# Learning objectives

Courses AP3AF and AK3AF

**Audience**

The audience are the students who have studied basic programming and OOP courses, but not all of them have successfully passed these subjects. They have mostly no experience with software projects.

**Current state of teaching**

Each week:

* The teacher gives the lecture with materials in PowerPoint or in Markdown.
* The teacher leads the lesson and assign the assignment to students. The assignment corresponds to the lecture topic. The assignment is independent form the preview lesson.

**Goal**

We want the students to apply their knowledge and comprehension of course topics on the semestral project so that students can use this project as a template for their own applications.

**Learning outcomes**

Knowledge

* The student lists client-side frameworks.
* The student lists backend frameworks.

Comprehension

* The student explains the Model-View-ViewModel design pattern for creating a user interface.
* The student explains the concepts of object serialization and deserialization.
* The student explains how to access a relational database in code.
* The student explains the concept of object-relational mapping (ORM).
* The student explains how to call a web API on a client application.
* The student justifies the benefits of separating the code for creating the user interface from the actual application logic.

Application

* The student applies the Model-View-ViewModel design pattern to the user interface design.
* The student defines the user interface independently of the application logic.
* The student serializes and deserializes objects.
* The student accesses relational database in code.
* The student uses an object-relational mapping (ORM) library.
* The student calls a web API on a client application.

# Functional requirements

Pizzeria **Self-service kiosk**.

Glossary

* **A kiosk** is a hardware device with touch display for a new kiosk session.
* **A kiosk session** is a personalized ordering experience**.**
* **An order fulfillment option** is a way in which the order is delivered and served.
* **A pizza** is a dish that is in the pizzeria menu.
* **A pizza's selection** is a set of options available for a specific kind of pizza**.**
* **A pizza’s option** is a choice for a pizza including toppings, crust types, sauces, and other customizable features.
* **A pizza’s configuration** is a set of options for a specific kind of pizza.
* **A pizza menu** is a non-empty set of various types of pizza and their configurations**.**
* **A shopping cart** is a multiset of pizzas and their configurations**.**
* **An order** is a non-empty multiset containing duplicate of pizzas and their configurations from the shopping cart, and current state of the order**.**

Roles

A customer wants to order and configure pizzas and eat it in the restaurant or take it away. An operator wants to periodically checks for orders and process orders. A manager wants to prepare pizza description and its configuration option for a customer.

Entities example states:

* A kiosk session: state: state: ready, inuse, , Dine-in, takeout
* The pizza: Name: PEPPERONI PIZZA, Description: tomato base shredded mozzarella

salami Pepperoni

Scenarios

The customer

* The customer wants to have the **kiosk** ready for them.
* The customer wants to navigate through the menu easily to select their desired options.
* The customer wants to specify the order fulfillment options and begin the new personalized **kiosk** **session**.
* The customer wants to have the same menu during the **kiosk session**.
* The customer wants to **select** the desired **pizza** from the **pizza menu**.
  + **Then** the costumer wants to **add** pizzas and their configurations to the **cart**.
  + **Or** the costumer wants to **delete** pizzas and their configuration from the cart.
* The customer wants to **delete** non-empty multiset of **pizza** and its **pizza’s configurations** in the **cart**.
* The customer wants to **order** non-empty multiset of **pizza** and its **pizza’s configurations** in the **cart**.
* The customer wants to see the **total order cost** and receive a **summary** before placing the order.

The operator

* The operator wants to see the list of non-empty multiset of **orders** ordered by **date ascending**.
* The operator wants to complete the **order**.
* The operator wants to cancel the **order**.

The manager

* The manager wants to create the pizza menu.
  + **Then** the manager wants toedit pizza menu.

# Non-functional requirements

* Tests: Only one example of unit test to make project achievable in time.
* Payment: No payment implementation, just order confirmation to make project easier.
* Language: one language only, English or Czech.
* Accessibility: application should support or in near future supports individuals with disabilities, adhering to accessibility guidelines.
* Security:
* Client-side Multiplatform applications (Windows, Linux, Mac).
* Desktop application with touch support for a customer
* Desktop application with touch support for an operator.
* Desktop application for a manager.
* Web API for the backend pre-implemented by a teacher.
* Entity Framework Core for the database (SQLite)
* Local database so that students do not need to setup server.
* Latest standard .NET core
* Easy and readable code, essential code only.
* Time simulation for development

Tasks

|  |  |
| --- | --- |
|  | Description |
|  | Rewrite to markdown, remove colors |
|  | Describe Order, rewrite Operator and Manager |
|  | Create: Utb.PizzaKiosk (solution) -> Utb.PizzaKiosk.Models (class library) -> db models, code first. Fluent API for everything. |
|  | Create GitHub repository. |
|  | Create entities examples and configuration options. |
|  | Create models. |
|  | Create Business Process Modeling and Notation (BPMN) diagram to describe processes. |
|  | Create wireframes. |
|  | Create SWOT analysis. |

Notes:

* My deadline is 9. 9. 2023 23:59.

Revisions

|  |  |
| --- | --- |
| Date | Description |
| 27. 7. 2023 19:00 | Entities example states, Tasks in table |
| 27. 7. 2023 15:00 | Order, Pizza menu, the operator, the manager, simplification |
| 27. 7. 2023 12:16 | Entities, Scenarios |
| 26. 7. 2023 9:00 | Minor changes |
| 24. 7. 2023 | Web Api Learning objectives, time simulation for development, swot, model task |
| 21. 7. 2023 8:00 | Web api pre-implemented by teacher |
| 21. 7. 2023 7:30 | Manager pizza edit, knowledge to comprehension, list frameworks, minor text fix, platform specification, language and framework specification |
| 20. 7. 2023 | Initial version |