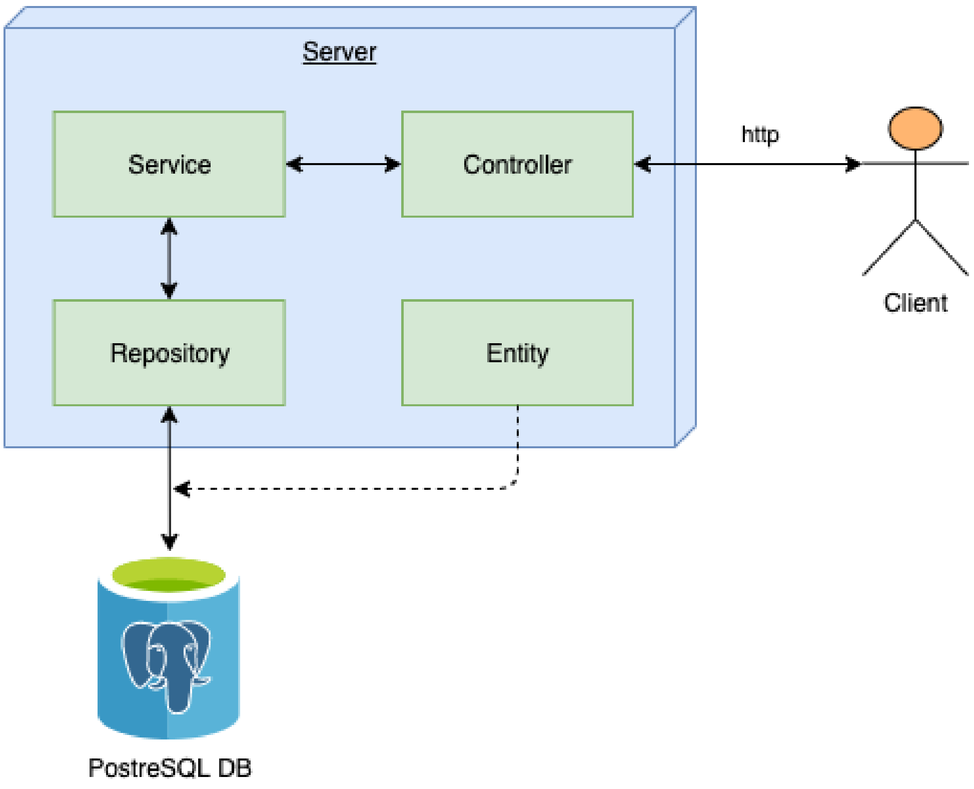


Figure 6.11: Server – architecture

Repository

Each table in the database responds to the entity classes. Repository grants that databases tables are mapped to the Entity classes and provides basic methods, such as:

- `findAll` – find all records in the database of specific entity,
- `findById` – find record in the database according to specific id of specific entity,
- `save` – create or update record in the database of specific entity,
- `deleteById` – delete record in the database of specific entity according to specific id.

To make a class Repository there have to be added `@Repository` annotation and class must be extended by `JpaRepository<T, ID>` where `T` is Entity class and `ID` is data type of Entity id. For example:

```
@Repository
interface ProductRepository : JpaRepository<Product, Long>
```

This interface provides basic CRUD methods which were introduced in the previous list.

To implement Entities and Repositories in the `pom.xml` file must be added following dependency:

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
```

Service

Service is responsible for all business logic, that means data transforming, data validation, state validation etc. Service is called from the Controller or other Service and can communicate with the Database through the Repository interface. Each Service has `@Service` annotation.

Controller

Controller provides the Server's REST endpoints that the Client communicates with. Each request to the Server is managed by Controller. The Controller delegates Client's request to the Service. Controller is also bean because it is annotated by `@Controller` annotation. Controllers are annotated by `@RequestMapping("")` by their functionality.

Spring Security

There are not many alternatives how to secure Spring Boot application. The best practise is to use Spring Security framework. The whole authorization and authentization is managed by Spring Security. This framework basically provides security based on session but it was configured for the authorization and the authentization based on JWT. Because more instances of the Server will run in the Cloud, the solution based on the JWT is the right choice. For the use of Spring Security in the `pom.xml` file there has to be the following dependency added:

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-security</artifactId>
</dependency>
```

Some endpoints require authorization and that is managed by `@PreAuthorize` annotation. This annotation allows only the user with specific role or roles to execute this endpoint.

6.3.3 Database

As has been said, there is no problem to run PostgreSQL at many systems. For application development on macOS, PostgreSQL Docker container was used to run localhost database system and pgAdmin for database management, such as, DB backups, DB restore and data view. The PostgreSQL Docker container is available at `https://hub.docker.com/_/postgres` and to get it run is only Docker Desktop required which is available at `https://docs.docker.com/get-docker/`. For communication between Server and Database it is necessary to define PostgreSQL driver in `pom.xml` file:

```
<dependency>
    <groupId>org.postgresql</groupId>
    <artifactId>postgresql</artifactId>
    <scope>runtime</scope>
</dependency>
```

then define connection in `application.properties` as database address with port, database name, username and password. This configuration is shown in the following figure:

```
spring.datasource.url=jdbc:postgresql://localhost:5432/db-name
spring.datasource.username=username
spring.datasource.password=password
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect
.PostgreSQLDialect
```

6.4 Testing

It is really hard not to make any mistakes in the process of System development. There are lots of cases and scenarios which System must respond to. Users make lots of mistakes while using the System, and because of that it is very important to cover all cases and scenarios which can occur. This can be sometimes almost impossible to reach that.

A nice way how to avoid bugs is to test the System during development and also before the System is deployed to the production. There are lots of types of System testing, such as:

- **Functional testing** – Is a software testing that validates the software System against the functional requirements [38], for example:
  - unit testing,
  - integration testing,
  - blackbox testing,
  - whitebox testing.
- **User Acceptance Testing** – Is performed by the client to verify the software System before moving the software application to the production environment. [39]
- **Usability testing** – Is all about getting regular people to interact with a System and observing their behavior and reactions to it. [40]

6.4.1 Functional testing

Unit testing was used as functional testing which is described in Section 6.4.1 below and also Integration testing which is described below the Unit testing in Section 6.4.1.

Unit testing

Unit testing is a level of software testing where individual units or components of a software are tested. The goal of this type of testing is to validate that each unit of the software performs as designed. [41]

Unit testing was performed at the Server layer where all services were tested. For Unit testing only services are tested as a unit. Mocking is used for Repository simulation. For example a test of getting all products method is in the following example:

```
@Test
fun getAll_shouldPass() {
    val product1 = Product("product1", 100, "description1")
    val product2 = Product("product2", 200, "description2")
    val product3 = Product("product3", 300, "description3")
    val products = listOf(product1, product2, product3)
    `when`(productRepository.findAll()).thenReturn(products)

    productService.all

    verify(productRepository, times(1)).findAll()

    assertThat(
        "Size is not equal to 3.",
        productService.all.size,
        `is`(3)
    )
}
```

Integration testing

Integration testing is a software testing where individual units or components are combined and tested as a group. It helps to expose faults in the interaction among integrated units. [42]

Server layer was tested in the System, starting from the Controller to the Database. For this kind of testing a special empty in-memory database is used. In Integration testing tests run in a Spring context. The example of the test for a product creation API is in the following example:

```
@Test
fun create_shouldPass() {
    val product = Product("product1", 100, "description1")

    mockMvc.perform(MockMvcRequestBuilders.post("/product")
        .contentType(MediaType.APPLICATION_JSON)
        .content(objectMapper.writeValueAsString(product)))
        .andExpect(MockMvcResultMatchers.status().isOk)
        .andExpect(MockMvcResultMatchers
            .content()
            .contentTypeCompatibleWith(
                MediaType.APPLICATION_JSON))
        .andExpect(MockMvcResultMatchers
            .jsonPath("$.name").value("product1"))
        .andExpect(MockMvcResultMatchers
            .jsonPath("$.price").value(1))
        .andExpect(MockMvcResultMatchers
            .jsonPath("$.description").value("description1"))

    AssertionsForClassTypes
        .assertThat(productController.getOne(1L))
        .isNotNull
}
```

Tester ID	Age	Experience in IT	Profession
Anonym_1	25	5	Programmer
Anonym_2	23	4	Student
Anonym_3	26	5	Programmer
Anonym_4	24	5	Student
Anonym_5	25	6	Student
Anonym_6	24	5	Student
Anonym_7	29	5	Student
Anonym_8	20	1	Student
Anonym_9	24	5	Student

Table 6.1: Information about the testers

6.4.2 Usability testing

The System was tested within a team project in the lecture *NI-NUR* by nine users. The information about the testers are in Table 6.1.

Tasks

Each tester takes the following instructions:

1. Visit website at address: http://46.28.110.43:3000/ .
2. Choose one product and order 25 pieces of this product.
3. Order 5 different products and one custom product.
4. Select a wedding cake in the gallery and order a similar one with any parameters. Then to go to the *Cart page* and verify cake's parameters. After that alter some parameters.
5. Add several products to the cart, then go to the *Cart page* and change cake's parameters, delete a cake from the cart and add a next cake to the cart.
6. Delete all items which are in the cart.
7. Find the address where the cakes will be picked up.

Report

All testers successfully completed all tasks with minimal problems. Some tasks had to be specified so that the testers knew what exactly to do.

The biggest problem was with task 2. Testers tried to edit input but the System did not support that operation. The testers with small displays had a problem with huge photographs which covered their entire screen.

Some Testers expected that if they would take inspiration of a cake in the gallery the parameters would be automatically filled in the form. Three of Testers did not know what the *number of pieces* parameter in the form at *Custom cake page* was. Two testers tried to find the button which would empty the entire cart at the same time.

Feedback implementation

New following features were implemented based on this feedback:

- The number of product pieces can be edited like input.
- Photographs are responsive.
- Added note how custom cakes form works.
- Added note what *number of pieces* parameter in the custom cakes form means.

6.4.3 User Acceptance Testing

Before the System is deployed to the production environment the customer will do Acceptance Testing to make sure that functionality, usability, performance, and user interface of the System correspond to the previous arrangement.

6.5 Deployment

A production version of the System has to be available through the internet network. The System consists of three runnable parts (Server, Client, Database) which must be deployed. The Database and the Server are deployed on the Cloud using Kubernetes and Docker. While the Client does not need the cloud benefits, such as, scalability he is deployed on the VPS because the author of this thesis subscribes one.

6.5.1 Docker

As it was mentioned to run PostgreSQL on Docker is very easy. PostgreSQL Docker image can be pulled by executing `docker pull postgres` command. To run this container, there have to be configured environment variables:

- POSTGRES_PASSWORD=password
- POSTGRES_USER=username
- POSTGRES_DB=db-name

There can be configured on which port in container PostgreSQL server will run. Also a port can be configured on which PostgreSQL server will be exposed on the localhost machine.

Volumes provide persisting data generated by and used by Docker containers. [43] Every time when a new PostgreSQL container runs, without specifying a `--volume` option, Docker automatically creates a new volume. Because of that `--volume` parameter is set to `pgdata:/var/lib/postgresql/data`.

To run the Server the Docker image has to be created first. To create Docker image Dockerfile must be defined. The Server's Dockerfile is shown in the following code:

```
FROM openjdk:14-jdk-alpine
EXPOSE 8080
ARG JAR_FILE=target/*.jar
ADD ${JAR_FILE} app.jar
ENTRYPOINT ["java","-jar","/app.jar"]
```

For simple Docker build `dockerfile-maven-plugin` was used. With this plugin a Docker image is automatically created with executing `mvn install` command.

Both images run in the different containers and to enable communication with each other both containers must run on the same network. The network is created by executing `docker network create network-name` command and configuration `-network=network-name` is added to both Docker run commands.

There are lots of configurations in both parts and because of that the `docker-compose.yaml` file was added. This file contains configuration which is required for both containers launching. With this file it is possible to run both containers only by executing `docker-compose up` command.

6.5.2 Cloud

As the cloud provider the Google Cloud was chosen and it is available at `https://cloud.google.com/` address. Kubernetes Cluster with three nodes was created in the Cloud. To deploy, run and manage Docker images in the Cloud the Gcloud and the Kubectl were installed and they are available at `https://cloud.google.com/sdk/docs/downloads-interactive` and `https://kube\rnetes.io/docs/tasks/tools/`. With these tools it is simple to manage containers in the Cloud through the command line in the local machine.

Docker images are deployed from the Dockerhub to the Cloud, which means that the Docker image must be first uploaded to the Dockerhub. To do that the Dockerhub account must be created and then by executing `docker push username/image-name` command the Docker image is upoaded to the Dockerhub repository. To deploy a new Docker image version from the Dockerhub to the Cloud, Docker Kompose was used and it is available at `https://kompose.io/`. This tool translates `docker-compose.yaml` file into Kubernetes resources. With this tool is possible to upload a new Docker image from Dockerhub to the Cloud only by executing `kubectl apply -f app-deployment.yam` command.

Chapter 7

Further System development and Discussion

In this Chapter in Section 7.1 and 7.2 the current state of the Information System is described. Next, in Sections 7.3, 7.4, and 7.5 it is mentioned how the System will be maintained and updated in the future.

7.1 Hi-fi prototype

Based on Wireframes, Analysis and Architecture the Hi-fi prototype was created, which covered all functional and non-functional requirements and Use Cases. This prototype was presented to the customer. The customer had a few functional requirements:

- each product should have information about production time,
- smaller photographs in the product detail and gallery detail,
- custom form should have select boxes containing cake parameters which can be managed by the administrator.

These all customer's requirements were successfully implemented. Hi-fi prototype is available at `http://46.28.110.43:3000/` and screenshots of the prototype are in the Appendix C.

7.2 Current System state

The System is ready to be presented to the customer as a final version. The customer should accomplish User Acceptance Testing of the this version before it is deployed to the production.

If the customer is satisfied with all functional and non-functional requirements the System will be deployed to the cloud as product version and it should be ready to be used by Confectionery customers. Otherwise customer can have some issues or additional functional requirements. Issues should be fixed as soon as possible and in case of additional requirements it depends on the customer if they would like to run the System without these requirements and wait for another version of the System with these requirements or wait for implementation of requirements before the System is deployed to the production.

7.3 Maintenance services

After the System is deployed to the product version, maintenance and support will begin. Maintenance is necessary to modify and continuously update the System to eliminate all possible errors, malfunctions and to improve System performance.

7.4 Support

The System can have some bugs and can became inapplicable for customers or administrators. Therefore support of the System is necessary. Bugs are divided to the four categories by their seriousness:

- **Priority 1** – The System does not work and any service cannot be used.
- **Priority 2** – Some services do not work and the System is partly unfunctional.
- **Priority 3** – Bug limiting usage of the System and all services are still available.
- **Priority 4** – Minor shortcomings and the System can be used without any problems.

To fix these bugs, the customer must contact the author and report the bug with corresponding priority. The author should fix this bug in the time which is defined for each priority in Table 7.1.

7.5 New functionality

If the customer needs some new functionality or change the user interface, this does not come under the maintenance or support. To implement new features will be charged extra. The costs of the new features will be charged by the time being spent by implementation of specific requirement and there is not a fixed deadline to implement these requirements.

Priority	Reaction time	Reporting method
1	Within 8 hours of reporting the failure on working days during normal working hours (8-16). Within 1 day of reporting the failure on weekdays except for business hours, on weekends and public holidays.	Must be performed by the customer at the telephone number: 774 218 772 and reported as an INCIDENT issue.
2	Within 1 working day from the reporting of the failure on working days during normal working hours (8-16). Within 2 working days of reporting the failure on weekdays except for business hours, on weekends and public holidays.	Must be performed by the customer at the telephone number: 774 218 772 and reported as a CRITICAL issue.
3	Within 4 working days of reporting the failure.	Must be made by the customer at the telephone number: 774 218 772 and reported as a NORMAL issue.
4	Within 6 working days of reporting the failure.	Must be performed by the customer at the telephone number: 774 218 772 and reported as a MINOR issue.

Table 7.1: Troubleshooting

Chapter 8
Conclusion

The main goal of this thesis was to design the Information System for the Confectionery. This goal was successfully fulfilled. The analysis of the requirements was accomplished. As a result of this analysis functional and non-functional Requirements, Use Cases, and Scenarios were created. Then the user interface Wireframe form was created. The System was presented in the form of this reactive paper prototype and consulted with the client. On the basis of this feedback the functional requirements and user interface were adjusted according to the customer's requests. After that, the architecture of the System was designed. On the basis of the architecture, the System was implemented and tested. The testing was performed by automation tests and by the real users. The current version of the System is deployed as Hi-fi prototype and is available at the address `http://46.28.110.43:3000/`. The System is currently in the state before acceptance tests by the customer. After finishing these tests, the System can be deployed to the production.

In this thesis the business plan is created, too. Both, on the basis of this experience and the following reference to start with setting up a new business with the development of Information Systems for a smaller entrepreneur. The author of this thesis created the Business Model for the customer and designed the timetable for this project. Then the risks including their mitigations are considered. In the end, the Return of Investment is described on the basis of the Financial Plan. In the Return of Investments an the optimistic as well as pessimistic model for the following 5 years is proposed.

The author followed the timetable that had been designed. All goals of this timetable were successfully fulfilled except for the deployment. Although the system is prepared for the production usage it has not been able to be deployed into the production because of the COVID-19 situation. For the System production deployment it is necessary to wait until the situation calms down.

Provided the customer was not interested in the System in this situation for any reason, the author would try to find another entrepreneur to whom the System would help to improve their business. It would only be sufficient to adjust the System to the new customer and then deploy it to the production. Even if such an entrepreneur was not found the author could use the current Hi-fi prototype as a smaller reference and continue according to the plan.

This thesis was of benefit to the Author. Not only did he gain knowledge in modern technologies, such as, Docker, Kubernetes, Cloud but he also improved his experience with technologies, such as, Spring Boot and React that he has already worked with. The next immense experience for the author was the cooperation with the real customer. The author found the customer, collected his requirements for the System, and in the course of the development consulted the current state of the Information System with the customer.

Bibliography

[1] Free Gantt Chart Excel Template. [online]. 2021, [cit. 4. 5. 2021]. Available from: https://www.teamgantt.com/free-gantt-chart-excel-template

[2] Jakub Romanowski. Which Major Companies Use PostgreSQL? What Do They Use It for?. [online]. 2021, [cit. 26. 4. 2021]. Available from: https://learnsql.com/blog/companies-that-use-postgresql-in-business/

[3] Adam Hayes. Business Plan. [online]. 2021, [cit. 11. 4. 2021]. Available from: https://www.investopedia.com/terms/b/business-plan.asp

[4] What are some business model examples?. [online]. 2021, [cit. 11. 4. 2021]. Available from: https://www.aha.io/roadmapping/guide/product-strategy/what-are-some-examples-of-a-business-model

[5] Communicate Your Idea Clearly and Concisely. [online]. 2021, [cit. 16. 4. 2021]. Available from: https://leanstack.com/lean-canvas

[6] What is a Gantt Chart?. [online]. 2021, [cit. 16. 4. 2021]. Available from: https://www.gantt.com/

[7] Software Analysis Design Tools. [online]. 2021, [cit. 16. 4. 2021]. Available from: https://www.tutorialspoint.com/software_engineering/software_analysis_design_tools.htm

[8] Functional and Nonfunctional Requirements: Specification and Types. [online]. 2021, [cit. 17. 4. 2021]. Available from: https://www.altexsoft.com/blog/business/functional-and-non-functional-requirements-specification-\and-types/

[9] Use Cases. [online]. 2021, [cit. 17. 4. 2021]. Available from: https://www.usability.gov/how-to-and-tools/methods/use-cases.html

[10] Use Case Diagram. [online]. 2021, [cit. 19. 4. 2021]. Available from: https://www.smartdraw.com/use-case-diagram/

[11] What’s is the difference between include and extend in use case diagram?. [online]. 2021, [cit. 19. 4. 2021]. Available from: https://courses.cs.ut.ee/MTAT.03.083/2015_fall/uploads/Main/ExtendInclude.pdf

[12] Jaye Hannah. What Exactly Is Wireframing? A Comprehensive Guide. [online]. 2021, [cit. 21. 4. 2021]. Available from: https://careerfoundry.com/en/blog/ux-design/what-is-a-wireframe-guide/

[13] Maria Myre. The 8 best wireframe tools in 2021. [online]. 2021, [cit. 5. 6. 2021]. Available from: https://zapier.com/blog/best-wireframe-tools/

[14] Bass, L.; Clements, P.; et al. *Software Architecture in Practice*. Addison-Wesley Professional, third edition, 2012, ISBN 0321815734.

[15] Architecture Models. [online]. 2021, [cit. 5. 6. 2021]. Available from: https://www.tutorialspoint.com/software_architecture_design/architecture_models.htm

[16] Prototyping. [online]. 2021, [cit. 22. 4. 2021]. Available from: https://www.usability.gov/how-to-and-tools/methods/prototyping.html

[17] Eleonora Ibragimova. High-fidelity prototyping: What, When, Why and How?. [online]. 2021, [cit. 22. 4. 2021]. Available from: https://blog.prototypr.io/high-fidelity-prototyping-what-when-why-and-how-\f5bbde6a7fd4

[18] FAQ. [online]. 2021, [cit. 23. 4. 2021]. Available from: https://kotlinlang.org/docs/faq.html

[19] What Spring can do. [online]. 2021, [cit. 1. 5. 2021]. Available from: https://spring.io/

[20] Tutorial: Intro to React. [online]. 2021, [cit. 26. 5. 2021]. Available from: https://reactjs.org/tutorial/tutorial.html

[21] Redux Essentials, Part 1: Redux Overview and Concepts. [online]. 2021, [cit. 26. 5. 2021]. Available from: https://redux.js.org/tutorials/essentials/part-1-overview-concepts

[22] Idempotent. [online]. 2021, [cit. 23. 4. 2021]. Available from: https://developer.mozilla.org/en-US/docs/Glossary/Idempotent

[23] What is REST?. [online]. 2021, [cit. 25. 4. 2021]. Available from: https://www.codecademy.com/articles/what-is-rest

[24] Doyle, K.; Ferguson, K.; et al. REST (REpresentational State Transfer). [online]. 2021, [cit. 25. 4. 2021]. Available from: https://searchapparchitecture.techtarget.com/definition/REST-REpresentational-State-Transfer

[25] ETag. [online]. 2021, [cit. 25. 5. 2021]. Available from: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/ETag

[26] Tkalec, T. JSON Web Token Tutorial: An Example in Laravel and AngularJS. [online]. 2021, [cit. 25. 4. 2021]. Available from: https://www.toptal.com/web/cookie-free-authentication-with-json-web-tokens-an-example\-in-laravel-and-angularjs

[27] What is Spring Security?. [online]. 2021, [cit. 26. 4. 2021]. Available from: https://docs.spring.io/spring-security/site/docs/5.0.19.RELEASE/reference/htmlsingle/#what-is-acegi-security

[28] Spring Security. [online]. 2021, [cit. 26. 4. 2021]. Available from: https://spring.io/projects/spring-security

[29] Spring Data JPA. [online]. 2021, [cit. 26. 4. 2021]. Available from: https://spring.io/projects/spring-data-jpa

[30] Zoltan Raffai. What Is Spring Data JPA?. [online]. 2021, [cit. 26. 4. 2021]. Available from: https://dzone.com/articles/what-is-spring-data-jpa

[31] Isha Upadhyay. Top 10 Major Characteristics of Cloud Computing. [online]. 2021, [cit. 26. 4. 2021]. Available from: https://www.jigsawacademy.com/blogs/cloud-computing/characteristics-of-cloud-computing/

[32] What is Docker? How Does it Work?. [online]. 2021, [cit. 26. 4. 2021]. Available from: https://devopscube.com/what-is-docker/

[33] What is Docker?. [online]. 2021, [cit. 26. 4. 2021]. Available from: https://opensource.com/resources/what-docker

[34] Kubernetes. [online]. 2021, [cit. 14. 4. 2021]. Available from: https://kubernetes.io/

[35] What is Kubernetes?. [online]. 2021, [cit. 14. 4. 2021]. Available from: https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/

[36] About. [online]. 2021, [cit. 26. 4. 2021]. Available from: https://www.postgresql.org/about/

[37] Net Present Value (NPV). [online]. 2021, [cit. 25. 4. 2021]. Available from: https://corporatefinanceinstitute.com/resources/knowledge/valuation/net-present-value-npv/

[38] What is Functional Testing? Types Examples (Complete Tutorial). [online]. 2021, [cit. 27. 4. 2021]. Available from: https://www.guru99.com/functional-testing.html

[39] What is User Acceptance Testing (UAT)?. [online]. 2021, [cit. 27. 4. 2021]. Available from: `https://www.guru99.com/user-acceptance-testing.html`

[40] A beginner's guide to user usability testing. [online]. 2021, [cit. 28. 4. 2021]. Available from: `https://www.hotjar.com/usability-testing/`

[41] JUnit Tutorial for Beginners: Learn in 3 Days. [online]. 2021, [cit. 28. 4. 2021]. Available from: `https://www.guru99.com/junit-tutorial.html`

[42] Integration Testing. [online]. 2021, [cit. 28. 4. 2021]. Available from: `https://softwaretestingfundamentals.com/integration-testing/`

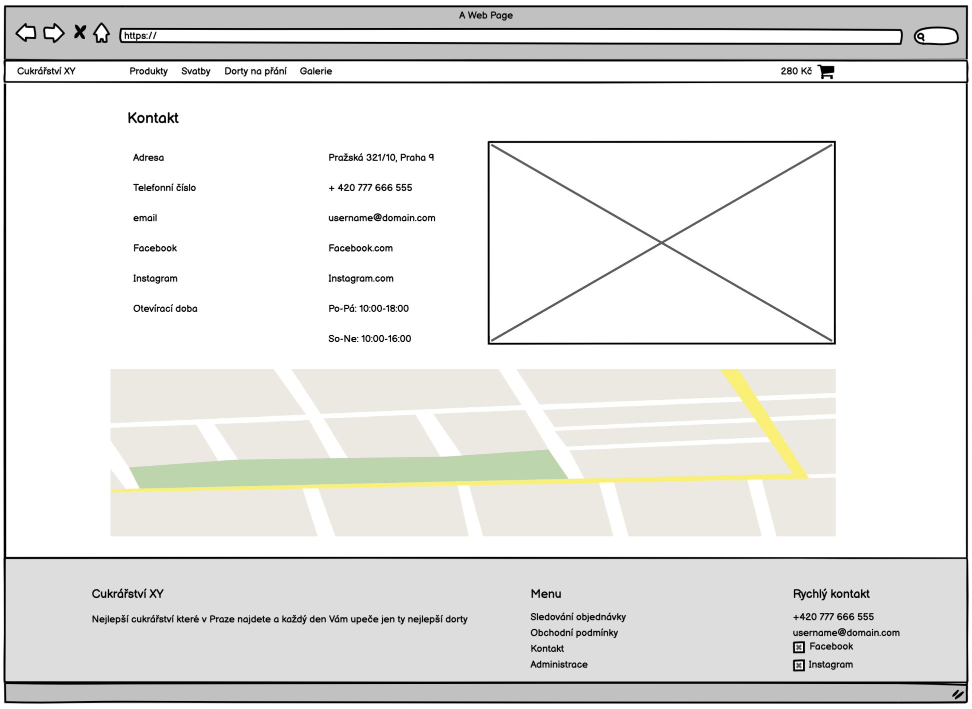
[43] Use volumes. [online]. 2021, [cit. 29. 4. 2021]. Available from: `https://docs.docker.com/storage/volumes/`

Appendices

Appendix A
Acronyms

Hi-fi	High-fidelity
VPS	Virtual Private Server
MS	Maintenance Services
UML	Unified Modeling Language
ADL	Architecture Description Language
JVM	Java Virtual Machine
POJO	Plain old Java object
JSON	JavaScript Object Notation
HTTP	Hypertext Transfer Protocol
REST	Representational State Transfer
JWT	JSON Web Token
LDAP	Lightweight Directory Access Protocol
MD	Man-Day
NPV	Net Present Value

Appendix B
Wireframes



Adresa	Pražská 321/10, Praha 9
Telefonní číslo	+ 420 777 666 555
email	username@domain.com
Facebook	Facebook.com
Instagram	Instagram.com
Otevírací doba	Po-Pá: 10:00-18:00
	So-Ne: 10:00-16:00

Cukrářství XY

Nejlepší cukrářství které v Praze najdete a každý den Vám upeče jen ty nejlepší dorty

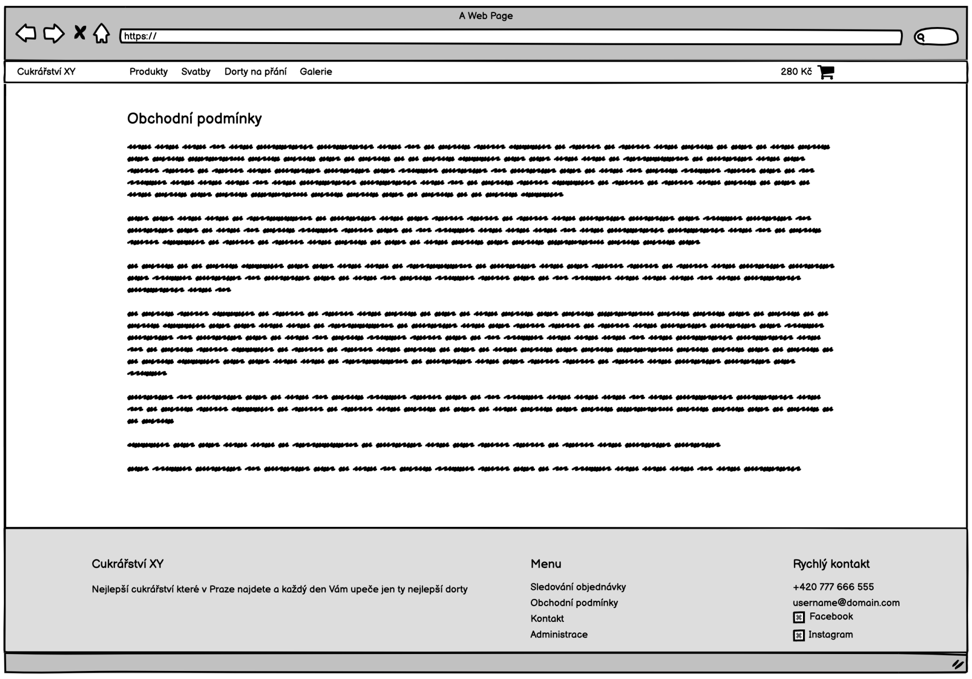
Menu

Sledování objednávky
Obchodní podmínky
Kontakt
Administrace

Rychlý kontakt

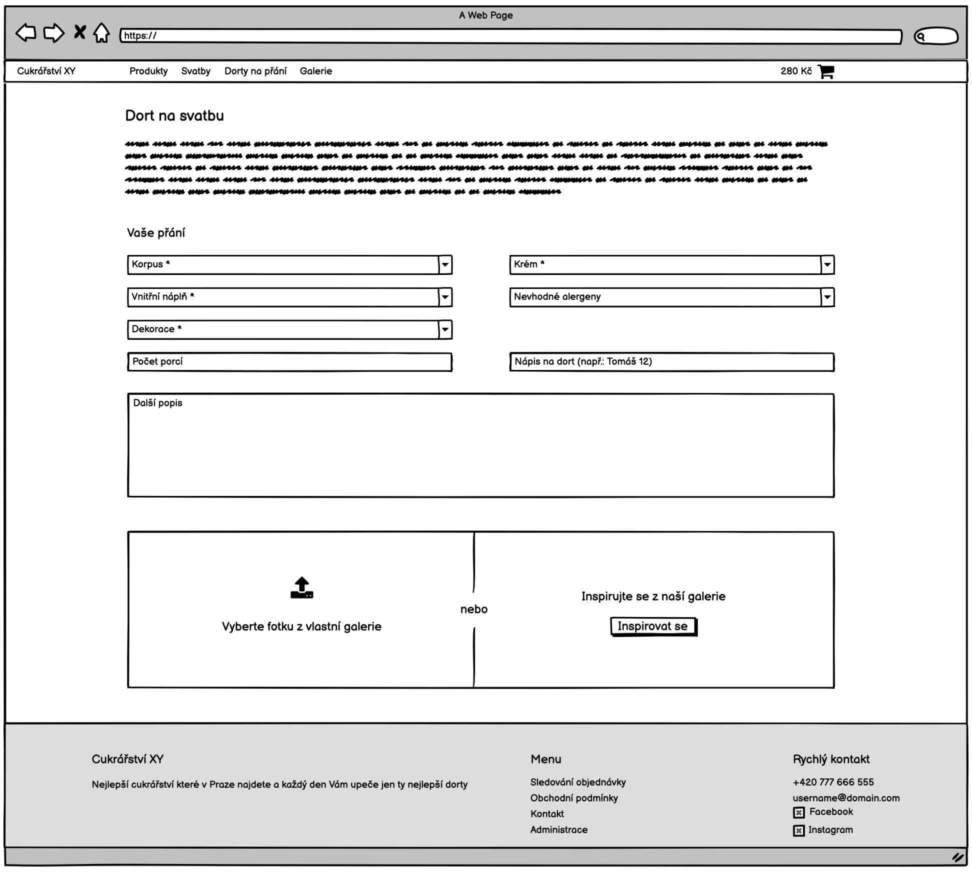
+420 777 666 555
username@domain.com
Facebook
Instagram

Figure B.1: Contacts



Nejlepší cukrářství které v Praze najdete a každý den Vám upeče jen ty nejlepší dorty

Figure B.2: Terms and contitions



Nejlepší cukrářství které v Praze najdete a každý den Vám upeče jen ty nejlepší dorty

Figure B.3: Creating a wedding cake

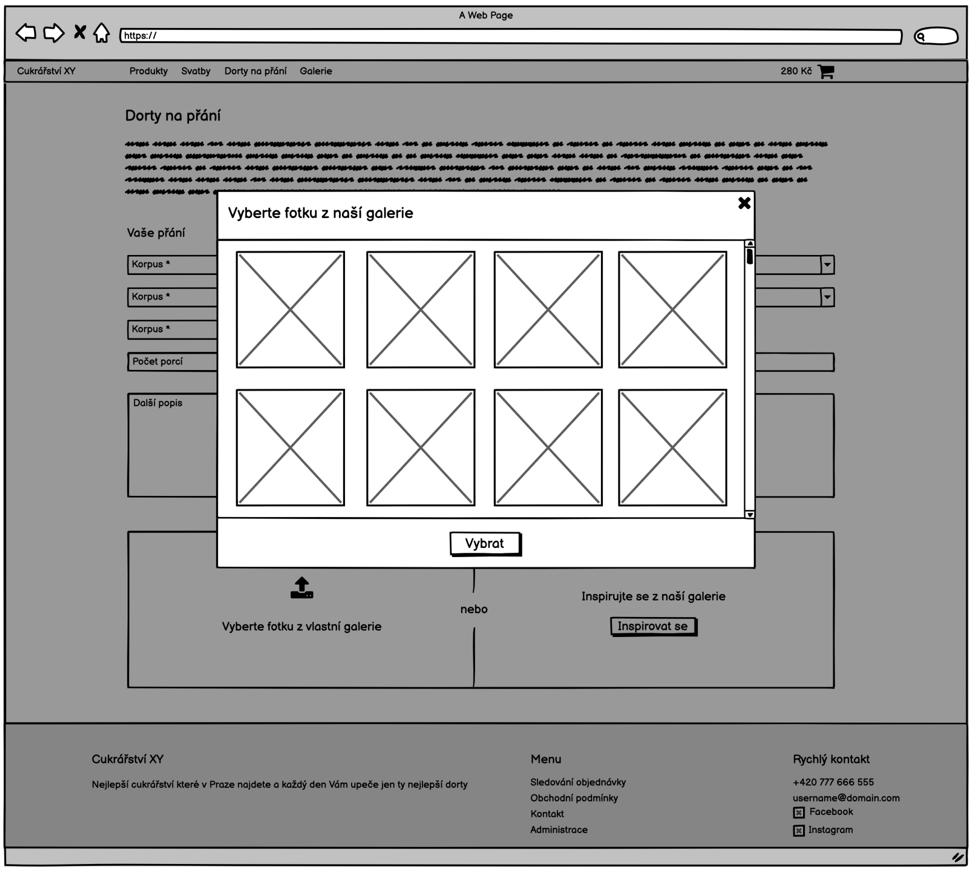
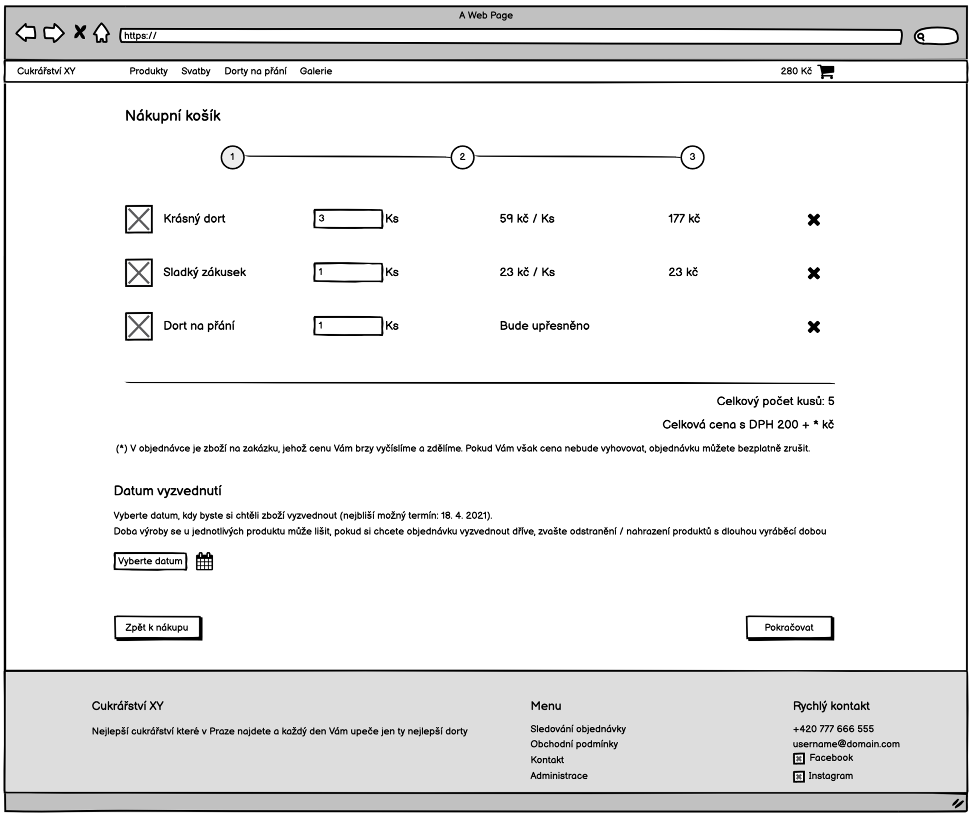


Figure B.4: Custom product – inspiration in the gallery



Nákupní košík

Krásný dort	3 Ks	59 kč / Ks	177 kč	✖
Sladký zákusek	1 Ks	23 kč / Ks	23 kč	✖
Dort na přání	1 Ks	Bude upřesněno		✖

Celkový počet kusů: 5

Celková cena s DPH 200 + * kč

(*) V objednávce je zboží na zakázku, jehož cenu Vám brzy vyčíslíme a zdělíme. Pokud Vám však cena nebude vyhovovat, objednávku můžete bezplatně zrušit.

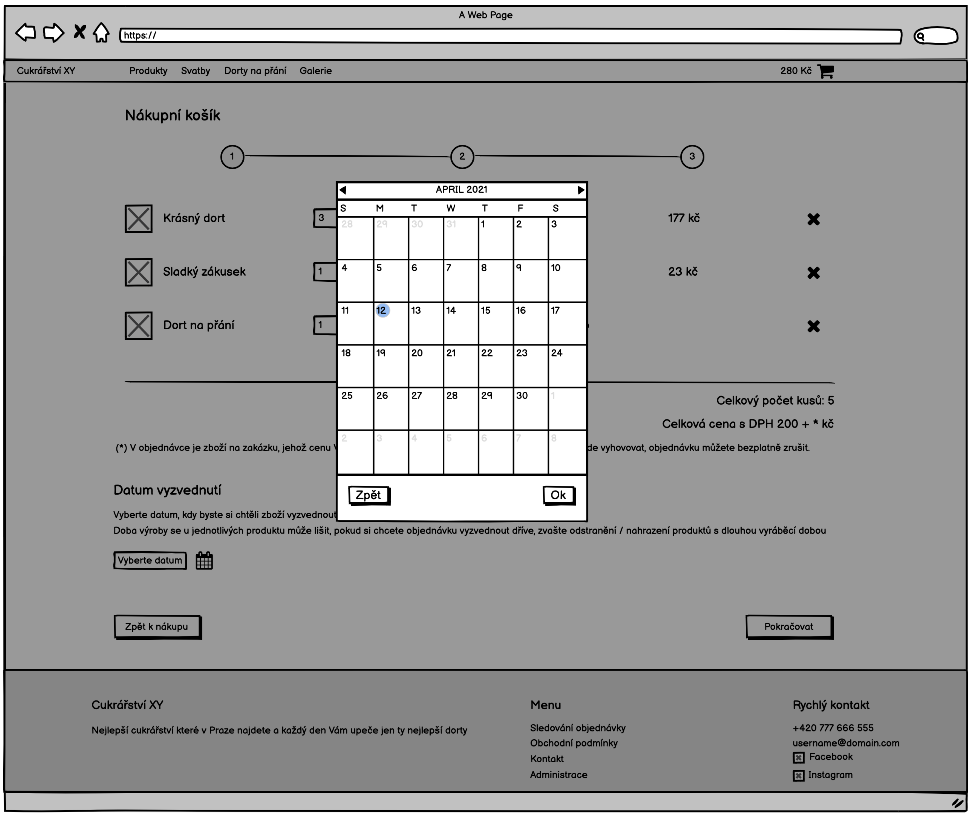
Datum vyzvednutí

Vyberte datum, kdy byste si chtěli zboží vyzvednout (nejbliší možný termín: 18. 4. 2021).
Doba výroby se u jednotlivých produktu může lišit, pokud si chcete objednávku vyzvednout dříve, zvašte odstranění / nahrazení produktů s dlouhou vyráběcí dobou

Cukrářství XY

Nejlepší cukrářství které v Praze najdete a každý den Vám upeče jen ty nejlepší dorty

Figure B.5: Order creating – shopping cart


A Web Page

https://

Cukrářství XY · Produkty · Svatby · Dorty na přání · Galerie · 280 Kč

Nákupní košík

1 — 2 — 3

Krásný dort	3	177 kč	✖
Sladký zákusek	1	23 kč	✖
Dort na přání	1		✖

Celkový počet kusů: 5

Celková cena s DPH 200 + * kč

(*) V objednávce je zboží na zakázku, jehož cenu V … de vyhovovat, objednávku můžete bezplatně zrušit.

Datum vyzvednutí

Vyberte datum, kdy byste si chtěli zboží vyzvednout

Doba výroby se u jednotlivých produktu může lišit, pokud si chcete objednávku vyzvednout dříve, zvašte odstranění / nahrazení produktů s dlouhou vyráběcí dobou

Vyberte datum

APRIL 2021

S	M	T	W	T	F	S
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1
2	3	4	5	6	7	8

Zpět · Ok

Zpět k nákupu · Pokračovat

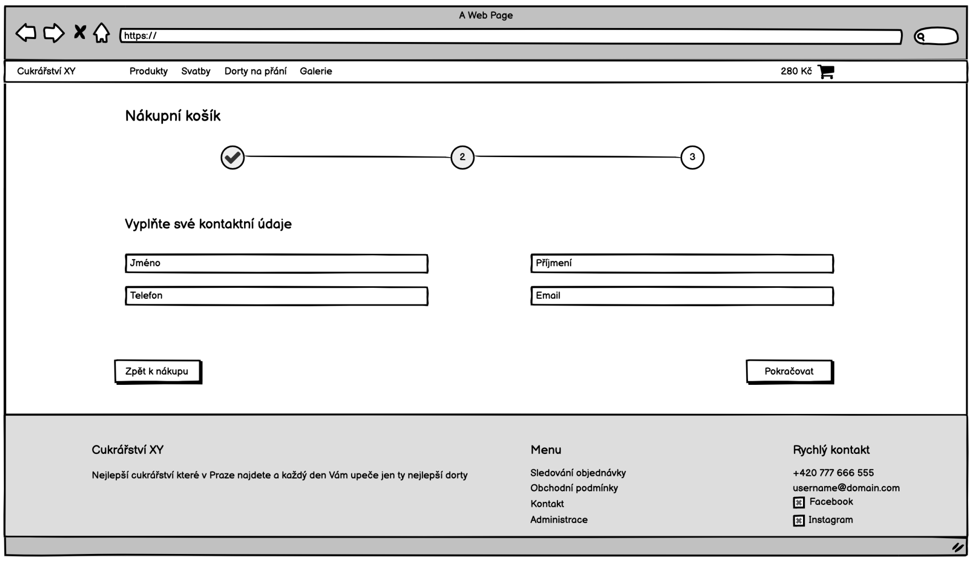
Cukrářství XY

Nejlepší cukrářství které v Praze najdete a každý den Vám upeče jen ty nejlepší dorty

Menu: Sledování objednávky · Obchodní podmínky · Kontakt · Administrace

Rychlý kontakt: +420 777 666 555 · username@domain.com · Facebook · Instagram

Figure B.6: Order – selecting date


A Web Page

https://

Cukrářství XY · Produkty · Svatby · Dorty na přání · Galerie · 280 Kč

Nákupní košík

✓ — 2 — 3

Vyplňte své kontaktní údaje

Jméno	Příjmení
Telefon	Email

Zpět k nákupu · Pokračovat

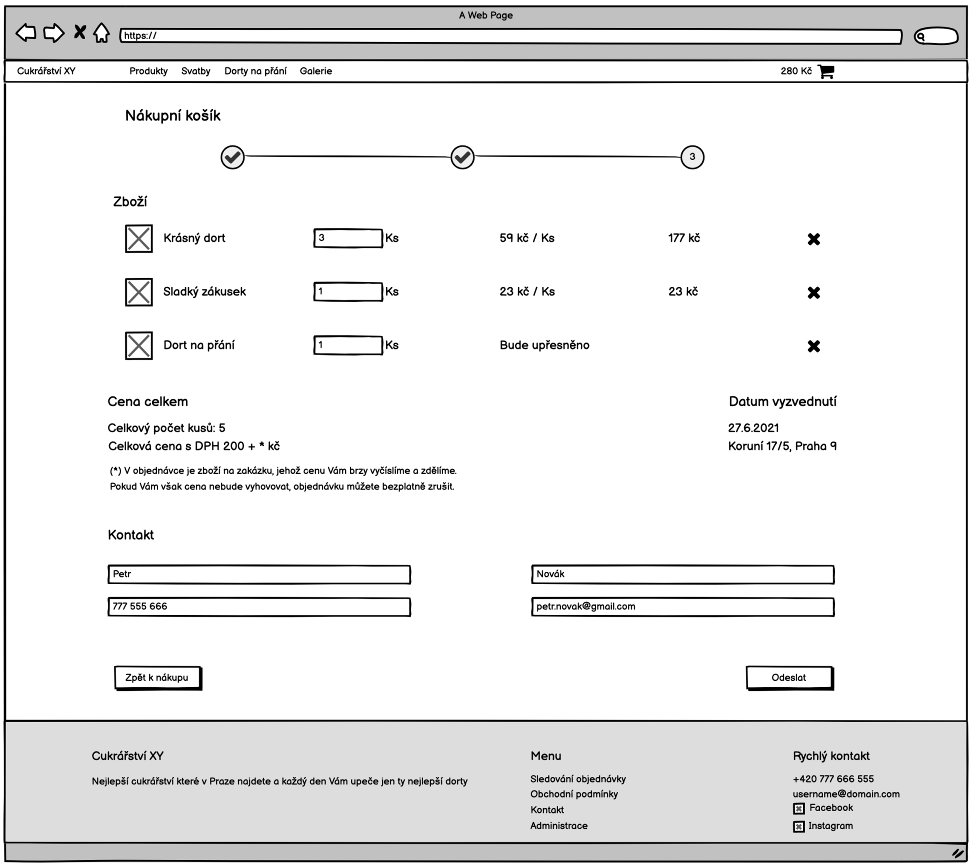
Cukrářství XY

Nejlepší cukrářství které v Praze najdete a každý den Vám upeče jen ty nejlepší dorty

Menu: Sledování objednávky · Obchodní podmínky · Kontakt · Administrace

Rychlý kontakt: +420 777 666 555 · username@domain.com · Facebook · Instagram

Figure B.7: Order creating – contact information


A Web Page

https://

Cukrářství XY Produkty Svatby Dorty na přání Galerie 280 Kč

Nákupní košík

3

Zboží

Krásný dort	3 Ks	59 kč / Ks	177 kč	✖
Sladký zákusek	1 Ks	23 kč / Ks	23 kč	✖
Dort na přání	1 Ks	Bude upřesněno		✖

Cena celkem

Celkový počet kusů: 5

Celková cena s DPH 200 + * kč

(*) V objednávce je zboží na zakázku, jehož cenu Vám brzy vyčíslíme a zdělíme.
Pokud Vám však cena nebude vyhovovat, objednávku můžete bezplatně zrušit.

Datum vyzvednutí

27.6.2021

Koruní 17/5, Praha 9

Kontakt

Petr	Novák
777 555 666	petr.novak@gmail.com

Zpět k nákupu Odeslat

Cukrářství XY

Nejlepší cukrářství které v Praze najdete a každý den Vám upeče jen ty nejlepší dorty

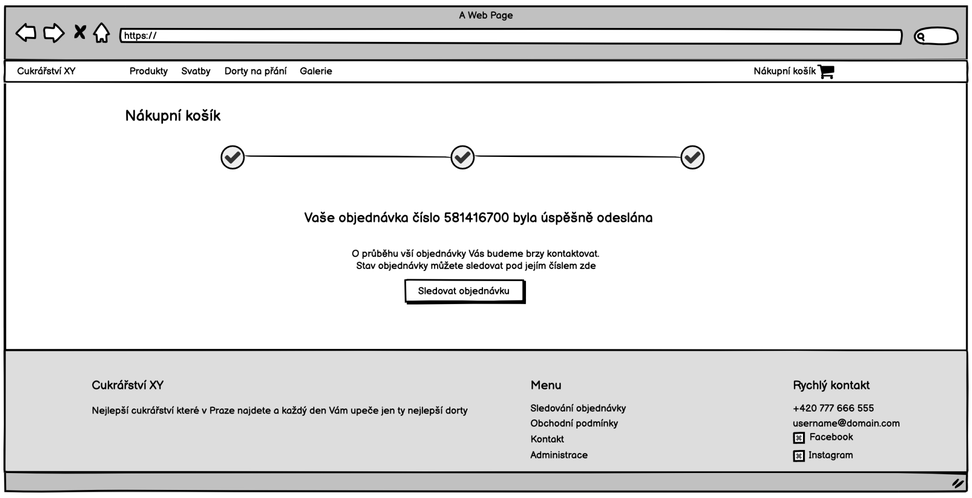
Menu

Sledování objednávky
Obchodní podmínky
Kontakt
Administrace

Rychlý kontakt

+420 777 666 555
username@domain.com
Facebook
Instagram

Figure B.8: Order creating – summary


A Web Page

https://

Cukrářství XY Produkty Svatby Dorty na přání Galerie Nákupní košík

Nákupní košík

Vaše objednávka číslo 581416700 byla úspěšně odeslána

O průběhu vší objednávky Vás budeme brzy kontaktovat.
Stav objednávky můžete sledovat pod jejím číslem zde

Sledovat objednávku

Cukrářství XY

Nejlepší cukrářství které v Praze najdete a každý den Vám upeče jen ty nejlepší dorty

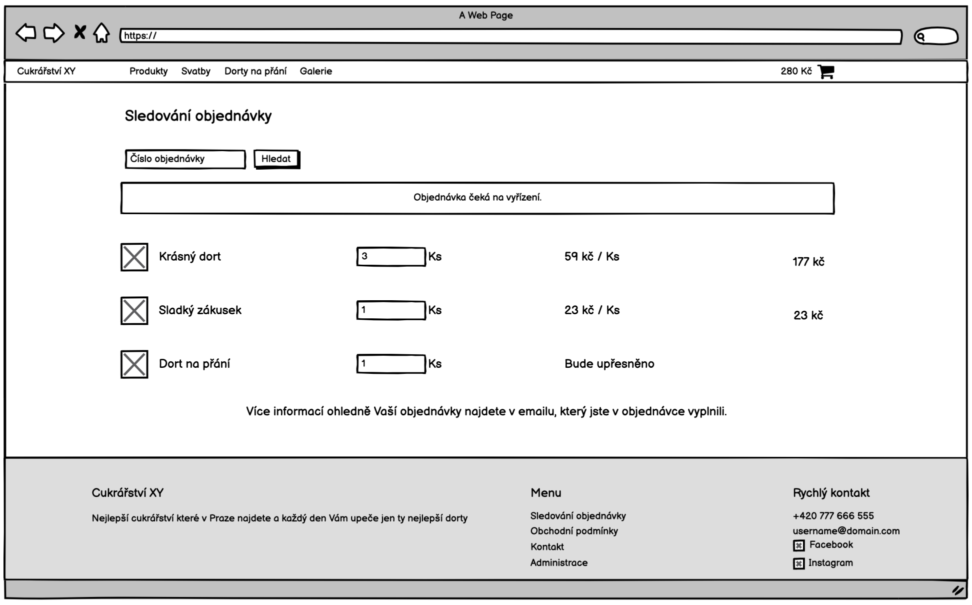
Menu

Sledování objednávky
Obchodní podmínky
Kontakt
Administrace

Rychlý kontakt

+420 777 666 555
username@domain.com
Facebook
Instagram

Figure B.9: New order created



Objednávka čeká na vyřízení.

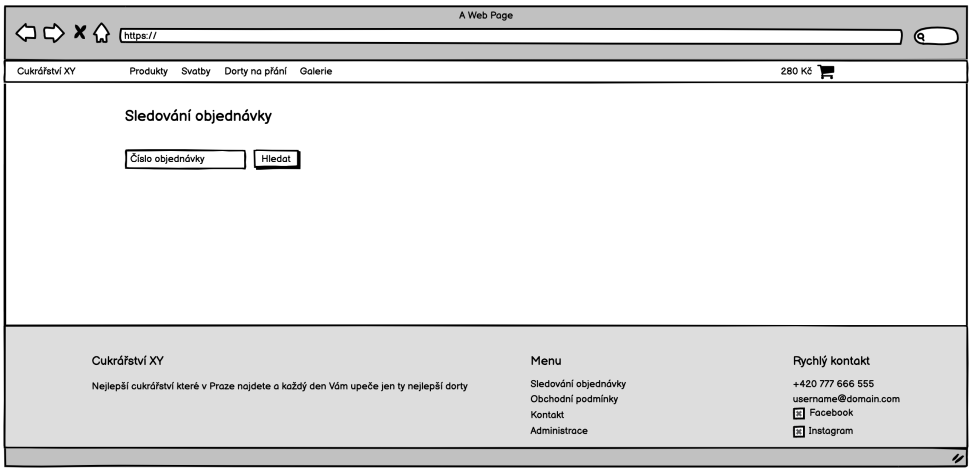
Krásný dort	3	Ks	59 kč / Ks	177 kč
Sladký zákusek	1	Ks	23 kč / Ks	23 kč
Dort na přání	1	Ks	Bude upřesněno	

Více informací ohledně Vaší objednávky najdete v emailu, který jste v objednávce vyplnili.

Cukrářství XY

Nejlepší cukrářství které v Praze najdete a každý den Vám upeče jen ty nejlepší dorty

Figure B.10: Check order status



Cukrářství XY

Nejlepší cukrářství které v Praze najdete a každý den Vám upeče jen ty nejlepší dorty

Figure B.11: Order status checking page

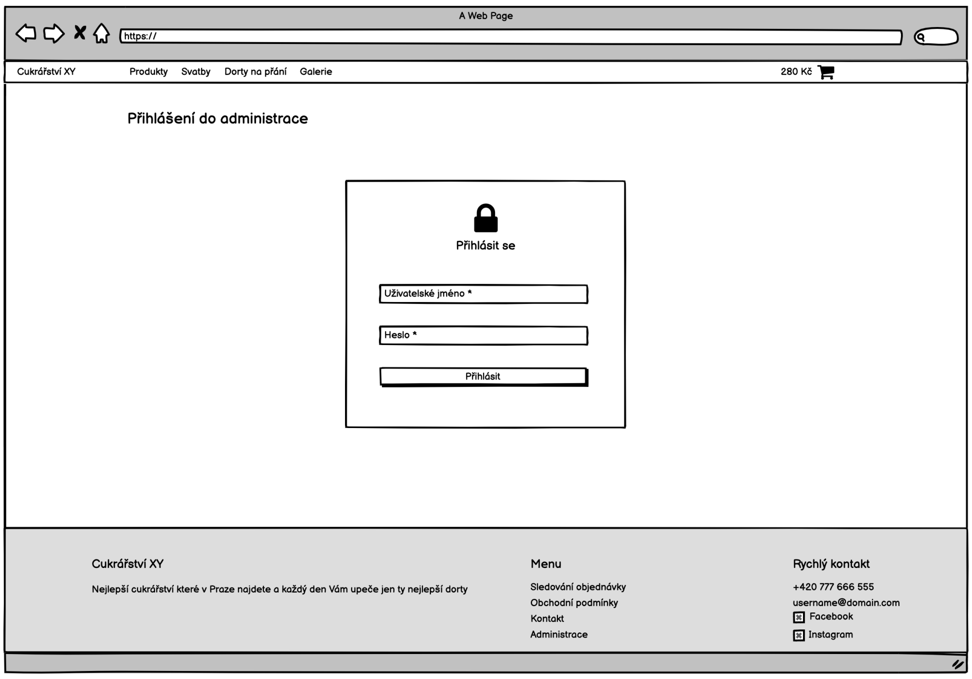


Figure B.12: Login to administration

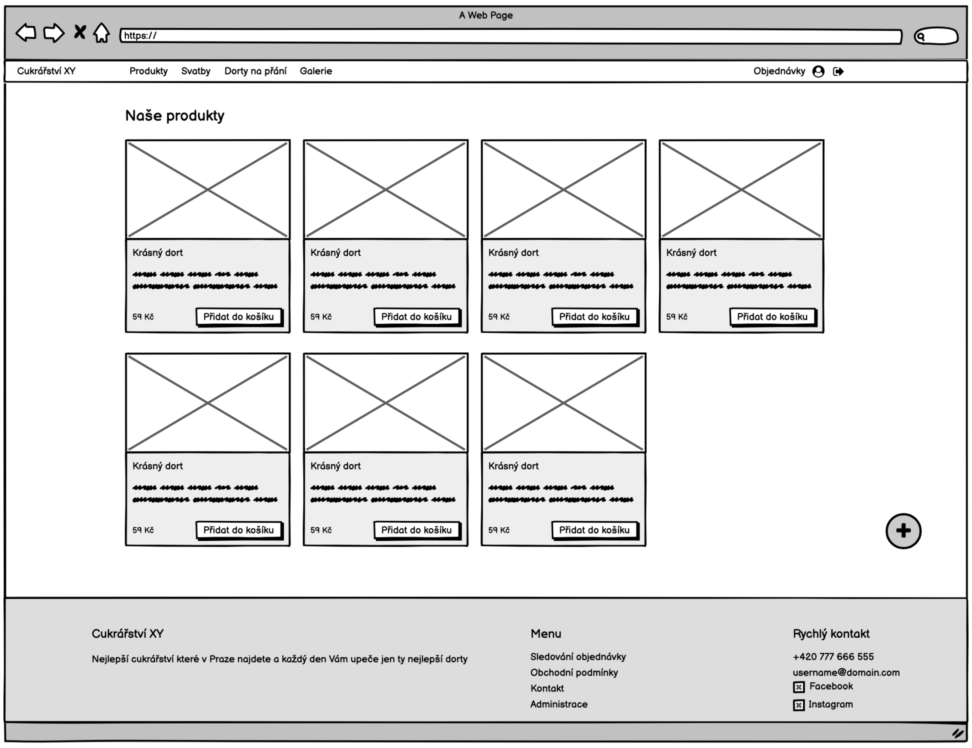


Figure B.13: Administrator – product list

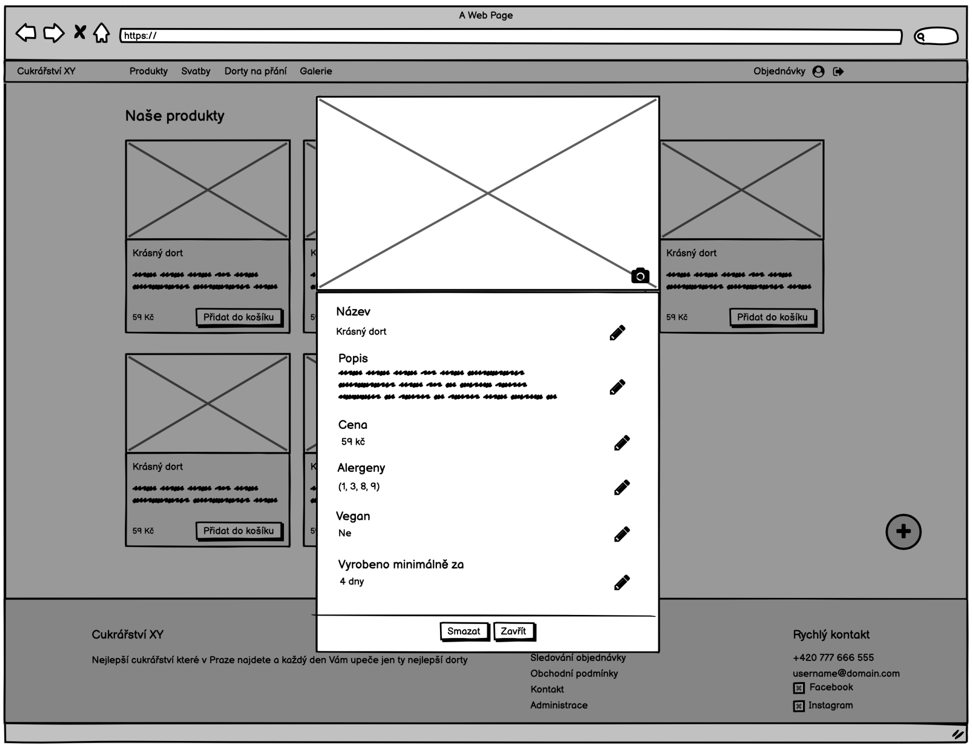


Figure B.14: Administrator – product detail

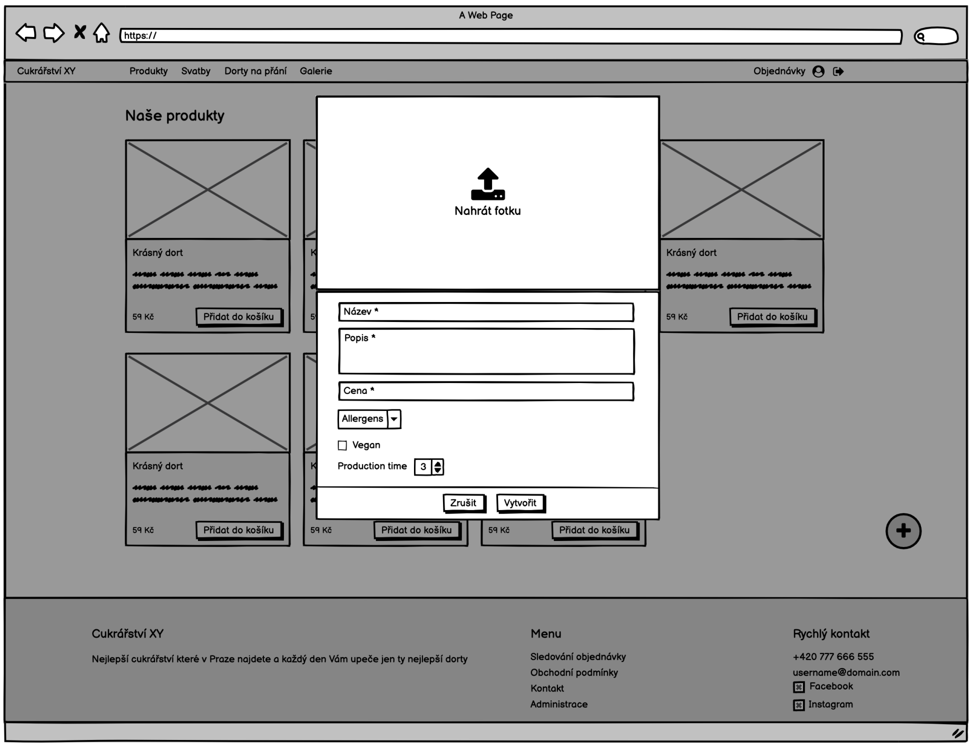
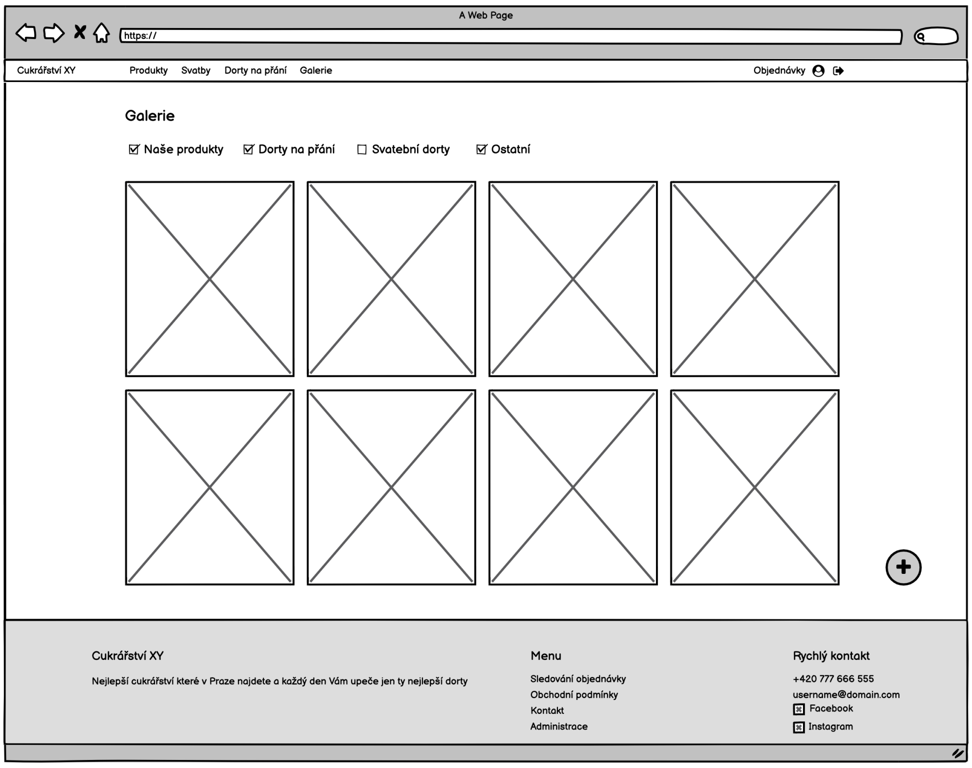
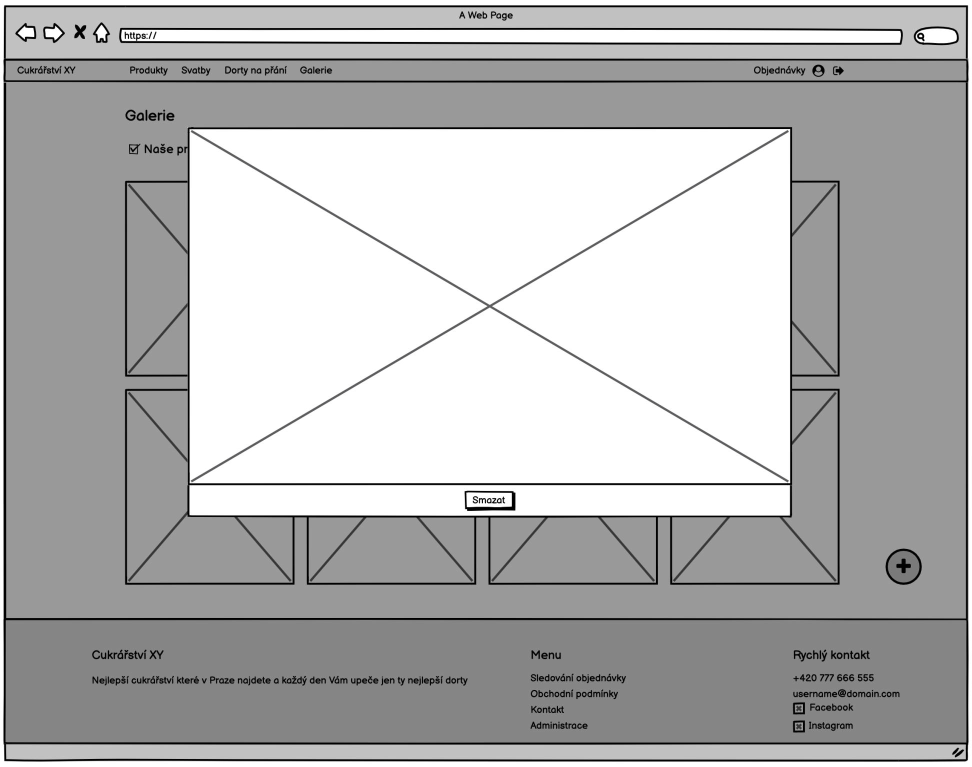


Figure B.15: Administrator – create a new product



Nejlepší cukrářství které v Praze najdete a každý den Vám upeče jen ty nejlepší dorty

Figure B.16: Administrator – gallery



Nejlepší cukrářství které v Praze najdete a každý den Vám upeče jen ty nejlepší dorty

Figure B.17: Administrator – gallery's picture detail

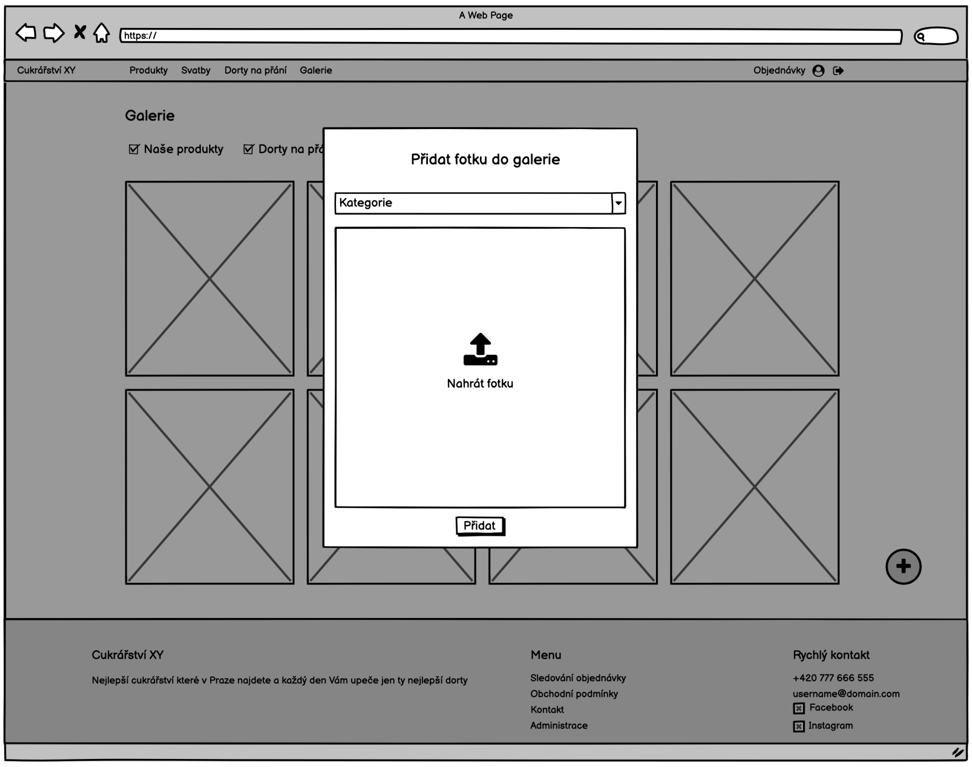
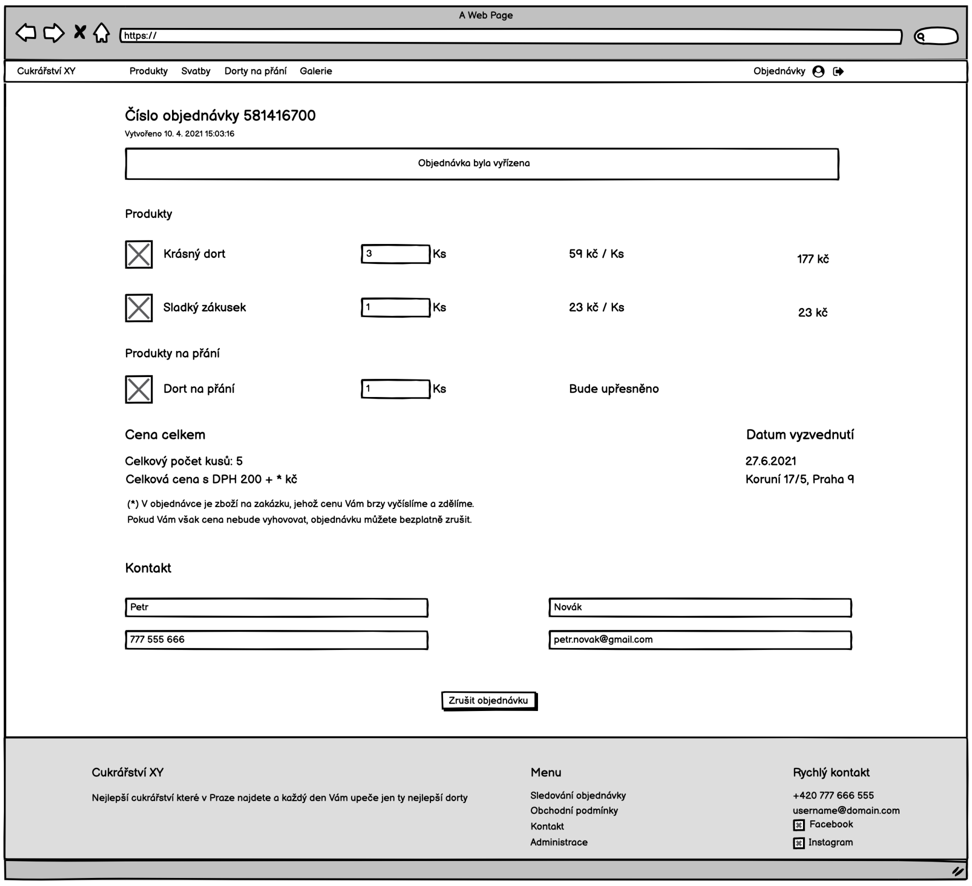


Figure B.18: Administrator – add picture to gallery



Číslo objednávky 581416700

Vytvořeno 10. 4. 2021 15:03:16

Objednávka byla vyřízena

Produkty

Krásný dort	3 Ks	59 kč / Ks	177 kč
Sladký zákusek	1 Ks	23 kč / Ks	23 kč

Produkty na přání

Dort na přání	1 Ks	Bude upřesněno

Cena celkem

Celkový počet kusů: 5
Celková cena s DPH 200 + * kč

(*) V objednávce je zboží na zakázku, jehož cenu Vám brzy vyčíslíme a zdělíme.
Pokud Vám však cena nebude vyhovovat, objednávku můžete bezplatně zrušit.

Datum vyzvednutí

27.6.2021
Koruní 17/5, Praha 9

Kontakt

Petr	Novák
777 555 666	petr.novak@gmail.com

Zrušit objednávku

Cukrářství XY

Nejlepší cukrářství které v Praze najdete a každý den Vám upeče jen ty nejlepší dorty

Menu

Sledování objednávky
Obchodní podmínky
Kontakt
Administrace

Rychlý kontakt

+420 777 666 555
username@domain.com
Facebook
Instagram

Figure B.19: Administrator – detail of the accepted order

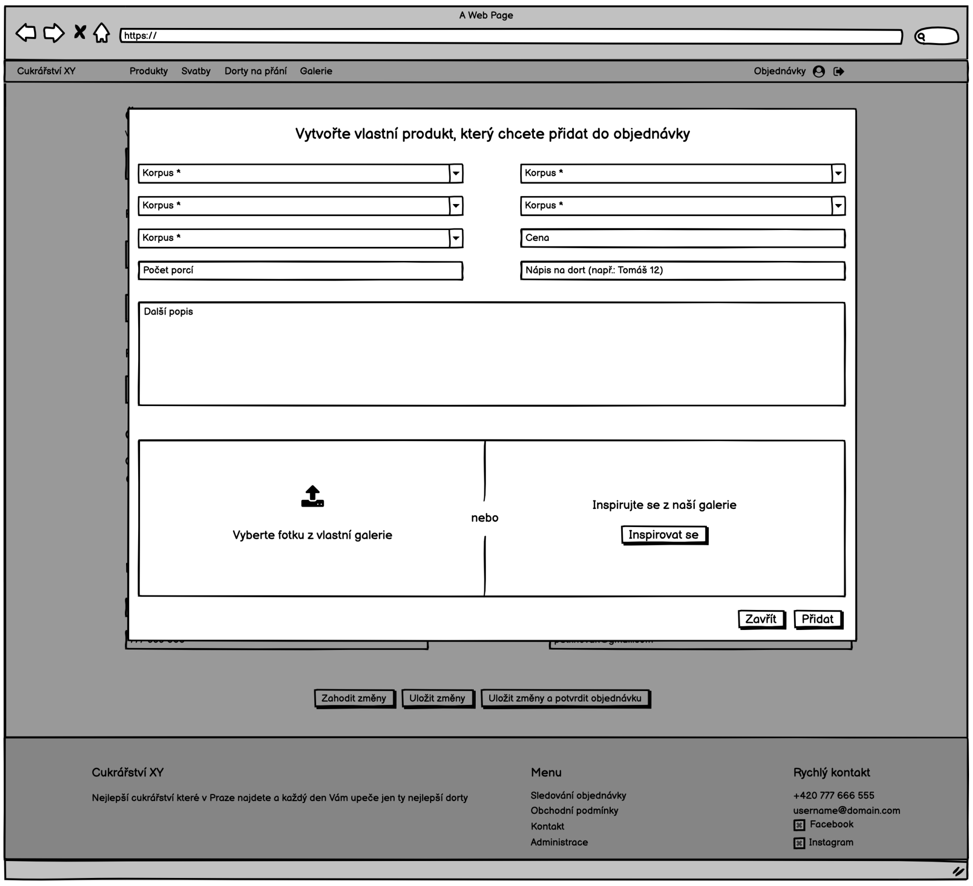


Figure B.20: Administrator – add custom product to the order

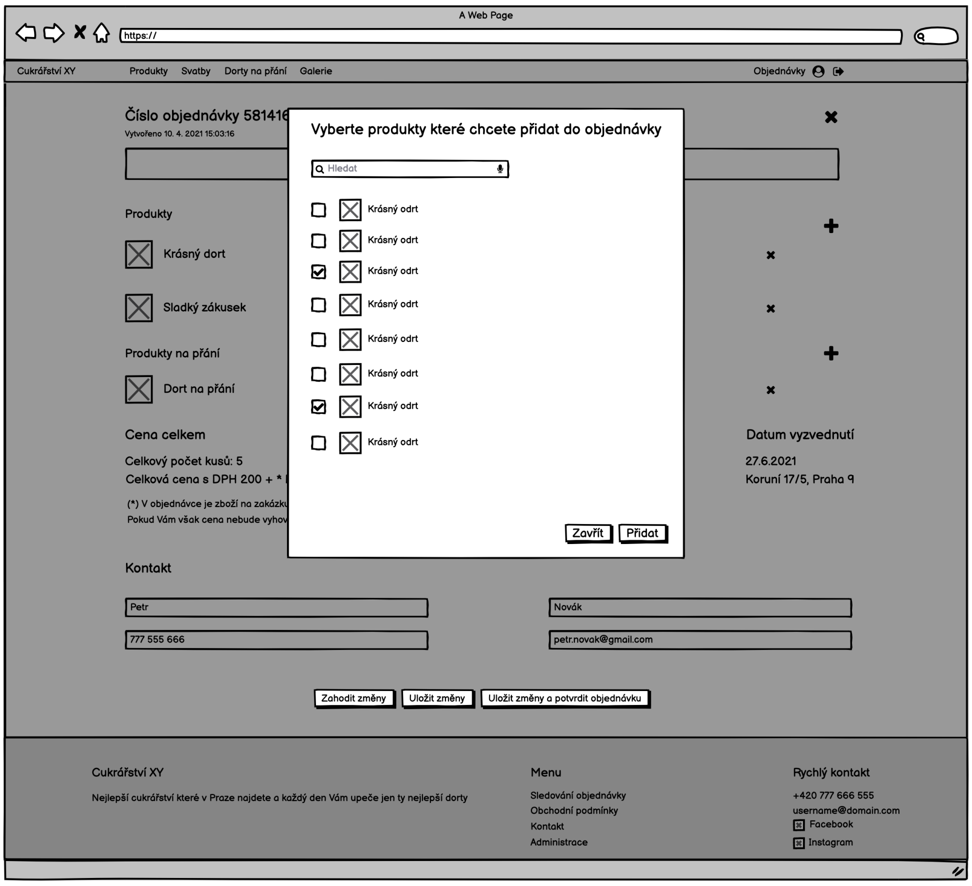
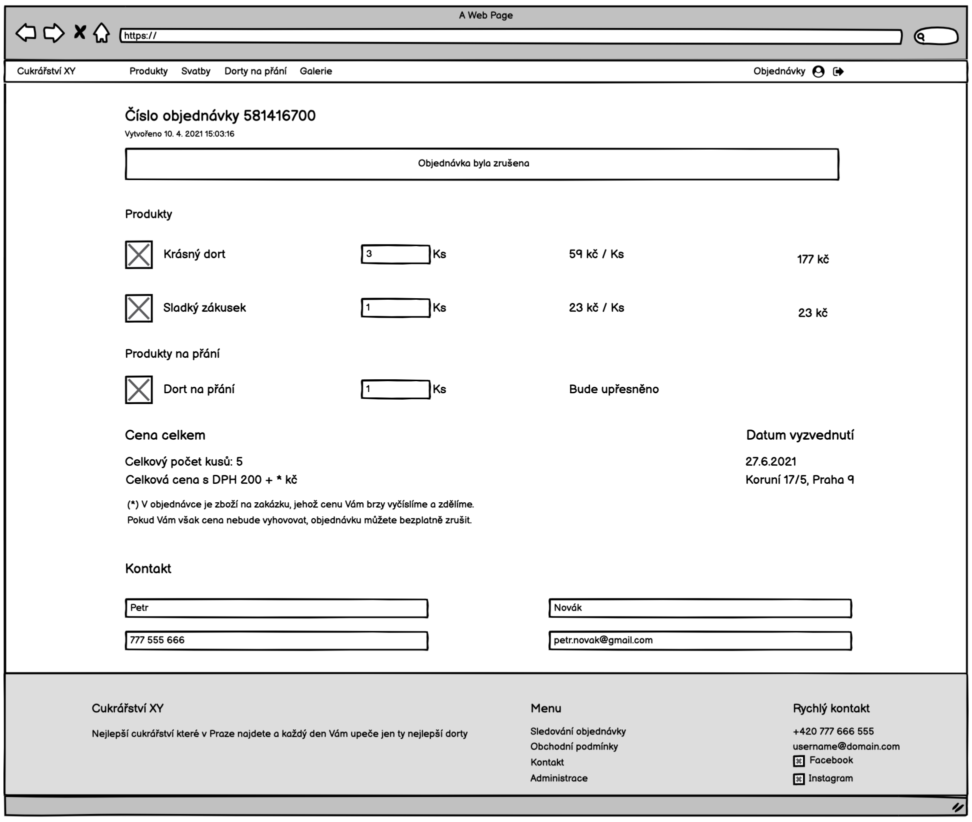


Figure B.21: Administrator – add product to the order



Číslo objednávky 581416700

Vytvořeno 10. 4. 2021 15:03:16

Objednávka byla zrušena

Produkty

Krásný dort	3 Ks	59 kč / Ks	177 kč
Sladký zákusek	1 Ks	23 kč / Ks	23 kč

Produkty na přání

Dort na přání	1 Ks	Bude upřesněno

Cena celkem

Celkový počet kusů: 5

Celková cena s DPH 200 + * kč

(*) V objednávce je zboží na zakázku, jehož cenu Vám brzy vyčíslíme a zdělíme.
Pokud Vám však cena nebude vyhovovat, objednávku můžete bezplatně zrušit.

Datum vyzvednutí

27.6.2021

Koruní 17/5, Praha 9

Kontakt

Petr	Novák
777 555 666	petr.novak@gmail.com

Cukrářství XY

Nejlepší cukrářství které v Praze najdete a každý den Vám upeče jen ty nejlepší dorty

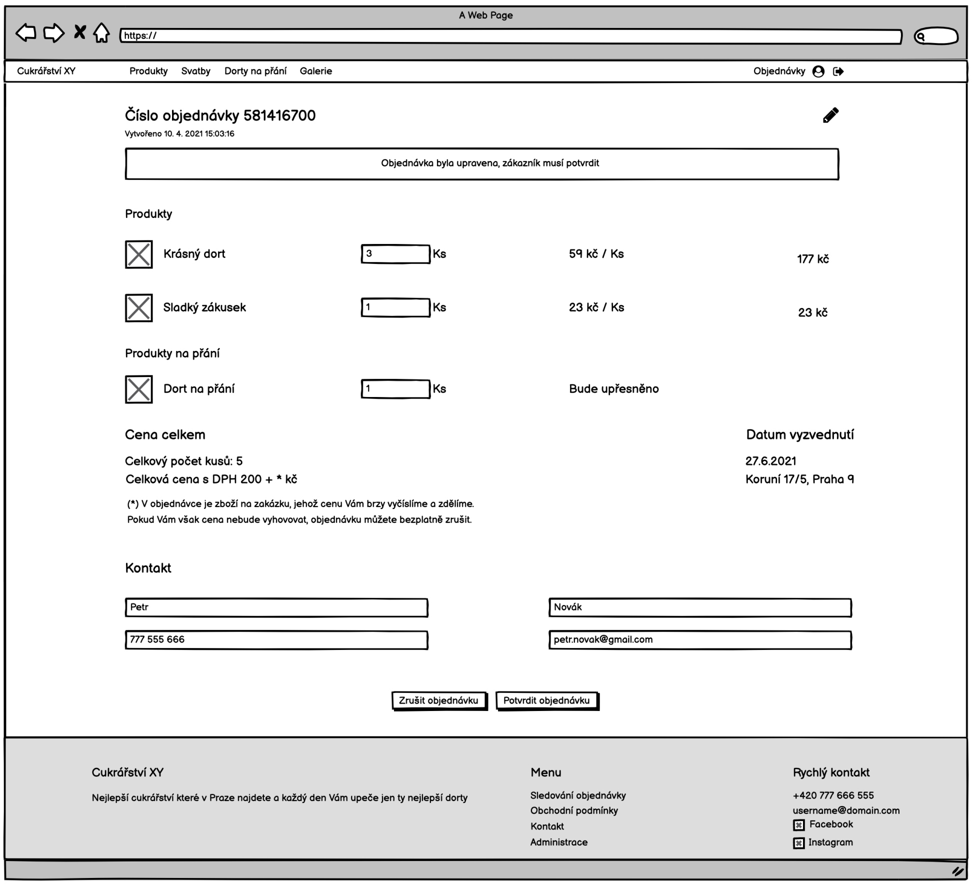
Menu

Sledování objednávky
Obchodní podmínky
Kontakt
Administrace

Rychlý kontakt

+420 777 666 555
username@domain.com
Facebook
Instagram

Figure B.22: Administrator – detail of the canceled order



Číslo objednávky 581416700

Vytvořeno 10. 4. 2021 15:03:16

Objednávka byla upravena, zákazník musí potvrdit

Produkty

Krásný dort	3 Ks	59 kč / Ks	177 kč
Sladký zákusek	1 Ks	23 kč / Ks	23 kč

Produkty na přání

Dort na přání	1 Ks	Bude upřesněno	

Cena celkem

Celkový počet kusů: 5
Celková cena s DPH 200 + * kč

(*) V objednávce je zboží na zakázku, jehož cenu Vám brzy vyčíslíme a zdělíme.
Pokud Vám však cena nebude vyhovovat, objednávku můžete bezplatně zrušit.

Datum vyzvednutí

27.6.2021
Koruní 17/5, Praha 9

Kontakt

Petr	Novák
777 555 666	petr.novak@gmail.com

Zrušit objednávku | Potvrdit objednávku

Cukrářství XY

Nejlepší cukrářství které v Praze najdete a každý den Vám upeče jen ty nejlepší dorty

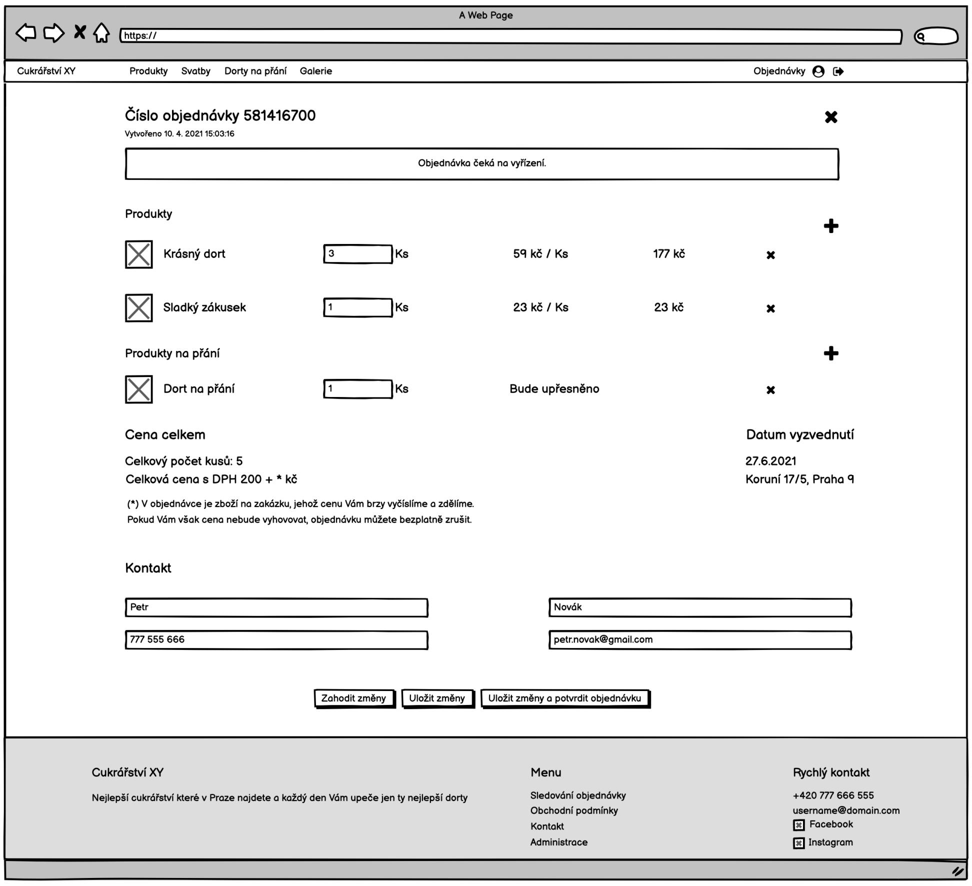
Menu

Sledování objednávky
Obchodní podmínky
Kontakt
Administrace

Rychlý kontakt

+420 777 666 555
username@domain.com
Facebook
Instagram

Figure B.23: Administrator – detail of the updated order



Číslo objednávky 581416700

Vytvořeno 10. 4. 2021 15:03:16

Objednávka čeká na vyřízení.

Produkty

Krásný dort	3 Ks	59 kč / Ks	177 kč
Sladký zákusek	1 Ks	23 kč / Ks	23 kč

Produkty na přání

Dort na přání	1 Ks	Bude upřesněno

Cena celkem

Celkový počet kusů: 5

Celková cena s DPH 200 + * kč

(*) V objednávce je zboží na zakázku, jehož cenu Vám brzy vyčíslíme a zdělíme.
Pokud Vám však cena nebude vyhovovat, objednávku můžete bezplatně zrušit.

Datum vyzvednutí

27.6.2021

Koruní 17/5, Praha 9

Kontakt

Petr	Novák
777 555 666	petr.novak@gmail.com

Cukrářství XY

Nejlepší cukrářství které v Praze najdete a každý den Vám upeče jen ty nejlepší dorty

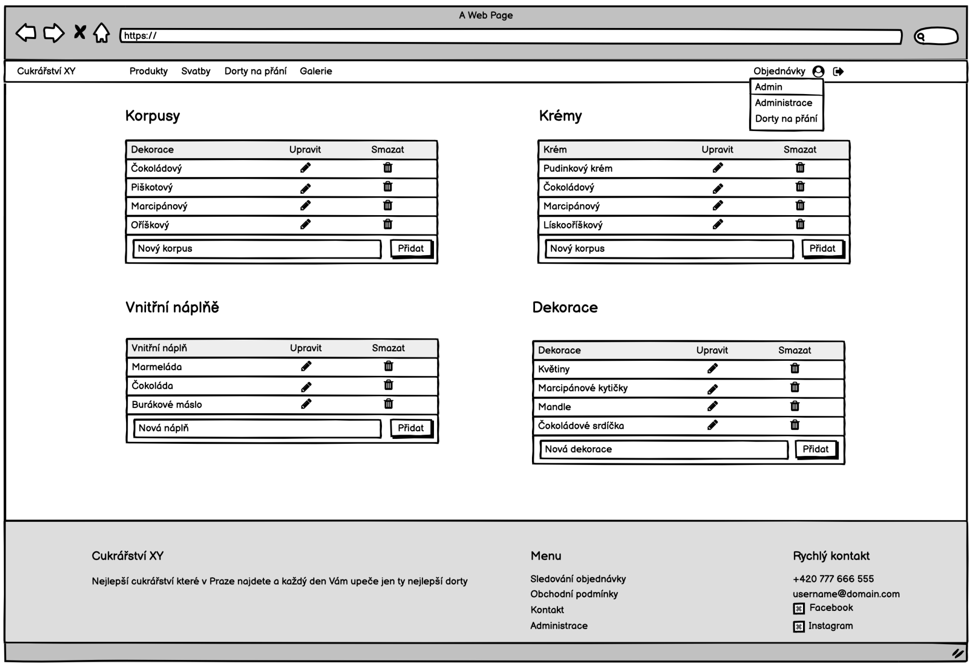
Menu

Sledování objednávky
Obchodní podmínky
Kontakt
Administrace

Rychlý kontakt

+420 777 666 555
username@domain.com
Facebook
Instagram

Figure B.24: Administrator – order editing



Korpusy

Dekorace	Upravit	Smazat
Čokoládový		
Piškotový		
Marcipánový		
Oříškový		
Nový korpus	Přidat	

Krémy

Krém	Upravit	Smazat
Pudinkový krém		
Čokoládový		
Marcipánový		
Lískooříškový		
Nový korpus	Přidat	

Vnitřní náplňě

Vnitřní náplň	Upravit	Smazat
Marmeláda		
Čokoláda		
Burákové máslo		
Nová náplň	Přidat	

Dekorace

Dekorace	Upravit	Smazat
Květiny		
Marcipánové kytičky		
Mandle		
Čokoládové srdíčka		
Nová dekorace	Přidat	

Cukrářství XY

Nejlepší cukrářství které v Praze najdete a každý den Vám upeče jen ty nejlepší dorty

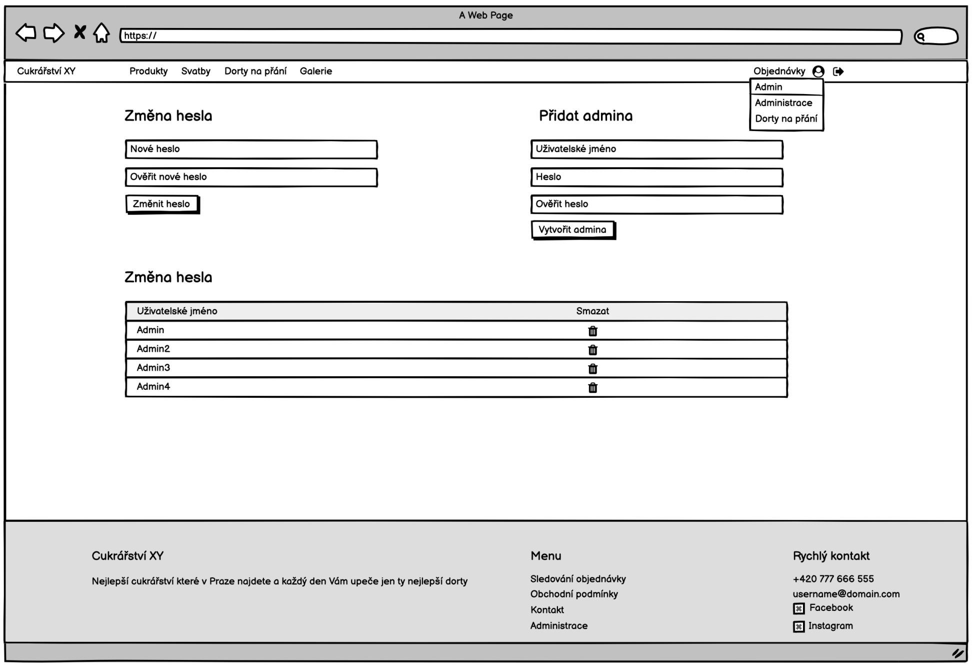
Menu

Sledování objednávky
Obchodní podmínky
Kontakt
Administrace

Rychlý kontakt

+420 777 666 555
username@domain.com
Facebook
Instagram

Figure B.25: Administrator – setting custom product parameters



A Web Page

https://

Cukrářství XY   Produkty   Svatby   Dorty na přání   Galerie   Objednávky

Admin
Administrace
Dorty na přání

Změna hesla

Nové heslo

Ověřit nové heslo

Změnit heslo

Přidat admina

Uživatelské jméno

Heslo

Ověřit heslo

Vytvořit admina

Změna hesla

Uživatelské jméno	Smazat
Admin	🗑
Admin2	🗑
Admin3	🗑
Admin4	🗑

Cukrářství XY

Nejlepší cukrářství které v Praze najdete a každý den Vám upeče jen ty nejlepší dorty

Menu

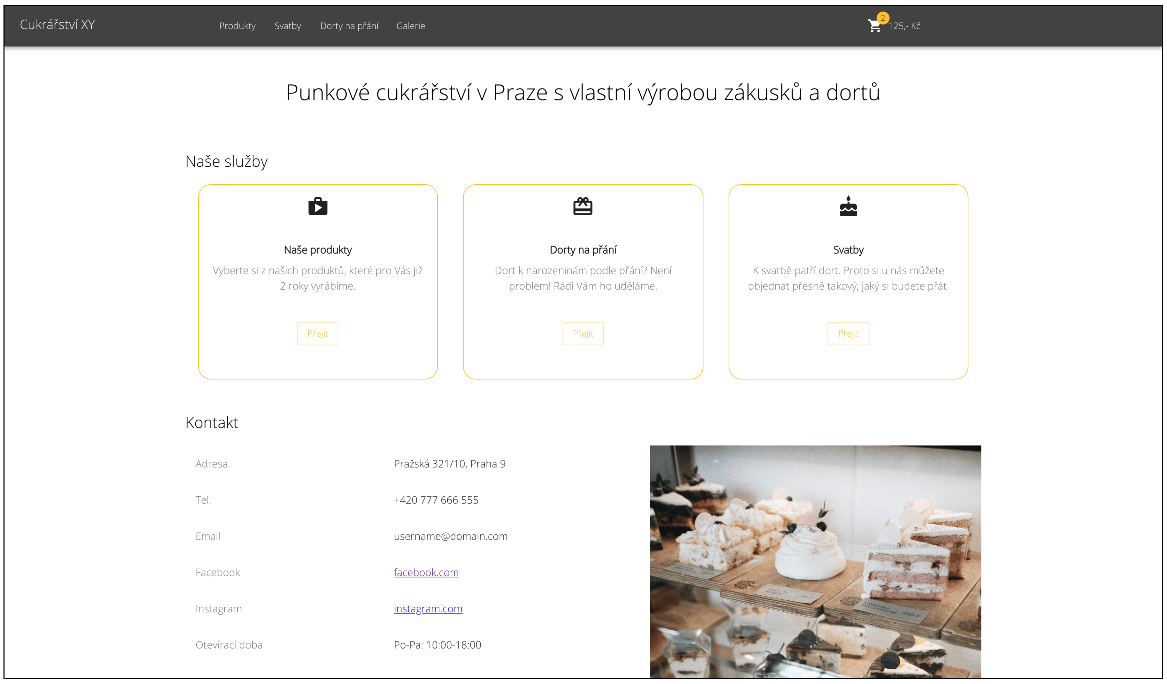
Sledování objednávky
Obchodní podmínky
Kontakt
Administrace

Rychlý kontakt

+420 777 666 555
username@domain.com
Facebook
Instagram

Figure B.26: Administrator – account administration

Appendix C
Hi-fi prototype



Punkové cukrářství v Praze s vlastní výrobou zákusků a dortů

Naše služby

Naše produkty

Vyberte si z našich produktů, které pro Vás již 2 roky vyrábíme.

Dorty na přání

Dort k narozeninám podle přání? Není problem! Rádi Vám ho uděláme.

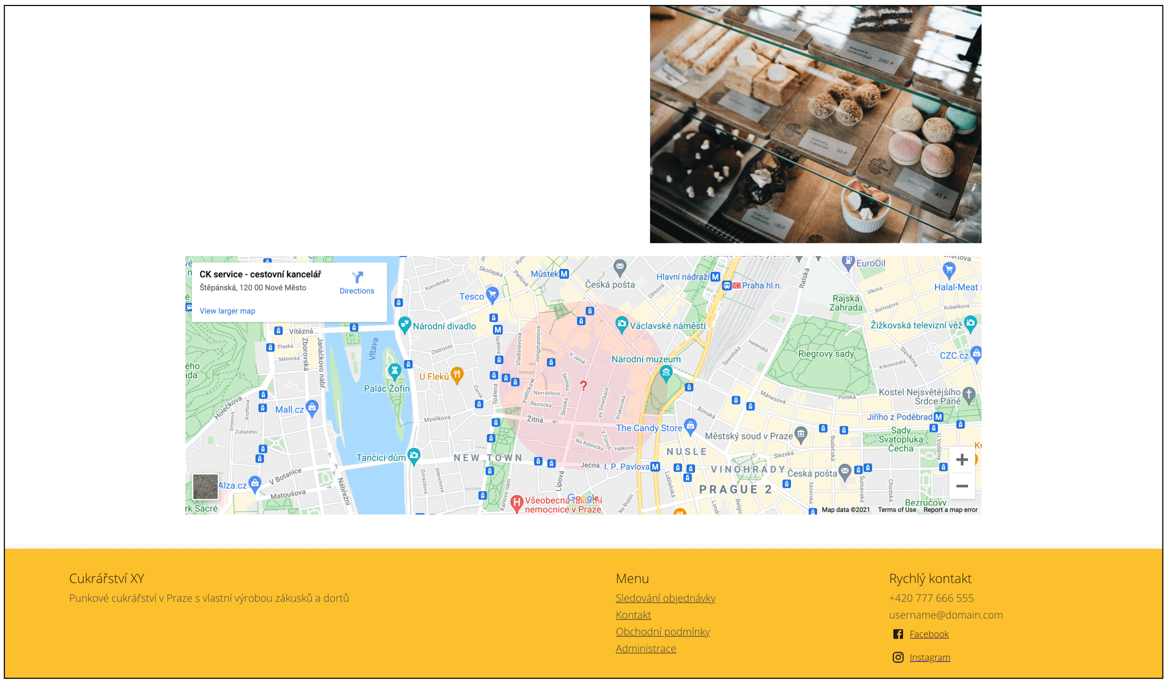
Svatby

K svatbě patří dort. Proto si u nás můžete objednat přesně takový, jaký si budete přát.

Kontakt

Adresa	Pražská 321/10, Praha 9
Tel.	+420 777 666 555
Email	username@domain.com
Facebook	facebook.com
Instagram	instagram.com
Otevírací doba	Po-Pa: 10:00-18:00

Figure C.1: Hi-fi – Homepage


CK service - cestovní kancelář
Štěpánská, 120 00 Nové Město
View larger map
Directions
Národní divadlo
Václavské náměstí
Národní muzeum
NEW TOWN
NUSLE
VINOHRADY
PRAGUE 2
Map data ©2021 Terms of Use Report a map error

Cukrářství XY

Punkové cukrářství v Praze s vlastní výrobou zákusků a dortů

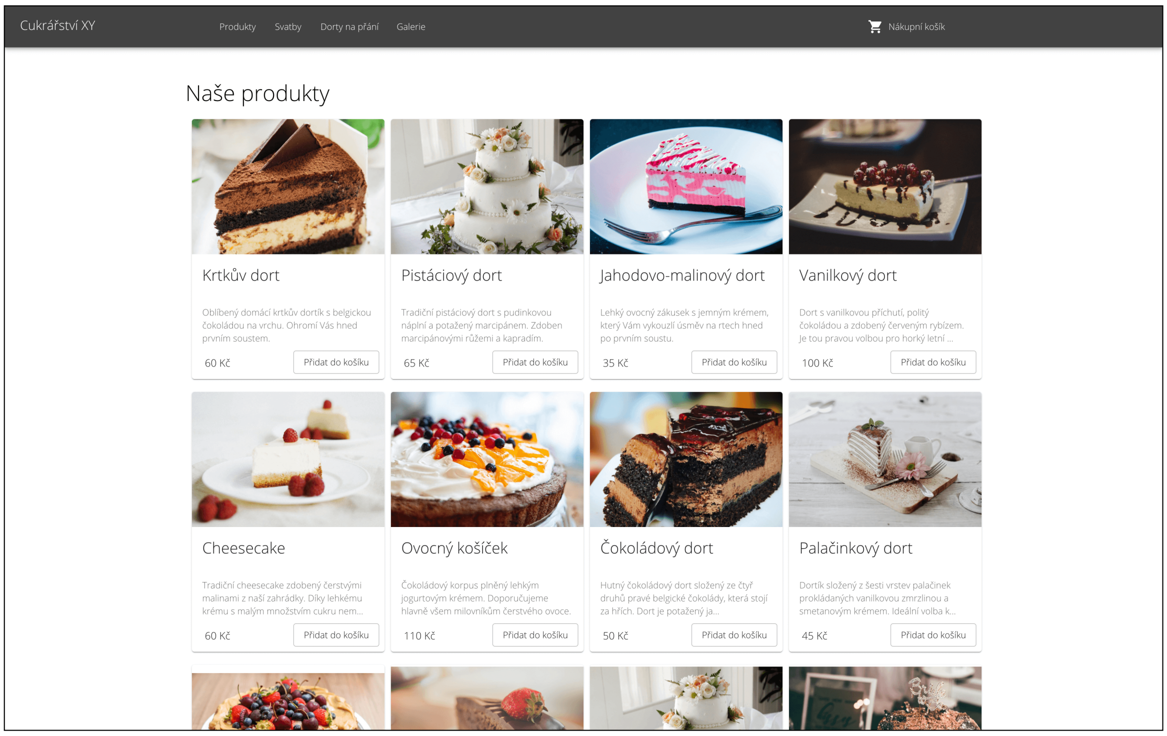
Menu

Sledování objednávky
Kontakt
Obchodní podmínky
Administrace

Rychlý kontakt

+420 777 666 555
username@domain.com
Facebook
Instagram

Figure C.2: Hi-fi – Homepage



Naše produkty

Krtkův dort
Oblíbený domácí krtkův dortík s belgickou čokoládou na vrchu. Ohromí Vás hned prvním soustem.
60 Kč Přidat do košíku

Pistáciový dort
Tradiční pistáciový dort s pudinkovou náplní a potažený marcipánem. Zdoben marcipánovými růžemi a kapradím.
65 Kč Přidat do košíku

Jahodovo-malinový dort
Lehký ovocný zákusek s jemným krémem, který Vám vykouzlí úsměv na rtech hned po prvním soustu.
35 Kč Přidat do košíku

Vanilkový dort
Dort s vanilkovou příchutí, politý čokoládou a zdobený červeným rybízem. Je tou pravou volbou pro horký letní ...
100 Kč Přidat do košíku

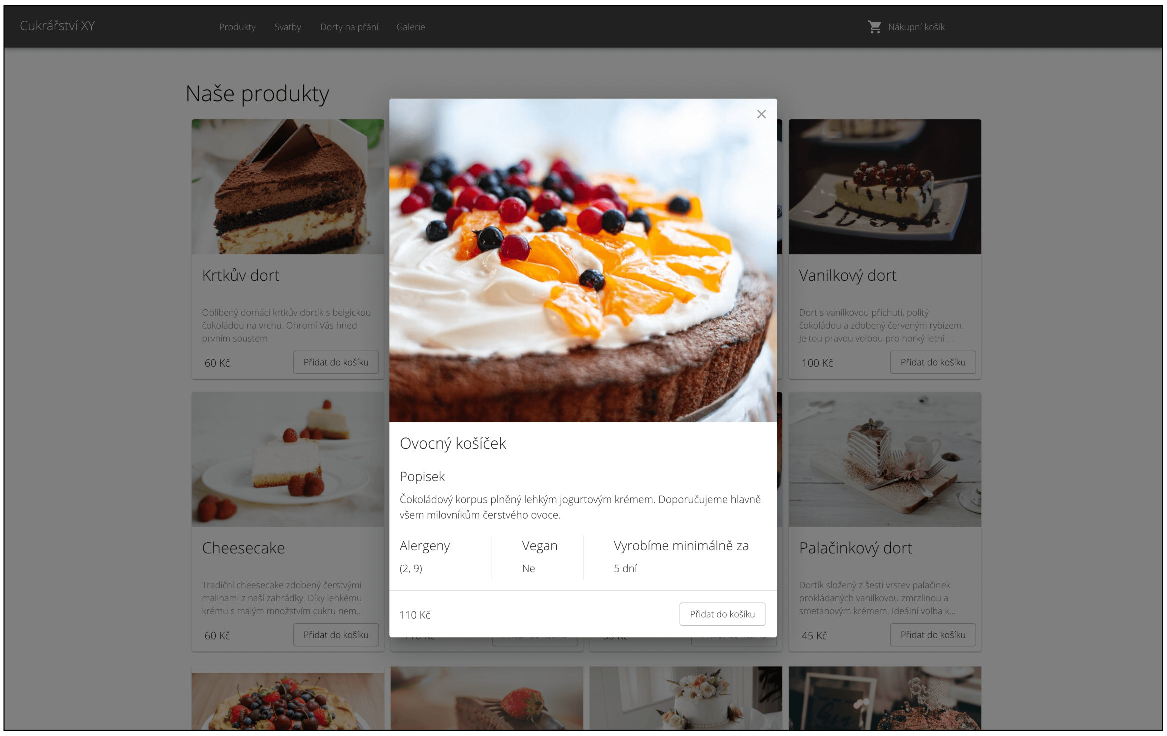
Cheesecake
Tradiční cheesecake zdobený čerstvými malinami z naší zahrádky. Díky lehkému krému s malým množstvím cukru nem...
60 Kč Přidat do košíku

Ovocný košíček
Čokoládový korpus plněný lehkým jogurtovým krémem. Doporučujeme hlavně všem milovníkům čerstvého ovoce.
110 Kč Přidat do košíku

Čokoládový dort
Hutný čokoládový dort složený ze čtyř druhů pravé belgické čokolády, která stojí za hřích. Dort je potažený ja...
50 Kč Přidat do košíku

Palačinkový dort
Dortík složený z šesti vrstev palačinek prokládaných vanilkovou zmrzlinou a smetanovým krémem. Ideální volba k...
45 Kč Přidat do košíku

Figure C.3: Hi-fi – Product list



Naše produkty

Krtkův dort

Oblíbený domácí krtkův dortík s belgickou čokoládou na vrchu. Ohromí Vás hned prvním soustem.

60 Kč Přidat do košíku

Vanilkový dort

Dort s vanilkovou příchutí, politý čokoládou a zdobený červeným rybízem. Je tou pravou volbou pro horký letní ...

100 Kč Přidat do košíku

Ovocný košíček

Popisek

Čokoládový korpus plněný lehkým jogurtovým krémem. Doporučujeme hlavně všem milovníkům čerstvého ovoce.

Alergeny	Vegan	Vyrobíme minimálně za
(2, 9)	Ne	5 dní

110 Kč Přidat do košíku

Cheesecake

Tradiční cheesecake zdobený čerstvými malinami z naší zahrádky. Díky lehkému krému s malým množstvím cukru nem...

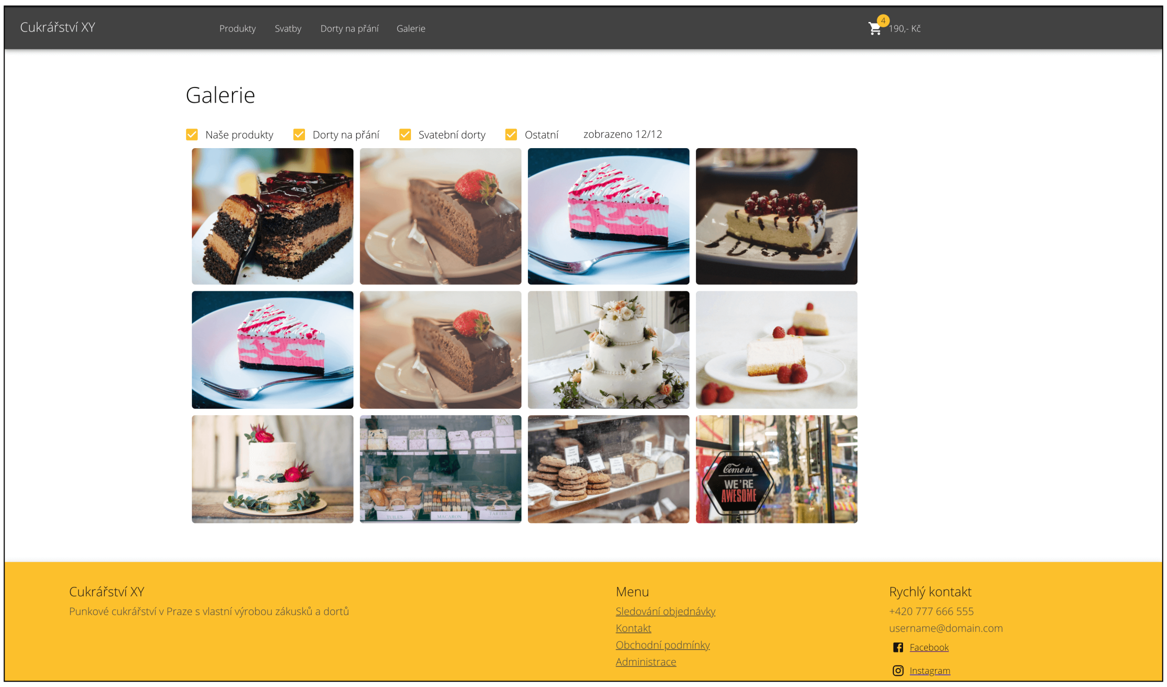
60 Kč Přidat do košíku

Palačinkový dort

Dortík složený z šesti vrstev palačinek prokládaných vanilkovou zmrzlinou a smetanovým krémem. Ideální volba k...

45 Kč Přidat do košíku

Figure C.4: Hi-fi – Product detail



Galerie

✓ Naše produkty ✓ Dorty na přání ✓ Svatební dorty ✓ Ostatní zobrazeno 12/12

Cukrářství XY

Punkové cukrářství v Praze s vlastní výrobou zákusků a dortů

Menu

Sledování objednávky

Kontakt

Obchodní podmínky

Administrace

Rychlý kontakt

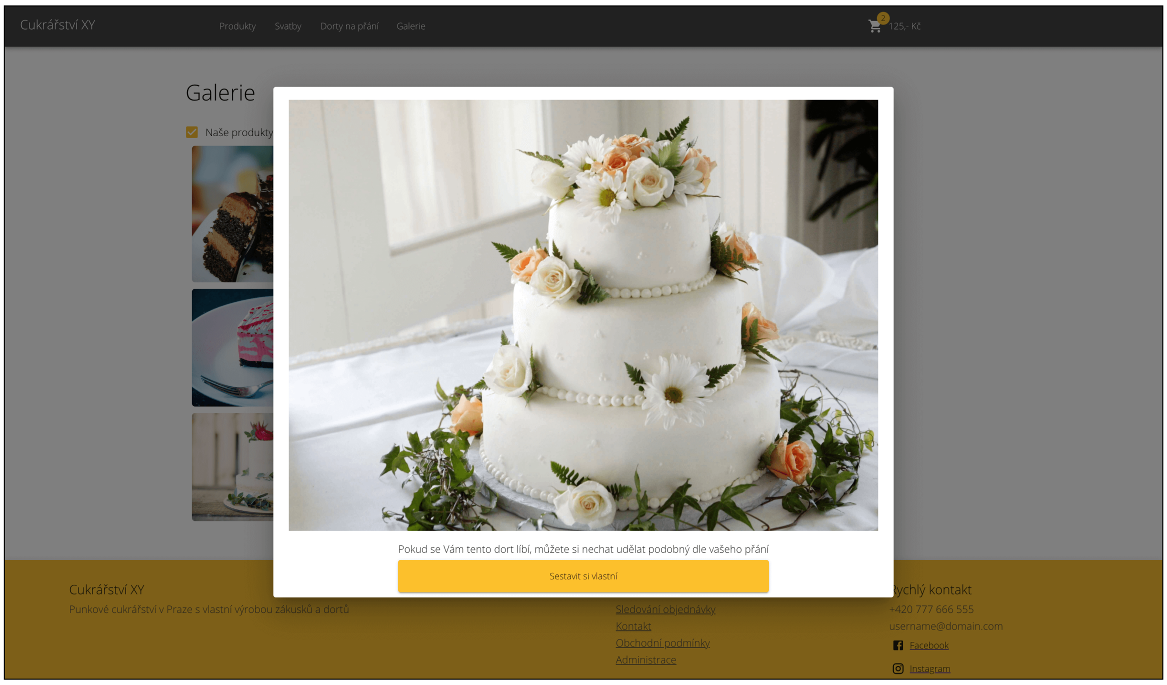
+420 777 666 555

username@domain.com

Facebook

Instagram

Figure C.5: Hi-fi – Gallery



Galerie

Naše produkty

Pokud se Vám tento dort líbí, můžete si nechat udělat podobný dle vašeho přání

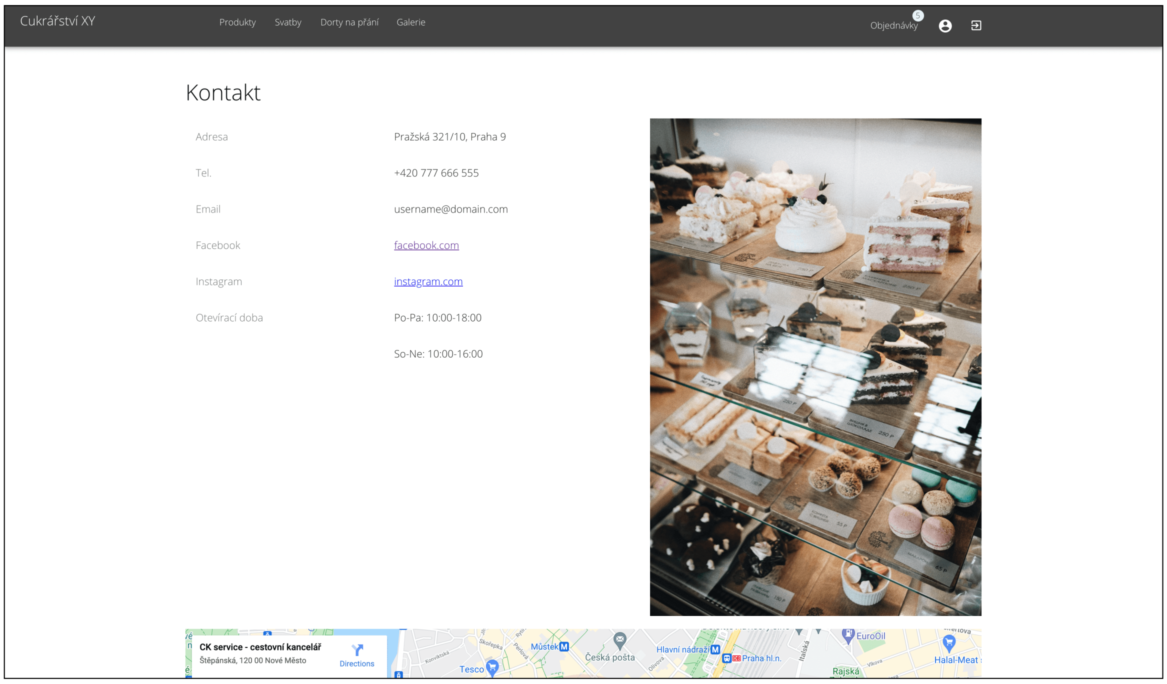
Cukrářství XY

Punkové cukrářství v Praze s vlastní výrobou zákusků a dortů

Sledování objednávky
Kontakt
Obchodní podmínky
Administrace

Rychlý kontakt
+420 777 666 555
username@domain.com
Facebook
Instagram

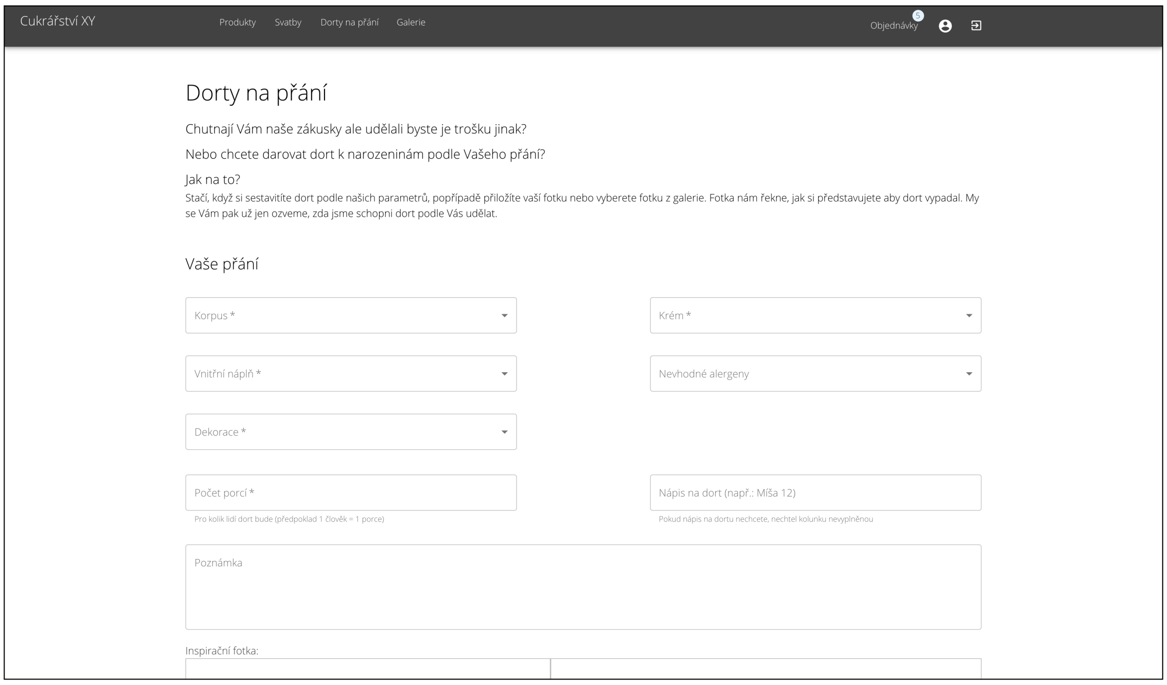
Figure C.6: Hi-fi – Gallery's picture detail



Kontakt

Adresa	Pražská 321/10, Praha 9
Tel.	+420 777 666 555
Email	username@domain.com
Facebook	facebook.com
Instagram	instagram.com
Otevírací doba	Po-Pa: 10:00-18:00
	So-Ne: 10:00-16:00

Figure C.7: Contacts


Cukrářství XY
Produkty Svatby Dorty na přání Galerie
Objednávky

Dorty na přání

Chutnají Vám naše zákusky ale udělali byste je trošku jinak?

Nebo chcete darovat dort k narozeninám podle Vašeho přání?

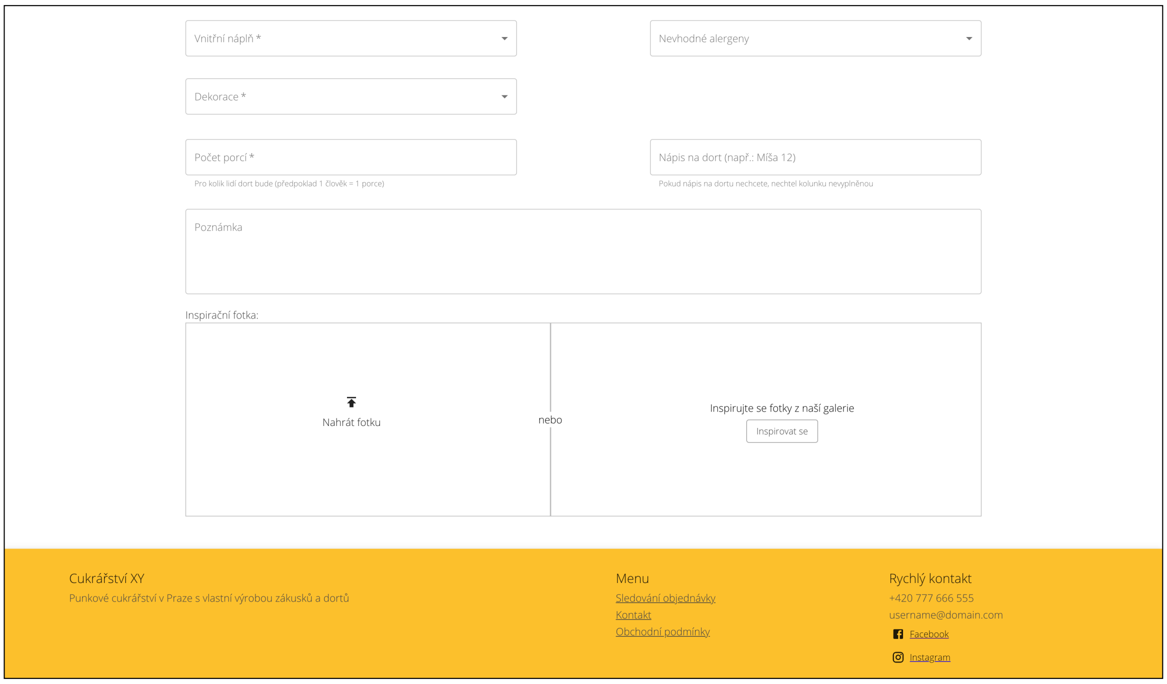
Jak na to?

Stačí, když si sestavitíte dort podle našich parametrů, popřípadě přiložíte vaší fotku nebo vyberete fotku z galerie. Fotka nám řekne, jak si představujete aby dort vypadal. My se Vám pak už jen ozveme, zda jsme schopni dort podle Vás udělat.

Vaše přání

Korpus *
Krém *
Vnitřní náplň *
Nevhodné alergeny
Dekorace *
Počet porcí *
Pro kolik lidí dort bude (předpoklad 1 člověk = 1 porce)
Nápis na dort (např.: Míša 12)
Pokud nápis na dortu nechcete, nechtel kolunku nevyplněnou
Poznámka
Inspirační fotka:

Figure C.8: Creating a wedding cake

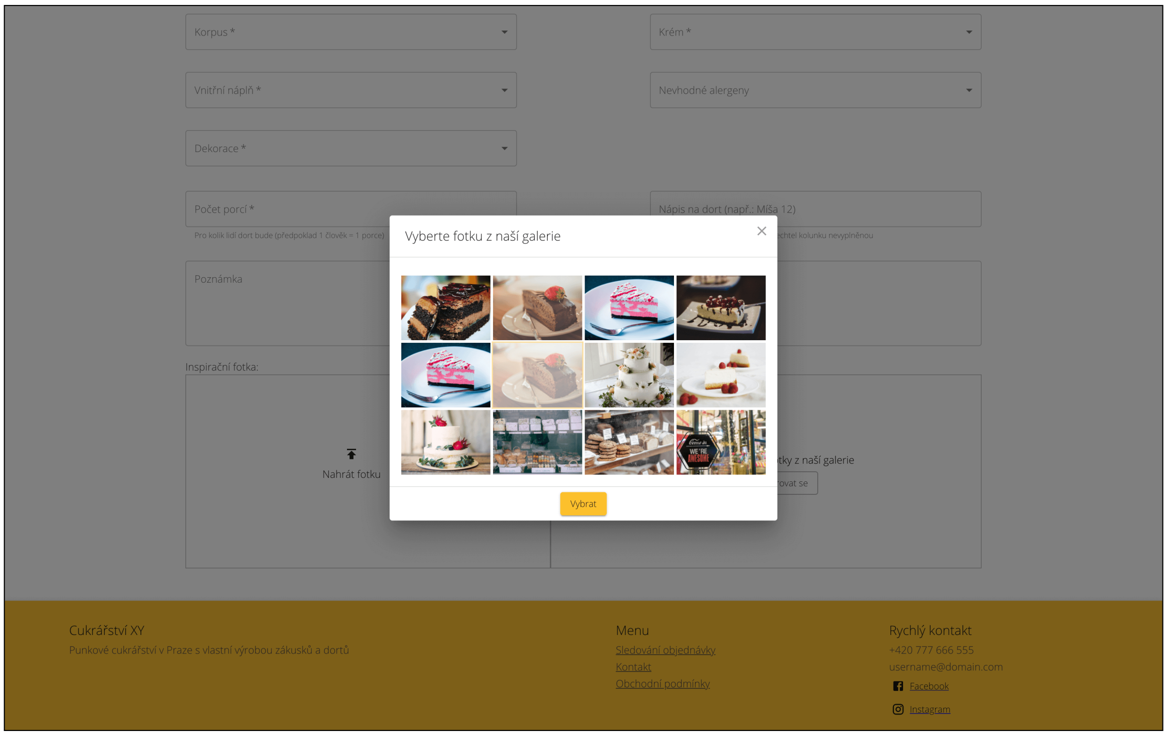

Vnitřní náplň *
Nevhodné alergeny
Dekorace *
Počet porcí *
Pro kolik lidí dort bude (předpoklad 1 člověk = 1 porce)
Nápis na dort (např.: Míša 12)
Pokud nápis na dortu nechcete, nechtel kolunku nevyplněnou
Poznámka
Inspirační fotka:
Nahrát fotku
nebo
Inspirujte se fotky z naší galerie
Inspirovat se

Cukrářství XY
Punkové cukrářství v Praze s vlastní výrobou zákusků a dortů

Menu
Sledování objednávky
Kontakt
Obchodní podmínky

Rychlý kontakt
+420 777 666 555
username@domain.com
Facebook
Instagram

Figure C.9: Custom product – inspiration in the gallery


Korpus *

Krém *

Vnitřní náplň *

Nevhodné alergeny

Dekorace *

Počet porcí *

Pro kolik lidí dort bude (předpoklad 1 člověk = 1 porce)

Nápis na dort (např.: Míša 12)

chtel kolunku nevyplněnou

Poznámka

Inspirační fotka:

Nahrát fotku

tky z naší galerie

ovat se

Vyberte fotku z naší galerie

Vybrat

Cukrářství XY

Punkové cukrářství v Praze s vlastní výrobou zákusků a dortů

Menu

Sledování objednávky

Kontakt

Obchodní podmínky

Rychlý kontakt

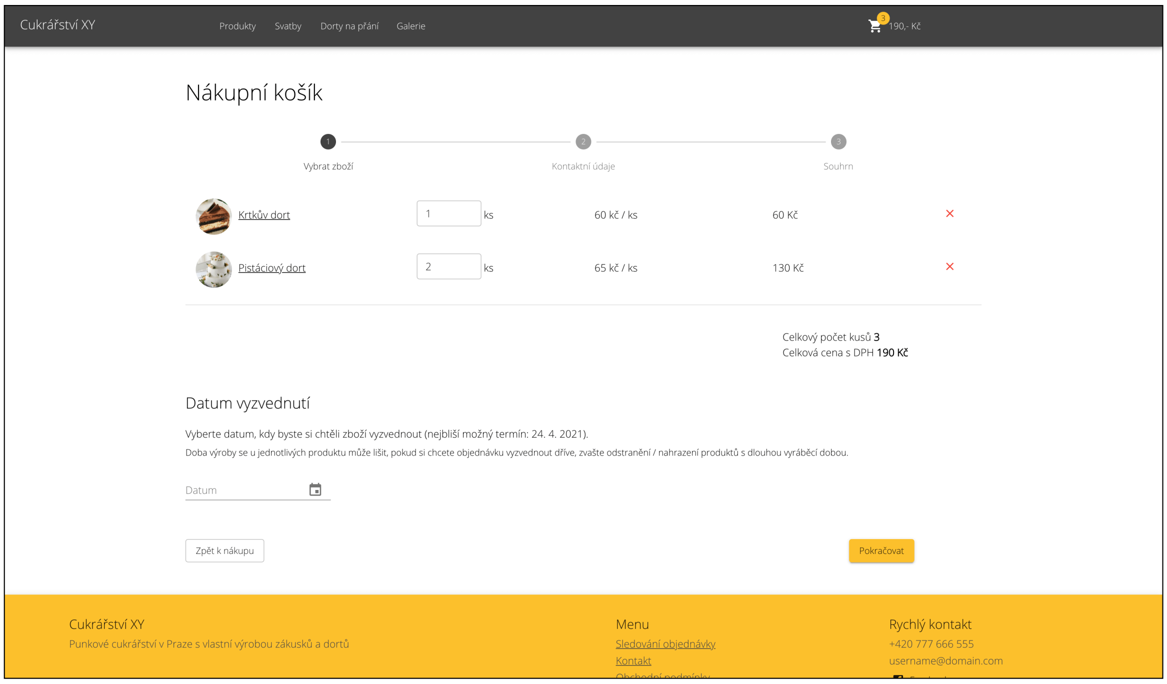
+420 777 666 555

username@domain.com

Facebook

Instagram

Figure C.10: Custom product – inspiration in the gallery


Cukrářství XY

Produkty Svatby Dorty na přání Galerie

3 190,- Kč

Nákupní košík

1 Vybrat zboží — 2 Kontaktní údaje — 3 Souhrn

Krtkův dort	1 ks	60 kč / ks	60 Kč	×
Pistáciový dort	2 ks	65 kč / ks	130 Kč	×

Celkový počet kusů **3**
Celková cena s DPH **190 Kč**

Datum vyzvednutí

Vyberte datum, kdy byste si chtěli zboží vyzvednout (nejbliší možný termín: 24. 4. 2021).

Doba výroby se u jednotlivých produktu může lišit, pokud si chcete objednávku vyzvednout dříve, zvašte odstranění / nahrazení produktů s dlouhou vyráběcí dobou.

Datum

Zpět k nákupu

Pokračovat

Cukrářství XY

Punkové cukrářství v Praze s vlastní výrobou zákusků a dortů

Menu

Sledování objednávky

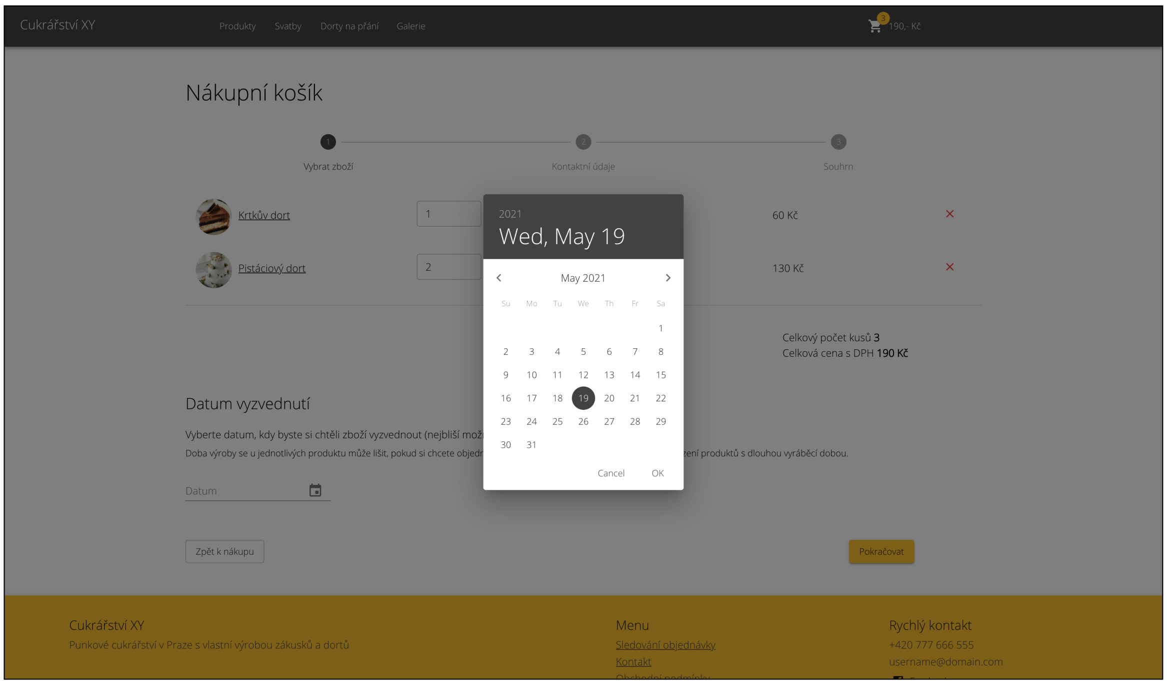
Kontakt

Rychlý kontakt

+420 777 666 555

username@domain.com

Figure C.11: Order creating – shopping cart



Nákupní košík

Krtkův dort	1	60 Kč
Pistáciový dort	2	130 Kč

2021

Wed, May 19

May 2021

Su	Mo	Tu	We	Th	Fr	Sa
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Celkový počet kusů **3**
Celková cena s DPH **190 Kč**

Datum vyzvednutí

Vyberte datum, kdy byste si chtěli zboží vyzvednout (nejbliší mož

Doba výroby se u jednotlivých produktu může lišit, pokud si chcete objedn ení produktů s dlouhou vyráběcí dobou.

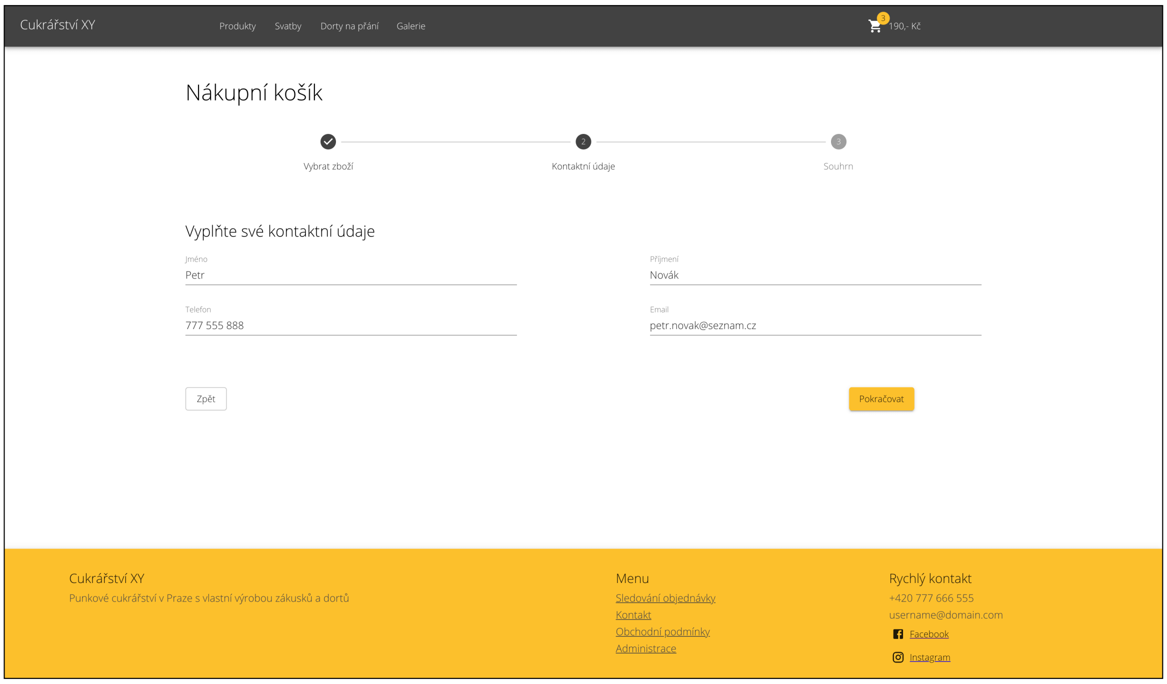
Datum

Cukrářství XY
Punkové cukrářství v Praze s vlastní výrobou zákusků a dortů

Menu
Sledování objednávky
Kontakt

Rychlý kontakt
+420 777 666 555
username@domain.com

Figure C.12: Order – selecting date



Nákupní košík

Vyplňte své kontaktní údaje

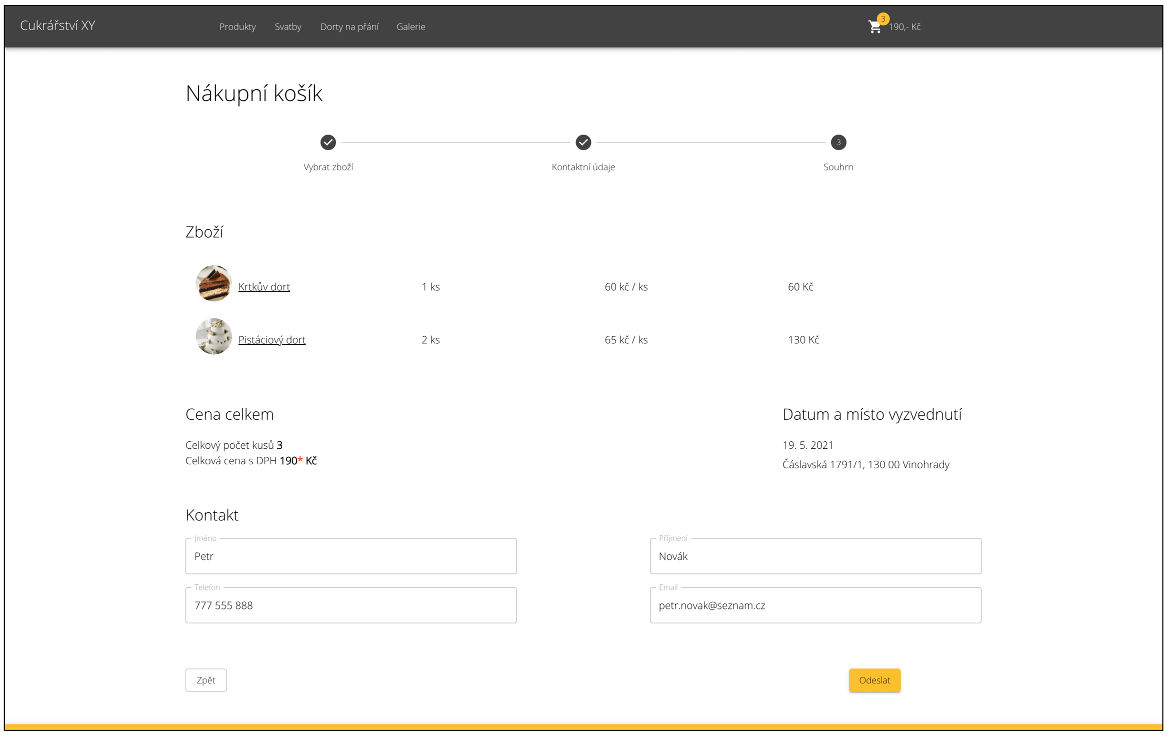
Jméno: Petr	Příjmení: Novák
Telefon: 777 555 888	Email: petr.novak@seznam.cz

Cukrářství XY
Punkové cukrářství v Praze s vlastní výrobou zákusků a dortů

Menu
Sledování objednávky
Kontakt
Obchodní podmínky
Administrace

Rychlý kontakt
+420 777 666 555
username@domain.com
Facebook
Instagram

Figure C.13: Order creating – contact information



Nákupní košík

Zboží

Krtkův dort	1 ks	60 kč / ks	60 Kč
Pistáciový dort	2 ks	65 kč / ks	130 Kč

Cena celkem

Celkový počet kusů **3**
Celková cena s DPH **190*** **Kč**

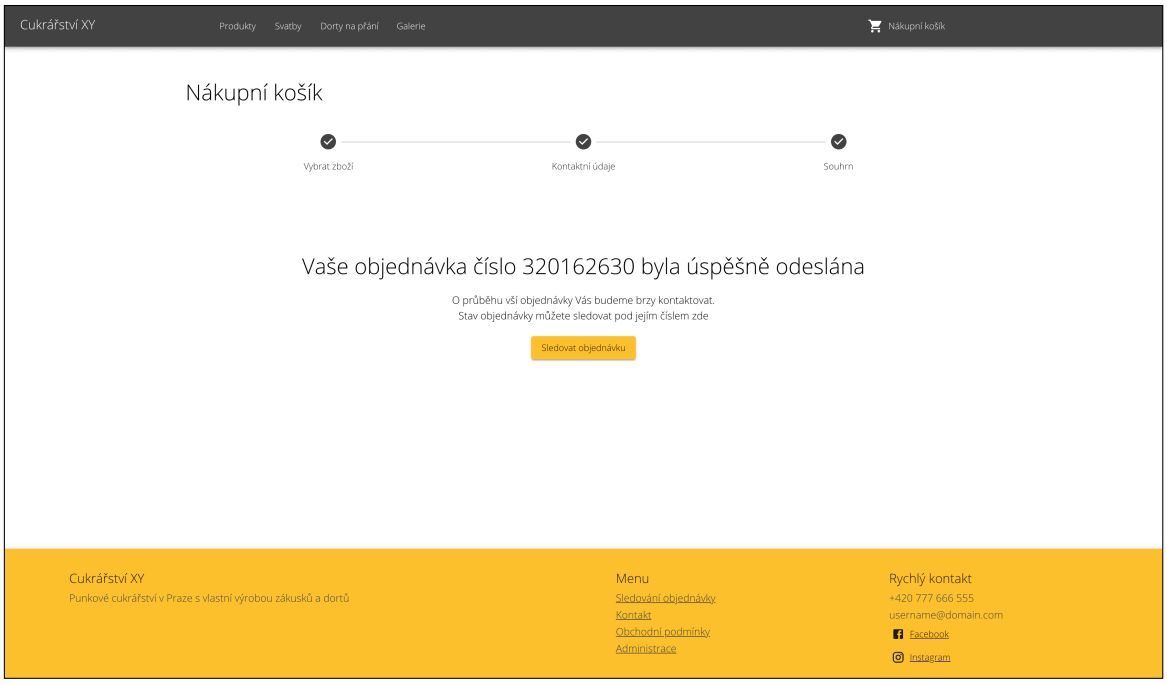
Datum a místo vyzvednutí

19. 5. 2021
Čáslavská 1791/1, 130 00 Vinohrady

Kontakt

Jméno: Petr	Příjmení: Novák
Telefon: 777 555 888	Email: petr.novak@seznam.cz

Figure C.14: Order creating – summary



Nákupní košík

Vaše objednávka číslo 320162630 byla úspěšně odeslána

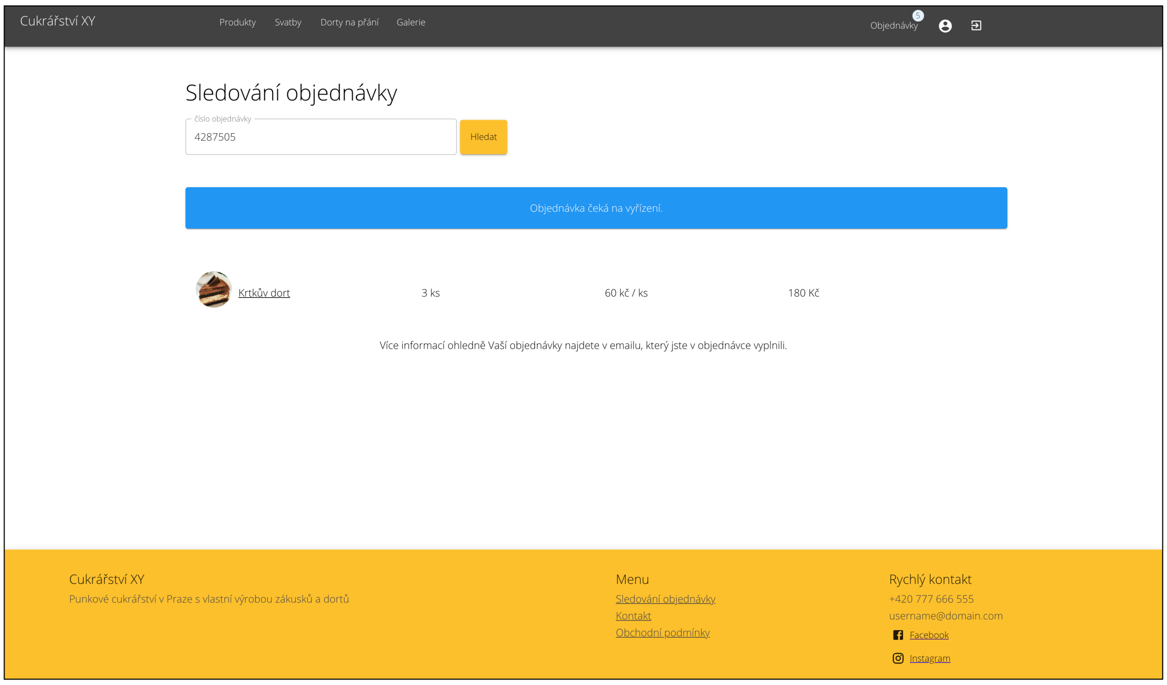
O průběhu vší objednávky Vás budeme brzy kontaktovat.
Stav objednávky můžete sledovat pod jejím číslem zde

Cukrářství XY
Punkové cukrářství v Praze s vlastní výrobou zákusků a dortů

Menu
Sledování objednávky
Kontakt
Obchodní podmínky
Administrace

Rychlý kontakt
+420 777 666 555
username@domain.com
Facebook
Instagram

Figure C.15: New order created



Sledování objednávky

Číslo objednávky: 4287505

Hledat

Objednávka čeká na vyřízení.

Krtkův dort	3 ks	60 kč / ks	180 Kč

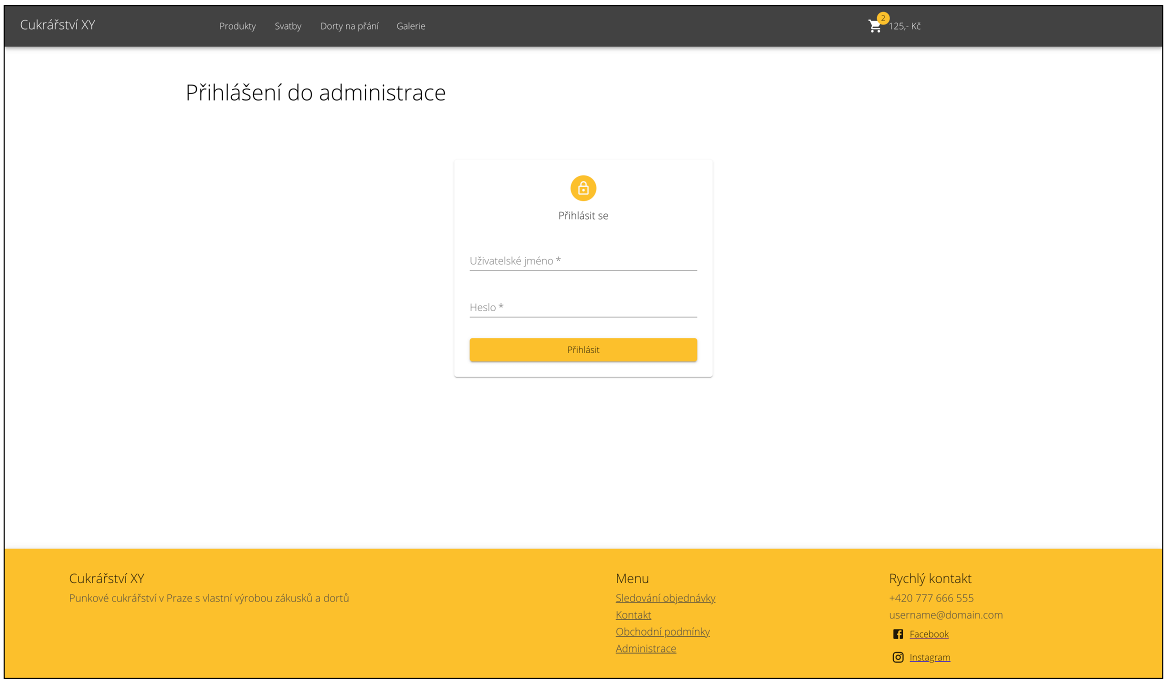
Více informací ohledně Vaší objednávky najdete v emailu, který jste v objednávce vyplnili.

Cukrářství XY
Punkové cukrářství v Praze s vlastní výrobou zákusků a dortů

Menu
Sledování objednávky
Kontakt
Obchodní podmínky

Rychlý kontakt
+420 777 666 555
username@domain.com
Facebook
Instagram

Figure C.16: Check order status



Přihlášení do administrace

Přihlásit se

Uživatelské jméno *

Heslo *

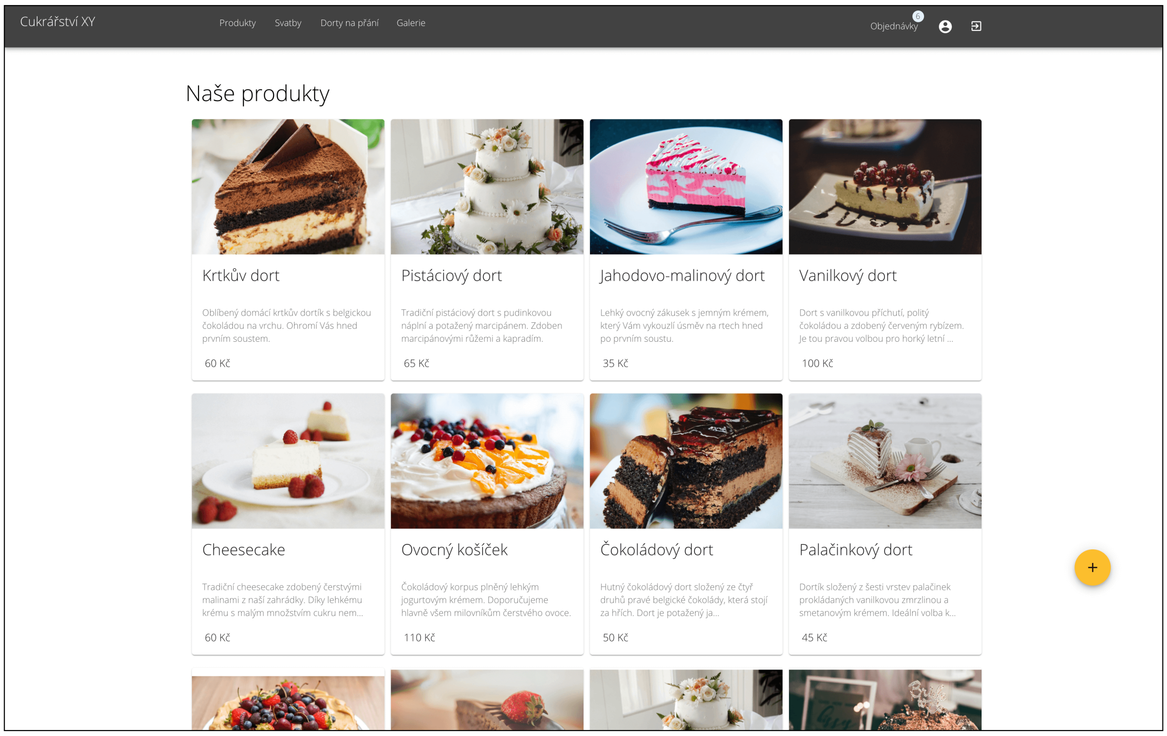
Přihlásit

Cukrářství XY
Punkové cukrářství v Praze s vlastní výrobou zákusků a dortů

Menu
Sledování objednávky
Kontakt
Obchodní podmínky
Administrace

Rychlý kontakt
+420 777 666 555
username@domain.com
Facebook
Instagram

Figure C.17: Login to administration



Naše produkty

Krtkův dort

Oblíbený domácí krtkův dortík s belgickou čokoládou na vrchu. Ohromí Vás hned prvním soustem.

60 Kč

Pistáciový dort

Tradiční pistáciový dort s pudinkovou náplní a potažený marcipánem. Zdoben marcipánovými růžemi a kapradím.

65 Kč

Jahodovo-malinový dort

Lehký ovocný zákusek s jemným krémem, který Vám vykouzlí úsměv na rtech hned po prvním soustu.

35 Kč

Vanilkový dort

Dort s vanilkovou příchutí, politý čokoládou a zdobený červeným rybízem. Je tou pravou volbou pro horký letní ...

100 Kč

Cheesecake

Tradiční cheesecake zdobený čerstvými malinami z naší zahrádky. Díky lehkému krému s malým množstvím cukru nem...

60 Kč

Ovocný košíček

Čokoládový korpus plněný lehkým jogurtovým krémem. Doporučujeme hlavně všem milovníkům čerstvého ovoce.

110 Kč

Čokoládový dort

Hutný čokoládový dort složený ze čtyř druhů pravé belgické čokolády, která stojí za hřích. Dort je potažený ja...

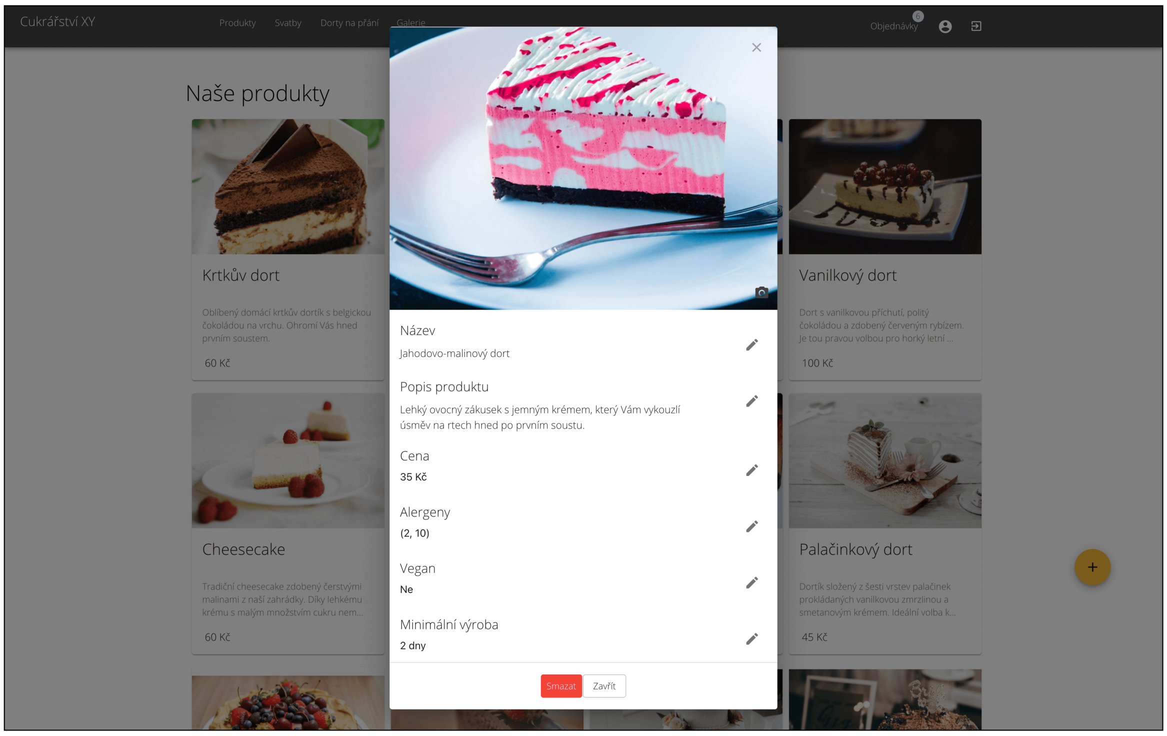
50 Kč

Palačinkový dort

Dortík složený z šesti vrstev palačinek prokládaných vanilkovou zmrzlinou a smetanovým krémem. Ideální volba k...

45 Kč

Figure C.18: Administrator – product list



Název

Jahodovo-malinový dort

Popis produktu

Lehký ovocný zákusek s jemným krémem, který Vám vykouzlí úsměv na rtech hned po prvním soustu.

Cena

35 Kč

Alergeny

(2, 10)

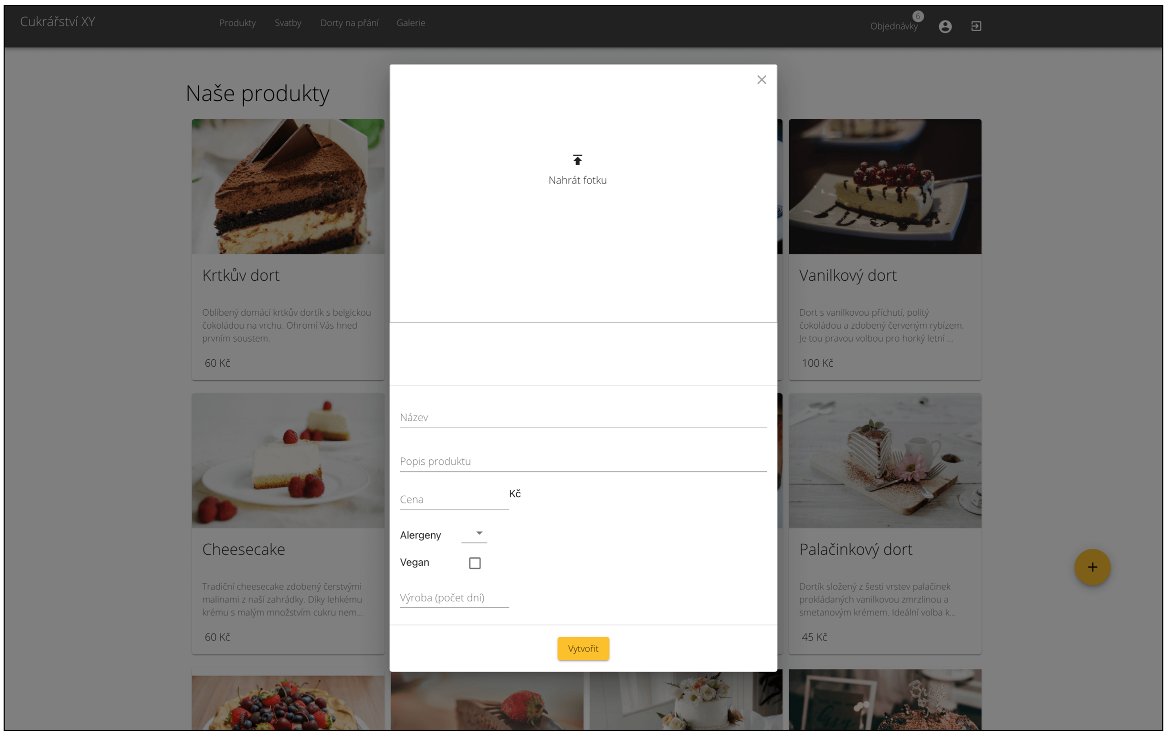
Vegan

Ne

Minimální výroba

2 dny

Figure C.19: Administrator – product detail



Naše produkty

Krtkův dort

Oblíbený domácí krtkův dortík s belgickou čokoládou na vrchu. Ohromí Vás hned prvním soustem.

60 Kč

Vanilkový dort

Dort s vanilkovou příchutí, politý čokoládou a zdobený červeným rybízem. Je tou pravou volbou pro horký letní ...

100 Kč

Cheesecake

Tradiční cheesecake zdobený čerstvými malinami z naší zahrádky. Díky lehkému krému s malým množstvím cukru nem...

60 Kč

Palačinkový dort

Dortík složený z šesti vrstev palačinek prokládaných vanilkovou zmrzlinou a smetanovým krémem. Ideální volba k...

45 Kč

Název

Popis produktu

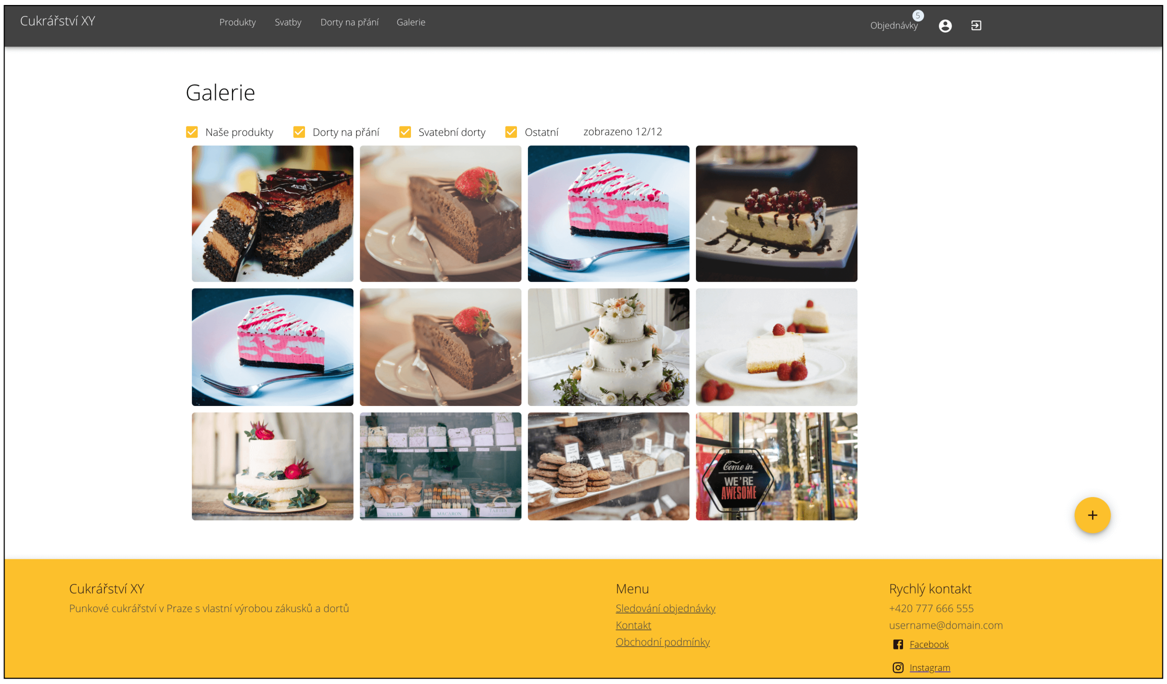
Cena Kč

Alergeny

Vegan ☐

Výroba (počet dní)

Figure C.20: Administrator – create a new product



Galerie

✓ Naše produkty ✓ Dorty na přání ✓ Svatební dorty ✓ Ostatní zobrazeno 12/12

Cukrářství XY

Punkové cukrářství v Praze s vlastní výrobou zákusků a dortů

Menu

Sledování objednávky

Kontakt

Obchodní podmínky

Rychlý kontakt

+420 777 666 555

username@domain.com

Facebook

Instagram

Figure C.21: Administrator – gallery

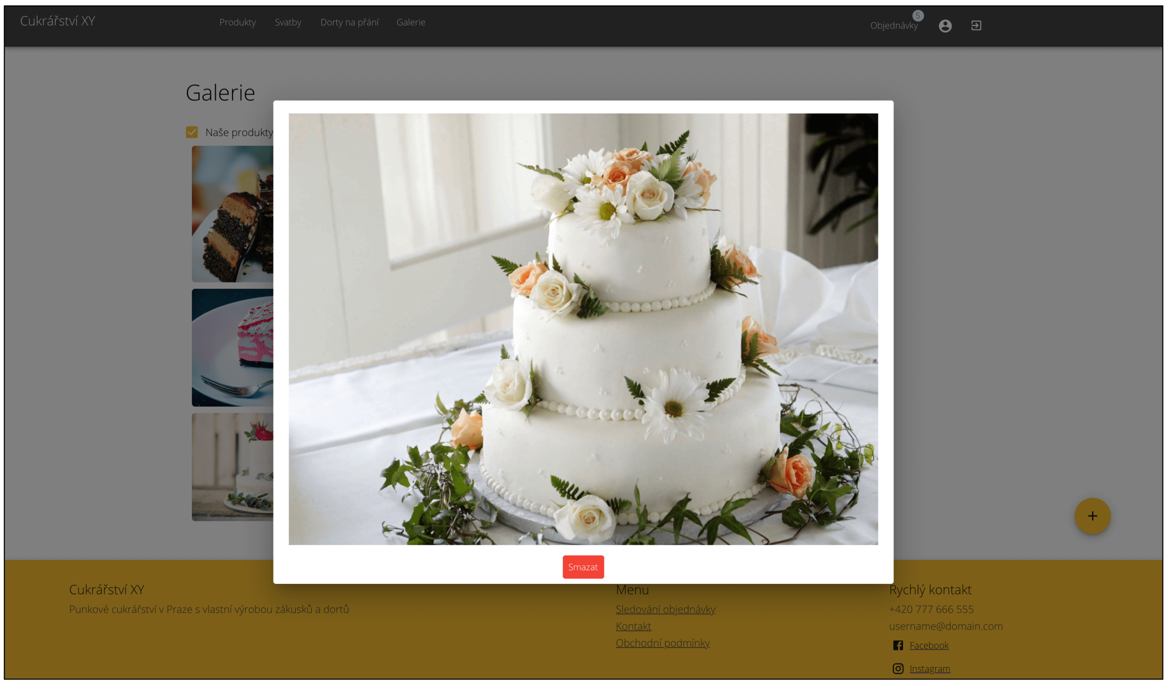


Figure C.22: Administrator – gallery's picture detail

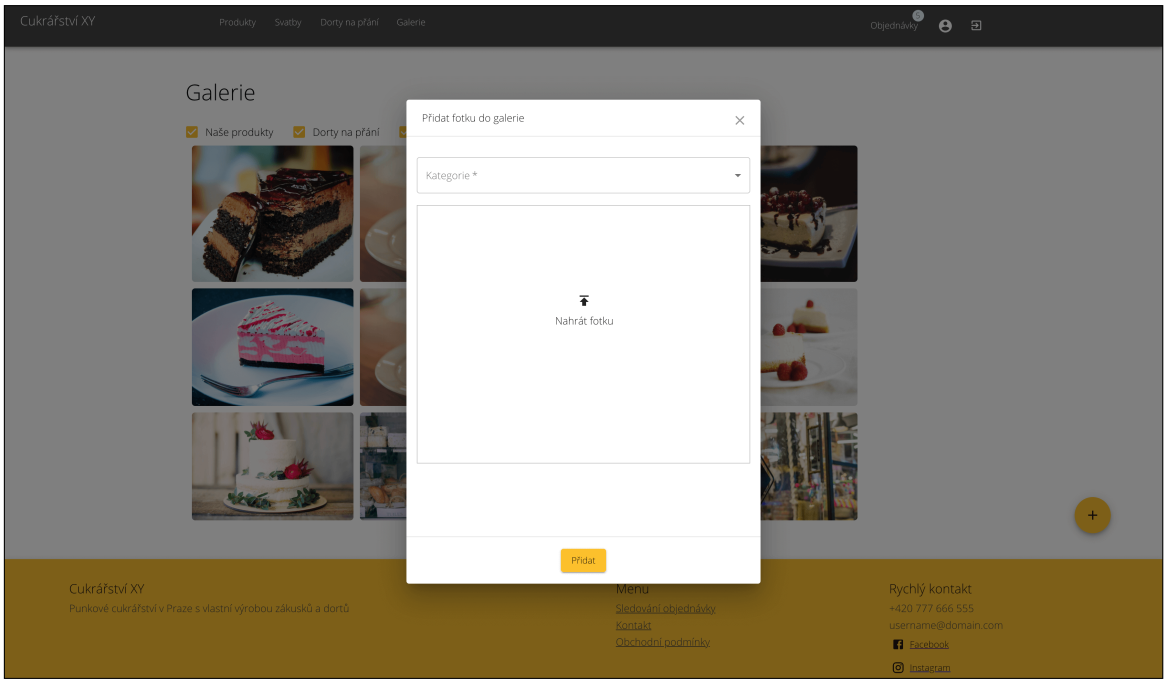
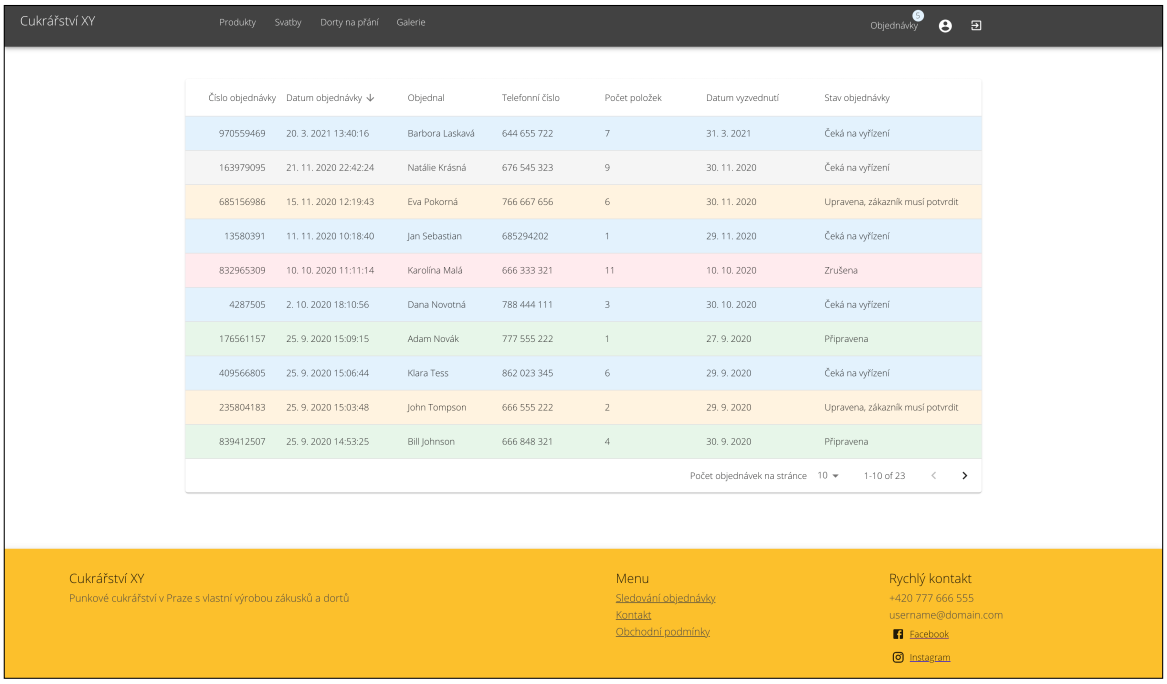


Figure C.23: Administrator – add picture to gallery



Číslo objednávky	Datum objednávky ↓	Objednal	Telefonní číslo	Počet položek	Datum vyzvednutí	Stav objednávky
970559469	20. 3. 2021 13:40:16	Barbora Laskavá	644 655 722	7	31. 3. 2021	Čeká na vyřízení
163979095	21. 11. 2020 22:42:24	Natálie Krásná	676 545 323	9	30. 11. 2020	Čeká na vyřízení
685156986	15. 11. 2020 12:19:43	Eva Pokorná	766 667 656	6	30. 11. 2020	Upravena, zákazník musí potvrdit
13580391	11. 11. 2020 10:18:40	Jan Sebastian	685294202	1	29. 11. 2020	Čeká na vyřízení
832965309	10. 10. 2020 11:11:14	Karolína Malá	666 333 321	11	10. 10. 2020	Zrušena
4287505	2. 10. 2020 18:10:56	Dana Novotná	788 444 111	3	30. 10. 2020	Čeká na vyřízení
176561157	25. 9. 2020 15:09:15	Adam Novák	777 555 222	1	27. 9. 2020	Připravena
409566805	25. 9. 2020 15:06:44	Klara Tess	862 023 345	6	29. 9. 2020	Čeká na vyřízení
235804183	25. 9. 2020 15:03:48	John Tompson	666 555 222	2	29. 9. 2020	Upravena, zákazník musí potvrdit
839412507	25. 9. 2020 14:53:25	Bill Johnson	666 848 321	4	30. 9. 2020	Připravena

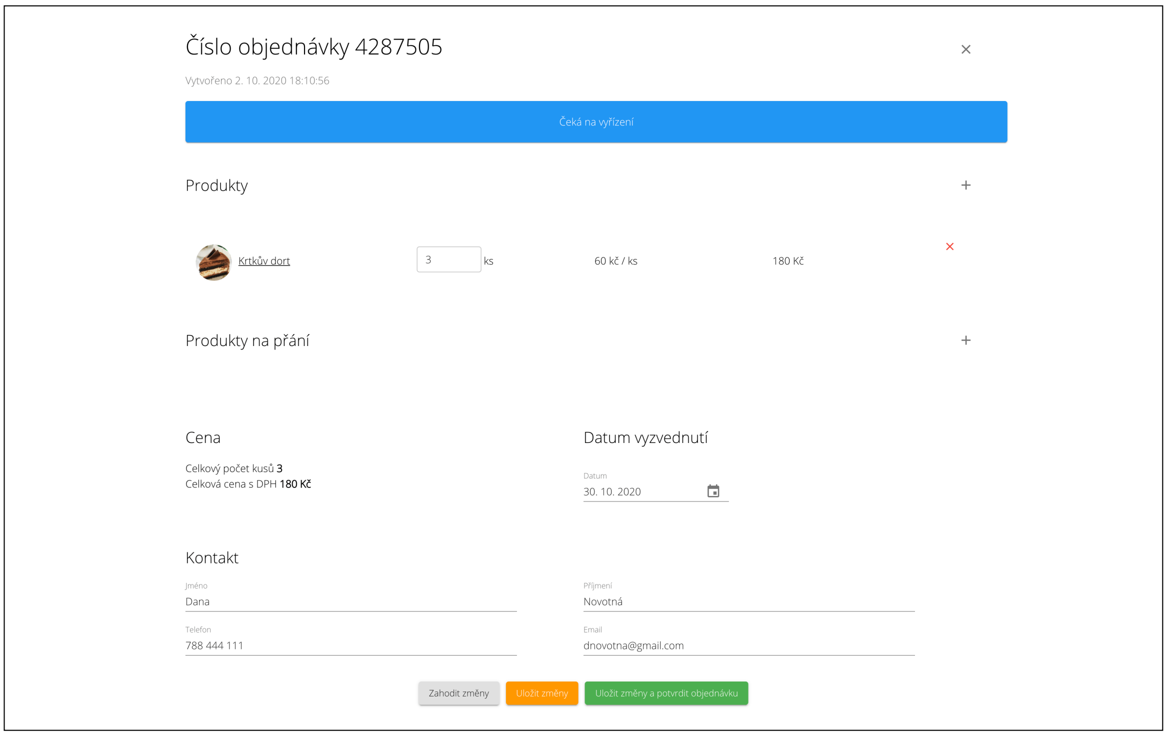
Počet objednávek na stránce 10 ▾ 1-10 of 23

Cukrářství XY
Punkové cukrářství v Praze s vlastní výrobou zákusků a dortů

Menu
Sledování objednávky
Kontakt
Obchodní podmínky

Rychlý kontakt
+420 777 666 555
username@domain.com
Facebook
Instagram

Figure C.24: Hi-fi – Order list


Číslo objednávky 4287505

Vytvořeno 2. 10. 2020 18:10:56

Čeká na vyřízení

Produkty

Krtkův dort 3 ks 60 kč / ks 180 Kč

Produkty na přání

Cena

Celkový počet kusů **3**
Celková cena s DPH **180 Kč**

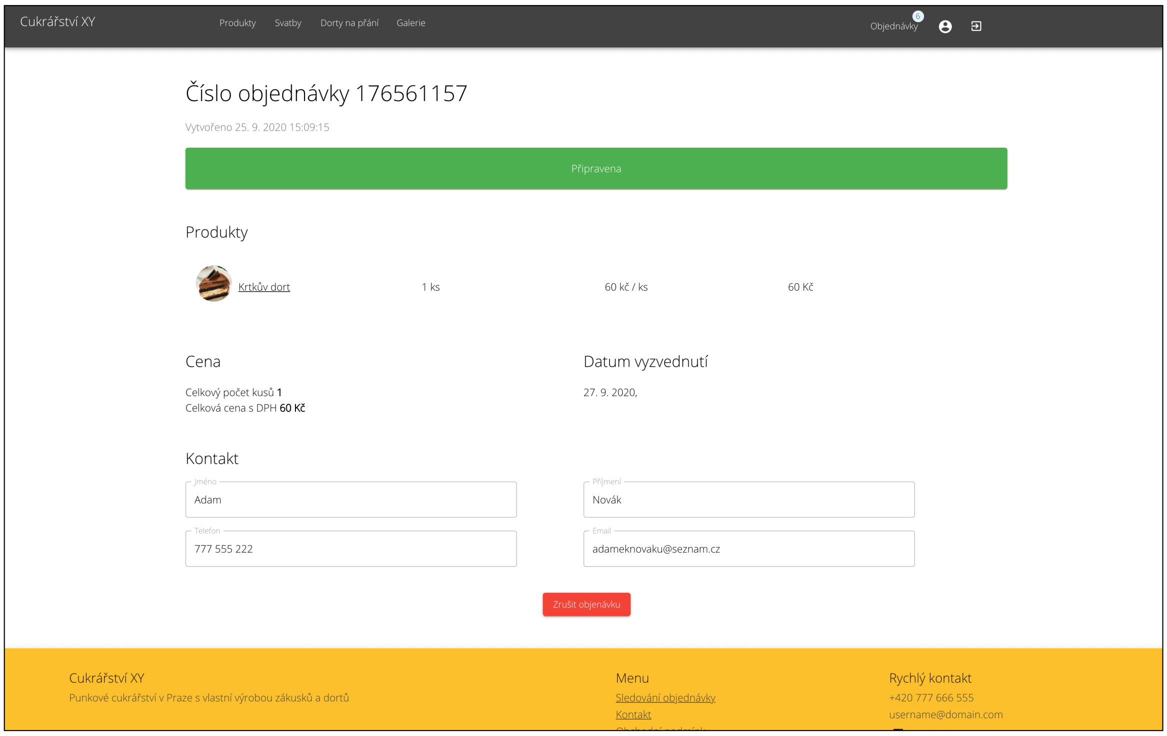
Datum vyzvednutí

Datum: 30. 10. 2020

Kontakt

Jméno: Dana
Příjmení: Novotná
Telefon: 788 444 111
Email: dnovotna@gmail.com

Figure C.25: Hi-fi – Administrator – detail of a new order



Číslo objednávky 176561157

Vytvořeno 25. 9. 2020 15:09:15

Připravena

Produkty

Krtkův dort	1 ks	60 kč / ks	60 Kč

Cena

Celkový počet kusů **1**
Celková cena s DPH **60 Kč**

Datum vyzvednutí

27. 9. 2020,

Kontakt

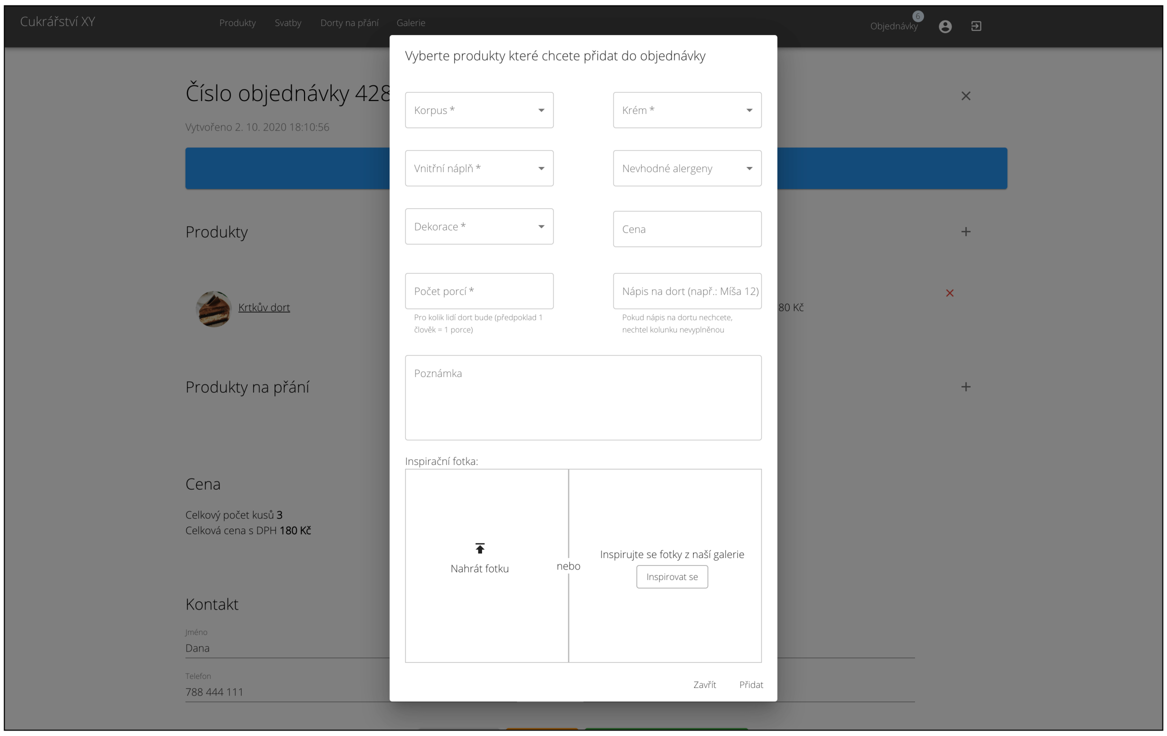
Jméno: Adam
Příjmení: Novák
Telefon: 777 555 222
Email: adameknovaku@seznam.cz

Cukrářství XY
Punkové cukrářství v Praze s vlastní výrobou zákusků a dortů

Menu
Sledování objednávky
Kontakt

Rychlý kontakt
+420 777 666 555
username@domain.com

Figure C.26: Administrator – detail of the accepted order



Číslo objednávky 428

Vytvořeno 2. 10. 2020 18:10:56

Produkty

Krtkův dort 80 Kč

Produkty na přání

Cena

Celkový počet kusů **3**
Celková cena s DPH **180 Kč**

Kontakt

Jméno: Dana
Telefon: 788 444 111

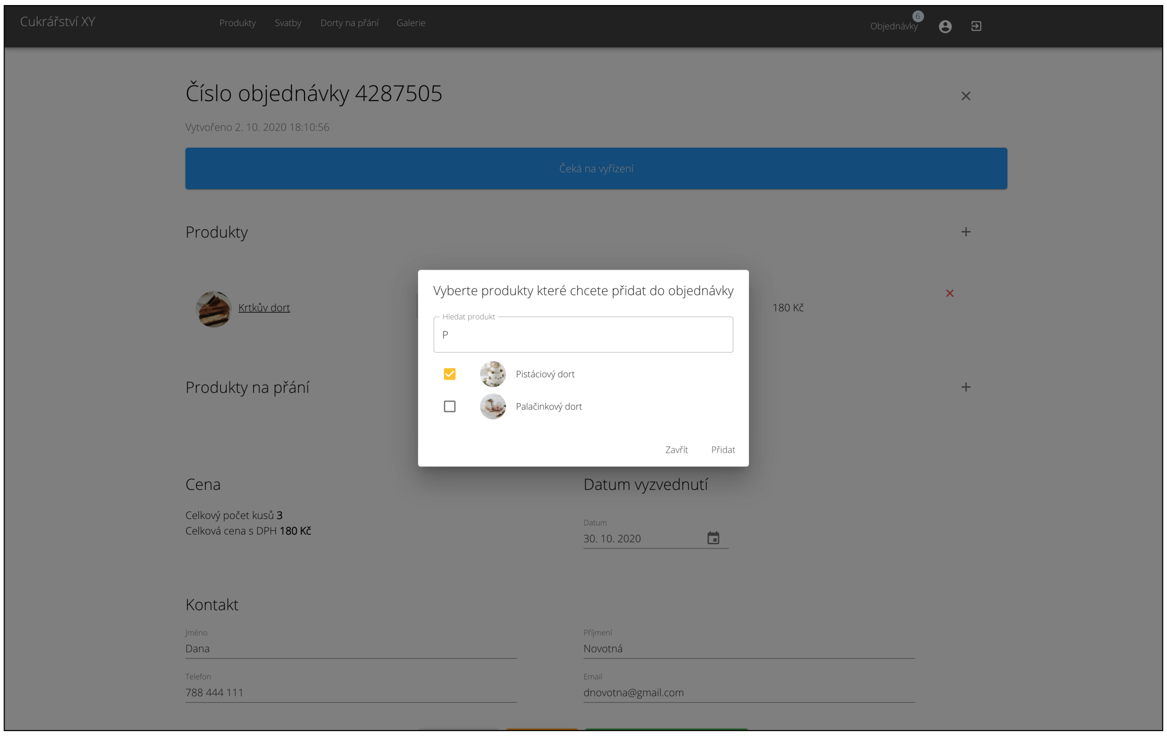
Vyberte produkty které chcete přidat do objednávky

Korpus *
Krém *
Vnitřní náplň *
Nevhodné alergeny
Dekorace *
Cena
Počet porcí *
Pro kolik lidí dort bude (předpoklad 1 člověk = 1 porce)
Nápis na dort (např.: Míša 12)
Pokud nápis na dortu nechcete, nechtel kolunku nevyplněnou
Poznámka

Inspirační fotka:

Nahrát fotku
nebo
Inspirujte se fotky z naší galerie

Figure C.27: Administrator – add custom product to the order



Cukrářství XY Produkty Svatby Dorty na přání Galerie Objednávky

Číslo objednávky 4287505

Vytvořeno 2. 10. 2020 18:10:56

Čeká na vyřízení

Produkty

Krtkův dort 180 Kč

Produkty na přání

Vyberte produkty které chcete přidat do objednávky

Hledat produkt: P

- [✓] Pistáciový dort
- [ ] Palačinkový dort

Zavřít Přidat

Cena

Celkový počet kusů **3**
Celková cena s DPH **180 Kč**

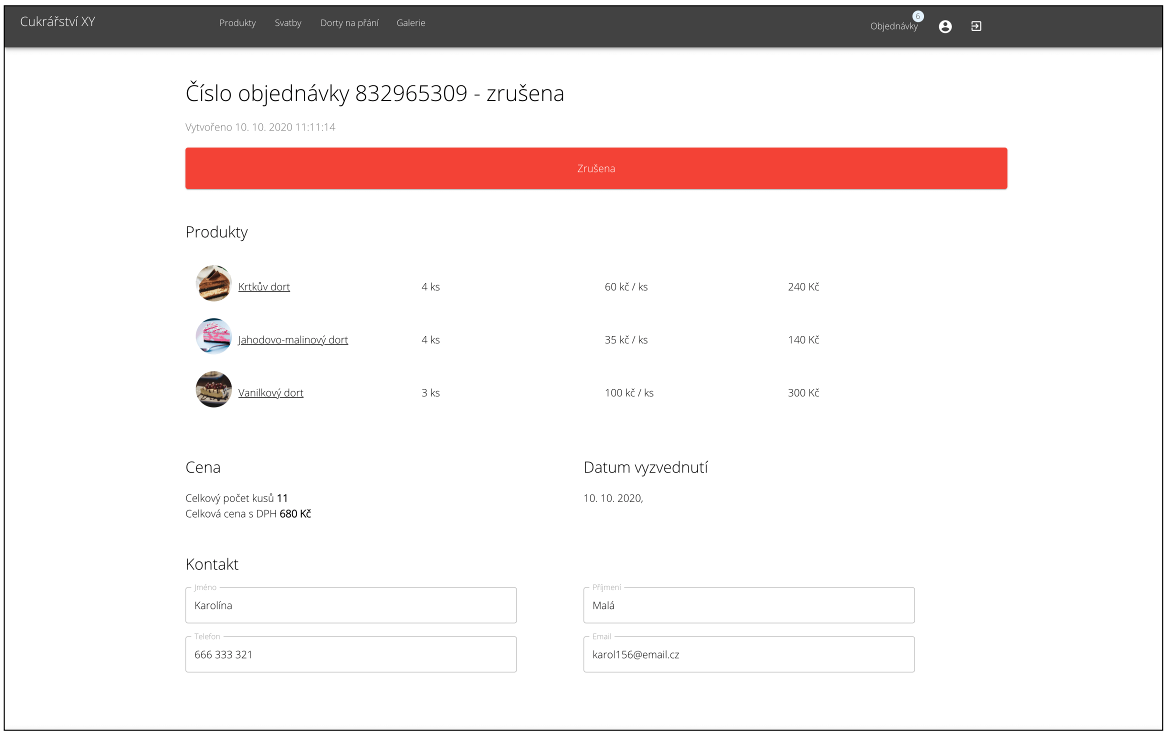
Datum vyzvednutí

Datum: 30. 10. 2020

Kontakt

Jméno: Dana
Příjmení: Novotná
Telefon: 788 444 111
Email: dnovotna@gmail.com

Figure C.28: Administrator – add product to the order



Cukrářství XY Produkty Svatby Dorty na přání Galerie Objednávky

Číslo objednávky 832965309 - zrušena

Vytvořeno 10. 10. 2020 11:11:14

Zrušena

Produkty

Krtkův dort	4 ks	60 kč / ks	240 Kč
Jahodovo-malinový dort	4 ks	35 kč / ks	140 Kč
Vanilkový dort	3 ks	100 kč / ks	300 Kč

Cena

Celkový počet kusů **11**
Celková cena s DPH **680 Kč**

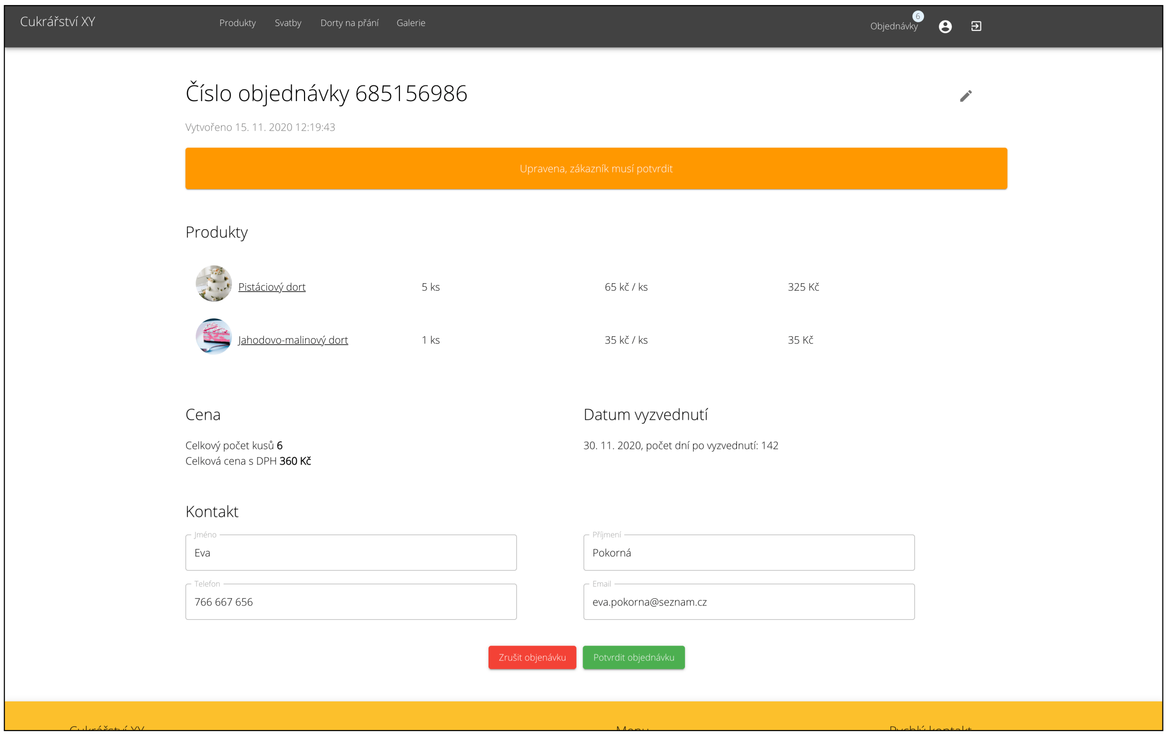
Datum vyzvednutí

10. 10. 2020,

Kontakt

Jméno: Karolína
Příjmení: Malá
Telefon: 666 333 321
Email: karol156@email.cz

Figure C.29: Administrator – detail of the canceled order



Číslo objednávky 685156986

Vytvořeno 15. 11. 2020 12:19:43

Upravena, zákazník musí potvrdit

Produkty

Pistáciový dort	5 ks	65 kč / ks	325 Kč
Jahodovo-malinový dort	1 ks	35 kč / ks	35 Kč

Cena

Celkový počet kusů **6**
Celková cena s DPH **360 Kč**

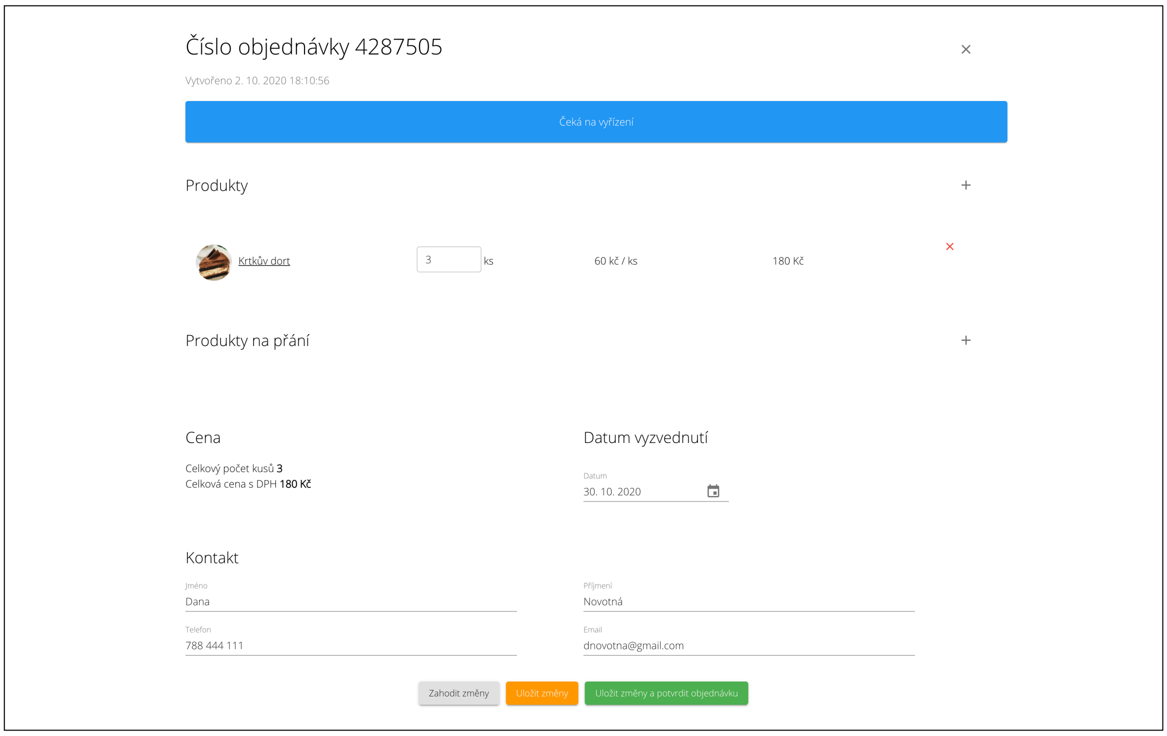
Datum vyzvednutí

30. 11. 2020, počet dní po vyzvednutí: 142

Kontakt

Jméno: Eva	Příjmení: Pokorná
Telefon: 766 667 656	Email: eva.pokorna@seznam.cz

Figure C.30: Administrator – detail of the updated order



Číslo objednávky 4287505

Vytvořeno 2. 10. 2020 18:10:56

Čeká na vyřízení

Produkty

Krtkův dort	3 ks	60 kč / ks	180 Kč

Produkty na přání

Cena

Celkový počet kusů **3**
Celková cena s DPH **180 Kč**

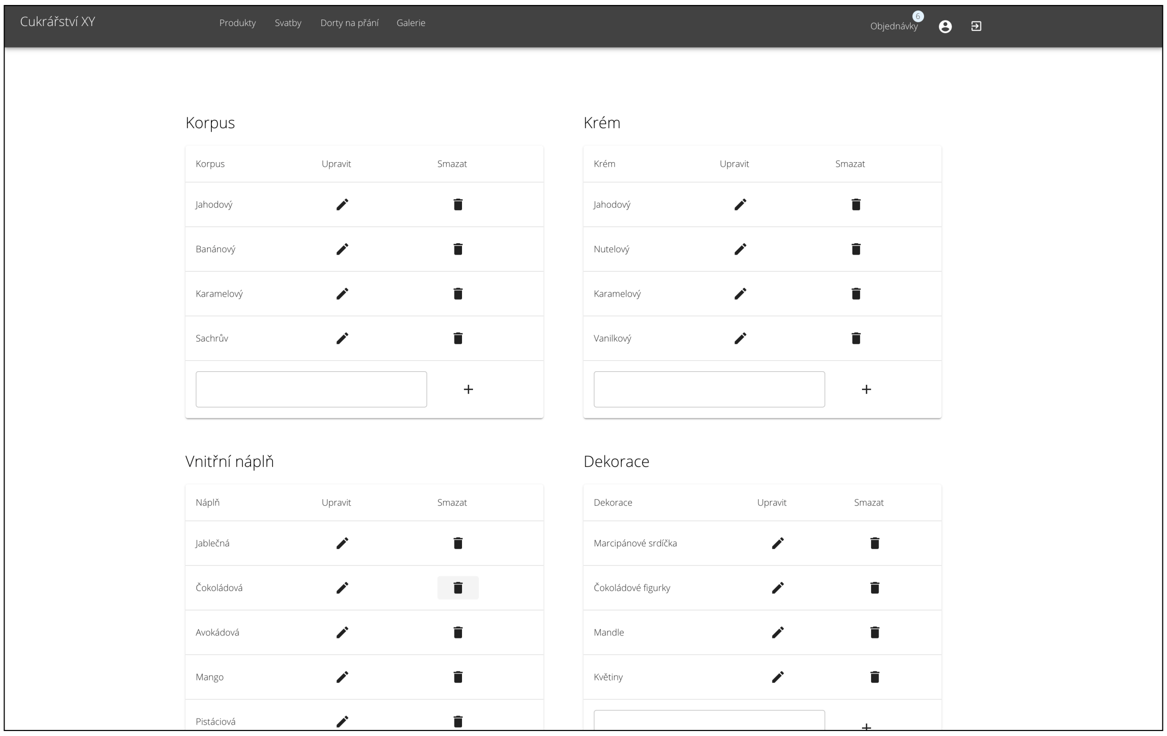
Datum vyzvednutí

Datum: 30. 10. 2020

Kontakt

Jméno: Dana	Příjmení: Novotná
Telefon: 788 444 111	Email: dnovotna@gmail.com

Figure C.31: Administrator – order editing



Cukrářství XY — Produkty · Svatby · Dorty na přání · Galerie — Objednávky

Korpus

Korpus	Upravit	Smazat
Jahodový	✎	🗑
Banánový	✎	🗑
Karamelový	✎	🗑
Sachrův	✎	🗑
	+	

Krém

Krém	Upravit	Smazat
Jahodový	✎	🗑
Nutelový	✎	🗑
Karamelový	✎	🗑
Vanilkový	✎	🗑
	+	

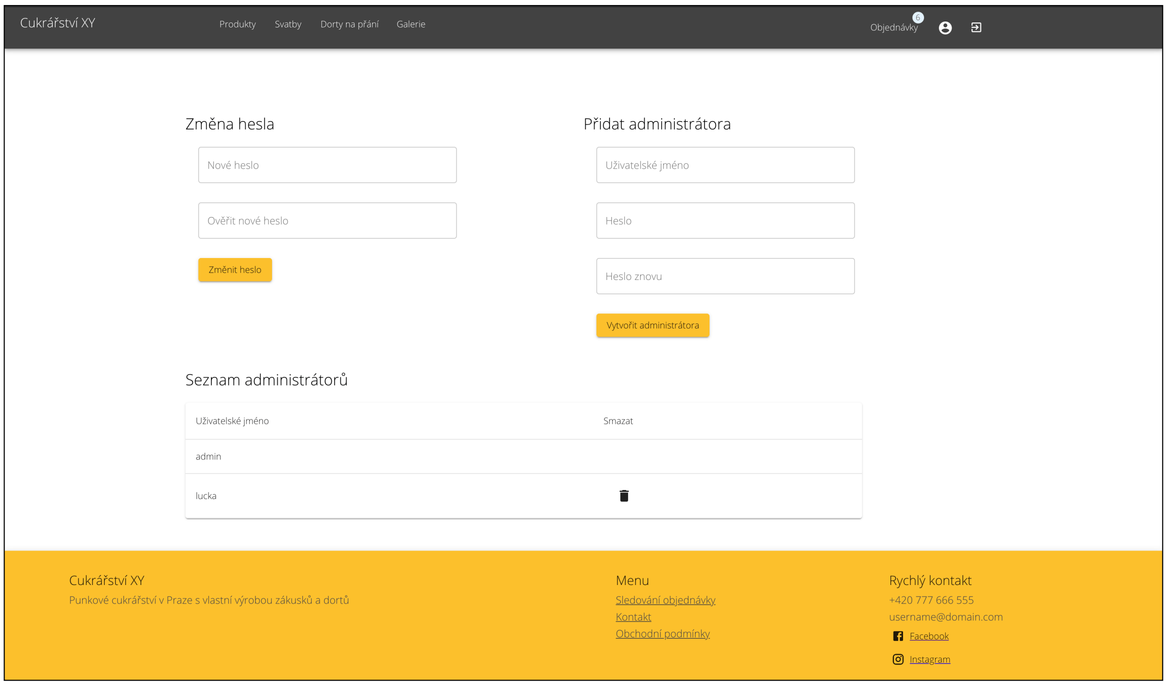
Vnitřní náplň

Náplň	Upravit	Smazat
Jablečná	✎	🗑
Čokoládová	✎	🗑
Avokádová	✎	🗑
Mango	✎	🗑
Pistáciová	✎	🗑

Dekorace

Dekorace	Upravit	Smazat
Marcipánové srdíčka	✎	🗑
Čokoládové figurky	✎	🗑
Mandle	✎	🗑
Květiny	✎	🗑
	+	

Figure C.32: Administrator – setting custom product parameters



Cukrářství XY — Produkty · Svatby · Dorty na přání · Galerie — Objednávky

Změna hesla

Nové heslo

Ověřit nové heslo

Změnit heslo

Přidat administrátora

Uživatelské jméno

Heslo

Heslo znovu

Vytvořit administrátora

Seznam administrátorů

Uživatelské jméno	Smazat
admin	
lucka	🗑

Cukrářství XY
Punkové cukrářství v Praze s vlastní výrobou zákusků a dortů

Menu
Sledování objednávky
Kontakt
Obchodní podmínky

Rychlý kontakt
+420 777 666 555
username@domain.com
Facebook
Instagram

Figure C.33: Administrator – account administration

Appendix D

User manual

The Information System consists of tree runnable parts.

D.1 Database

First the PostgreSQL server needs to be installed and run with the following configuration:

```
port: 5432
database name: sweet-shop
username: postgres
password: app
```

Then `pg-dump.backup` has to be restored in the PostgreSQL server. This backup file is in the attachment of this thesis.

D.2 Server

In the attachment `/exe` of this thesis there is the Server compiled executable file `server.jar`. To run the Server the following command has to be executed:

```
nohup java -jar -Dspring.profiles.active=dev server.jar &
```

The Server is exposed on port 8080.

D.3 Client

In the attachment `/exe` of this thesis there is the Client direcotry with executable files. To run the Client the following command has to be executed:

```
serve -s -n -l 3000 client &
```

The Client is exposed on port 3000.

Appendix E

Contents of enclosed CD

```
readme.txt ....................... the file with CD contents description
├─ user-manual.pdf .................. the user manual for the installation
├─ pg-dump.backup .............. the script for the Database initialization
├─ src ..................................... the directory of source codes
│  └─ thesis ............ the directory of LaTeX source codes of the Thesis
├─ exe .................. the directory with executable parts of the System
│  ├─ server.jar ............................ the executable Server part
│  └─ client ..................... the directory with compiled Client part
└─ text ........................................ the thesis text directory
   └─ farajan2-diploma-thesis.pdf ..... the Thesis text in PDF format
```