BookLeaf

Software Development Plan

Version <2.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 22/10/2024 | 1.0 |  | Truong Thanh Toan |
| 10/11/2024 | 2.0 |  | Pham Thanh Vinh |
|  |  |  |  |
|  |  |  |  |

Table of Contents

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

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

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

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

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

[**3.**](#_heading=h.3dy6vkm) **Project Organization 4**

[*3.1*](#_heading=h.1t3h5sf) *Organizational Structure 4*

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

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

[*4.1*](#_heading=h.17dp8vu) *Project Estimates 4*

[*4.2*](#_heading=h.3rdcrjn) *Project Plan 4*

[4.2.1](#_heading=h.26in1rg) Phase and Iteration Plan 5

[4.2.2](#_heading=h.35nkun2) Releases 5

[4.2.3](#_heading=h.1ksv4uv) Project Schedule 5

[4.2.4](#_heading=h.1ci93xb) Project Resourcing 5

[*4.3*](#_heading=h.44sinio) *Project Monitoring and Control 5*

[4.3.1](#_heading=h.2jxsxqh) Reporting 5

[4.3.2](#_heading=h.3j2qqm3) Risk Management 5

[4.3.3](#_heading=h.4i7ojhp) Configuration Management 6

Software Development Plan

# Introduction

These days, the demand for learning and discovering new knowledge has been and will ever be more popular. However, to store and manage a personal library with physical books, articles, and publications is never an effortless task, in contrast to a nearly insurmountable one as the expense and space needed grow exponentially. Therefore, we decided to build a web application that can quench the thirst for storing, managing, and maintaining a library for book lovers with minimum costs and synchronization enabled. Thus came the idea of Bookleaf, an e-library for bookworms that require minimum cost, as well as storage space.

This Software Development Plan outlines the development approach, processes, and management strategies for BookLeaf, a web-based e-library application. BookLeaf addresses the challenges of digital library management by providing a platform for storing, reading, and managing digital books with advanced annotation features and community-driven functionalities. This document details the project organization, development methodology, timelines, and resource allocation across the 10-week development cycle. It serves as the primary reference for all stakeholders involved in the project's planning and execution.

# Project Overview

## Project Purpose, Scope, and Objectives

**Purpose**

***BookLeaf*** is an e-library that offers its users customizability. Users can upload their favorite and beloved books, articles, and publications. Additionally, they can share their library and download books in the form of either PDF or EPUB. While reading, users can edit documents or highlight memorable quotes within Bookleaf.

**Scope**

Web-based application with responsive design

Support for PDF and EPUB file formats

User authentication and library management

Reading features (highlighting, note-taking, bookmarks)

Social features (progress sharing, reading goals)

**Objectives**

Complete development within 10-week timeframe

Launch with zero budget using free-tier services

Enable seamless book uploading and organization

Provide customizable reading experience

## Assumptions and Constraints

***\* Constraints***

* + Zero-budget project(being dependent on **free APIs and resources**)
  + Fixed 10-week schedule
  + Books’ categorization and cloud storage
  + Cannot check for liability and legality of users and their uploaded
  + Limited to web platform only
  + Fixed team size of 4 members
  + Maximum 20 hours per week per team member
  + Part-time availability due to academic commitments

**\*Assumptions**

* Files uploaded are either PDF or EPUB
* Users’ intentions are to store, read and maintain an e-library on our cloud.
* Users are comfortable with web applications
* Users will provide their own content
* Users have: Stable internet connection (minimum 3G), Modern web browser

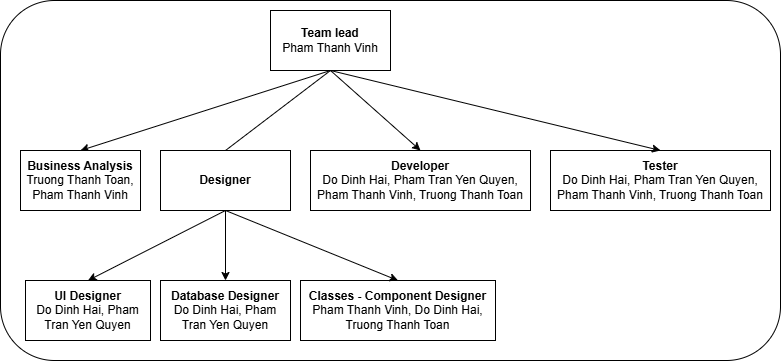
## Project Deliverables

List of documents and the corresponding delivery date for each sprint.

* **Sprint 1 (Inception):**
  + Software Development Plan (27/10/2024).
  + Vision Document (27/10/2024).
  + Project proposal (24/10/2024).
  + 2 weekly reports (15/10/2024 and 22/10/2024).
* **Sprint 2 (Elaboration):**
  + Use-case model and use-case specification (10/11/2024).
  + Database conceptual design: Entity Relationship Diagram (10/11/2024).
  + UI design: Color palette, landing page UI (10/11/2024).
    - ***Note: Coding can begin earlier than Sprint 4 once one of the UI is designed.***
  + 2 weekly reports (28/10/2024 and 4/11/2024).
* **Sprint 3 (Elaboration):**
  + UI design: key features’s screen (cont’d) (16/11/2024).
    - ***Note: Coding can begin earlier than Sprint 4 once one of the UI is designed.***
  + Based on the final ERD, choose a database system to go with (NoSQL or SQL), and then implement the data model design (deliver a ready-to-use database) (18/11/2024).
  + Test plan (Test Strategy, Test Cases, Testing Tools and Environments) (24/11/2024).
  + Software architecture document (24/11/2024).
  + 2 weekly reports (11/11/2024 and 18/11/2024).
  + UML diagram (18/11/2024).
  + Prototype (24/11/2024): prototypes will be delivered one by one so that coding phase can begin earlier once a prototype is produced.
  + Risk management document (24/11/2024).
* **Sprint 4 (Construction):**
  + Successfully built source code, including user interface and some well-performing basic features: beta release (8/12/2024)
  + Core Features Implementation: User Authentication, File Upload/Download, Reading Interface, Basic Social Features.
  + Test report (Unit Test, Integration Test) (8/12/2024).
  + 2 weekly reports (25/11/2024 and 2/12/2024).
* **Sprint 5 (Construction):**
  + Successfully built source code, including user interface and all features are well-performing: GA release (22/12/2024)
  + Complete User Manual (22/12/2024)
  + Test report (Final Test Reports) (22/12/2024).
  + 2 weekly reports (9/12/2024 and 16/12/2024).

# Project Organization

## Organizational Structure



## Roles and Responsibilities

|  |  |
| --- | --- |
| ***Person*** | ***Role*** |
| Đỗ Đình Hải (22127095) | UI Designer, Database Designer, Developer, Tester***\**** |
| Phạm Trần Yến Quyên (22127357) | UI Designer, Database Designer, Developer, Tester |
| Phạm Thanh Vinh (22127459) | Team Lead, Business Analysis, Classes - Component Designer, Developer, Tester |
| Trương Thanh Toàn (22127488) | Business Analysis, Classes - Component Designer, Developer, Tester |

# *\*Testers will test the functionalities which they did not contribute to the development.*

# Management Process

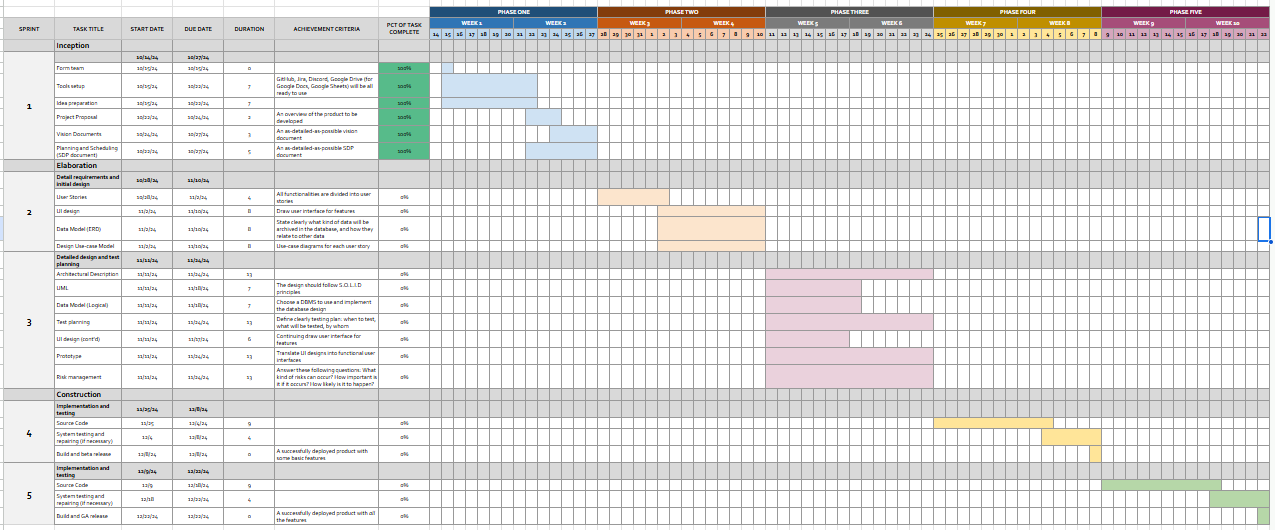
## Project Estimates

* Cost:
  + Less than 200.000 VND in total, may be used for non-free tools required for the development.
  + Approximately 6.000.000 VND/person, which is 24.000.000 VND in total, for the tuition fee of this course.
* Time: 12 weeks, including planning, preparing, designing, coding and testing; approximately 20 hours/week (potentially be more in later weeks).
* Staffs:
  + 4 members.
  + Necessary roles:
    - Team lead.
    - Business Analysis.
    - Designer (UI, Database, Class and Component).
    - Developer.
    - Tester.
* Re-Estimation
  + Schedule Re-Estimation
    - If team composition changes
    - After each sprint retrospective
  + Cost Re-Estimation
    - When free-tier limits are approached
    - If additional tools/services become necessary

## Project Plan

### Phase and Iteration Plan

Access this link ([Google Sheets](https://docs.google.com/spreadsheets/d/1n-J985vaShM67djaUq5lqBU2gfX8u_kDpyiQZJ5AWmY/edit?usp=sharing)), choose the sheet “Gantt” for a clearer view.



### Releases

* **Prototype** (ET: end of **sprint 3**): A working user interface.
* **Beta release** (ET: end of **sprint 4**): Upload, download books; highlight documents, sharing, edit. Addressing bugs, errors, conflicts, and stakeholders and testers’ feedback.
* **GA release** (ET: end of **sprint 5**): Fixing bugs, fine-tuning the application so it fits accordingly to stakeholders and testers, checking for critical errors, ready for releasing.

### Project Schedule

Project schedule following the above Gantt chart (in section 4.2.1).

## Project Monitoring and Control

### Reporting

* Weekly meeting:
  + Held online/offline on Monday or Tuesday (Tuesday is prefered).
  + Aims to review done work, assign new tasks, and review progress (what the assignees have done in the previous week, how it is going on, reassigning tasks if necessary, assigning new tasks for the week, adjusting the deadlines as needed,...).
* Weekly project status report:
  + Is given in the weekly meeting.
  + Oral report. Crucial information will be noted in the weekly meeting report.
  + Aims to review what has been done in the previous week, and outline the task for the upcoming week.
* Informal chats: can happen anytime, either via Discord (in #homework-help, #general, or creating a new channel if the chat is expected to last longer) or in person

### Risk Management

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***Risk ID*** | ***Risk Description*** | ***Probability*** | ***Impact*** | ***Risk Exposure*** | ***Priority*** | ***Mitigation Strategy or Contingency Plan*** |
| 1 | Lack of knowledge for the assigned task | Likely | Moderate | 3750 | 1 | Immediately study a crash course to make up for the lack of knowledge or having another team member with experience and knowledge assigned as a pair-programming duo / mentor - mentee. |
| 2 | Non-discovered **critical** bugs / errors | Occasional | Serious | 3750 | 1 | If discovered soon enough, then **immediately** fix it (them), otherwise, ensure that other components/functions do not suffer from the system failure. |
| 3 | Changes project's of requirements (functionalities or vision) | Seldom | Serious | 1875 | 2 | It depends on how the requirements are changed:   * Modification on requirements which were built or are building: if rebuild consumes a reasonable amount of time, then rebuild it; otherwise, we need to consider how important the modification is - rebuild it if it is really important, or ignore it is not. * Add new functionalities: adapt them if the functionalities are sufficiently suitable and the timing is appropriate. * Remove functionalities: try not to do this, and only remove if absolutely essential. |
| 4 | Disagreement between team members | Seldom | Serious | 1875 | 2 | Resolving conflict by democratically voting between team members. If ties, then continue to explain ideas until the team can converge into one idea. |
| 5 | Task estimation | Likely | Insignificant | 1875 | 2 | If time for the task was overestimated, immediately assign a new task to make up for the free time. Otherwise, assign more time and also might consider utilizing pair-programming to fasten the pace. |
| 6 | No reasonable priced APIs | Seldom | Moderate | 1250 | 3 | Either accept the price or reconstruct the APIs ourselves. Prioritize the option that is effective in cost, quality and time. |
| 7 | Hardware limitation | Seldom | Moderate | 1250 | 3 | Save up money for buying and installing new hardware devices (RAM, ROM,...). |
| 8 | Component conflicts | Unlikely | Catastrophic | 900 | 4 | Replace a conflict causing component by either a new component or a self-constructed one.  May be resolved by Component - Classes Designer. |
| 9 | Team members' health issue/absence | Unlikely | Moderate | 500 | 5 | If severe and hospitalizable illness occurred, then allow them to rest/absence and break-down and redistribute the remaining task for all other team members.  Moreover, each task is assigned to at least 2 members to make sure that there is a backup when someone is absent. |

### Configuration Management

* **Git** and **Github**: files exchanging, version control, collaboration, cloud-based code storage, CI/CD.
* **Google Drive** and **Confluence**: files and documents sharing (for Google Docs and Google Sheets).
* **Discord**: sharing resources, allowing collaboration, sharing ideas, calling and meeting, chatting. Allowing conference calls and online meetings, interviews.
* **Jira**: works management for all scales, from tasks to stories to epics. Creating Gantt charts, utilizing management philosophies such as Scrum, Kanban or Waterfall.
* **Moodle**: posting and submitting assignments.
* **Figma:** Use for UI/UX design of the application
* **Slack:** Use for discussions and interactions with teachers and TAs.