Introduction to Software Engineering

Project Proposal

The student team is required to complete the **Project Proposal** documentation for the assigned course project, following the attached template.

Software Engineering Department
Faculty of Information and Technology
University of Science

Table of Contents

Project Proposal

Objectives

This document focus on the following topics:

- ✔ Completing the Project Proposal document with the following sections:
  - Preliminary Problem Statement
  - Proposed Solution
  - Development Plan
  - Human Resources & Costing Plan
- ✔ Understanding the Project Proposal document.

1 Member Contribution Assessment

ID	Name	Contribution (%)	Signature
23127128	Nguyễn Thành Tiến	100%	
23127157	Nguyễn Hoàng Gia Bảo	100%	
23127184	Lê Phạm Kiều Duyên	100%	
23127396	Lương Linh Khôi	100%	

2 Preliminary Problem Statement

Project Title: *SweetieBakery – Food Delivery Website using MERN Stack*

1. Introduction and Background

In recent years, online food ordering and delivery systems have become an essential part of modern life. With the increasing demand for convenience, customers prefer to order food and bakery products online instead of visiting stores physically. However, most small and family-owned bakeries still rely on traditional methods such as phone calls or direct visits to receive and manage orders. This manual process can lead to errors in communication, difficulty in tracking orders, and inefficient management of customer data.

SweetieBakery aims to provide a modern, web-based solution for small bakery businesses to manage their products, receive customer orders, and track delivery processes in real-time. The system will allow customers to easily browse bakery products, place orders online, and monitor their order status step-by-step, while administrators (shop owners) can efficiently manage the store's operations through a single online platform.

2. Problem Description

Currently, many small bakeries face several challenges in their daily operations:

- Orders are received manually via phone or social media, leading to confusion or loss of information.
- There is no centralized system to manage product listings, stock, or customer orders.
- Customers cannot track their order progress after placing an order, which reduces satisfaction and trust.
- Business owners lack tools to monitor sales performance or analyze customer preferences.

To address these challenges, *SweetieBakery* will be developed as an integrated online ordering and management system. The proposed solution will automate the process of order handling,

product management, and delivery tracking to improve customer experience and operational efficiency.

3. Operating Environment

- **Frontend:** React.js (HTML5, CSS3, JavaScript ES6)
- **Backend:** Node.js with Express.js framework
- **Database:** MongoDB (NoSQL, hosted on MongoDB Atlas)
- **Server Environment:** Deployed on a cloud server (e.g., Render)
- **Supported Browsers:** Chrome, Firefox, Edge, and other HTML5-compatible browsers
- **Operating Systems:** Windows, macOS, Android, and iOS (via mobile browser)

5. Design and Implementation Constraints

- The project must be implemented using **MERN stack technologies** only.
- The database design will follow NoSQL schema modeling practices.
- The system should follow RESTful API design principles for backend communication.
- The frontend will be responsive and adhere to UI/UX best practices.
- Documentation will follow IEEE software documentation standards.
- The project will focus on functionality, scalability, and usability but will not include payment gateway integration in the initial phase (optional for future improvement).

3 Proposed Solution

3.1 Software

3.1.1. Features

Admin features:

No.	Expectations	Software Requirements
1	As an admin, I want to log in securely, so that only authorized staff can manage the system.	Admin authentication & role-based access control.
2	As an admin, I want to view a dashboard with sales summaries, so that I can monitor business performance.	Dashboard analytics and reporting
3	As an admin, I want to manage the product catalog, so that I can add, update, or remove cakes to keep the menu accurate and up to date.	Product management system
4	As an admin, I want to manage customer orders, so that I can view, update their status, and filter or search them efficiently for smooth order processing.	Order management system
5	As an admin, I want to manage user accounts, so that I can view user	User management system

	information and control account activity to keep the system secure and reliable.	
6	As an admin, I want to manage customer feedback and reviews, so that I can respond to users and remove inappropriate content to maintain a positive website environment.	Feedback and review management system
7	As an admin, I want to create promotional campaigns, so that I can boost sales.	Promotion & discount management
8	As an admin, I want to generate and export reports, so that I can analyze sales performance.	Report export (CSV/PDF)
9	As an admin, I want to back up data regularly, so that I can recover in case of system failure.	Database backup and restore
10	As an admin, I want to monitor website logs, so that I can track system activities or errors.	Log monitoring system
11	As an admin, I want the system should load the admin dashboard within 3 seconds.	Performance optimization (page load < 3 seconds)
12	As an admin, I want the admin interface must be intuitive and user-friendly, requiring no complex training.	User-friendly UI/UX design
13	As an admin, I want the system to be available 99.9% of the time (high availability), so that I can manage the	Availability

	business operations without unexpected downtime.	
14	As an admin, I want to ensure all customer data is handled according to privacy laws, so that the business avoids legal penalties.	Compliance (Data Privacy Regulations)

User features:

No.	Expectations	Software Requirements
1	As a user, I want to create and manage my account (register, log in/out, update personal information, reset password), so that I can securely access and control my personal profile.	User account and authentication system
2	As a user, I want to browse and search for cakes with detailed information, so that I can easily find products that match my preferences.	Product catalog display, search and filtering
3	As a user, I want to add, modify, and purchase items through a shopping cart and checkout process, so that I can place my orders conveniently.	Shopping cart and checkout system
4	As a user, I want to track, receive notifications, and cancel my orders if necessary, so that I stay informed about the order status.	Order tracking and notification system
5	As a user, I want to read and submit product reviews and ratings, so that I can share and learn from other customers' experiences.	Review and rating management system

6	As a user, I want to add items to a wishlist, so that I can save products for later.	Wishlist feature
7	As a user, I want to view ongoing promotions, so that I can buy cakes at discounted prices.	Promotion display module
8	As a user, I want to contact the bakery for questions, so that I can get support.	Contact form or chat system
9	As a user, I want to receive newsletters, so that I can get updates about new products or sales.	Newsletter subscription system
10	As a user, I want my personal and payment information to be protected, so that I can shop securely.	Use encryption for sensitive data.
11	As a user, I want the website to load quickly, so that I can browse cakes without delays.	Performance optimization (page load < 5 seconds)
12	As a user, I want the interface to be simple and easy to navigate, so that I can find what I need without confusion.	User-friendly UI/UX design
13	As a user, I want the system to save my actions automatically, so that I don't lose my shopping cart if I refresh the page.	Session persistence (localStorage, cache)
14	As a user, I want the website to work properly on different browsers, so that I can access it anywhere.	Cross-browser compatibility (Chrome, Firefox, Edge, Safari).

3.1.2. Software Architecture

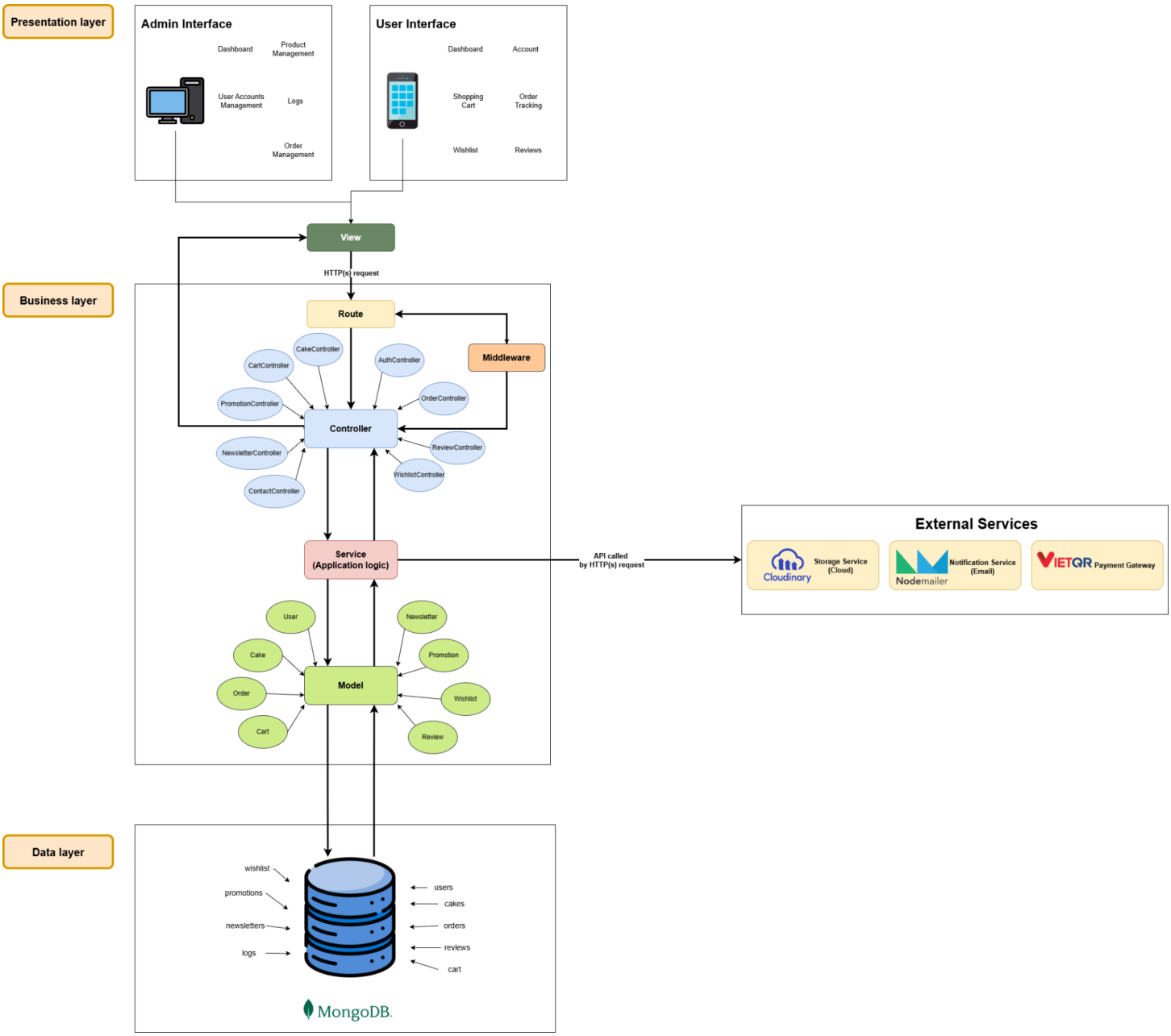


Diagram of the 3-tier system architecture using the MVC pattern, showing interactions between presentation, business, and data layers with external services integration.

3.2 Hardware

3.2.1. Client-Side Hardware Requirements

Component	Minimum Requirement	Recommended Requirement
Processor (CPU)	Dual-core 2.0 GHz	Quad-core 2.5 GHz or higher
Memory (RAM)	4 GB	8 GB or higher
Storage	500 MB free space	1 GB free space
Display	1280×720 resolution	Full HD (1920×1080)
Operating System	Windows 10 / macOS / Linux / Android / iOS	Latest OS version with updated browser
Web Browser	Chrome, Edge, Firefox, Safari (supporting HTML5 & ES6)	Latest version of Chrome (recommended)
Internet Connection	Stable 2 Mbps	Broadband ≥ 5 Mbps for smooth UI updates

3.2.2. Server-Side Hardware Requirements

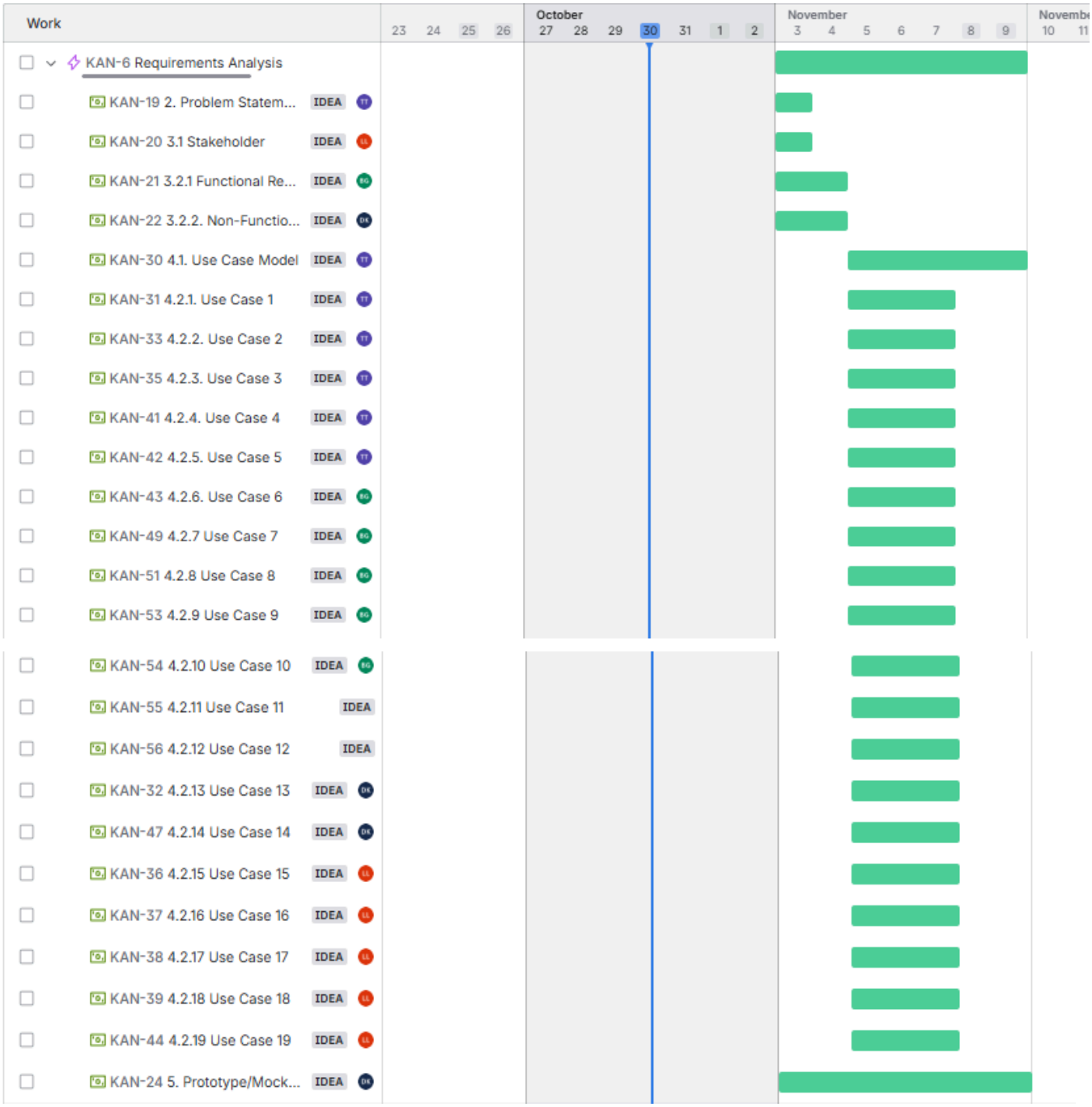
Component	Minimum Requirement	Recommended Requirement
Processor (CPU)	Quad-core 2.4 GHz	8-core 3.0 GHz (Virtual CPU)
Memory (RAM)	8 GB	16 GB or higher
Storage	100 GB HDD	200 GB SSD for faster I/O
Operating System	Ubuntu Server 20.04 / Windows Server 2019	Latest stable Linux distribution
Network	100 Mbps bandwidth	1 Gbps dedicated connection
Hosting Option	Localhost / VPS / Cloud (e.g., Render, Vercel, AWS EC2)	Cloud hosting with auto backup and SSL
Database Server	MongoDB Atlas or local MongoDB instance	Cloud-hosted MongoDB cluster (Replica Set)

3.2.3. Cloudinary Configuration

Component	Description
Cloud Provider	Cloudinary
Service Type	Cloud-based image storage and management
Account Type	Free tier (Up to 25,000 transformations / 2 GB storage)
Cloud Name	SweetieBakery
API Key / Secret	Provided by Cloudinary Dashboard (for secure server-side integration)
Integration Method	Using official cloudinary npm package and multer-storage-cloudinary for Node.js
Image Folder	/cakes/ – store all bakery product images
Usage Purpose	Store product images uploaded by admin, automatically optimize and deliver them via CDN links to frontend users

4 Development Plan

4.1 Requirements Analysis



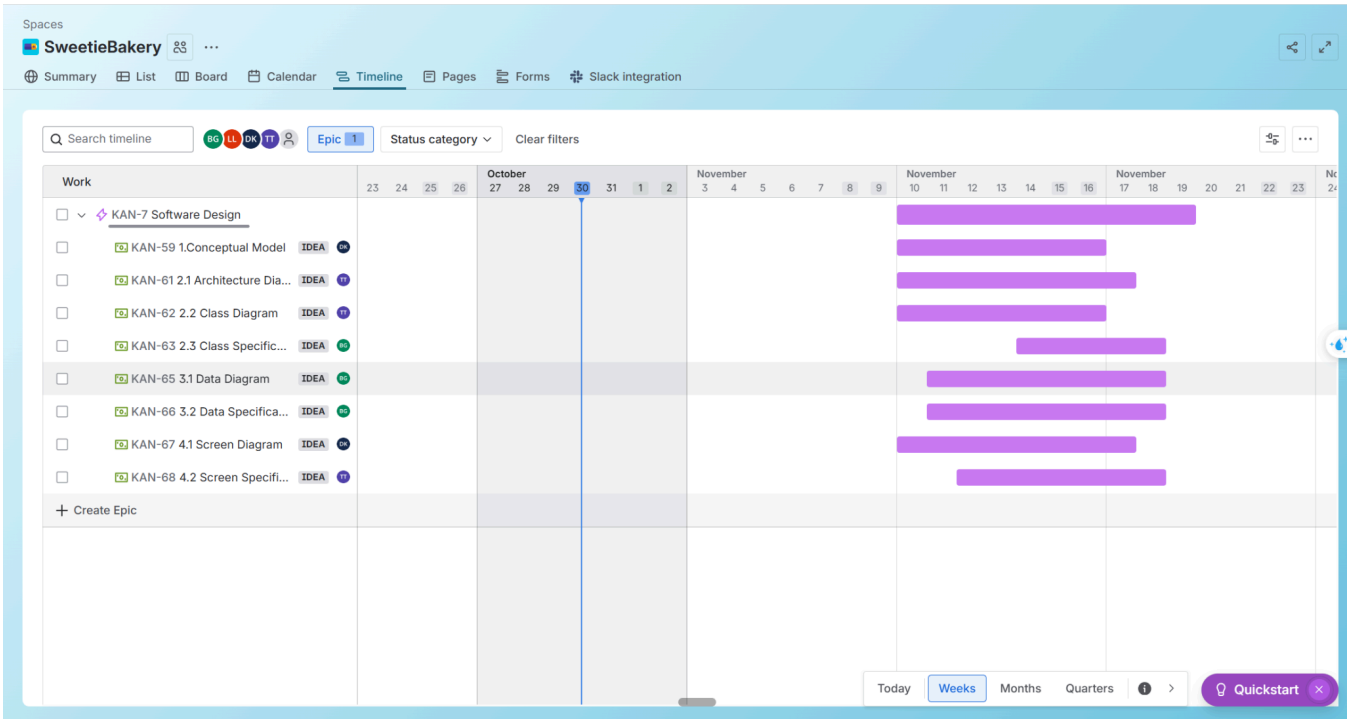
Schedule of future plans for each member during the requirements analysis phase.

Goal: To understand and document what the software needs to do

Key Deliverables:

- Software Requirements Specification (SRS):
  - Project Scope & Objectives.
  - Functional Requirements (What the system does).
  - Non-Functional Requirements (How well the system works - e.g., Speed, Security).
- Use Cases:
  - Descriptions of user interaction

4.2 Software Design



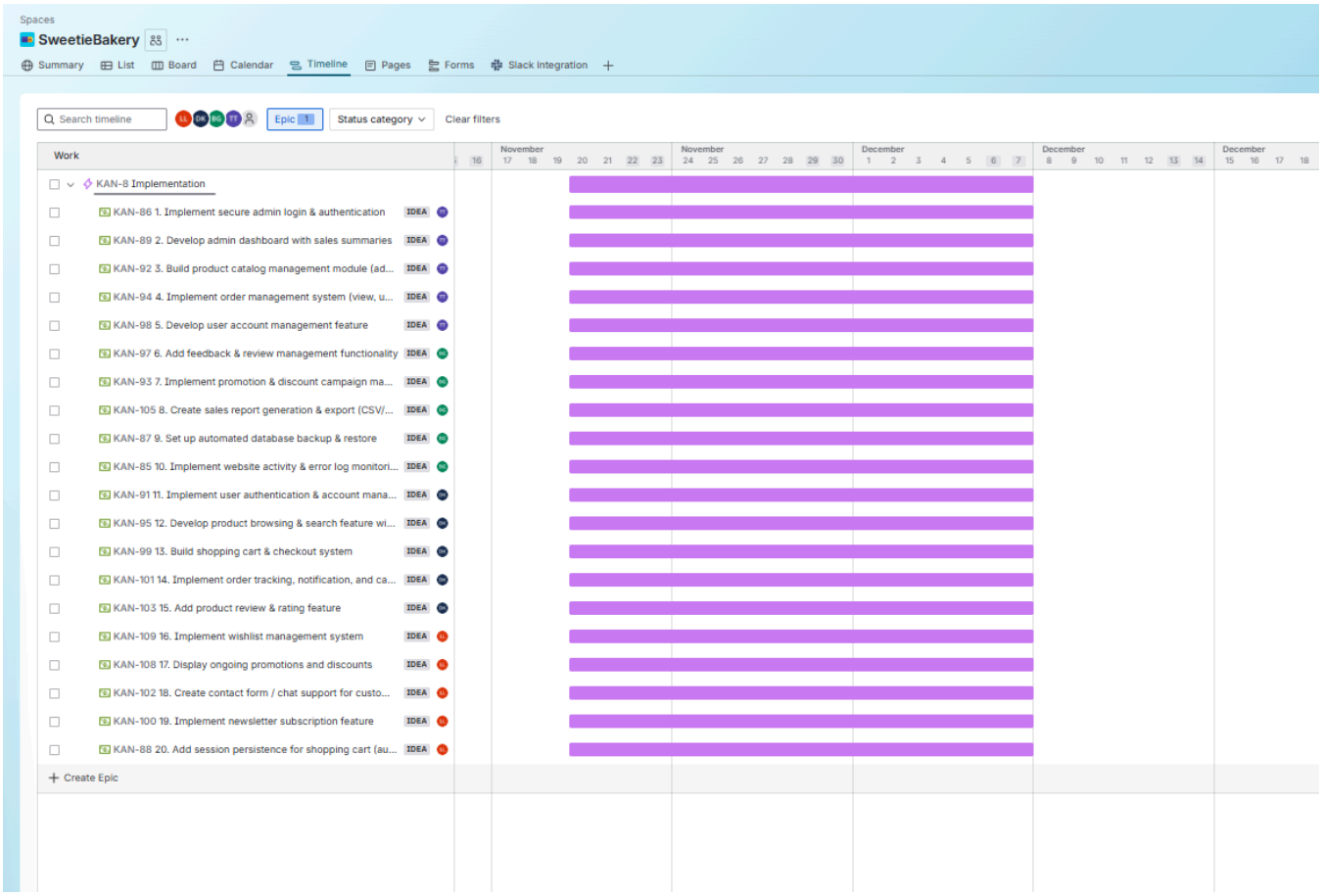
Schedule of future plans for each member during the software design phase.

Goal: To create the technical blueprint for how the software will be built.

Key Deliverables:

- Conceptual Model: (e.g., EER Diagram illustrating semantic entities)
- Architectural Design: (Decomposition Tree, Overall Architecture Diagram, Class Diagram)
- Class Specifications: (Detailed specifications for main classes)
- Data Design: (Data Diagram/Schema, Data Specifications/Table definitions)
- User Interface and User Experience Design: (Screen flow diagrams, Screen specifications, and Mockups)

4.3 Implementation



Schedule of future plans for each member during the implementation phase.

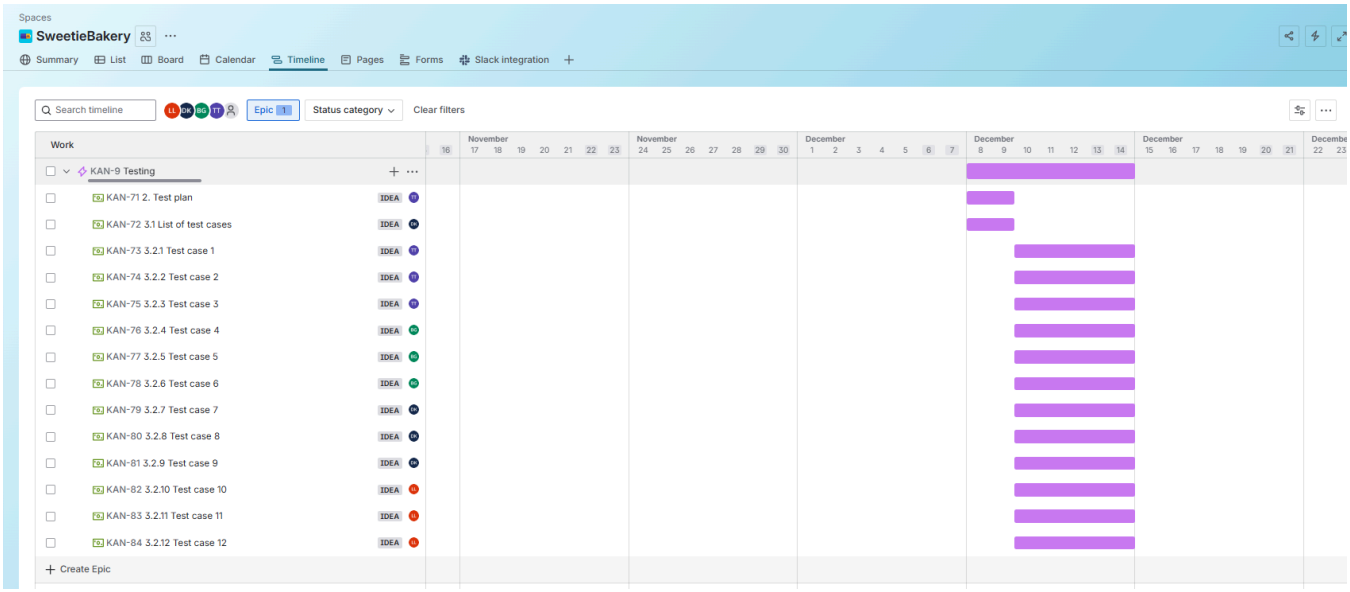
Goal: To write the code and build the software based on the design.

Key Deliverables:

- Source Code Repository: (e.g., GitHub link with commit history)

- Software Builds: (Runnable versions)
- Unit Tests

4.4 Testing



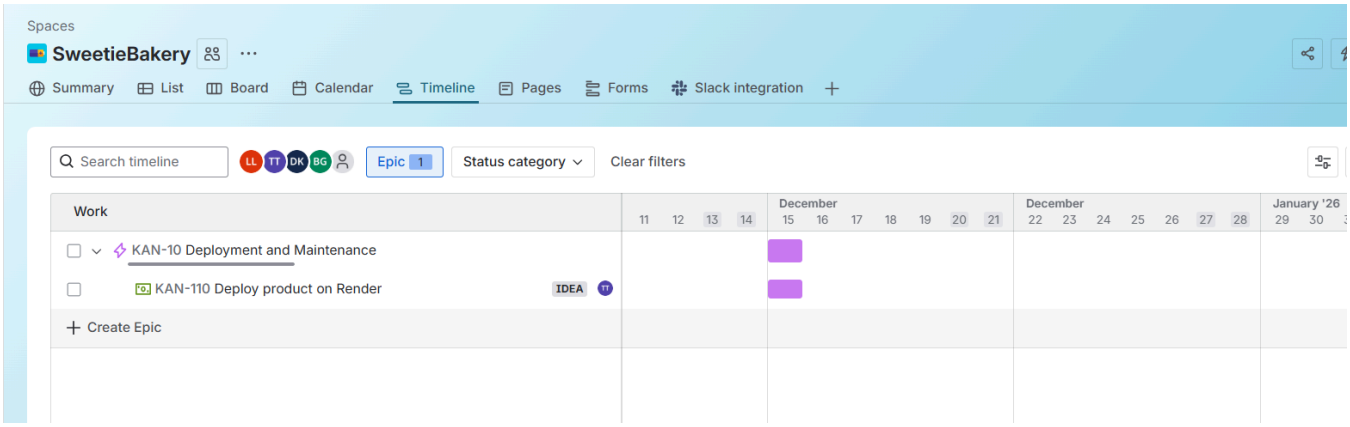
Schedule of future plans for each member during the testing phase.

Goal: To find and fix bugs, ensuring the software meets all requirements.

Key Deliverables:

- Test Plan: (Documenting testing scope, techniques, and test objects).
- Test Cases: (Including a list of test cases and detailed specifications for key cases).
- Bug Tracker / Report: (A report of all bugs found, their priority, and status).

4.5 Deployment and Maintenance



Schedule of future plans for each member during the deployment and maintenance phase.

5 Human Resources & Costing Plan

5.1. Human Resources

Role	Member	Responsibilities
Project Manager	Thành Tiến	• Plans, organizes, and monitors project progress. • Assigns tasks, manages schedule, and coordinates communication. • Handles risk management and ensures project goals are met.
Frontend Developer	Kiều Duyên Thành Tiến	• Develops user interface using React.js and related libraries. • Implements responsive design for web and mobile devices.

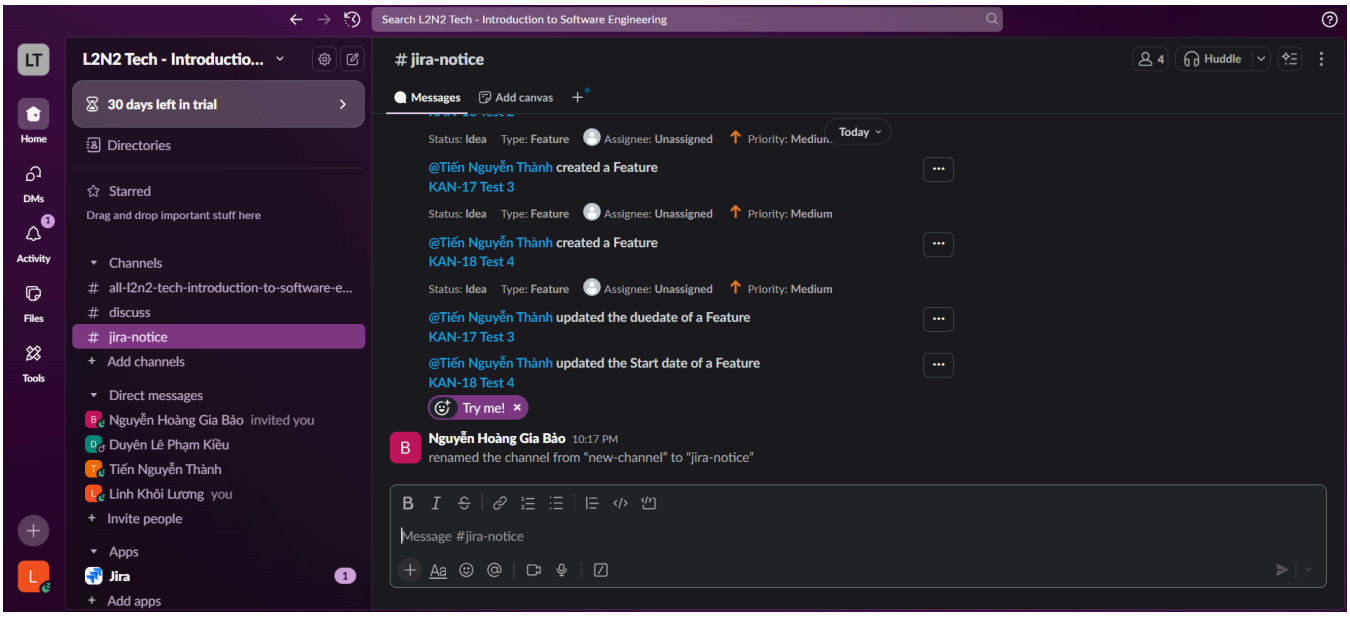
		• Integrates APIs and ensures smooth user interaction.
Backend Developer	Linh Khôi Gia Bảo	• Builds and maintains server-side logic using Node.js/Express. • Develops RESTful APIs and manages authentication. • Connects backend with MongoDB and ensures system security.
Testing	Linh Khôi	• Designs and executes test cases for functionality and performance. • Detects, reports, and verifies bugs or system issues. • Ensures final product meets quality standards.
UI/UX Designer	Kiều Duyên	• Designs website layout, components, and navigation flow. • Creates wireframes, mockups, and prototypes. • Ensures a consistent and user-friendly interface.
Database Engineer	Gia Bảo	• Designs database schema and relationships in MongoDB. • Manages data storage, indexing, and optimization. • Ensures data consistency, integrity, and backup.
Business Analyst	Kiều Duyên	• Gathers and documents user and system requirements. • Analyzes business processes and defines functional specifications. • Acts as a bridge between users and the development team.

5.2. Costing plan

Resource Type	Unit	Estimated Hours	Estimated Cost (for reference only)
Manpower (4 members)	Hours	≈ 4 × 1.5 x 40 ≈ 240 total hours	$0 (academic project)
Cloud Hosting (Render, MongoDB Atlas, Cloudinary)	*Free-tier*	-	*$0*
Development Tools (VS Code, GitHub, JIRA)	*Free*	-	*$0*
Total Estimated Cost	-	≈ 240 total hours	***$0 (time-based only)***

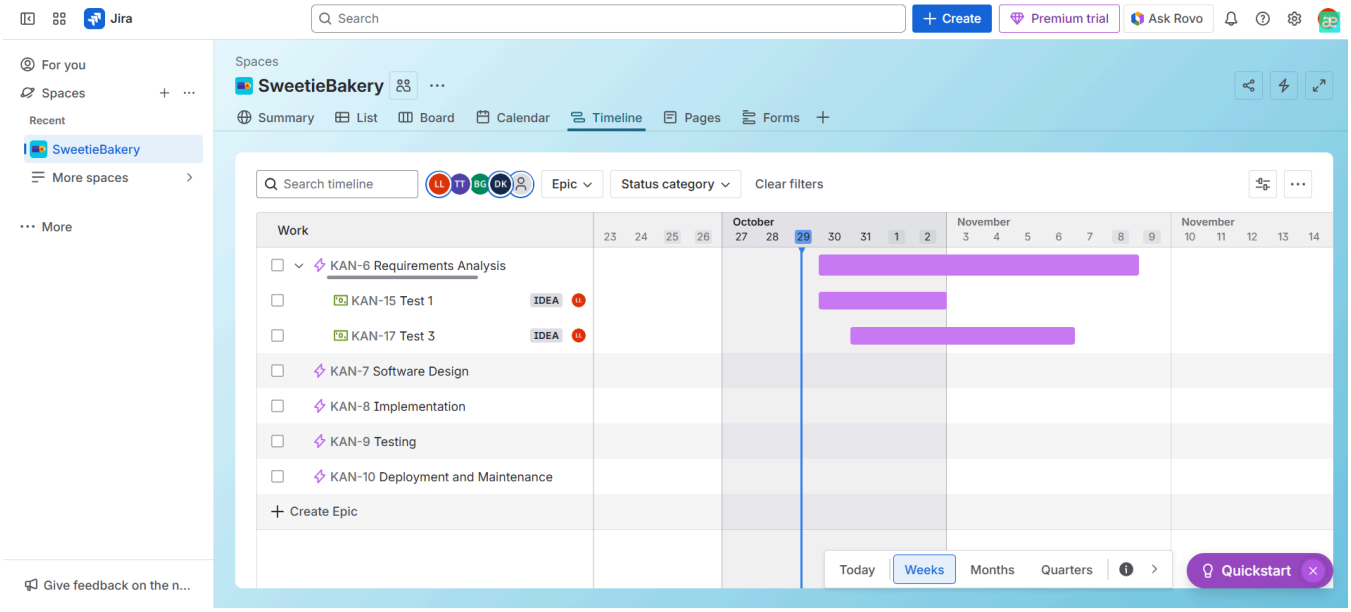
6 Tools setup

6.1. Slack



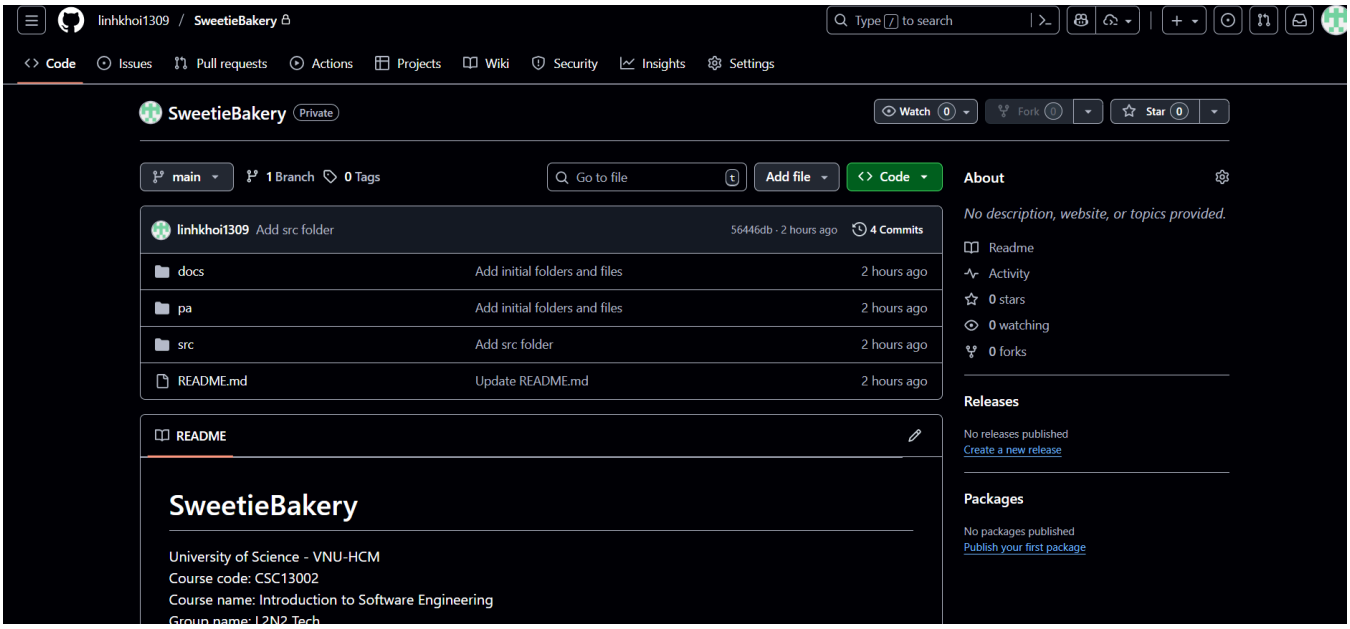
Slack workspace integrated with Jira for communicating among project members and getting latest information

6.2. JIRA

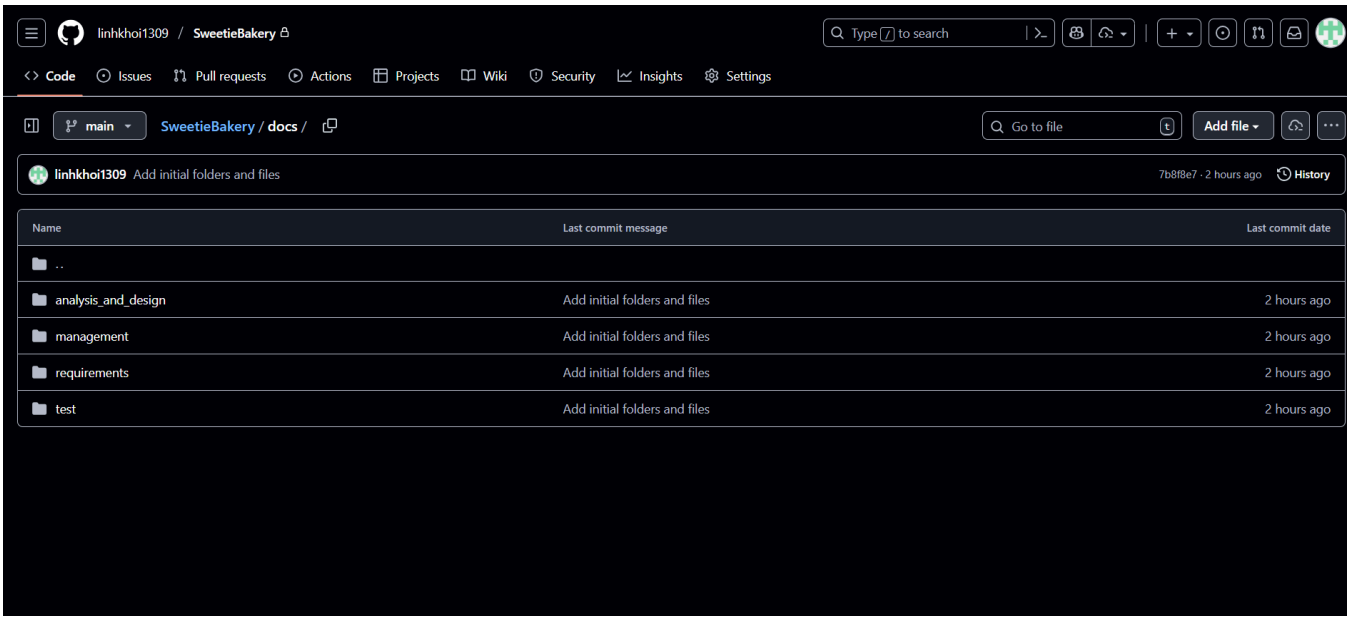


Jira project management tool for tracking project progress

6.3. Github



SweetieBakery Project folders structure



Documentation folders structure