Brute Force

BruteWardrobe
Software Development Plan (Small Project)
Version 1.0

Revision History

Date	Version	Description	Author
28/10/23	1.0	First version of Software Development Plan document	Team Leader (Nguyen Minh Quan)

Table of Contents

Software Development Plan (Small Project)

●

1.   Introduction

1.1   Purpose

The purpose of the Software Development Plan (SDP) is to communicate to team members and stakeholders the approach to be taken when developing software on a program and how a program manager will utilize direct resources.

The following people use the *Software Development Plan*:

●   The **project manager** uses it to plan the project schedule and resource needs, and to track progress against the schedule.

●   **Project team members** use it to understand what they need to do, when they need to do it, and what other activities they are dependent upon.

1.2   Scope

This *Software Development Plan* describes the overall plan to be used by the **BruteWardrobe** project, including deployment of the product. The details of the individual iterations will be described in the Iteration Plans.
The plans as outlined in this document are based upon the product requirements as defined in the *Vision Document*.

1.3   Overview

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

Purpose:

Creating an e-commerce website dedicated to selling brand clothes. The website's purpose is to provide a seamless and user-friendly online shopping experience for customers, while also helping the brand expand its online presence and reach a broader audience.

Scope:

The project includes designing and developing a secure and user-friendly e-commerce website for selling brand clothes. It covers website development, content management, user experience, security, payment integration, inventory management, search and filtering, order processing, testing, and documentation.

Objectives:

The project's main goals are to create a functional e-commerce website that provides a secure and seamless shopping experience for customers, increases brand visibility and online sales. Ongoing support and maintenance, as well as tracking and improving website performance, are also part of the objectives.

2.2   Assumptions and Constraints
Assumptions:

1.   Clear requirements: clearing and well-defined set of requirements for the website, including features, design, and functionality.

2.   Brand assets: existing brand assets such as logos, color schemes and product images.

3.   Product information: providing enough product's details, images and pricing.

4.   E-commerce platform: building professional websites like sale pages of famous brands such as Nike, Adidas, Puma,...

5.   Payment processing: having arrangements for payment including customer, merchant accounts and payment gateway.

Constraints:

1.   Budget: optimize the available resources, technology choices, and development options.

2.   Staffing: availability and expertise of the developing team including project manager, UI/UX designer, developers and tester, may be constrained and impacted to the project timeline.

3.   Equipment: software tools for development and testing may be limited by budget or technical constraints.

4.   Schedule: there may be time constraints such as a specific deadline time which need to be adhered to.

2.3   Project Deliverables

The final product is the most complete version which satisfies the user's needs.
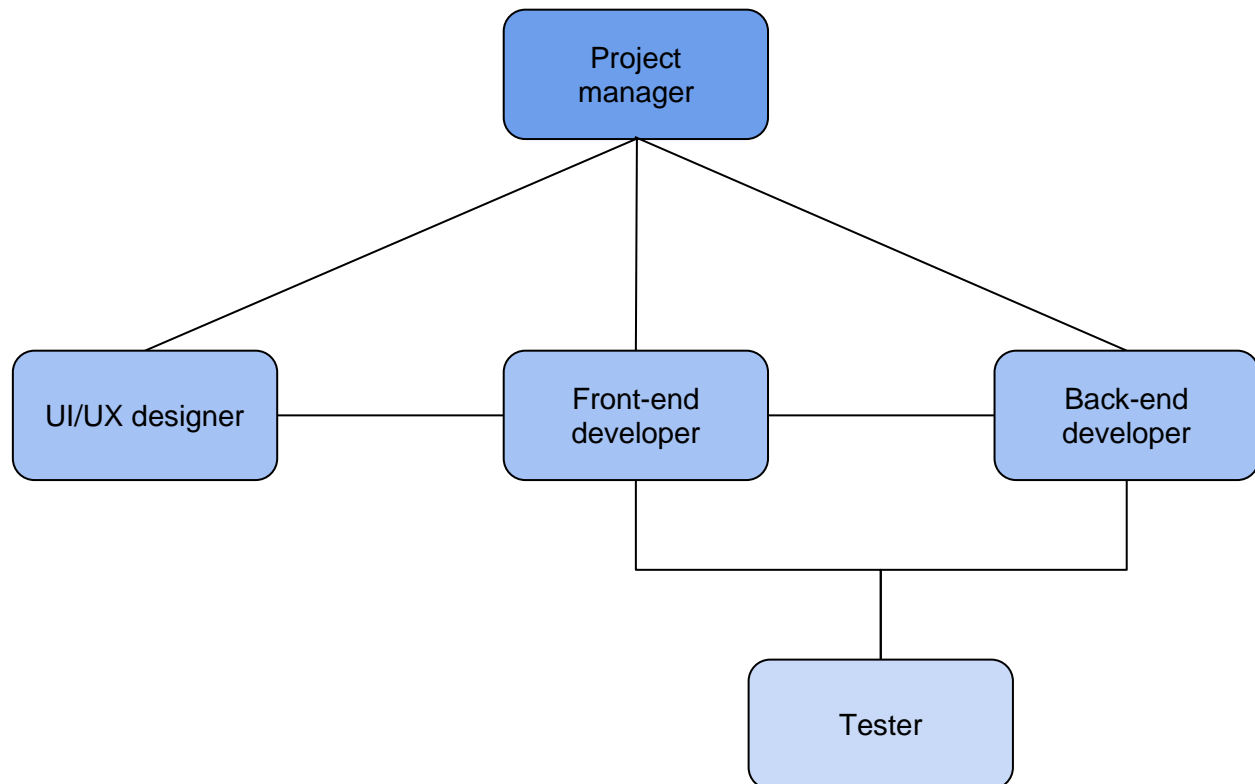
The following deliverables will be produced during the project.

● 	User interface prototype
● 	Supplementary specifications
● 	Business use cases
● 	Creative design briefs
● 	Navigation map
● 	Data model
● 	Design model
● 	Database design
● 	Use case survey
● 	Software architecture document
● 	Implementation subsystem
● 	Test package
● 	Change requests
● 	Test summary

3.   Project Organization

3.1   Organizational Structure

```
                          ┌─────────────┐
                          │   Project   │
                          │   manager   │
                          └─────────────┘
                      ╱          │          ╲
              ┌────────────┐ ┌────────────┐ ┌────────────┐
              │ UI/UX      │ │ Front-end  │ │ Back-end   │
              │ designer   │─│ developer  │─│ developer  │
              └────────────┘ └────────────┘ └────────────┘
                                    │              │
                                  ┌──────────────┐
                                  │    Tester    │
                                  └──────────────┘
```

3.2 Roles and Responsibilities

Role	Responsibilities
Project manager (Team leader)	Being responsible for projects from initiation to close, making sure the work gets done efficiently and satisfactorily: ● Plan and develop the project idea ● Monitor project progress and set deadlines ● Solve thorny issues that arise throughout the developing process ● Evaluate project performance
UI/UX designer	● Gathering and evaluating user requirements, in collaboration with product managers and engineers ● Illustrating design ideas using storyboards, process flows and sitemaps ● Designing graphic user interface elements, like menus, tabs and widgets
Back-end developer	● Creating, maintaining, testing, and debugging the entire back-end of the website ● This includes the core application logic, databases, data and application integration, API, and other processes taking place behind the scenes

Front-end developer	● Implementing visual and interactive elements that users engage with ● Creating user interfaces, test websites' usability, troubleshoot coding issues and change interfaces
Tester	● Testing websites for usability and reporting bugs or usability issues to developers ● Reporting on website performance and user experience ● Reporting any security risks or website errors to the appropriate department for resolution

Person	Roles
Nguyễn Minh Quân	Project manager (Team leader), Front-end developer
Trần Hải Phát	Front-end developer, UI/UX designer
Nguyễn Phát Đạt	Back-end developer, Tester
Phạm Sĩ Phú	Front-end developer, UI/UX designer
Phạm Trần Tuấn Tú	Back-end developer, Tester

4. Management Process

4.1 Project Estimates

The project will take 11 weeks to carry out.
- **Week 1 (05/10 - 11/10):** Group registration and planning the schedule for the next few weeks.
- **Week 2 (12/10 - 18/10):** PA0 - Preparing project proposal and tools setup.
- **Week 3 (19/10 - 25/10):** PA1 - Writing project plan, vision document.
- **Week 4 (26/10 - 01/11):** PA1 - Writing project plan, vision document.
- **Week 5 (02/11 - 08/11):** PA2 - Revised project plan and detailed vision document.
- **Week 6 (09/11 - 15/11):** PA2 - Use-case model and specification.
- **Week 7 (16/11 - 22/11):** PA3 - Revising the use-case model and specifications from TAs's feedback.
- **Week 8 (23/11 - 29/11):** PA3 - Defining software architecture, class diagrams.
- **Week 9 (30/11 - 06/12):** PA4 - Revising software architecture that was submitted in PA3 and drawing a deployment diagram(s) using UML.
- **Week 10 (07/12 - 13/12):** PA4 - Designing user-interface prototype.
- **Week 11 (14/12 - 23/12):** PA5 - Preparing test plan, designing and executing test cases, summarizing and reporting test results and submitting project presentation.
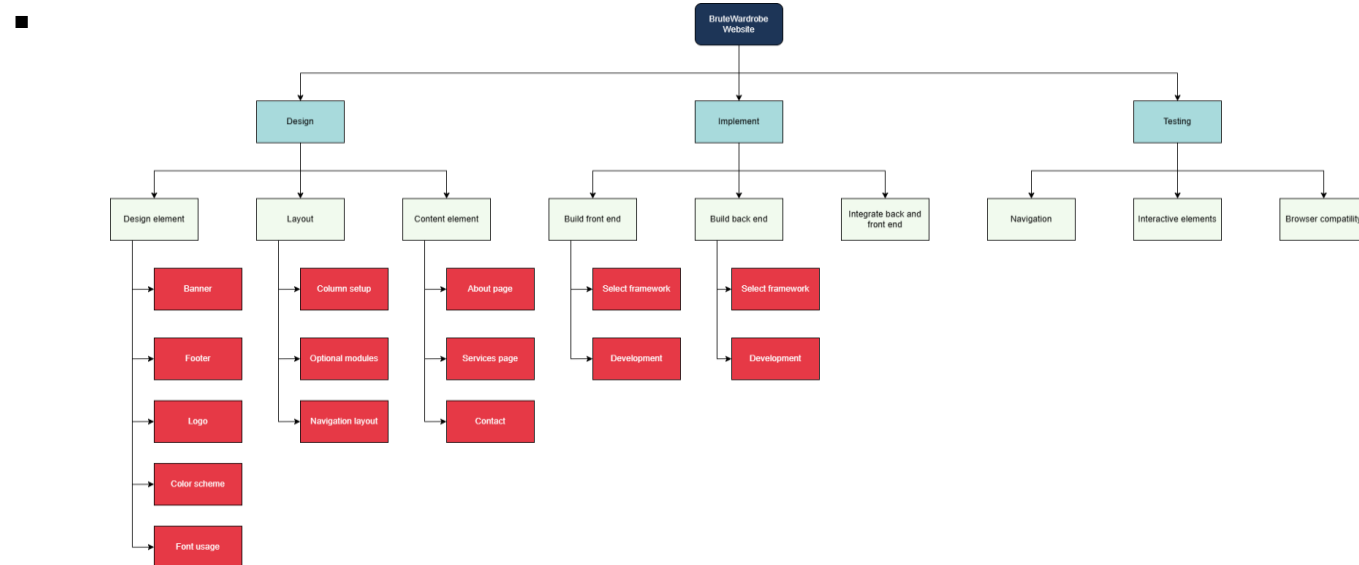
4.2 Project Plan

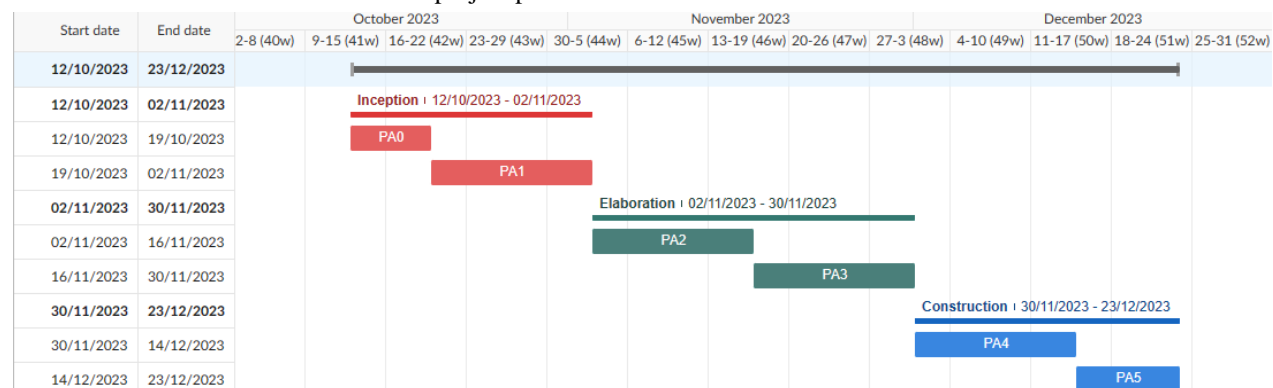*[This section contains the schedule and resources for the project.]*

4.2.1   Phase Plan

- Work Breakdown Structure:

■

■   - The allocation of time to the project phases:

- Major milestones:

Milestones are established at the end of each phase to define the particular goals to be met as well as the project's progress.

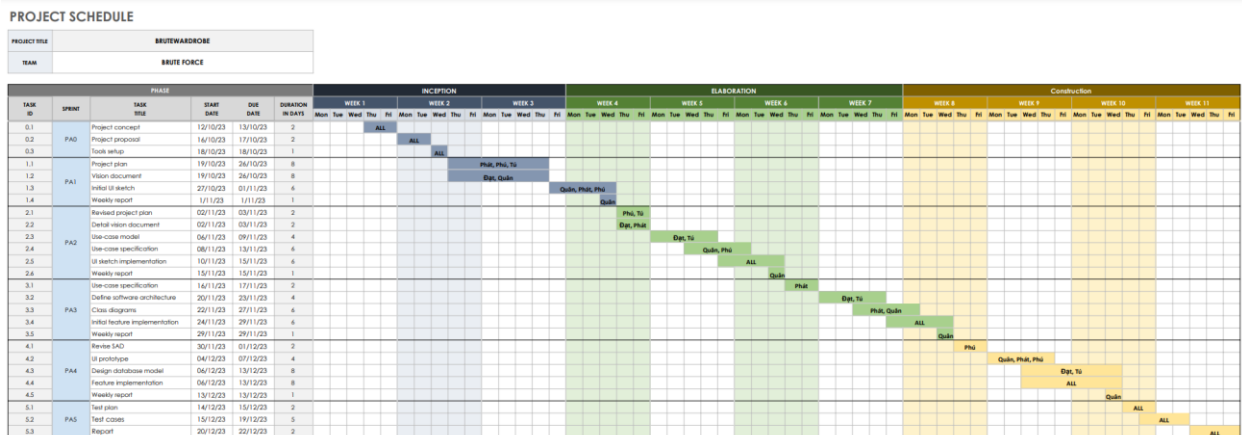| Milestone (At the end of the phase)	Achievement criteria
Inception	- Reach an agreement on project concepts. - Outline the fundamental product requirements and implementation plan.
Elaboration	- Complete the analysis and design of use-case diagrams.
Construction	- Implement the product successfully with all proposed features.

4.2.2 Iteration Objectives

Phase	Iteration	Objectives
Inception	Sprint 1	- The group's software project concept. - An early draft of the project plan. - An initial version of the vision document.
Elaboration	Sprint 2	- Use-case diagrams show use-cases, actors, and their relationships. - The use-case specification document for each use-case in use-case diagram
Construction	Sprint 3	- The software architecture document. - Class diagrams for each of the key components defined in the architecture.
	Sprint 4	- The sketch of the user interface. - The initial release.
	Sprint 5	- The test plan and test cases - Full release

4.2.3 Releases

There will be two releases. The first, a beta version, will be completed at the end of sprint 4. Any issues detected will be fixed so that the full release may be completed before the conclusion of sprint 5.

4.2.4   Project Schedule

-    **Overall**:

-    **Specific**:
●   Inception

TASK ID	SPRINT	TASK TITLE	START DATE	DUE DATE	DURATION IN DAYS	INCEPTION																WEEK 4		
						WEEK 1					**WEEK 2**					**WEEK 3**								
						Mon	Tue	Wed	Thu	Fri	Mon	Tue	Wed	Thu	Fri	Mon	Tue	Wed	Thu	Fri	Mon	Tue	Wed
0.1	PA0	Project concept	12/10/23	13/10/23	2				ALL														
0.2		Project proposal	16/10/23	17/10/23	2							ALL											
0.3		Tools setup	18/10/23	18/10/23	1								ALL										
1.1	PA1	Project plan	19/10/23	26/10/23	8									Phát, Phú, Tú									
1.2		Vision document	19/10/23	26/10/23	8									Đạt, Quân									
1.3		Initial UI sketch	27/10/23	01/11/23	6											Quân, Phát, Phú							
1.4		Weekly report	1/11/23	1/11/23	1																Quân		
2.1	PA2	Revised project plan	02/11/23	03/11/23	2																		
2.2		Detail vision document	02/11/23	03/11/23	2																		
2.3		Use-case model	06/11/23	09/11/23	4																		
2.4		Use-case specification	08/11/23	13/11/23	6																		
2.5		UI sketch implementation	10/11/23	15/11/23	6																		
2.6		Weekly report	15/11/23	15/11/23	1																		
3.1	PA3	Use-case specification	16/11/23	17/11/23	2																		
3.2		Define software architecture	20/11/23	23/11/23	4																		
3.3		Class diagrams	22/11/23	27/11/23	6																		
3.4		Initial feature implementation	24/11/23	29/11/23	6																		
3.5		Weekly report	29/11/23	29/11/23	1																		
4.1	PA4	Revise SAD	30/11/23	01/12/23	2																		
4.2		UI prototype	04/12/23	07/12/23	4																		
4.3		Design database model	06/12/23	13/12/23	8																		
4.4		Feature implementation	06/12/23	13/12/23	8																		
4.5		Weekly report	13/12/23	13/12/23	1																		
5.1	PA5	Test plan	14/12/23	15/12/23	2																		
5.2		Test cases	15/12/23	19/12/23	5																		
5.3		Report	20/12/23	22/12/23	2																		

- Elaboration

- Construction

4.2.5 Project Resourcing

The needed staff for the project is 5. Members come from the same course. During the inception phase, there will be training sections to guarantee that all staff acquire the necessary project skills.

4.3 Project Monitoring and Control

4.3.1 Requirements Management

The requirements for this system are captured in the Vision document. Requested changes to requirements are captured in Change Requests, and are approved as part of the Configuration Management process.

4.3.2 Reporting and Measurement

Updated cost and schedule estimates, and metrics summary reports, will be generated at the end of each iteration.

The Minimal Set of Metrics, as described in the RUP Guidelines: Metrics, will be gathered on a weekly basis. These include:

- Earned value for completed tasks. This is used to re-estimate the schedule and budget for the remainder of the project, and/or to identify need for scope changes.

- Total defects open and closed – shown as a trend graph. This is used to help estimate the effort remaining to correct defects.

- Acceptance test cases passing – shown as a trend graph. This is used to demonstrate progress to stakeholders.

- In addition, overall costs will be monitored against the project budget.

4.3.3 Risk Management

Risks will be identified in the Inception Phase using the steps identified in the RUP for Small Projects activity "Identify and Assess Risks". Project risk is evaluated at least once per iteration and documented in this table. The risks of the greatest magnitude are listed first in the table.

Risk Ranking (High, Medium, Low)	Risk Description and Impact	Mitigation Strategy and/or Contingency Plan
High	There are members who are ill and unavailable at a key time -> Slow down project progress	- Prepare a backup time - Reorganize team so that there is more overlap of work and people therefore understand each other's jobs
Medium	Design modifications to match the needs of the instructor -> Slow down project progress	- Prepare a backup time - Request feedback on project from instructors on a regular basis
Low	The database used in the system cannot process as many transactions per second as expected -> Increase total budget	- Investigate the possibility of buying a higher-performance database or ignore the problem
High	Inaccurate time estimates for project components -> The project will not be completed on schedule	- Prepare a backup time - Adjust schedule regularly
Medium	Software tools cannot be integrated -> Related features are difficult to develop or canceled	- Prepare alternate tools or features
High	The time required to fix bugs is longer than anticipated -> Slow down project progress	- Prepare a backup time - Use code generation tools
Medium	The size of the software is underestimated -> Some features must be eliminated or left incomplete	- Prepare a backup time - Determine the priority of features in order to eliminate some when time is limited
Low	Code generated by code generation tools is inefficient -> not significantly impacted	- Ignore

4.3.4 Configuration Management

Appropriate tools will be selected which provide a database of Change Requests and a controlled versioned repository of project artifacts.

All source code, test scripts, and data files are included in baselines. Documentation related to the source code is also included in the baseline, such as design documentation. All customer deliverable artifacts are included in the final baseline of the iteration, including executables.