# PepeFit

# Project Plan

# Introduction

This plan covers the content and enablement portions of the PepeFit Closed-Alpha project.

# Project organization

|  |  |  |  |
| --- | --- | --- | --- |
| **Team Member** | **Role A** | **Role B** | **E-mail** |
| Ege Uçak | Software Project Manager | Software Developer | b21527467@cs.hacettepe.edu.tr |
| Bahadir Adak | Software Analyst | Software Developer | b21526619@cs.hacettepe.edu.tr |
| Eyüpcan Bodur | Software Architect | Software Developer | b21526753@cs.hacettepe.edu.tr |
| Serhat Sağlık | Software Configuration Manager | Software Developer | b21527285@cs.hacettepe.edu.tr |
| Berk Can Özen | Software Tester | Software Developer | b21485591@cs.hacettepe.edu.tr |

# Development process and measurements

The PepeFit team will use OpenUP; adapted to address the fact that we are also doing content development, not just coding. And the development process model we use is WaterFall development model. We will track the progress using Asana and Discord. We will have weekly review meetings. We will control our codes using GitHub.

# Project milestones and objectives

[Define and describe the high-level objectives for the phases and define milestones. For example, use the following table to lay out the schedule ]

This section covers objectives for the entire PepeFit project.

1. Automating a gym
2. Having different kinds of memberships
3. Group training
4. Enabling people to see their progress for losing weight or building muscle mass.
5. Seeing and setting training schedule
6. Achievements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Phase** | **Iteration** | **Primary objectives** (risks and use case scenarios) | **Scheduled start or milestone** | **Target velocity** |
| Inception /  Warm-up | I1 | 1. Visit a gym (Gym owners might not help) 2. Determine requirements (Requirements might not be clear) 3. Determining technologies to be used (There might be not many documentation) 4. Create schedule (Schedule might change) | 06/03/2018  13/03/2018 | 7 days |
| Elaboration | I2 | 1. Train people for using git (They might not understand) 2. Database architecture (Design might be inefficient) 3. Design architecture (Design might be inefficient) 4. UI design (It might be hard to use) | 14/03/2018  18/03/2018 | 4 days |
| Elaboration | I3 | 1. Determine use cases (It might be hard to determine) 2. Create risk management report (We might not see all risks) 3. Create config/change management report 4. Make coding standards (Not every member will follow them) | 19/03/2018  21/03/2018 | 4 days |
| Construction | I4 | 1. Construct database 2. Create login screen 3. Create front-end | 21/03/2018  28/03/2018 | 7 days |
| Construction | I5 | 1. Add more features | 29/03/2018  19/04/2018 | 21 days |
| Verification | I6 | 1. Test software 2. Risk management report 3. Config change report | 20/04/2018  18/05/2018 | 27 days |

# Deployment

* The project plan is improved according to customers / stakeholders requests.
* Ensure to optimize usage of service according to customers and users .
* It must be traceable by a team to ensure that the project is effective and efficient.
* Ensure that all release and deployment packages can be tested and verified, if appropriate.
* Ensure that change is managed during the release and deployment activities.

# Lessons learned

At this point, we learned how to plan a project.