

**CAPSTONE PROJECT REPORT**

**Report 2 – Project Management Plan**

– Hanoi, Jan 2024–

**Table of Contents**

[I. Record of Changes 3](#_heading=h.gjdgxs)

[II. Project Management Plan 4](#_heading=h.30j0zll)

[1. Overview 4](#_heading=h.1t3h5sf)

[1.1 Scope & Estimation 4](#_heading=h.4d34og8)

[1.2 Project Objectives 6](#_heading=h.2s8eyo1)

[1.3 Project Risks 7](#_heading=h.17dp8vu)

[2. Management Approach 8](#_heading=h.35nkun2)

[2.1 Project Process 8](#_heading=h.1ksv4uv)

[2.2 Quality Management 9](#_heading=h.44sinio)

[2.3 Training Plan 10](#_heading=h.1pxezwc)

[3. Project Deliverables 10](#_heading=h.49x2ik5)

[4. Responsibility Assignments 12](#_heading=h.2p2csry)

[4.1 Team & Structure 12](#_heading=h.147n2zr)

[4.2 Role & Responsibility 13](#_heading=h.3o7alnk)

[5. Project Communications 16](#_heading=h.23ckvvd)

[6. Configuration Management 17](#_heading=h.ihv636)

[6.1 Document Management 17](#_heading=h.32hioqz)

[6.2 Source Code Management 17](#_heading=h.1hmsyys)

[6.3 Tools & Infrastructures 17](#_heading=h.41mghml)

# I. Record of Changes

| **Date** | **A\*, M, D** | **In charge** | **Change Description** |
| --- | --- | --- | --- |
| 09/01/2024 | A | Report Create | Create basic items of the report |
| 01/02/2024 | M | Modify Feature |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

\*A - Added M - Modified D - Deleted

# II. Project Management Plan

## 1. Overview

### 1.1 Scope & Estimation

| **#** | **WBS Item** | **Complexity** | **Est. Effort**  **(man-days)** |
| --- | --- | --- | --- |
| **1** | **Authentication & Authorization** |  | **45** |
| 1.1 | Login | Medium | 10 |
| 1.2 | Logout | Simple | 5 |
| 1.3 | Register | Medium | 10 |
| 1.4 | Forgot Password | Simple | 5 |
| 1.5 | Reset Password | Simple | 5 |
| 1.6 | Edit Profile | Medium | 10 |
| **2** | **User interface** |  | **90** |
| 2.1 | Home Page | Medium | 10 |
| 2.2 | Shelves | Simple | 5 |
| 2.3 | Filter Stories | Medium | 10 |
| 2.4 | View Story | Simple | 10 |
| 2.5 | View Author | Simple | 5 |
| 2.6 | View Chapter | Complex | 15 |
| 2.7 | Like or follow | Simple | 5 |
| 2.8 | Personal Shelves | Medium | 10 |
| 2.9 | Comment | Medium | 10 |
| 2.10 | Report | Medium | 10 |
| **3** | **Author interface** |  | **90** |
| 3.1 | Manage Stories Published | Medium | 10 |
| 3.2 | Manage Story Volumes & Chapter | Medium | 10 |
| 3.3 | Add story | Medium | 10 |
| 3.4 | Edit Story | Medium | 10 |
| 3.5 | Add Chapter | Medium | 10 |
| 3.6 | Edit Chapter | Medium | 10 |
| 3.7 | Add volume | Simple | 5 |
| 3.8 | Edit volume | Simple | 5 |
| 3.9 | View Story Analyzation | Medium | 10 |
| 3.10 | Create Prints | Medium | 10 |
| **4** | **Reviewer Interface** |  | **25** |
| 4.1 | Reviewer Request | Simple | 5 |
| 4.2 | View stories need reviewed | Simple | 5 |
| 4.3 | View reviewed story’s chapters | Simple | 5 |
| 4.4 | Review story’s chapters | Medium | 10 |
| **5** | **Purchase** |  | **90** |
| 5.1 | View Wallet | Simple | 5 |
| 5.2 | Purchase story | Complex | 15 |
| 5.3 | Purchase chapter | Complex | 15 |
| 5.4 | Purchase chapters | Complex | 15 |
| 5.5 | View Transaction History | Medium | 10 |
| 5.6 | Top up | Complex | 15 |
| 5.7 | Withdraw | Complex | 15 |
| **6** | **Admin** |  | **70** |
| 6.1 | Add Genres | Simple | 5 |
| 6.2 | Edit Genres | Simple | 5 |
| 6.3 | Stories management | Medium | 10 |
| 6.4 | Users management | Simple | 5 |
| 6.5 | Tickets Management | Medium | 10 |
| 6.6 | Assign Role | Simple | 5 |
| 6.7 | Reports management | Medium | 10 |
| 6.8 | Transaction Management | Medium | 10 |
| 6.9 | Handle Refund Request | Medium | 10 |
| **Total Estimated Effort (man-days)** | | | 410 |

### 1.2 Project Objectives

* Objectives:
* This project must be finished by 21/04/2024.
* All team members need to follow the task assigned by the PM.
* All team members give their best effort to complete the project.
* Project team members learn new knowledge, and new technology to apply to the project.
* Target Metrics:
* Timelines: 100%
* Allocated Effort: 410 mans-days
* Defect Distribution:

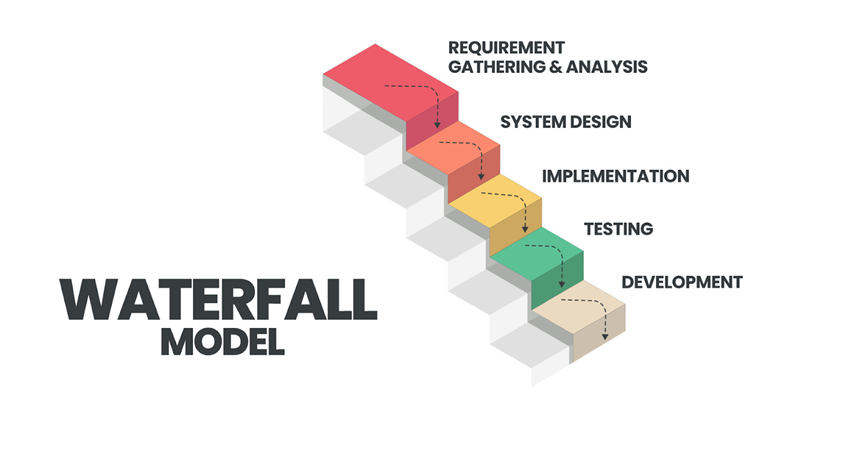
| **#** | **Testing Stage** | **Test Coverage** | **No. of Defects** | **% of Defect** | **Notes** |
| --- | --- | --- | --- | --- | --- |
| 1 | Reviewing | 100% | 10 | 9.1% | Estimates are based on experiences drawn from projects of previous subjects and experience working at top technology company |
| 2 | Unit Test | 100% | 15 | 13.6% |
| 3 | Integration Test | 100% | 30 | 27.3% |
| 4 | System Test | 100% | 35 | 31.8% |
| 5 | Performance Testing | 100% | 10 | 9.1% |
| 6 | Acceptance Test | 100% | 10 | 9.1% |
|  | **Total** |  | **110** | **100%** |  |

### 1.3 Project Risks

| **#** | **Risk Description** | **Impact** | **Possibility** | **Response Plans** |
| --- | --- | --- | --- | --- |
| 1 | Failure to meet the deadline | Medium | High | Assign tasks carefully |
| 2 | Requirement changes | Medium | Medium | Members need to discuss to clearly understand the requirements before implementing. |
| 3 | The time and manpower of the project may not be able to meet the need to build a complete project | Medium | High | Focus on building and perfecting the main functions of the system |
| 4 | Conflicts between team members do not reach consensus. | Low | High | Regularly hold meetings to reach a consensus. |
| 5 | Compatibility and integration risks | Medium | High | Build APIs to increase interoperability and efficiently communicate data with other applications and systems |

## 2. Management Approach

### 2.1 Project Process



**The Genesis** project uses Waterfall development.

The Waterfall Model is a linear and sequential approach to software development, where each phase must be completed before moving on to the next. Here are the steps to implement the project's development using the Waterfall Model:

**Requirements Analysis**

* Gather and document all project requirements comprehensively at the beginning of the project.
* Define the scope, goals, and objectives of the project based on the gathered requirements.

**System Design**

Design the user interface and interactions based on the system design

**Implementation**

* Code the software according to the detailed technical specifications and design documents.
* Integrate different modules and components to create the complete software system.

**Testing**

Perform system testing, integration testing, and user acceptance testing to validate the software's performance.

**Deployment**

* Deploy the Genesis system into the real environment.
* Train users and provide documentation for using the software effectively.

Particularly, the final phase, Maintenance, will not be implemented in the context of a graduation project.

### 2.2 Quality Management

To improve Project quality, we plan to use several approaches as follows:

* ***Reviewing***: Conduct review, inspection, and evaluation of key project documents. Besides, there is a design review, a source code review, and a function view to ensure the project is built according to the basic rules.
* ***Unit Testing***: Members perform unit tests for their own modules to ensure that basic errors do not occur.
* ***Integration Testing***: Conduct integrated testing of the combination of components and modules in the system to ensure that they can be compatible with each other without causing errors, and overcome problems of asynchrony between modules in the system.
* ***Performance Testing***: Evaluate and test the system's operability and performance, test the system's load capacity in many different situations. To ensure that the system has a good load capacity, meeting the requirements for processing speed and response time.
* ***Security Testing***: Test and verify the security of data, access rights, authentication and other security methods in the system.
* ***System Testing***: Perform comprehensive testing on **Genesis** systems to ensure operation, performance, and meet set quality requirements.

### 2.3 Training Plan

| **Training Area** | **Participants** | **When, Duration** | **Waiver Criteria** |
| --- | --- | --- | --- |
| Source Control Training on Git, Github | All team members | 15/01/2024 (1 day) | Mandatory |
| Coding Convention & Bug Logging Convention | All team members | 16/01/2024 (1 day) | Mandatory |
| ReactJs 18, Ant Design (FrontEnd) | DuongPC | 17/01/2024- 27/01/2024(10 days) | Mandatory |
| C#/.Net 7 (BackEnd) | All team members | 17/01/2024 - 24/05/2024(7 days) | Mandatory |

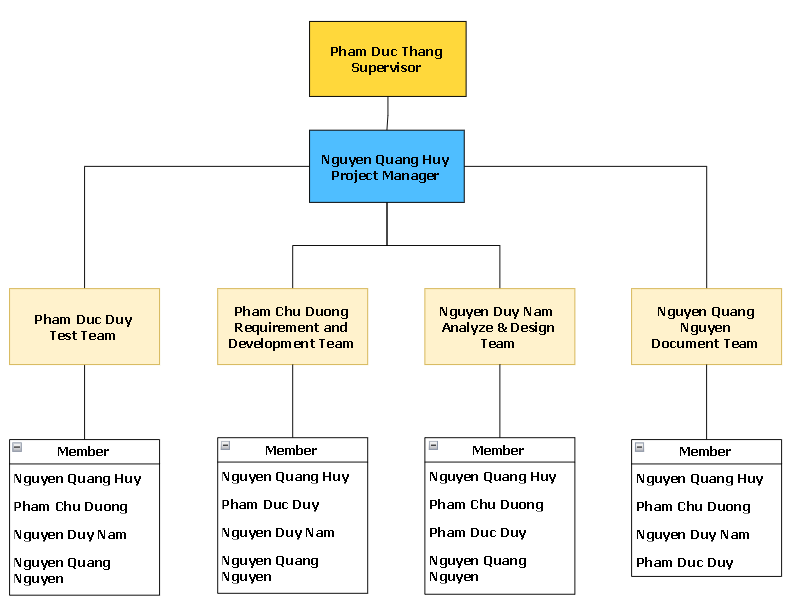
## 3. Project Deliverables

## 

| **#** | **Deliverable** | **Start Date** | **Due Date** | **Duration** | **Predecessor** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- |
| **1** | **Requirements Analysis** | **01/01/2024** | **21/01/2024** | **21 days** |  | **Week 1 - Week 3** |
| 1.1 | Prepare Project Introduction Document (Report 1) | 01/01/2024 | 06/01/2024 | 6 days |  |  |
| 1.2 | Review/Delivery Project Introduction Document | 07/01/2024 | 07/01/2024 | 1 day | 1.1 |  |
| 1.3 | Prepare Project Planning Document (Report 2) | 08/01/2024 | 13/01/2024 | 6 days |  |  |
| 1.4 | Review/Deliver Planning Document | 14/01/2024 | 14/01/2024 | 1 day | 1.3 |  |
| 1.5 | Prepare SRS Document (Report 3) | 15/01/2024 | 20/01/2024 | 6 days |  |  |
| 1.6 | Review/Delivery SRS Document (Report 3) | 21/01/2024 | 21/01/2024 | 1 day | 1.5 |  |
| **2** | **System Design** | **22/01/2024** | **04/02/2024** | **14 days** | **1** | **Week 4 - Week 5** |
| 2.1 | Building Source code Skeleton | 22/01/2024 | 23/01/2024 | 2 days |  |  |
| 2.2 | Building Database Schema | 24/01/2024 | 28/01/2024 | 5 days |  |  |
| 2.3 | Prepare SDS Document (Report 4) | 29/01/2024 | 03/02/2024 | 6 days |  |  |
| 2.4 | Review/Delivery SDS Document (Report 4) | 04/02/2024 | 04/02/2024 | 1 days | 2.3 |  |
| **3** | **Implementation** | **19/02/2024** | **17/03/2024** | **28 days** | **2** | **Week 6 - Week 9** |
| 3.1 | Workable Source Codes & Database | 19/02/2024 | 15/03/2024 | 26 days |  |  |
| 3.2 | Review/Delivery Source Codes & Database | 16/03/2024 | 17/03/2024 | 2 days | 3.1 |  |
| **4** | **Testing** | **18/03/2024** | **31/03/2024** | **14 days** | **3** | **Week 10 - Week 11** |
| 4.1 | Prepare Unit Test Case Documentation | 18/03/2024 | 26/03/2024 | 9 days |  |  |
| 4.2 | Review/Delivery Unit Test Case Documentation | 27/03/2024 | 27/03/2024 | 1 days | 4.1 |  |
| 4.3 | Prepare Integration Test Report | 18/03/2024 | 26/03/2024 | 9 days |  |  |
| 4.4 | Review/Delivery Integration Test Report | 27/03/2024 | 27/03/2024 | 1 days | 4.3 |  |
| 4.5 | Prepare Acceptance Test Report | 18/03/2024 | 26/03/2024 | 9 days |  |  |
| 4.6 | Review/Delivery Acceptance Test Report | 27/03/2024 | 27/03/2024 | 1 days | 4.5 |  |
| 4.7 | Prepare for Test Documentation (Report 5) | 18/03/2024 | 26/03/2024 | 9 days |  |  |
| 4.8 | Review/Delivery Test Documentation (Report 5) | 27/03/2024 | 27/03/2024 | 1 days | 4.6 |  |
| 4.9 | Fix bugs | 25/03/2024 | 31/03/2024 | 7 days |  |  |
| **5** | **Deployment** | **01/04/2024** | **14/04/2024** | **14 days** | **4** | **Week 12 - Week 13** |
| 5.1 | Building deployment environments for apps and databases | 01/04/2024 | 03/04/2024 | 3 days |  |  |
| 5.2 | Prepare User Guide (Report 6) | 01/04/2024 | 06/04/2024 | 6 days |  |  |
| 5.3 | Review/Delivery User Guide (Report 6) | 07/04/2024 | 07/04/2024 | 1 days | 5.2 |  |
| 5.4 | Prepare Final Report (Report 7) | 08/04/2024 | 13/04/2024 | 6 days |  |  |
| 5.5 | Review/Delivery Final Report (Report 7) | 14/04/2024 | 14/04/2024 | 1 days | 5.4 |  |
| 5.6 | Prepare Presentation File | 11/04/2024 | 13/04/2024 | 3 days |  |  |
| 5.7 | Review/Delivery Presentation File | 14/04/2024 | 14/04/2024 | 1 days |  |  |

## 4. Responsibility Assignments

### 4.1 Team & Structure



### 4.2 Role & Responsibility

| Role | Name | Responsibilities |
| --- | --- | --- |
| ***Supervisor*** | Pham Duc Thang | * Give instruction to the project team * Verify deliverables * Supervise project team’s status |
| ***Project Manager*** | Nguyen Quang Huy | * Tracking member’s work * Report working status to the instructor * Review, discuss each stage completed on time * Assign task to each member * Have overall responsibility of the project |
|  |  | **Documentation Report Team** |
| ***Leader*** | Nguyen Quang Nguyen | * Assign task to members * Responsible for SRS (Software Requirement Specification) * Review all of the documents * Support in other documents |
| ***Members*** | Nguyen Quang Huy  Pham Chu Duong  Nguyen Duy Nam  Pham Duc Duy | * Support, complete in other documents |
|  |  | **Analysis & Design Team** |
| ***Leader*** | Nguyen Duy Nam | * Analysis Requirements * Create SRS and define scopes * Define high level architecture based on SRS * Use case specification * System Architecture Design * Screen Design * Sequence Diagram * Workflow |
| ***Members*** | Nguyen Quang Huy  Pham Duc Duy  Pham Chu Duong  Nguyen Quang Nguyen | * Complete Screen Design * Screen Layout * Class Diagram * Flow Chart * Context Diagram * ERD Diagram * Database Design * Define business process flow and object state |
|  |  | **Requirement & Development Team** |
| ***Leader*** | Pham Chu Duong | * Decide technique and tools to be used * Ensure assignment or directly assigns specific tasks to Code Team members * Ensure that the team has sufficient performance levels and is also responsible for conflict prevention and resolution. * Provide feedback on quality plan, help determine metrics and criteria for this project * Control source code and merge code * Provide feedback on deliverables and quality reviews * Keeping track of development work |
| ***Members*** | Nguyen Quang Huy  Pham Duc Duy  Nguyen Duy Nam  Nguyen Quang Nguyen | * Coding back-end * Coding front-end * Provide the best possible application reception by the user * Help decide what technique and tools to be used * Deploy web application to host server in each release version |
|  |  | **Testing Team** |
| ***Leader*** | Pham Duc Duy | * Create test plan and define test strategy * Assign tasks to members * Create test case and testing documents * Controlling testing activities * Implement test case and report test result * Collect and analyse customer information |
| ***Member*** | Nguyen Quang Huy  Pham Chu Duong  Nguyen Duy Nam  Nguyen Quang Nguyen | * Create test case in Unit test * Create test case in Integration test * Create test case in System test * Create test case in Acceptance test * Implement test case, fix bugs and log defect |

## 5. Project Communications

| **Communication Item** | **Who/ Target** | **Purpose** | **When, Frequency** | **Type, Tool, Method(s)** |
| --- | --- | --- | --- | --- |
| Kick-off meeting | Project team + Supervisor | Introduce the project, confirm the project objectives, goals, and deliverables | A one-off event, at the beginning of the project | Online meetings in Google Meet, Discord |
| Meeting with Supervisor | Project team + Supervisor | Review project status and discuss potential issues. | Weekly  (Every Weekend) | Meet face-to-face |
| Weekly meeting schedule | Project team | Discuss what each team member did, and what they’ll do | Weekly  (Every Thursday, Friday) | Meetings in Google Meet, Discord |
| Daily meeting schedule | Each small team (Design, FrontEnd, BackEnd, Test,...) | Each team will have an online meeting to inform the others: “What did I do yesterday?”, “What will I do today?” | Everyday | Online meetings in Google Meet, Discord |

## 6. Configuration Management

### 6.1 Document Management

Documents are managed and stored on the drive so that everyone can access and view versions of each folder.

### 6.2 Source Code Management

The source code is hosted on GitHub and the leader manages the source under 3 main branches including:

* Shorten-flow branches: separate branches for each function in development
* Dev branch: branch developed and merged by function when completed and unit tested
* Testing branch: branch to perform integration testing for functions that have already been deployed on the dev branch
* Main branch: the branch that shows the application is completed and released through each version

The code merge process is done according to each merge request and is controlled and managed by the leader.

### 6.3 Tools & Infrastructures

| **Category** | **Tools / Infrastructure** |
| --- | --- |
| **Technology** | ReactJs 18, Ant Design (FrontEnd), C#/.Net 7 (BackEnd) |
| **Database** | SQL SERVER |
| **IDEs/Editors** | Visual Studio, VS Code, SSMS |
| **Diagramming** | Visio, DrawIO, Visual Paradigm |
| **Documentation** | Ms Office, Google Docs/Sheets/Slides |
| **Version Control** | GitHub (Source Codes), Google Drive (Documents) |