

dISH DIVE

PROJECT PLAN



HELLO FRESH

|  |
| --- |
| **Date : 19-09-23** |
| **Version : 0.4** |
| **State : In-Review** |
| **Author : Slobodan Starcevic** |

#### Version history

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Author(s)** | **Changes** | **State** |
| 0.1 | 12-09-23 | Slobodan | Creation | Draft |
| 0.2 | 14-09-23 | Slobodan | Changed project plan to follow the idea of a social media app linked with a music app | In-Review |
| 0.3 | 17-09-23 | Slobodan | Changed topic direction to a recipe website linked to Food corp. as the client | In-Review |
| 0.4 | 19-09-23 | Slobodan | Changed client to Hello Fresh with more context to the goal the client is trying to reach | In-Review |

**Distribution**

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Sprint** |
| 0.4 | 22-09-23 | Sprint 1 |
|  |  |  |

Contents

[1. Project assignment 4](#_Toc42673512)

[1.1 Context 4](#_Toc42673513)

[1.2 Goal of the project 4](#_Toc42673514)

[1.3 Scope and preconditions 4](#_Toc42673515)

[1.4 Strategy 4](#_Toc42673516)

[1.5 Research questions 4](#_Toc42673517)

[1.6 End products 4](#_Toc42673518)

[2. Project Organisation 6](#_Toc42673519)

[2.1 Stakeholders and team members 6](#_Toc42673520)

[2.2 Communication 6](#_Toc42673521)

[3. Activities and time plan 7](#_Toc42673522)

[3.1 Phases of the project 7](#_Toc42673523)

[3.2 Time plan and milestones 7](#_Toc42673524)

[4. Testing strategy and configuration management 8](#_Toc42673525)

[4.1 Testing strategy 8](#_Toc42673526)

[4.2 Test environment and required resources 8](#_Toc42673527)

[4.3 Configuration management 8](#_Toc42673528)

[5. Finances and Risk 9](#_Toc42673529)

[5.1 Project budget 9](#_Toc42673530)

[5.2 Risk and mitigation 9](#_Toc42673531)

# Project assignment

## Context

The company Hello Fresh is an established food delivery service that wants to branch out and have people interact more with their brand. The idea is to let people post their recipes and be able to then order the said ingredients via their already existing systems.

## Goal of the project

The major goal is to make a platform where people can share their recipes and have other users interact with these recipes. Additionally having forum features would be welcome so people can discuss more and also have more person-to-person interaction that don’t necessarily need to be based around an existing recipe on the website, like forum-based conversation or chat functions.

To let people, find many different recipes the plan is to have some interesting searching features like for example searching for a specific ingredient and use that as a filter to search for all recipes for inspiration, or maybe search for dinner, breakfast or lunch.

The value this add to Hello Fresh is that they can let people expand their choices of meals themselves, while still being the one selling the ingredients. The interaction between people will give them an interest to try each other’s’ recipes, and what could be more convenient than being able to press a button “order ingredients” so there is no need to search multiple stores to find the right ingredients.

Hello Fresh has a set of recipes they deliver to people, the only problem is that is isnt infinite so it is inevitable to get a recipe multiple times. This can cause frustration with the subscriber since they want more variation. Dish Dive solves this repitition by giving the ability for users to supplement the amount of recipes available. This still keeps the door open for the services of Hello Fresh to be used while users offer their recipes.

## Scope and preconditions

|  |  |
| --- | --- |
| **In scope** | **Outside of scope/optional** |
| 1. Recipe Posting | 1. Cloud storage |
| 1. Rate Recipes | 1. Mobile compatibility |
| 1. Comment on recipes | 1. Verifying recipes by admins |
| 1. Make community posts | 1. Verifying users as trusted cooks |
| 1. Make community post comments | 1. User profiles |
| 1. User system | 1. Chat function between users |
| 1. Order ingredients system |  |
| 1. Search system for recipe searching with filters |  |

## Strategy

The strategy is Agile SCRUM. I will work in 3 week sprints. At the end of the sprints I will have a feedback moment with my supervisor and steer my project from there for the next sprint. It is important to make a retrospective and reflect on the previous sprint to make better decisions for the following sprint.

I will be using JIRA to keep track of the project. Here I will be archiving things like the product backlog, sprints planning and sprint backlog. Using JIRA I can keep track of the backlog, tasks and bugs to keep on top of the project progress. This tool also makes it possible to keep an overview of user stories and keep track of those.

As this is a one-person project it is still import to have an equivalent of standups to keep track of progress.

## Research questions and methodology

What type of recipe website is something that could enrich Hello Fresh?

What do people want from a recipe website?  
-Field – Observation

-To answer this question I will do thorough research on existing recipe website to find out what the most popular ones use to draw their community. This should result in a list of features.

What makes DishDive different from competitors?

-Library – Competitive analysis  
-To answer this question, as I will get the feature list from the previous question, I will be able to find holes that our website would be able to fill. Possibly from the unique opportunity of being able to offer ingredient delivery for chosen recipes.

How do you make a website streamlined for users when you offer many features?

-Library – Best good and bad practices

-To answer this question I will dive into the many websites available on the web and analyse websites that have a similar amount of features and understand what makes their website streamlined or not.

How do you smoothly implement 2 things and make them interact with eachother, like recipes and community posts?  
-Library – Available product analysis  
-To find out how to make the recipes interact with the community posts I will do analysis on already existing food forums and find out how people like to share recipes.

## End products



# Project organisation

## Stakeholders and team members

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Abbreviation** | **Role and functions** | **Availability** |
| *Slobodan Starcevic*  *Teachers* | *-*  *-* | *Sole developer*  *Supervisors* | *24/7*  *During their respective lessons* |

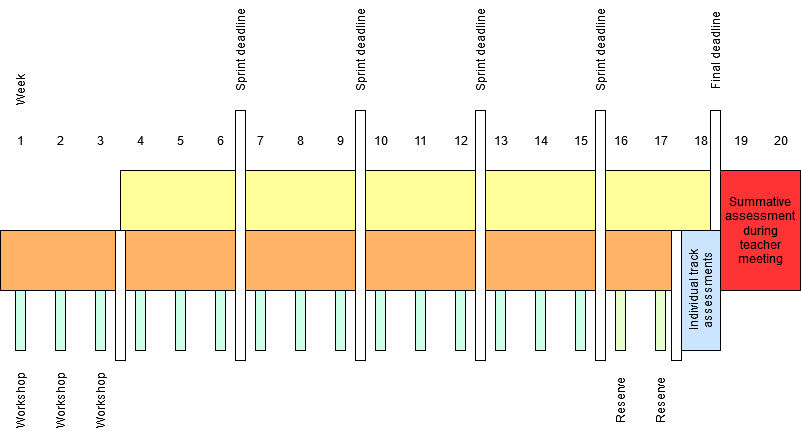
## Communication

The big meetings will be done triweekly at the end of every sprint with the supervisors.

Besides these meetings I will continuously ask feedback to the appropriate teacher.

# Activities and time plan

## Phases of the project



## Time plan and milestones

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Week | Topic | Who initiates | Link to topic | Teacher |
| 0 | Kickoff | Course owner | [Kick-off recordingLinks to an external site.](https://stichtingfontys-my.sharepoint.com/:v:/g/personal/886548_fontys_nl/Eaez_WbxVgRLgesfGVoYC7kBWQS9nkPqg1OtqeVegvVWcg?e=d8dtLu)  [SlidesLinks to an external site.](https://stichtingfontys-my.sharepoint.com/:p:/g/personal/886548_fontys_nl/EUzi4fQfgYNBrJk5KYQDGxUB-DUbZRYdEvQjY5EctlVogQ?e=rQ9OLH) |  |
| 1 | Technical Day 1: Java programming language intro | Technical teacher | [Java Introduction](https://fhict.instructure.com/courses/13328/pages/java-introduction) | Tech. Teacher 1 |
|  | Technical Day 2: Lombok and Gradle | Technical teacher | [Lombok and Gradle](https://fhict.instructure.com/courses/13328/pages/lombok-and-gradle) | Tech. Teacher 2 |
|  | Agile and scrum | Semester coach | [Agile and Scrum](https://fhict.instructure.com/courses/13328/pages/agile-and-scrum) |  |
| 2 | Technical Day 1: Spring and Dependency injection | Technical teacher | [Dependency Injection](https://fhict.instructure.com/courses/13328/pages/dependency-injection) | Tech. Teacher 1 |
|  | Technical Day 2:  RESTful services | Technical teacher | [RESTful Services](https://fhict.instructure.com/courses/13328/pages/restful-services) | Tech. Teacher 2 |
|  | Agile Requirements | Semester coach | [What's Required?](https://fhict.instructure.com/courses/13328/pages/whats-required) |  |
| 3 | Technical Day 1: Continuous integration | Technical teacher | [Continuous integration](https://fhict.instructure.com/courses/13328/pages/continuous-integration) | Tech. Teacher 1 |
|  | Applied Research | Semester Coach | [Research methods](https://fhict.instructure.com/courses/13328/pages/research-methods) |  |
| 4 | Technical Day 1: Javascript introduction | Technical teacher | [Javascript Introduction](https://fhict.instructure.com/courses/13328/pages/javascript-introduction-2) | Tech. Teacher 1 |
|  | Technical Day 2: Javascript asynchronous programming | Technical teacher | [Javascript asynchronous programming](https://fhict.instructure.com/courses/13328/pages/javascript-asynchronous-programming) | Tech. Teacher 2 |
| 5 | Technical Day 1: React introduction | Technical teacher | [React](https://fhict.instructure.com/courses/13328/pages/react) | Tech. Teacher 1 |
|  | Technical Day 2: Architecture using C4 | Technical teacher | [Architecture using C4](https://fhict.instructure.com/courses/13328/pages/architecture-using-c4) | Tech. Teacher 2 |
|  | Organizational structures | Semester coach | [Organizational structures](https://fhict.instructure.com/courses/13328/pages/organizational-structures) |  |
|  | Specialization | Semester coach | [Specialization preparation](https://fhict.instructure.com/courses/13328/pages/specialization-preparation) | Placeholder for news on specialization. No classes here. |
|  | Internship | Semester coach | [Internship preparation](https://fhict.instructure.com/courses/13328/pages/internship-preparation) | Placeholder for news on internship. No classes here. |
| 6 | Technical Day 1: React + Spring Boot API Demo | Technical teacher | [React + Spring Boot API Demo](https://fhict.instructure.com/courses/13328/pages/react-+-spring-boot-api-demo) | Tech. Teacher 1 |
|  | Technical Day 2: Sequence diagrams | Technical teacher | [Sequence diagrams](https://fhict.instructure.com/courses/13328/pages/sequence-diagrams) | Tech. Teacher 2 |
| 7 | Technical Day 1: JPA introduction | Technical teacher | [Data Persistence](https://fhict.instructure.com/courses/13328/pages/data-persistence) | Tech. Teacher 1 |
|  | Technical Day 2: Software quality metrics | Technical teacher | [Software Quality Metrics](https://fhict.instructure.com/courses/13328/pages/software-quality-metrics) | Tech. Teacher 2 |
| 8 | Database queries with JPA | Technical teacher | [Database queries with JPA](https://fhict.instructure.com/courses/13328/pages/database-queries-with-jpa) | Tech. Teacher 1 |
|  | Cultural awareness | Semester coach | [Cultural Awareness](https://fhict.instructure.com/courses/13328/assignments/227623) |  |
| 9 | Taking it to the test | Technical teacher | [Taking it to the test.](https://fhict.instructure.com/courses/13328/pages/taking-it-to-the-test) | Tech. Teacher 2 |
| 10 | Back end: Authentication & authorization | Technical teacher | [Authentication & Authorization](https://fhict.instructure.com/courses/13328/pages/authentication-and-authorization) | Tech. Teacher 1 |
|  | Demo day #1 | Technical teacher | [Demo day](https://fhict.instructure.com/courses/13328/pages/demo-day) | Tech. Teacher 2 |
| 11 | Front end: Authentication & authorization | Technical teacher | [Frontend authentication and authorization](https://fhict.instructure.com/courses/13328/pages/frontend-authentication-and-authorization) | Tech. Teacher 1 |
|  | Front end: UI/UX | Technical teacher | [The UX development cycle](https://fhict.instructure.com/courses/13328/pages/the-ux-development-cycle) | Tech. Teacher 2 |
| 12 | Websockets | Technical teacher | [WebSockets](https://fhict.instructure.com/courses/13328/pages/websockets) | Tech. Teacher 2 |
| 13 | Continuous delivery/deployment | Technical teacher | [Continuous delivery and deployment](https://fhict.instructure.com/courses/13328/pages/continuous-delivery-and-deployment) | Tech. Teacher 1 |
|  | Demo day #2 | Technical teacher | [Demo day](https://fhict.instructure.com/courses/13328/pages/demo-day) | Tech. Teacher 2 |
| 14 | Back end: Asynchronous services | Technical teacher | [Asynchronous Services](https://fhict.instructure.com/courses/13328/pages/asynchronous-services) | Tech. Teacher 2 |
|  | Web security | Technical teacher | [Security of web apps](https://fhict.instructure.com/courses/13328/pages/security-of-web-apps) | Tech. Teacher 1 |
| 15 | End-to-end Testing | Technical teacher | [End-to-end testing](https://fhict.instructure.com/courses/13328/pages/end-to-end-testing) | Tech. Teacher 2 |
| 16 | Web performance | Technical teacher | [Performance of web apps](https://fhict.instructure.com/courses/13328/pages/performance-of-web-apps) | Tech. Teacher 1 |
|  | Demo day #3 | Technical teacher | [Demo day](https://fhict.instructure.com/courses/13328/pages/demo-day) | Tech. Teacher 2 |
| 17 | Applied research presentation | Semester Coach | [Applied research presentation](https://fhict.instructure.com/courses/13328/pages/applied-research-presentation) |  |
| 18 | Individual track assessment |  |  |  |
|  | Final sprint demo group project |  |  |  |

# Testing strategy and configuration management

## 

## Testing strategy

The testing strategy will be unit testing. This was I will ensure that all components work individually. This is important as my application will have many systems working together to make the whole, and in this case it is very detrimental for each component to work properly and not break other systems.

Besides that Unit testing is quite an agile way of testing as it is based on being able to continuously test newly added features over and over again and ensure it works with the rest of the system.

Making quality code and early on ensuring that the code is bug free is important while developing, which unit testing makes possible by for example test driven development.

## Test environment and required resources

I will make use of a CI/CD pipeline to test and deploy releases. For this I will make use of a gitlab runner ran by docker docker.

## Configuration management

For version control git will be used with gitlab. To ensure that committed data that is work in progress does not interfere with the working code base I will make use of a dev branch, or feature branches to ensure it only gets merged with main when Im sure it works correctly with the rest of the code.

# Finances and risk

## Risk and mitigation

These are the risks we have found as a group.

1. **Absent or sick**, work on the project will be set back and planning will be changed. Meaning the I need to replan the sprint and ensure I can still deliver at the end of the sprint.
2. **Tasks not finished on time can be harmful for the project's completion.** Tasks can stack up and delay the completion of following tasks further.
3. **Data loss**, documents and files can be lost due to team members’ computers breaking etc., setting the group back on progress
4. **Delayed attendance**, it can be harmful for the completion of tasks on time, as valuable information from lessons might be lost.
5. **Equipment failure**, when a group members’ equipment for example a laptop breaks, development / progress will be delayed due to the group member not being able to work on their tasks
6. **What if I cannot work at school?** It could be that the school locks down or the facilities are not accessible, what do we do in that case?
7. **What if someone does not or cannot attend a meeting?** Life can be in the way, or for that matter laziness, so how do we handle someone not turning up to a meeting where decisions need to be made?
8. **Not understanding a task and getting to a stalemate**, this is always possible as we are still learning, but how do we deal with a member not being able to progress?

**(On the next page you can find the risk assessment table)**

|  |  |  |  |
| --- | --- | --- | --- |
| Risk | probability | impact | Mitigation |
| 1 | medium | medium | The project team will keep each other informed about their presence and work remotely if needed. Each member is responsible for reporting absence or sickness so changes can be made to the task. |
| 2 | low | medium | The team will regularly check in and see how each task has been progressed and mitigate resources if necessary to complete tasks on time. This will be done through meetings. |
| 3 | high | high | Documents and files will be hosted in a shared environment and not saved locally |
| 4 | medium | low | Project members keep each other informed about delays so the members who are present can decide how to efficiently divide the workload for the time being. |
| 5 | low | high | Since all files are uploaded in shared folders, the member can rent the equipment necessary from the school. |
| 6 | low | high | If we cannot work together physically, we will start working remotely from discord or Teams. From there we can share our screen and be on calls to finish work. |
| 7 | medium | medium | If someone knows they will not be able to attend a meeting, they are required to communicate this. If someone does not turn up without a reason, the decisions that must be made will be made by the other members. Furthermore, this will be reported to the teacher supervising the meeting and written down for future reference. |
| 8 | medium | high | If a team member does not know how to go on with their task, first they will communicate this with the group and see if someone does know the solution. Even if the problem is not solved, we will take it to one of the teachers. |