PotPal

Software Development Plan

Version 1.4

Revision History

| **Date** | **Version** | **Description** | **Author** |
| --- | --- | --- | --- |
| 24/10/24 | 0.1 | Added Introduction, Project Purpose, Scope, and Objectives | Phạm Minh Mẫn |
| 24/10/24 | 0.2 | Added Assumption, Constraint and Project Deliverables | Nguyễn Khánh Toàn |
| 25/10/24 | 0.3 | Added Project Organization | Lê Thanh Minh Trí |
| 25/10/24 | 0.4 | Added Project Estimate, Phase and Iteration Plan | Cao Hoàng Lộc |
| 26/10/24 | 0.5 | Added Releases | Phạm Anh Văn |
| 26/10/24 | 0.6 | Added Project Schedule | Lê Thanh Minh Trí |
| 27/10/24 | 0.7 | Added Reporting and Configuration Management | Phạm Anh Văn |
| 27/10/24 | 1.0 | Initial Version | Lê Thanh Minh Trí |
| 08/11/24 | 1.1 | Added Risk Management | Nguyễn Khánh Toàn |
| 10/11/24 | 1.2 | Revise and update Project Deliverables | Cao Hoàng Lộc |
| 10/11/24 | 1.3 | Update Project Schedule | Phạm Anh Văn |
| 10/11/24 | 1.4 | Update Project Configuration and Proof | Phạm Anh Văn |

Table of Contents

[**1. Introduction 4**](#_heading=h.30j0zll)

[**2. Project Overview 4**](#_heading=h.1fob9te)

[2.1 Project Purpose, Scope, and Objectives 4](#_heading=h.3znysh7)

[2.2 Assumptions and Constraints 4](#_heading=h.2et92p0)

[2.3 Project Deliverables 4](#_heading=h.tyjcwt)

[**3. Project Organization 6**](#_heading=h.3dy6vkm)

[3.1 Organizational Structure 6](#_heading=h.1t3h5sf)

[3.2 Roles and Responsibilities 6](#_heading=h.4d34og8)

[**4. Management Process 7**](#_heading=h.2s8eyo1)

[4.1 Project Estimates 7](#_heading=h.17dp8vu)

[4.2 Project Plan 7](#_heading=h.3rdcrjn)

[4.2.1 Phase and Iteration Plan 7](#_heading=h.26in1rg)

[4.2.2 Releases 8](#_heading=h.35nkun2)

[4.2.3 Project Schedule 8](#_heading=h.1ksv4uv)

[4.3 Project Monitoring and Control 9](#_heading=h.44sinio)

[4.3.1 Reporting 9](#_heading=h.2jxsxqh)

[4.3.2 Risk Management 10](#_heading=h.3j2qqm3)

[4.3.3 Configuration Management 11](#_heading=h.4i7ojhp)

Software Development Plan

# Introduction

The main purpose of this Software Development Plan (SDP) is to outline overall the application and the management process for the development of the [**Potpal**] software.

# Project Overview

## Project Purpose, Scope, and Objectives

* **Purpose:** This project is a part of Intro to Software Engineer course, aims to provide practical experience in designing, developing, and testing a software application. Specifically, the project involves creating a game that demonstrates core principles of software engineering, including requirements gathering, system design, implementation, and testing.
* **Scope:** The scope of this project includes the development of a fully functional Ecommerce website. The website will be implemented, and tested within a predefined timeline, following the structured phases of the software development lifecycle
* **Objectives:** The objective of this project is to demonstrate the application of software engineering principles, showcasing the students' understanding of the software development process.

## Assumptions and Constraints

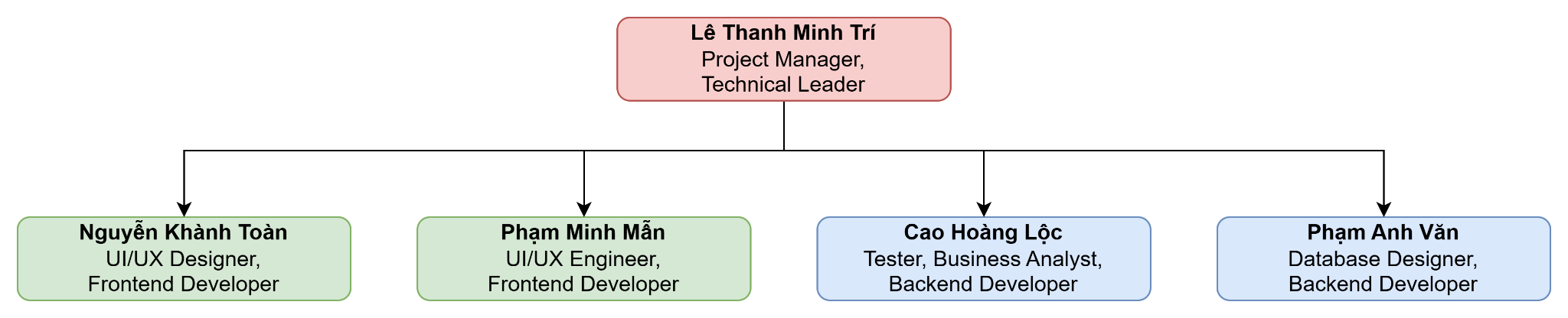
* The team consists of 5 people and there will be no more people added during the project.
* The project runs on a low budget.
* Project has a development time of around 10 weeks.

## Project Deliverables

| **Week** | **Artifacts** | **Description** |
| --- | --- | --- |
| 1 | Project Proposal | Outlining project’s objectives, key elements and functionalities of the laptop shopping website, focusing on creating a user-friendly platform for users. |
| 2 | Software Development Plan | Detailing the software development lifecycle including timelines, resource allocation, risk management and project monitoring. |
| Vision Document | Provides a foundational overview of the project's vision, highlighting the core issues it aims to solve, the main features required, and how it will deliver value to users through a smooth shopping experience. |
| Use-case Document | Detailed descriptions of how users will interact with the system, including specific flows, outcomes and user actions. |
| 3 | Detailed Requirements | Specifying the functional and nonfunctional requirements of the system. It outlines user needs, system features, and expected behavior, creating the basis for the system's design and implementation. |
| Initial Design | Specify functional and non-functional requirements. |
| 4 | Software Architecture Document | Create detailed software architecture diagrams, showing system components and their interactions. |
| UML Models | Develop comprehensive UML models to represent the system’s design. |
| UI Design | Design user interface mockups and wireframes |
|
|
| 5 | Testing Plan | A comprehensive testing plan to validate the functionalities and performance of the system. |
| Database Design | A document or diagram illustrating the structure of the database, including tables, relationships and data types, to ensure effective data storage and retrieval. |
| 6-7 | Front-end Development | Implementation for UI/UX and API calling. |
| Back-end Development | Detailed design and development of the backend system, including database setup, API development and integration with the frontend, ensuring the system runs smoothly and can handle user requests. |
|
| 8-9 | First Demo Product | Initial release for testing system and functionalities. |
| Testing | Test and create testing reports. |
| 10-11 | Second Demo Product | Second release for testing with users and get feedback for improvement. |
| User Feedback Analysis | Collecting and analyzing user feedback. |
| 12 | Final Product | Final release of the shopping platform after detailed testing and improvement from user feedback. |

# Project Organization

## Organizational Structure



## Roles and Responsibilities

| **Person** | **Role** | **Description** |
| --- | --- | --- |
| Lê Thanh Minh Trí | Project Manager,  Technical Leader | * Make project plan * Roles assignment and tasks distribution * Weekly report * Organize meetings * Validate codebase before deployment * Coordinate backend and frontend * Dev support |
| Nguyễn Khánh Toàn | UI/UX Designer,  Frontend Developer | * Design the user interface * Implement UI/UX design to codebase * Create Software Architecture Document * Cooperate with UI/UX Engineer to ensure user friendly interface * Write source code * Perform unit test * Communicate with backend devs |
| Phạm Minh Mẫn | UI/UX Engineer, Frontend Developer | * Design UI/UX system * Implement the UX system * Cooperate with UI/UX Designer to ensure user-friendly and quality interface * Write source code * Perform unit test * Communicate with backend devs * Review source code |
| Cao Hoàng Lộc | Tester, Business Analyst, Backend Developer | * Prepare and conduct test plan, test case and test report * Document requirements (interview customers, etc.) * Communicate with team members about the requirements * Coordinate to review requirements * Cooperate with Database Designer on implementing classes, components, etc. * Write source code * Perform unit test |
| Phạm Anh Văn | Database Designer, Backend Developer | * Design the backend system (components, classes, databases, etc.) * Create Software Architecture Document * Cooperate with Business Analyst to implement backend design * Write source code * Perform unit test * Communicate with frontend devs * Review source code |

# Management Process

## Project Estimates

- **Cost:** The cost varies depending on chosen domain and hosting service, commercial service, technologies and tool services.

- **Time:** Each team member is expected to spend approximately 21 hours per week over a 10 week period. With a team of 5 members, this amounts to 1050 hours total for the project.

- **Re-estimation points, circumstances:** After each sprint/phase or the previous estimation is not close.

- **Infrastructure:** The project will be developed, tested and deployed on personal computers of each member of the group then pushed and managed on GitHub.

## Project Plan

### Phase and Iteration Plan

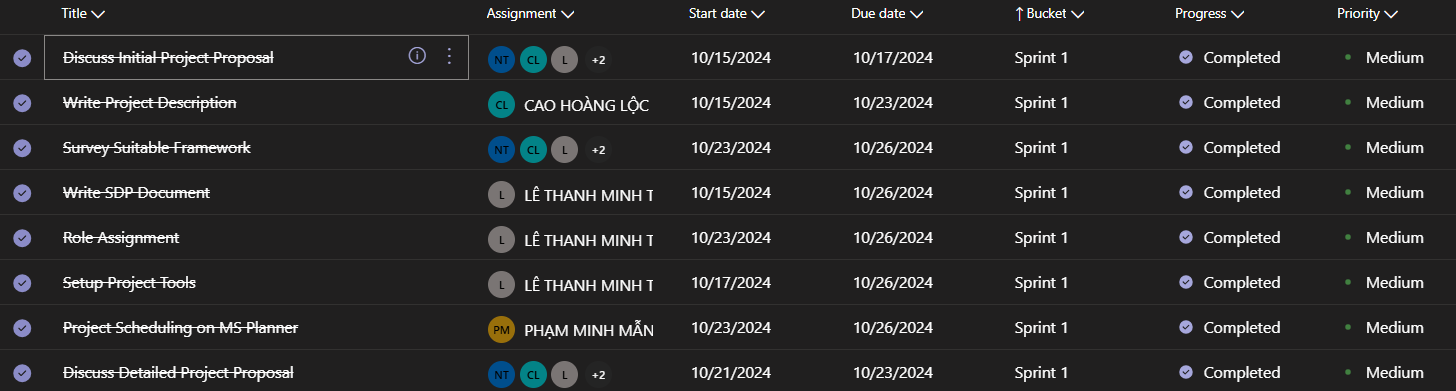
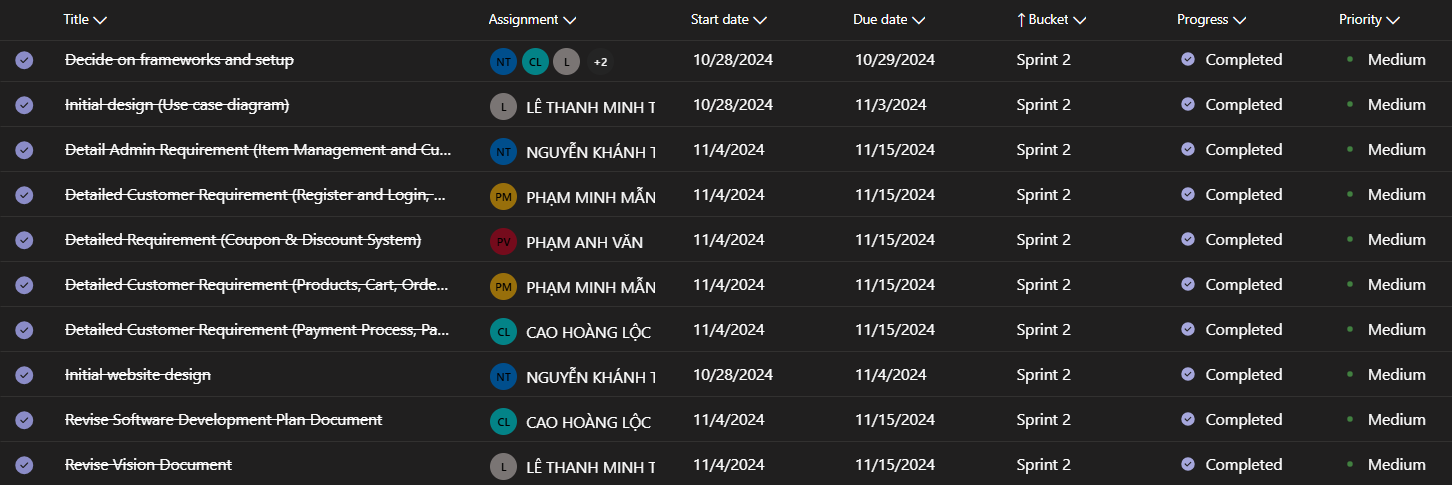
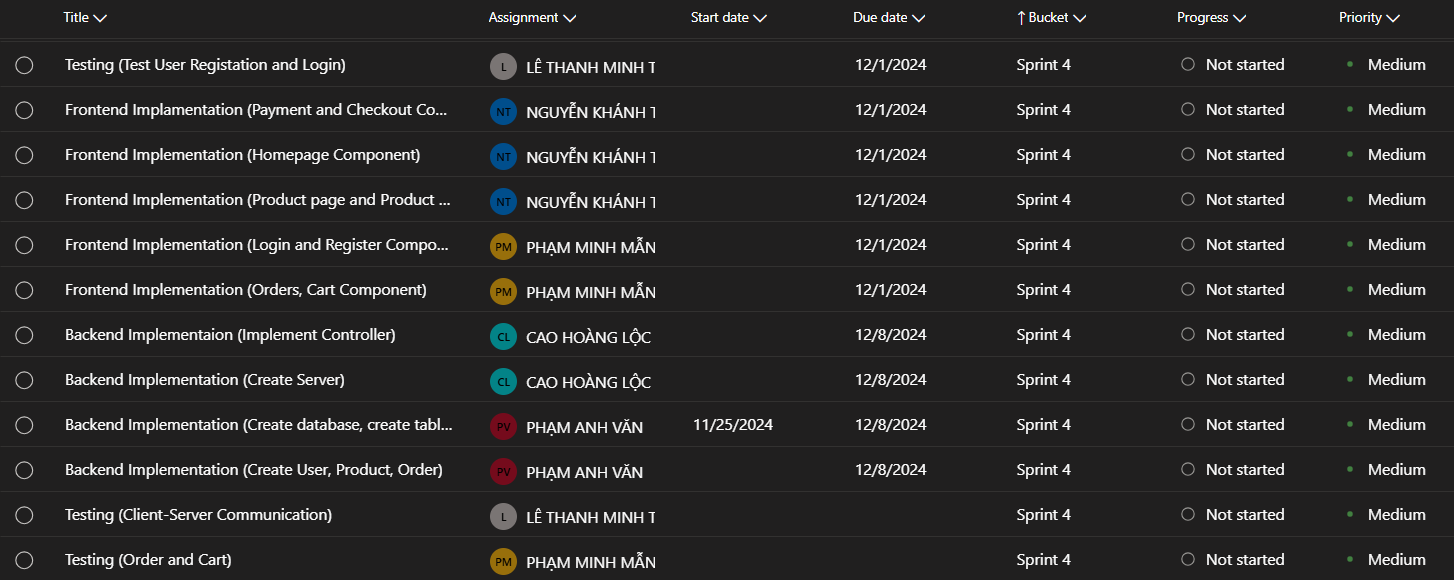
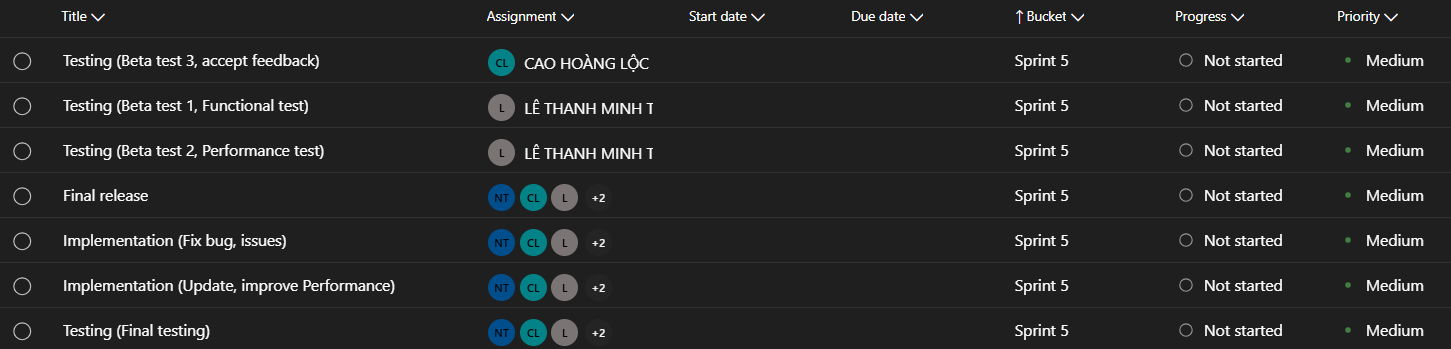
| **Phase** | **Sprint** | **Start day** | **End day** | **Main objectives** |
| --- | --- | --- | --- | --- |
| Inception | 1 | 14/10 | 27/10 | Project proposal, initial requirements |
| Elaboration | 2 | 28/10 | 10/11 | Detailed requirements and initial design |
| 3 | 11/11 | 24/11 | Detailed design and test planning |
| Construction | 4 | 25/11 | 8/12 | Implementation and testing |
| 5 | 9/12 | 22/12 | Implementation and testing |

### 

### Releases

* **Demo:** First 3 versions are intended for demonstration purposes, allow the project manager to visualize the progress and give feedback.
* **Product release:** This is where the website goes online for all users, requiring thorough testing and validation to ensure a smooth user experience.

### Project Schedule

* Sprint 1:  
  
* Sprint 2:  
  
* Sprint 3:  
  
* Sprint 4:  
  
* Sprint 5:  
  

## Project Monitoring and Control

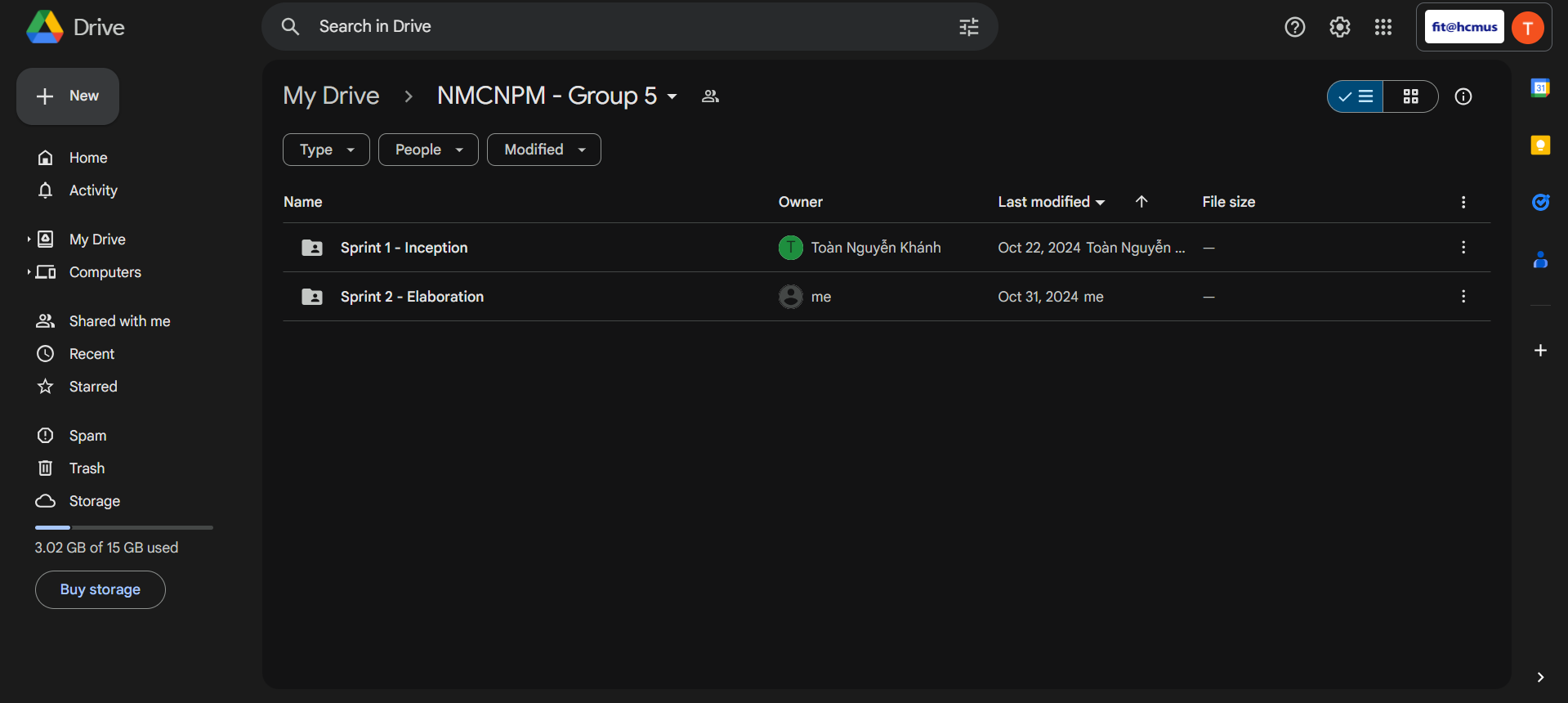
### Reporting

* **Weekly meeting:** 
  + Time: Saturday every week.
  + Method: Online meeting on Discord or Google Meet.
  + Participants: All team members.
  + Purpose: Review project progress, address challenges encountered, plan tasks and objectives for the upcoming week.
  + Documentation: The meeting result will be recorded, summarizing information and decisions made.
* **Weekly status report:**
  + Time: Once every week
  + Method: Project manager will write the documentation report.
  + Content: Summary of progress as well as project milestones and objectives, overview completed tasks and upcoming tasks, identification of any risks or issues that need attention, Updates on any changes to the project timeline or scope.
* **Informal chats:** 
  + Participants: All team members.
  + Method: Messenger, Discord (and Zalo for emergency).
  + Purpose: Provide a channel for team members to discuss, share and resolve issues.

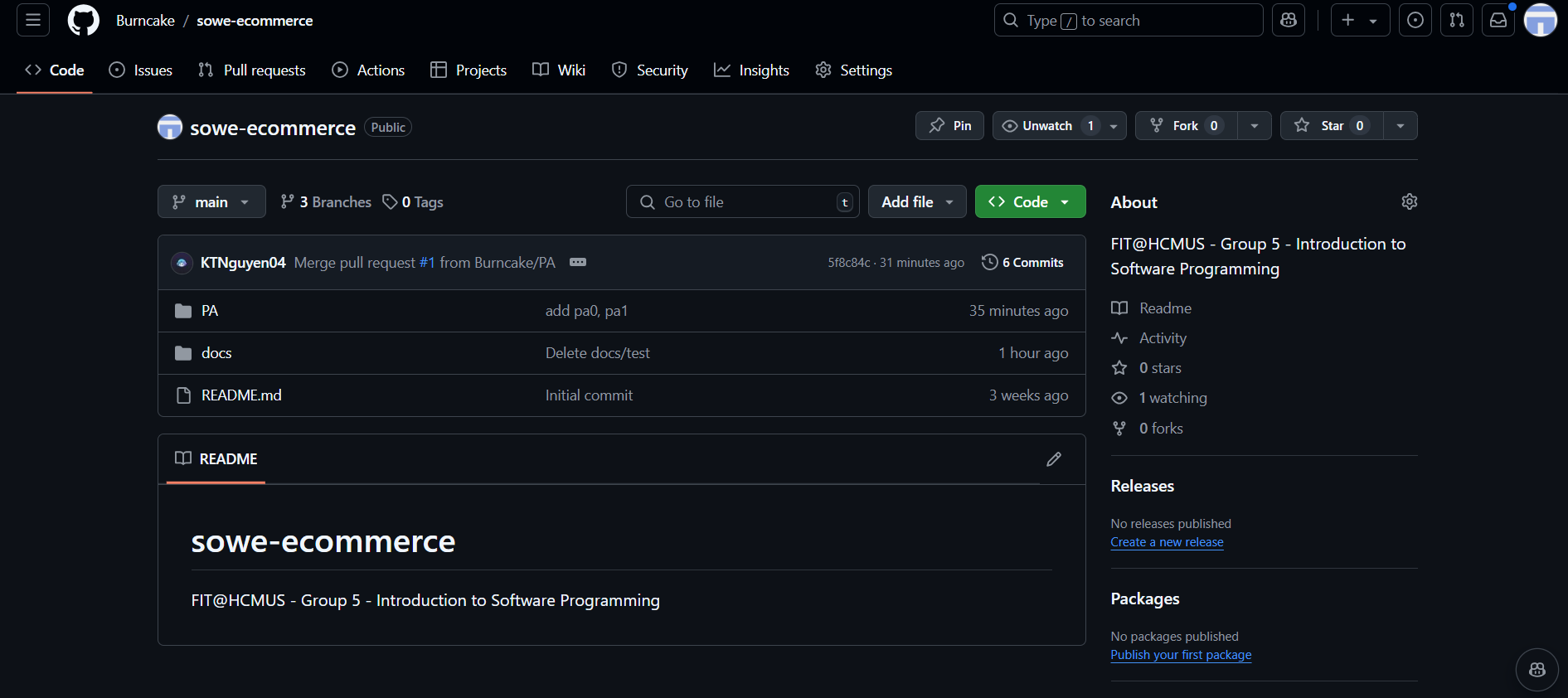
### Risk Management

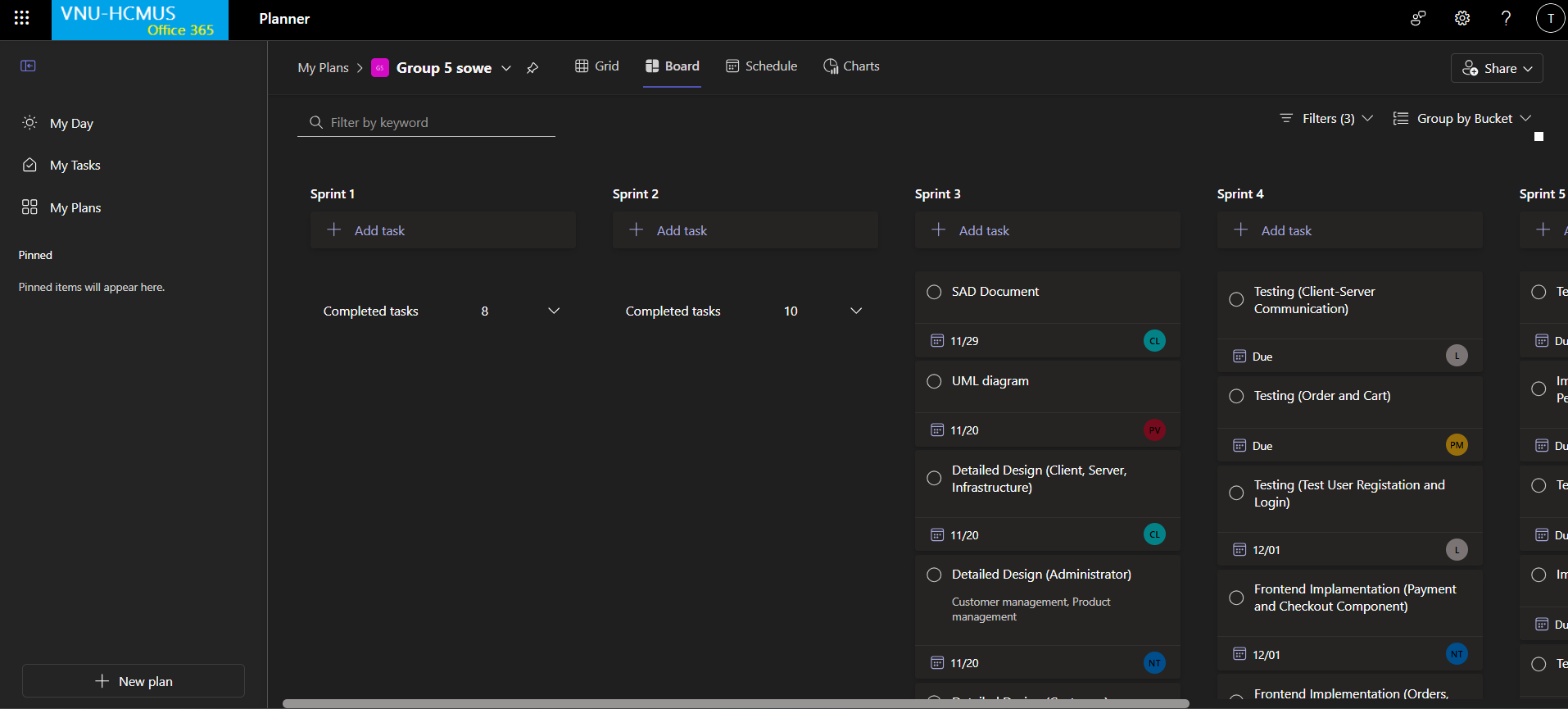
| **Risk ID** | **Risk Description** | **Probability** | **Impact** | **Risk Exposure** | **Priority** | **Mitigation Strategy or Contingency Plan** |
| --- | --- | --- | --- | --- | --- | --- |
| S3 | Poor time management | Likely | Serious | 5625 | 1 | * Can lead to catastrophe if neglected. * Members may change their priority to the closest problem to get the project’s progress back on track. * Project Manager must be agile on noticing bad time management and notify the entire team to react and resolve as soon as possible |
| M2 | Deadline changes | Frequently | Moderate | 4500 | 2 | * Project Manager adjusts plans to fit the new deadline. * Worst case scenario, the entire team shifts focus to finish the deadline as fast as possible. |
| S4 | Poor tasks distribution | Likely | Moderate | 3750 | 3 | * Project Manager re-evaluate if assigned tasks are already fair or not, then re-assign tasks if necessary. * Tasks distribution must be accepted by ALL members. |
| S1 | Staffs are unavailable during critical times | Occasional | Serious | 3750 | 4 | * Emergency contact (Direct call) to discuss with the unavailable staff. Project Manager overtake the task and split it for the entire team to handle. * In case Project Manager is unavailable, select a Co-Manager to overtake the project temporarily. |
| S2 | Lack of experiences | Occasional | Moderate | 2500 | 5 | * The member may ask Project Manager, AI assistant or any friends with experiences for help and support. * Members must be proactive to avoid wasting time. * If no choices are available, the member and Project Manager work with TA to find alternative solutions. |
| T1 | Data loss without backup | Seldom | Moderate | 1250 | 6 | * Cloud experiment data: Project Manager helps create new data to replace lost data. * Codebase on GitHub: GitHub codebase must be splitted into branches. Must be reviewed and updated regularly.  If any branch is lost or any member lost their in-progress code, that segment should only be for one specific task, simplifying the recovery process. * Documentation: Upload all documents on Drive and the finalized version on GitHub to prevent loss. |
| T3 | Technologies changes | Seldom | Moderate | 1250 | 7 | * Must carefully study and research the newly chosen technology or tool. * ALL members must agree on the changes. * Otherwise the member(s) that performs the change have to take responsibility and resolve in case conflicts arise. |
| T2 | Lack of technologies | Seldom | Moderate | 1250 | 8 | * The Project Manager should reconsider the tasks or the technologies supporting the project. * Either restart, replace, upgrade or refer to TA for guidance. |
| M3 | Natural disaster | Unlikely | Catastrophic | 900 | 9 | * Might lead to members being unavailable. * Project Manager and/or TA must be notified immediately if members would have to be unresponsive for days. |
| M1 | Requirement changes | Seldom | Insignificant | 625 | 10 | Project Manager minimizes time loss by proactively handling changes on their own if possible. |

### Configuration Management

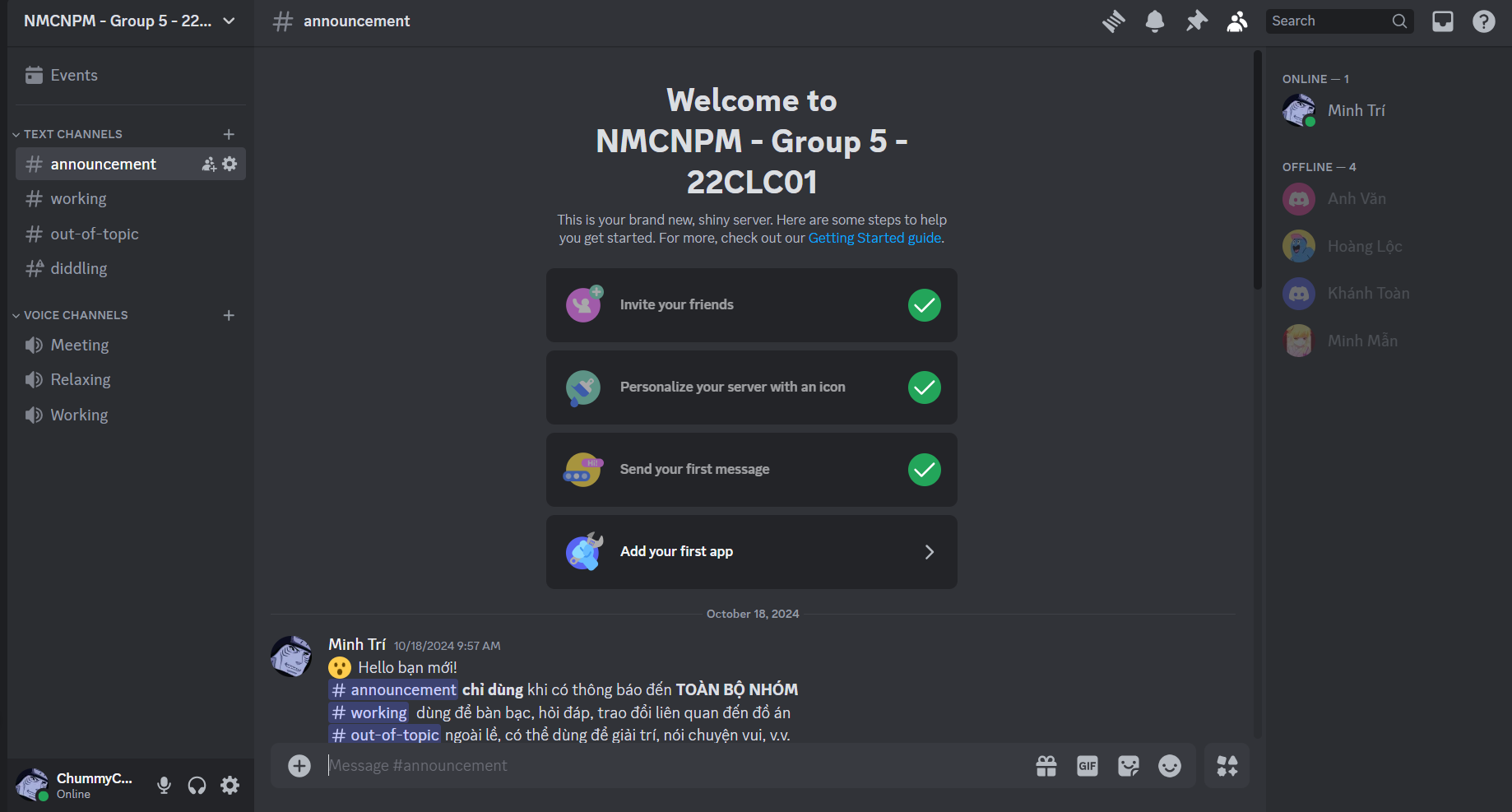
- Google Drive: [Google Drive Proof](https://drive.google.com/drive/folders/1LzSp4pshTCRXEJwvQcMpWDSktmDlS-HI?usp=drive_link)

- GitHub: [Github Proof](https://github.com/Burncake/sowe-ecommerce)

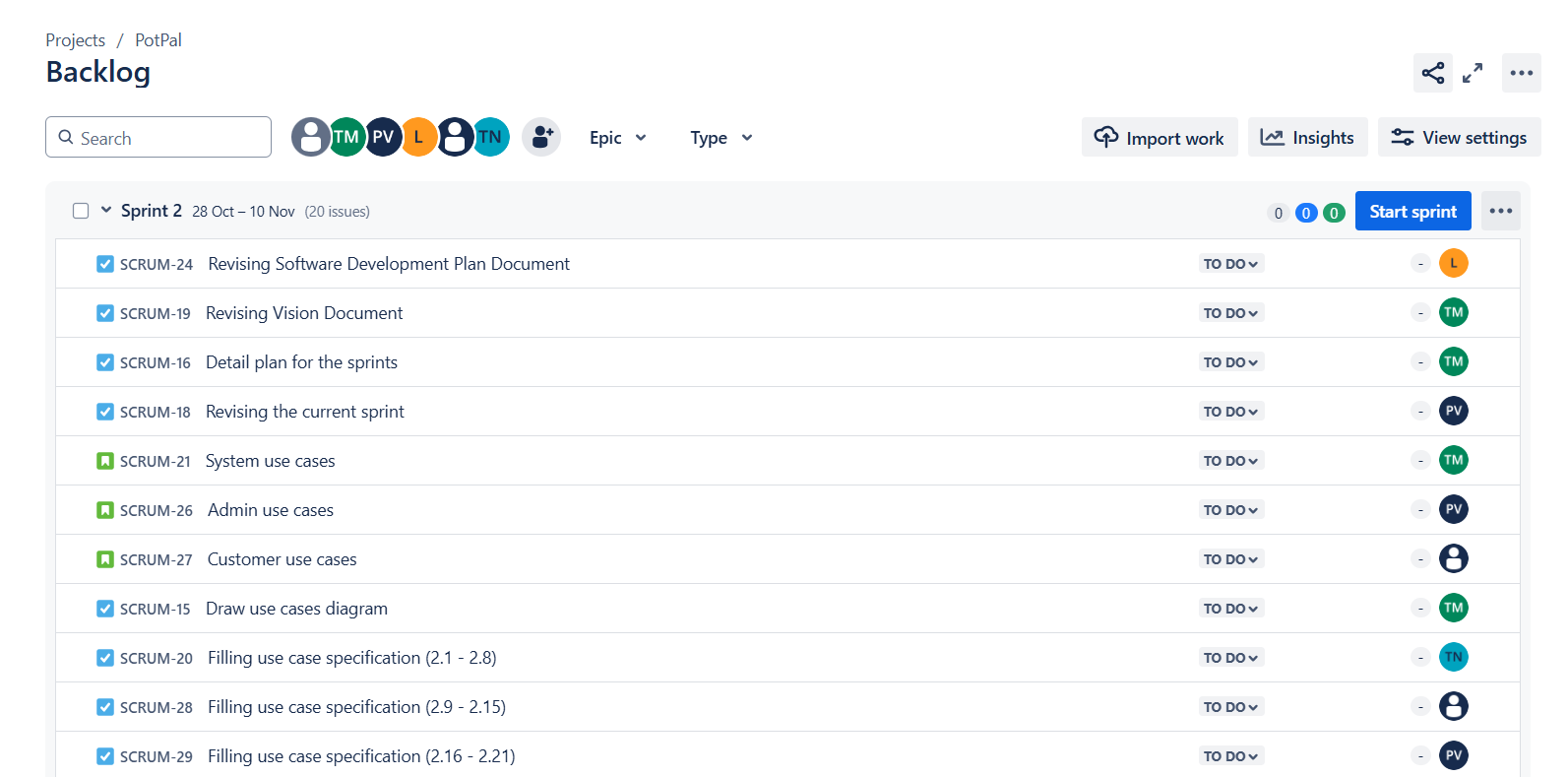


- MS Planner: [Link to MS Planner](https://planner.cloud.microsoft/student.hcmus.edu.vn/Home/PlanViews/FRRtfp8_N0qlaJ4-t9BBNsgABzbm?Type=PlanLink&Channel=Link&CreatedTime=638656307995820000)  


- Discord: [Link to Discord](https://discord.gg/YGfAcv2pkw)



- Jira: [Link to Jira](https://intro-to-se-potpal-22clc01.atlassian.net/jira/software/projects/SCRUM/boards/1/backlog?atlOrigin=eyJpIjoiZTJkMjFhZTJhZGFkNDE2Y2E2ZTY4NWY3MjIwNGYxNTIiLCJwIjoiaiJ9)



- Slack: [Link to Slack](https://hcmus-se.slack.com/archives/C07TZ32F72Q)

