Individual track project plan

Viktor Skachkov

19-02-2022

Tutors: Tim Kurvers and Paixão Márcio

Table of content

Съдържание

Versioning table

Version	Changes
1.1	I created the document

1. Introduction

Oliver wants to open a new pizzeria in Eindhoven and has tasked me to make an application which would allow clients to make orders and reserve tables and would also allow the employees to add, remove and edit different meals.

2. Project plan

2.1 Client: Our client is Oliver

2.2 Who is working on the application:

- Viktor Aleksandrov Skachkov – viktor.skachkov01@gmail.com or v.skachkov@student.fontys.nl

2.3 Current situation: I am a student in Fontys University and I am tasked to create an application for a new pizzeria in town which will manage the activities. I should write the application in Java and JavaScript using IntelliJ IDEA. I have the entire semester to work on it which is more than enough time.

2.4 Problem description: Oliver wants to open his first pizzeria in Eindhoven. It provides different types of meals and will allow customers to order food and reserve tables and employee to add, remove and edit meals.

2.5 Project goal: Creating a full application with CRUD operations for the meals. The project should also allow employees to see the sales for a specified period of time and allow customers to order food and to reserve table for a specific time.

2.6 Deliverables: I should deliver an application with a feature to add meals, a feature to delete meals, a feature to edit meals, a feature to see the sales for a specified period of time, a feature which allows ordering food and a feature which allows reserving a table.

2.7 Non-deliverables: We won’t be delivering a manual, training on the program and also we wont be delivering any type of hardware.

2.8 Constraints**:** I have to keep in mind the time because there is a strict deadline to deliver the project. I have to complete everything before the end of the third semester. Another constraint is that I must use Java and JavaScript for the application.

2.9 Risks**:** One potential risk is that if I get sick, it will slow down my progress. Another risk is the timeframe and whether I can finish everything on time. We should only focus on what is needed so that we can finish our priorities on time.

2.10 Phasing:

Number of sprint	Actions
Sprint 1	Preparing the documentation for the project and presenting it to the tutors.
Sprint 2	I will make high level architecture which explains how SOLID is guaranteed, architecture diagrams, important design decisions and applied research. I will also start working on the application itself and will make an applied research document.
Sprint 3	I will finish the back-end part of the application, include testing, and demonstrate connectivity with the backend database.

Sprint 4	I will make changes to the UX, include dependency injection and improve the documentation.
Sprint 5	I will implement websockets in the project, make charts, update the documentation, make fully operational CI/D pipelining and a security report on how the application deals with the OWASP top 10 security risks.
Sprint 6	Finishing documentation, honing the application and submitting everything.

3. Plan of coding

There will be a User class and two classes (Worker and Client) which would inherit from it. There will also be classes Category, Meal, Adding, OrderedMeal and Order. Every Meal will contain one category. Every OrderedMeal will contain one Meal and 0 or more Addings. One order will contain a list of all the meals a particular customer has ordered.

There will be class Table to mark each table in the restaurant and class ResevationRequest to mark each reservation request made by the customers. The employees can approve or reject them and assign tables to them.

There will be DataHelper classes which will extract information from the database, Administration classes which will contain methods that can affect the database itself and the Controller classes will contain methods which allows the users an access to certain data.

There will also be mock classes which will mirror the normal classes in almost every way, but they will be connected to another database. The normal Administration classes and the mock Administration classes will be connected to the same interface, containing their methods.