Team 05

muly
Software Development Plan
Version 1.0

Revision History

Date	Version	Description	Author
<12/11/2022>	1.0	First release	**muly**

Table of Contents

Software Development Plan

1. Introduction

The objective of this Software Development Plan is to define the development activities in terms of the phases and iterations required for implementing **muly**.
This *Software Development Plan* describes the overall plan to be used by the project, including the deployment of the product. The plans outlined in this document are based on the product requirements defined in the *Vision Document*.

This *Software Development Plan* contains the following information:

Project Overview — provides a description of the project's purpose, scope, and objectives. It also defines the deliverables that the project is expected to deliver.

Project Organization — describes the organizational structure of the project team.

2. Project Overview

2.1 Project Purpose, Scope, and Objectives

- Objectives and Purpose: There are many services for video streaming but a few of them focus on education and multi-purpose, and a lot of streaming platforms are just for games like Nimo and Twitch. This service will let everyone stream their own content with our reasonable censorship.
- Scope: There are two kinds of users on our platform. One is streamers and the other is users that watch streaming video.

2.2 Assumptions and Constraints

- Budget: Around 25 tasks, each task takes on average 3 hours, and each hour cost 20$ per developer. We double this number and have the value of 2 * 3 * 25 * 20 = 3000$ for five developers.
- The project has a fixed schedule of 10 weeks, equivalent to 5 sprints.
- Project has 5 people with no added during the project.

2.3 Project Deliverables

- Vision Document.
- Software Development Plan.
- Source code with architecture documentation.
- Database schemas.
- User interface design in each state.
- Sprintly progress report.
- Project budget report per sprint.

3. Project Organization

3.1 Organizational Structure

- This project will be evaluated at the end of the semester and we just have five people in our team, therefore our roles have no hierarchy. We mainly focus on developing the software so that it can be deployed and run successfully at the end of the semester. All of the team is developers with different aspects and split into three main types: Frontend, Backend, and DevOps.

Role	Name
Project manager, Backend Developer	Nguyễn Hoàng Khang
Backend Developer, DevOps	Tạ Quang Khôi
Tester, Frontend Developer	Võ Huỳnh
Business Analyst, Backend Developer	Nguyễn Huỳnh Sang
Business Analyst, Frontend Developer, Data Scientist	Bùi Duy Bảo

3.2 Roles and Responsibilities

All team member are required to attend team meetings, daily catch up

Role	Responsibility
Project manager	Design the project plan, assign tasks to members. Monitor process and report status. Plan meeting for the team. Manage and adjust the general project's timeline.
Designer	Design the system. Design user interface. Create Software Architecture Document.
Developer	Write source code for the system. Perform unit testing. Review source code if any bug reports. Split into 3 types: ● **Backend Developer**: Write server-side code. ● **Frontend Developer**: Implement UI code as available design. ● **Devops**: Deployed to AWS web service and implement CI/CD workflow
Tester	Write the test cases and test plans. Perform system test, and report to PM and developer if has any bugs.
Business analysis	Document requirements. Thinking about marketing solutions and then communicating with team members about the requirements. Find suitable customers if any.

4. Management Process

4.1 Project Estimates

4.2 Project Plan

[This section contains the schedule and resources for the project.]

4.2.1 Phase and Iteration Plan

Phase	Iterations	Start	End
Inception	1	Week 2	Week 3
Elaboration	2	Week 4	Week 6
Construction	2	Week 7	Week 10

In each phase we have the breakdown structure:

Phase	Milestones
Inception	Project plan Vision document Weekly reports
Elaboration	Revised project plan Revised vision document Use case document with a revised use case model Design document (Including UI prototypes, architecture document) Weekly reports
Construction	Revised artifacts from Elaboration phase Test plan Test cases Source code/prototype Test report Defects Weekly reports Presentation and Demo

4.2.2 Releases

The project demo is expected to be available on 15/12/2022.

The release **version 1.0** will have almost all the main features and can be used by users: main UI, stream system, and small social media for chatting,... And some other incomplete features will be updated in a later version soon.

4.2.3 Project Schedule

Task	Start	End

Inception	22/10	29/10
Elaboration 1	30/10	6/11
Elaboration 2	7/11	12/11
Construction 1	14/11	26/11
Construction 2	27/11	17/12
Demo		

4.3 Project Monitoring and Control

4.3.1 Reporting

Metrics summary reports will be generated at the end of each sprint.

The reports include

- Team workload. Workload reports visualize who is working on what and when they are working on it, and how much work they have over time.
- Earned value for completed tasks. Earned value provides a clear picture of where your project stands versus where it should have been as planned.

In addition, overdue tasks and upcoming deadlines will be alerted to keep up with the project plan.

4.3.2 Risk Management

Risk ID	***Risk Description***	***Risk exposure***	***Probability***	***Impact***	***Priority***	***Mitigation Strategy or Contingency Plan***
1	Project team members leaving	75 * 75 = 5625	Likely	Serious	High	Documenting the details of your project
2	Budget issues	50 * 90 = 4500	Occasional	Catastrophic	Medium	Be sure to adjust your project plan and budget whenever changes are made to avoid raising project costs.
3	Code Issues	50 * 75 = 3750	Likely	Moderate	Low	Testing code frequently
4	Disclosures issues	25 * 75 = 1875	Seldom	Serious	Medium	Make a nondisclosure agreement contract while hiring new people joining to the team
5	Change requirement issues	50 * 75 = 3750	Occasional	Serious	High	Make sure all design, technique with the client and all member in the team

6	Business risks	50 * 75 = 3750	Occasional	Serious	Medium	Prepare a backup plan, and alternative cútomers if risks occur

4.3.3 Configuration Management

All source code, test scripts, and data files are stored in a GitHub repository. All members have access to the repository. Documentation related to the source code is included in a folder in Google Drive, including the project plan, vision document, use case diagram, class diagram, and software architecture document.