**Project plan**

Food delivery web application

|  |
| --- |
| **Date : 13.01.2023** |
| **Version : 2.0** |
| **State : Complete** |
| **Author : Dragos Andrei Munteanu** |

#### Version history

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Author(s)** | **Changes** | **State** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Table of contents**

1. Project assignment …………………………………………………………………………………3

1.1 Context .……………………………………………………………………………………………3

1.2 Goal of the project ………………………………………………………………………….3

1.3 Strategy ……………………………………………………………………………………………3

2. Test plan …………………………………………………………………………………………….…10

2.1 Test strategies ……………………………………………………………………………….10

1. **Project assignment**

**1.1 Context**

The context of this project for the 3rd semester is to create a full stack, user friendly web application that is comprised out of a javascript framework for the front-end and a RESTful API for the back-end, but that also needs to have authentication, authorization, and security.

**1.2 Goal of the project**

In order to meet the requirements of this project, the idea that I came up with is to create a food delivery website that will have users that can order food, restaurants that can join the roster of the web application and start selling their products, managers (admins), that will have to manually approve restaurants that want to join our platform in order to prevent the website from malicious intent, and can also remove users or restaurants for security reasons, and possibly drivers that can apply to join the team from the website and start earning money by delivering orders.

* 1. **Strategy**

The approach for this project will be done in an agile manner, more specifically scrum, in order to get acquainted to this workflow.

Because scrum is being used, an initial product backlog with user stories, acceptance criteria, prioritization, and estimation is provided.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | User stories | Acceptance criteria | Prioritization | Estimation | Actual |
| 1. | As a user I want to be able to create an account and log in in order to save my address and contact details. | **Scenario** Creating an account  **Given**  The forms have valid data  **When**  The user presses create account  **Then**  Create account and send confirmation email to user | Must | Week 10 | **Done** |
| 2. | As a user I want to be able to customize the food I order in order to create a meal to my liking. | **Scenario** Customize food when ordering  **Given**  User selects food that can be customized  **When**  User orders food  **Then**  Order details will be updated | Could | Week 11 | In progress |
| 3. | As a user I want to be able to see the order information of the current order to make sure that the details are correct. | **Scenario**  Check order information for current order  **Given**  An order was placed and it is valid  **When**  A user has ordered food  **Then**  A screen with the current order information will be shown | Should | Week 12 | **Done** |
| 4. | As a user I want to be able to check my order history to quickly order the same thing I've ordered the last time. | **Scenario**  Check order history  **Given**  Orders were made on the account  **When**  The user clicks the “Order history“ button  **Then**  A list of previous orders is shown | Should | Week 13 | **Done** |
| 5. | As a user I want to easily navigate on the website in order to choose and get my food quick. | **Scenario** Navigate quickly on the website  **Given**  User has access to internet  **When**  User uses the website  **Then**  User can choose and order fast and with ease | Should | Week 16 | **Done** |
| 6. | As a user I want to see the estimated delivery time of the order to know approximately when it will arrive. | **Scenario**  Display delivery time  **Given**  User has ordered food  **When**  Order has been processed  **Then**  A screen with the estimated delivery time will be shown | Should | Week 15 | Not started |
| 7. | As a manager I want to be able to approve or reject restaurants that want to join our roster as a security check. | **Scenario**  Accept or decline restaurant request  **Given**  The restaurant meets the requirements or not  **When**  A restaurant requests to join the roster  **Then**  The restaurant will join the roster and ordering from it will be possible or it will be declined | Should | Week 12 | Not started |
| 8. | As a manager I want to be able to remove users and restaurants in case they are exploiting the web application to keep the website secure. | **Scenario**  Remove users/restaurants  **Given**  Users/restaurants try to exploit the application  **When**  The managers notice malicious intent from the users/restaurants  **Then**  The managers can remove (ban) the accounts | Should | Week 13 | **Done** |
| 9. | As a restaurant owner I want to be able to join the roster to increase my sales. | **Scenario**  Restaurant wants to join roster  **Given**  The restaurant meets the requirements  **When**  A restaurant requests to join the web application  **Then**  The owner of the restaurant will be notified if it has been approved or not by the managers | Must | Week 11 | **Done** |
| 10. | As a restaurant owner I want to have an account with the rights to edit items and prices in order to keep up with inflation. | **Scenario**  Restaurant owner wants authorization rights  **Given**  The restaurant meets the requirements  **When**  The restaurant has joined the roster  **Then**  The owner will receive a special account with authorization to make changes to prices and items | Must | Week 12 | **Done** |
| 11. | As a restaurant owner I want to be able to see all the orders that have been handed out and the users information | **Scenario**  Restaurant owner wants to see all the orders  **Given**  Users have ordered from the restaurant  **When**  The owner logs in with the restaurant account and clicks on “Orders”  **Then**  A list of all the orders handed out from the restaurant will be shown | Should | Week 17 | **Done** |
| 12. | As a driver I want to be able to join the delivery post in order to earn money. | **Scenario**  A person wants to become driver  **Given**  There are enough free posts for drivers and the person meets the requirements  **When**  The person applies for the job  **Then**  The person will receive a special account and can start working | Could | Week 15 | Not started |
| 13. | As a driver I want to be able to see how much money I made to know if I need to change my routine. | **Scenario** Checking income as a driver  **Given**  The driver has a driver account and has previous orders completed  **When**  The driver clicks the “Income” screen  **Then**  The driver will see income based on the deliveries he has made previously | Could | Week 16 | Not started |
| 14. | As a driver I want to have a live map with the ability to see trip details from the restaurant to the users’ address to not have to use an external application. | **Scenario**  Check live map as driver  **Given**  Driver picks up an order  **When**  Order has been picked up  **Then**  A live map with the trip from the restaurant to the users’ address is shown | Wont |  | Not started |
| 15. | As a driver I want to receive money for a completed delivery based on how long it took to complete the trip to be sure that payment is fair. | **Scenario**  Earn money based on trip time  **Given**  Driver has finished the trip  **When**  Driver has checked in the website  **Then**  Payment should be calculated based on trip time | Wont |  | Not started |

**2. Test plan**

**2.1 Test strategies**

The testing strategies for this application will be done through unit testing and end-to-end testing.

In unit testing, individual components of the application will be tested, and the purpose is to validate each unit of the software until code performs as expected.

End-to-end testing will be used as an integration test, being able to test functionality between front end and back end at the same time.

The service classes have a good amount of coverage on tests (>70%), the controller classes are almost fully tested, but the repository classes do not have too many tests, because they only test interactions with the database, which is also tested in the service classes.

There is only one End-to-end test, which covers the case of a user logging in and seeing their name on the home page.

Link to GitLab repository: <https://git.fhict.nl/I516586/dragos-food-delivery>