Group 01

KOOLB
Software Development Plan (Small Project)
Version 1.1

Revision History

Date	Version	Description	Author
12/11/22	1.0	The first version of Planning Document	Cát, Quỳnh
26/11/22	1.1	Fix Gantt Chart in Section 4.2	Cát, Quỳnh

Table of Contents

Software Development Plan (Small Project)

-

1. Introduction

1.1 Purpose

The purpose of this document is to gather all information necessary to control the project. It describes the development activities in terms of the phases and iterations required for project KOOLB

The following people use this document:

- 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, when they have to finish and what other activities they are dependent upon.
- The **stakeholders** use it to track the project

1.2 Scope

This Software Development Plan establishes the plan for software implementation, test, and qualification for the KoolB - an accommodation booking project. KoolB is being developed under the direction of Group01. Updates to this *Software Development Plan* will address future KoolB software upgrades. 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* identifies applicable policies, requirements, and standards for KoolB Project software development. It defines schedules, organization, resources, and processes to be followed for all software activities necessary to accomplish the development. This *Software Development Plan* contains no privacy considerations pertaining to the KoolB Project.

This *Software Development Plan* contains the following information:

Project Overview — provides an explanation of the project's aim, scope, and goals It also specifies the deliverables that are expected from the project.

Project Organization — outlines the project team's organizational structure

Management Process — provides the core categories for project management, as well as strategy and vision. The management process includes stages to plan, organize, staff, estimate cost and schedule, lead and control resources, and risk management.

2. Project Overview

2.1 Project Purpose, Scope, and Objectives

The project will create a rental-booking software that allows hosts to rent, renters to find places and rent.

2.2 Assumptions and Constraints

- This project will be available on android, ios and web (for admin and debugging)

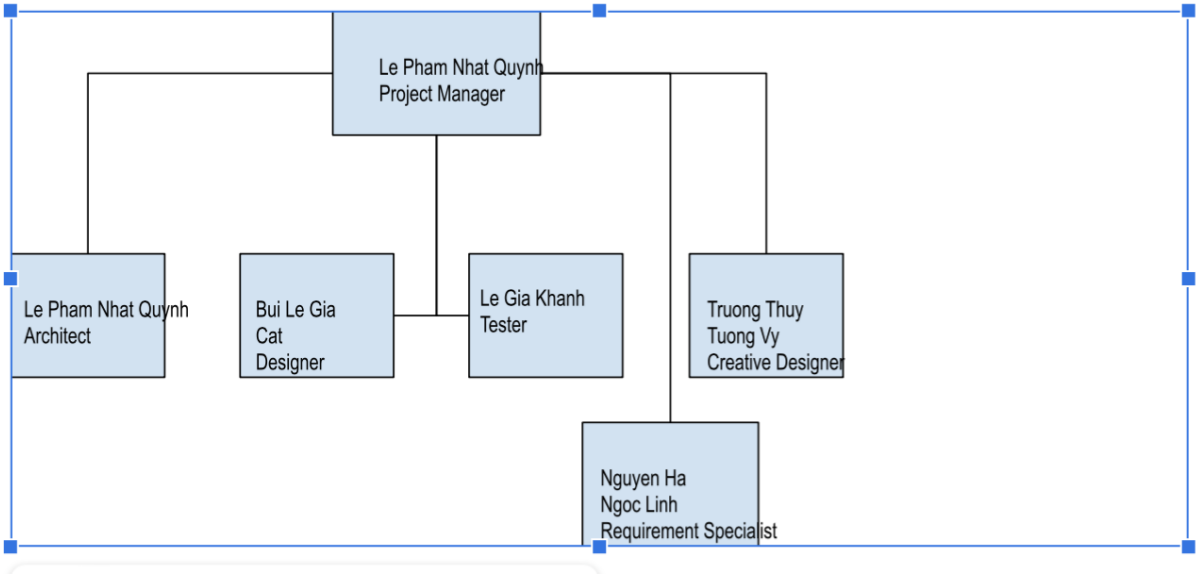
2.3 Project Deliverables

- Creative Design Briefs
- Navigation Map
- User Interface Prototype
- Use Case Survey
- Data Model
- Design Model
- Software Architecture Document
- Change Requests
- Test Summary

3. Project Organization

3.1 Organizational Structure

The project team for the Inception phases will be organized as follows:



3.2 Roles and Responsibilities

Role	**Responsibility**
Project Manager	The Project Manager distributes resources, establishes priorities, handles interactions with customers and users, and works to keep the project team on track. In addition, the Project Manager sets a set of procedures to assure the integrity and quality of project artifacts.
Architect	Throughout the project, the Architect directs and manages technical activities and artifacts. The Architect provides the general framework for each architectural vision: the decomposition of the view, the grouping of elements, and the interfaces between these primary groups.
Designer	The designer specifies one or more classes' responsibilities, actions, characteristics, and relationships and chooses how they should be changed to the implementation environment. Furthermore, the designer may be in charge of one or more design packages or design subsystems, as well as any classes owned by the packages or subsystems.
Creative Designer	The creative designer leads and coordinates the interface prototyping and design by capturing interface requirements, including usability requirements, building prototypes, involving other interface stakeholders, such as end users, in usability reviews and use testing sessions, and reviewing and providing appropriate feedback on the final implementation of the interface.
Tester	The Tester is in charge of carrying out testing, including test setup and execution, evaluating test execution and error recovery, reviewing test findings, and reporting found faults.

Requirements Specialist	The Requirements Specialist describes the Requirements element of one or more use cases and other supporting software requirements to capture the definition of a portion of the system's functionality. The Requirements Specialist is also in charge of a use-case package and ensuring its integrity.

4. Management Process

4.1 Project Estimates

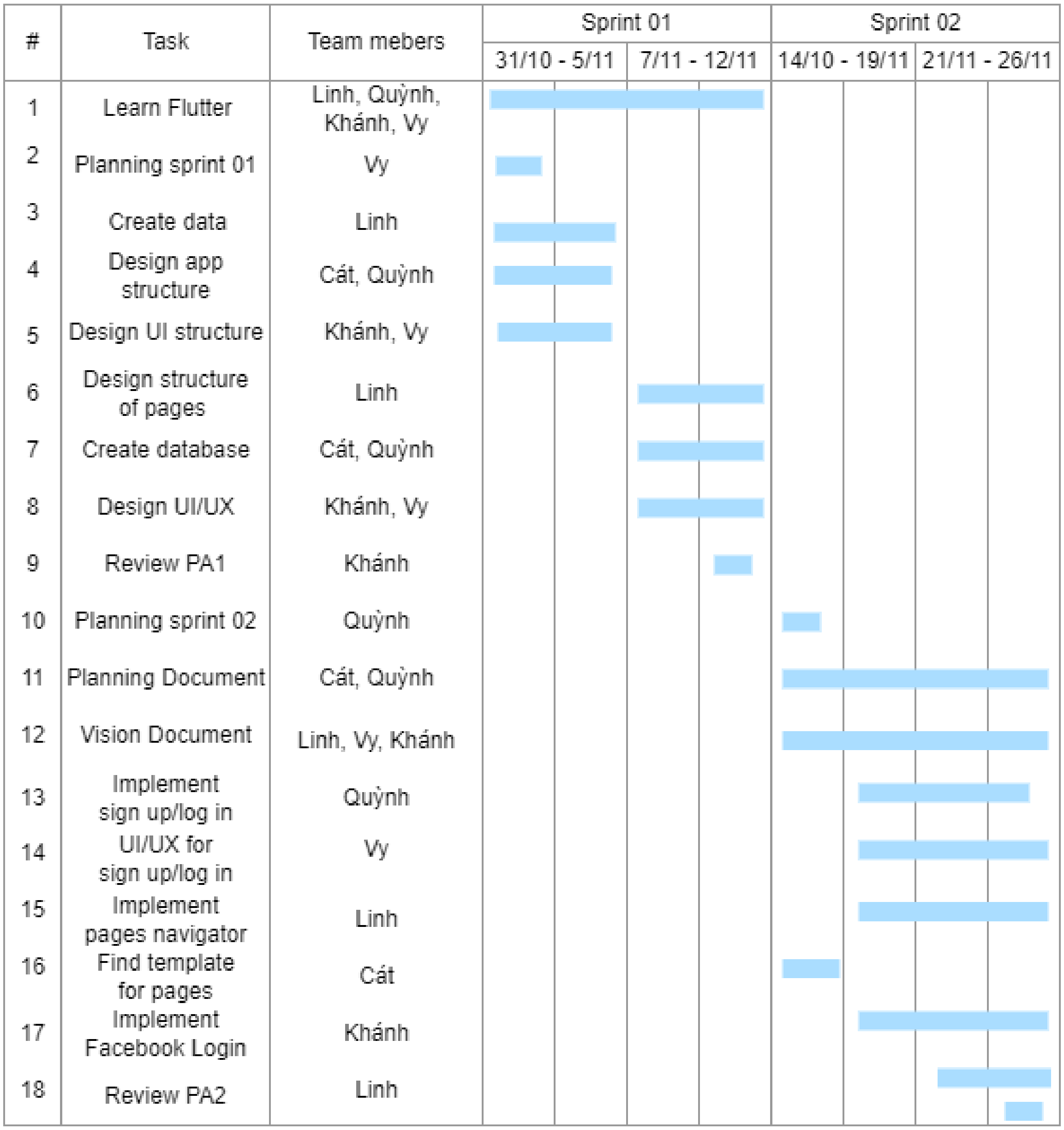
This project must be completed by January 14th, 2023.

The UI structure must be done before November 7th, 2022

4.2 Project Plan

4.2.1 Phase Plan

Gantt chart:



#	Task	Team mebers	Sprint 01		Sprint 02	
			31/10 - 5/11	7/11 - 12/11	14/10 - 19/11	21/11 - 26/11
1	Learn Flutter	Linh, Quỳnh, Khánh, Vy	■	■		
2	Planning sprint 01	Vy	■			
3	Create data	Linh	■			
4	Design app structure	Cát, Quỳnh	■			
5	Design UI structure	Khánh, Vy	■			
6	Design structure of pages	Linh		■		
7	Create database	Cát, Quỳnh		■		
8	Design UI/UX	Khánh, Vy		■		
9	Review PA1	Khánh		■		
10	Planning sprint 02	Quỳnh			■	
11	Planning Document	Cát, Quỳnh			■	■
12	Vision Document	Linh, Vy, Khánh			■	■
13	Implement sign up/log in	Quỳnh			■	■
14	UI/UX for sign up/log in	Vy			■	■
15	Implement pages navigator	Linh			■	■
16	Find template for pages	Cát			■	
17	Implement Facebook Login	Khánh			■	■
18	Review PA2	Linh				■ ■

Construction Phase: including sprint 03 and 04

- Sprint 03:
- Planning PA3
- Revising PA2
- Defining software architecture
- Drawing class diagrams
- Finishing implementing pages
- Reviewing PA3
- Sprint 04:
- Planning PA4
- Revising PA3
- Agreeing on UI structure and design

- Finishing implementing functions
- Reviewing PA4
- Transition Phase: including sprint 05:
- Planning PA5
- Revising PA4
- Planning test and test cases
- Getting ready presentation
- Reviewing PA5

Phase	Start	End
Inception Phase	31/10/2022	12/11/2022
Elaboration Phase	14/11/2022	26/11/2022
Construction Phase	28/11/2022	24/12/2022
Transition Phase	26/12/2022	13/01/2023

4.2.2 Iteration Objectives

Phase	Iteration	Description	Associated Milestones	Risks Addressed
Inception	Preliminary Iteration	Defines business model, product requirements, project plan, and business case.	Business Case Review	Clarifies user requirements up front. Develops realistic project plans and scope. Determines feasibility of project from a business point of view
Elaboration Phase	Develop Architectural Prototype	Completes analysis & design for all use cases. Develops the architectural	Architectural Prototype	Architectural issues clarified. Technical risks

		prototype.		mitigated. Early prototype for user review.
Construction Phase	C1 Iteration – Develop Beta	Implement and test use cases to provide the Beta Version.	Beta	All key features from a user and architectural prospective implemented in the Beta. User feedback prior to release of software.
	C2 Iteration – Develop initial Release	Implement and test remaining use cases, fix defects from Beta, and incorporate feedback from Beta. Develops the initial system.	Software	Software fully reviewed by user community. Product quality should be high. Defects minimized. Cost of quality reduced.
	C3 Iteration – Develop full Release	Incorporate enhancements and defects from initial release. Develops the full system.	Software	Quick release addresses customer satisfaction. All key functionality provided in System by full Release.
Transition phase	Software Release	Package, distribute, and install Release.	Software Released	

4.2.3 Releases

4.2.4 Project Schedule

4.2.5 Project Resourcing

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
Financial budget is not enough for the project (High)	The team may meet unexpected costs such as data storing service, hardware installation. This is serious if team members cannot afford it.	The team should learn more about the service or plan carefully before working on the project. These problems should be detected early so that the team can find an alternative solution.
Poor project management due to lack of experience (High)	The team leader is not experienced enough and decisive in assigning the task as well as reminding members, leading to the task not completed perfectly. Furthermore, disorientation can	Other team members should give timely constructive feedback to their leader so that they can improve and perform better.

	happen in the team.	
Development team cannot keep up with the technology (High)	The project might require a particular platform or framework that the team has not been proficient with. Thís may result in the project not being completed on time.	Team members should have a plan for training at the beginning of the course. Moreover, they can ask the teacher for advice about documents and tutorials.
Some members drop out of the project (High)	This happens when a team member is assigned a specific task, but they suddenly quit joining the project. This can make the remaining members not know how to complete his/her tasks.	The team should make sure that every member understands the tasks of other members so that anyone can undertake any task in case someone drops out of the team.
Database used in the system fails to meet expectations (data corruption) (High)	When the code writes incorrect data, or when something interrupts a write process, data of the system can be messed up and make the file corrupted.	It is necessary to take complete backup of all essential data.
The deadline may be delayed (Medium)	The development team are all beginners to software development, so they may not be familiar with the software development process. Therefore, some tasks may take more time than expected to complete.	Team members should read PAs in advance, and regularly discuss with the teacher about the project progress.
Framework is outdated or incompatible to modern devices/platforms (Medium)	Framework the team chooses to use does not have a sufficient level of development or is incompatible to different modern platforms. It may be hard or even impossible to customize the application and add new features.	The team should carefully choose a framework to integrate into their application. The problem should be detected soon so that the team can timely change to another framework.

Not enough physical device to test the application (Medium)	This happens when the disk does not have enough space for virtual devices. Besides, the application sometimes needs to be tested on a physical device for practical evaluation.	The team should plan the cross platform testing beforehand. Use cross-platform testing tools.
Wrong approach to the concepts proposed in project-related documents (Medium)	The team members may not deeply understand concepts proposed in project documents. Therefore, it is easy to encounter technical problems as well as design and structure in the process of developing	Make sure that every team member reads and understands project-related documents carefully. Contents in these documents should be examined to compare to and keep track of the project progress.
Poor communication between team members (low)	Team members who are friends of each other but they do not have much contact, which can lead to poor communication in the team	It is compulsory for every member to join all meetings of the team. Everyone is encouraged to express their opinions and problems. Everyone is welcome to ask any things about the projects as well as tasks if they feel confused.

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.