

VIETNAM NATIONAL UNIVERSITY
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING



INTEGRATION PROJECT (CO3103)

Project:

NoteFlow

Instructor: Trương Tuấn Anh, CSE-HCMUT
Students: Lê Thị Phương Thảo - 2252757
Huỳnh Nguyễn Ngọc Anh - 2252022
Trần Hoàng Phúc - 2252647
Nguyễn Hoàng Vũ - 2252919
Trần Xuân Hảo - 2252191
Đoàn Anh Quang - 2252666
Nguyễn Dương Khánh Tâm - 2252733

HO CHI MINH CITY, JANUARY 2025



Contents

1 Revision History	4
2 Requirement Elicitation	6
2.1 Description	6
2.2 Domain Context	6
2.3 Stakeholders and Needs	7
2.4.1 General Users	7
2.4.2 Organizations	7
2.4.3 Developers	8
2.4.4 Administrator	8
2.5 Functional Requirements	8
2.5.1 General Users	8
2.5.2 Single Users	9
2.5.3 Organizations	9
2.5.4 Developer	9
2.5.5 Administrator	10
2.6 Non- Functional Requirements	10
2.6.1 Performance	10
2.6.2 Reliability	10
2.6.3 Usability and Interface	10
2.6.4 Maintainability	11
2.6.5 Scalability	11
2.6.6 Security	11
3 Use case	12
3.1 Use case diagram for the whole system	12
3.2 Use case: Login & Sign-up module	12
3.3 Use case: Take Note module	14
3.3.1 Create Note	15
3.3.2 Edit Note	16
3.3.3 Delete Note	17
3.3.4 Share Note	18
3.4 Use case: Plan module	19
3.5 Use case: Team project	21
3.6 Use case: Chat module	22
3.6.1 Chat	23
3.6.2 Send Message	24
3.6.3 Receive Message	24
3.6.4 Search Message	25
3.6.5 View Message History	25
3.6.6 View Shared Content	26
4 System modeling	27
4.1 Activity Diagram	27
4.2 Sequence Diagram	32
4.2.1 Upload Document	32
4.2.2 Take Note	35



4.2.3	Plan	36
4.3	Class Diagram	38
4.4	User Interface	39
4.5	Welcome Page	40
4.6	Home Page	43
4.7	Take Notes	44
4.8	Plan & Add Tasks	45
5	Architecture Design	46
5.1	Presentation Strategy	46
5.1.1	Front-end Library and Framework	46
5.1.2	Responsive Design	46
5.1.3	User-Friendly Features	46
5.2	Data Storage Approach	47
5.3	API Management	48
5.3.1	Authorization and Authentication APIs	48
5.3.2	Task Management APIs	48
5.3.3	Note-taking and Collaboration APIs	48
5.3.4	Calendar and Scheduling APIs	48
5.3.5	Pomodoro Timer and Mini-Games APIs	48
5.3.6	System Control APIs	48
5.3.7	Integration with External Services	49
5.4	Component Diagram	49
6	Implementation - Sprint 1 (4)	51
6.1	Set up repository on Github	51
6.2	Steps to Run the Project	51
6.3	Usability test with the User Interface	51
6.3.1	Recruit participants/ testers	52
6.3.2	Define tasks	52
6.3.3	Define test strategy (qualitative vs. quantitative, remote vs. in person)	53
6.3.4	Conduct the test	53
6.3.5	Document the feedback from testers	55
7	Task 5: Implementation - Sprint 2	64
7.1	Set Up	64
7.2	Welcome Page	64
7.3	Login and Sign up	64
7.4	Welcome Page	64
7.4.1	Signup	64
7.4.2	Login	65
7.5	Homepage	65
7.6	Plan Component	65
7.6.1	Make Plan	66
7.6.2	Today Plan	66
7.6.3	Upcoming Plan	67
7.7	Note Component	67
7.7.1	Note Interface	68
7.7.2	Share Note	68



7.8 Settings	69
7.9 Lesson Learned	69



1 Revision History

Week	Tasks and Milestones
Week 35	Project Introduction <ul style="list-style-type: none">- Defined the scope and objectives for the Smart Printing Service, including required tasks and the final deliverable for demonstration.- Assigned roles within the team:<ul style="list-style-type: none">• Thao & Ngoc Anh & Hao: Searched for previous years' reports and assignments for reference.• Quang, Vu: Organized the team meeting for project introduction and task assignment.• Phuc: Researched the project structure and suitable programming languages.
Week 36	Technology Research and Decision <ul style="list-style-type: none">- Researched potential programming languages and tools:<ul style="list-style-type: none">• Thao, Tam, Vu, Phuc : Frontend frameworks (e.g., React, Vue.js).• Quang & Hao: Backend technologies (e.g., Flask, Django, Node.js).- Conducted a team meeting to evaluate and finalize the technology stack.
Week 37	Requirement Elicitation (Task 1.1, 1.2) <ul style="list-style-type: none">- Defined domain and stakeholders:<ul style="list-style-type: none">• Hao, Ngoc Anh: Wrote the domain context.• Vu, Phuc: Documented stakeholders.• Quang: Wrote stakeholders and current needs.- Drafted functional and non-functional requirements:<ul style="list-style-type: none">• Thao: Defined non-functional requirements.• Vu: Defined functional requirements.- Team: Met to clarify and review errors.
Week 38, 39	Use-case Diagrams (Task 1.3) <ul style="list-style-type: none">- Team: Assigned tasks and drafted initial use-case diagrams.- Thao, Vu, Phuc, Quang, Ngoc Anh, Hao: Worked on creating use-case diagrams and corresponding task descriptions.- Conducted a team meeting to review and refine the use-case diagrams.
Week 40, 41	System Modeling (Part 1) <ul style="list-style-type: none">- Hao: Developed sequence diagrams for the system architecture.- Thao: Developed class diagrams for the system architecture.- Phuc: Developed component diagrams for the system architecture.- Vu, Ngoc Anh, Vu: Designed Figma UIs for two modules: Students and SPSO.- Team: Met to discuss the user interface. System Modeling (Part 2) <ul style="list-style-type: none">- Finalized all diagrams and system design documents.- Conducted a team meeting to review progress and align on MVP1 deliverables.
Week 42	Midterm Break <ul style="list-style-type: none">- Checked and completed tasks.- Wrote descriptions for diagrams and user interface designs (Figma).- Thao established an online Git repository for version control.
Week 43, 44	Refinement of Designs and Feedback Incorporation, Start Coding



Week	Tasks and Milestones
	<ul style="list-style-type: none">- Reviewed updated deliverables, including revised Figma designs and diagrams. Collected feedback from instructors, adjusted errors, and rewrote descriptions.- Tam, Phuc & Vu: Updated Figma designs to improve usability and started coding the frontend using React. Divided tasks for pages and usability.- Ngoc Anh & Quang, Hao, Thao: Coded the database and backend.- Team: Met for clarification
Week 45, 46	Initial Development (Backend) <ul style="list-style-type: none">- Quang, Ngoc Anh: Continued developing backend functionalities.- Vu: Set up the frontend structure and connected it to backend APIs.- Tam, Vu, Phuc & Thao: Implemented core frontend features based on Figma designs.- Thao & Tam: Conducted usability tests on the user interface.- Held a team meeting to review progress and debug issues
Week 47, 48	Final Development and Testing <ul style="list-style-type: none">- Tested and validated MVP2 with functional backend and frontend.- Thao & Ngoc Anh: Finalized frontend development.- Quang, Ngoc Anh & Hao: Optimized backend processes for efficiency.- Thao: Compiled test results and documented fixes.- Team: Prepared and rehearsed the project report. Assigned tasks and practiced demonstration.
Week 49, 50	Submission <ul style="list-style-type: none">- Team: Finalized and rehearsed the project presentation, assigned tasks, and practiced the demonstration.- Submission: Checked the report, combined the code, and submitted the final version.



2 Requirement Elicitation

2.1 Description

Managing projects, notes, and tasks can be challenging for students and professionals, especially when using multiple tools that lack integration. This approach often leads to inefficiencies, confusion, and missed deadlines. The absence of a unified system for organizing knowledge, tracking progress, and collaborating with others only increases these frustrations.

NoteFlow is a web-based productivity platform designed to empower users to effectively manage their tasks, schedules, and monitor their progress.

Inspired by the popular note-taking and project management tool *Notion*, our platform offers a comprehensive suite of features to enhance productivity and streamline workflow. Some key features of the website will be:

- **Note-taking:** Users can create and organize notes, documents, and ideas in a flexible and customizable format.
- **Task management:** The platform provides robust task management capabilities, including task creation, prioritization, deadlines, and progress tracking.
- **Calendar and Scheduling:** Users can plan their days, weeks, and months by creating and managing events and appointments.
- **Performance tracking:** The platform calculates user performance based on completed tasks and set goals, providing valuable insights into productivity levels.
- **Mini-Games and Pomodoro Sessions:** To promote relaxation and focus, the platform incorporates mini-games and Pomodoro timers, allowing users to take breaks and improve concentration.

NoteFlow is primarily aimed at individuals and teams who require a versatile and efficient tool for managing their daily tasks and projects. This includes students, professionals, and anyone seeking to improve their productivity and organization.

Finally, we have carefully selected the latest technologies to build a website that is both visually appealing and highly responsive, ensuring a positive user experience.

- **Collaboration:** Enables users to collaborate on notes and tasks with team members or colleagues.
- **Integration:** Integrates with other popular social media applications (e.g., Google).

2.2 Domain Context

NoteFlow is an app designed to help general users, organizations organize their tasks, make plans, and take notes with various customizable designs and extra features. The key entities include General Users, Organizations (Companies, Industrial Corps, etc.), Developers. The application allows us to organize their tasks, even update their timetable, keep track of activities and homework from school or tasks from the company with the addition of taking notes, planning on many different templates. And also help teachers bring out exercises, lecture notes or tasks more easily. Organizations can also use this app to manage and arrange meetings, working plans for their employees and synchronize the usage of an app in a whole company for better supervision and optimization.

The app includes many features, helping users manage their time and work more easily. Documents, spreadsheets, pictures, graphs, etc. are all integrated in the app to ensure every users' need. Furthermore, it can filter tasks created according to any filter constraints made by users and also share it with others, making the mutual plans more accessible and manageable. The database will ensure efficient management of resources while making it easy for developers to manage and maintain.



With **NoteFlow**, users benefit from an intuitive and efficient platform that enhances productivity, reduces the need for multiple tools.

2.3 Stakeholders and Needs

The **NoteFlow** gives services and is maintained to support users. Therefore, there are various stakeholders in this system corresponding to their needs.

The first stakeholder is the General Personal Users. They are the main users of the system: making weekly, monthly plans, tracking homeworks given by teachers, etc. They can also view how productive they are based on their completion of the tasks they created to motivate their studies and work. Moreover, they can share tasks, messages which means team working is easier and also, through the colorful and customizable templates, it can also be more fun and motivational. The templates could be a great help for people to design their timetable however they want, creating a better feeling of going to school or doing extracurricular activities and working. They can also play games on the app.

The second stakeholder is the Organisations. The organizations need to supervise the employees and make sure that meetings must go according to plans. The features provided for users of the organization are: creating tasks and sending notifications for the tasks for the employees, planning meetings more efficiently and also provides an overview of the tasks' progress, tracking and notifying if anything goes wrong or not according to plan. For example, teachers can also use this app to create homework and spread them out to students more efficiently, also planning extra classes more easily.

The third stakeholder is the Developers. The developers will be in charge of maintaining the app, fixing bugs, receiving feedback, and providing a better user experience. Should any problems arise regarding the app malfunctioning or not working properly, the developer will be responsible for it. This ensures that the app always works smoothly and up to date with users' needs.

The fourth stakeholder is the Administrator. The administrator plays a critical role in ensuring the system's availability and security. The administrator will ensure that only authorized users can access the system, and also can extract reports from the app and supervise the overall state of the usage of the app. The administrator is in charge of the app's stability and credibility.

2.4 Benefits of the System

2.4.1 General Users

For general personal users, the system helps them: manage tasks easily by allowing them to create, organize, and prioritize their daily, weekly, or monthly plans. It tracks productivity with evaluations to check daily, weekly, and monthly efficiency, keeping users motivated. The app's customization features enable users to personalize their timetables and plans with templates, enhancing engagement and making the planning process more enjoyable. Users can share tasks and messages with others, facilitating teamwork and group projects. This will make collaboration much simpler. Additionally, the inclusion of basic integrated games provides a source of relaxation, promoting a balanced lifestyle. (Integration with Google accounts allows for seamless login and synchronization, adding convenience to the user experience.) Daily reminders and advanced editing tools like text formatting, tables, and image insertion help users stay on track with their tasks and deadlines.

2.4.2 Organizations

For organizations, the system streamlines workflow management by enabling them to create and assign tasks to employees with automatic notifications. This ensures that all team members are aware of their responsibilities and deadlines, enhancing overall productivity. The app also helps plan meetings with calendar features, keeping all participants informed and schedules aligned. Organizations can track task



progress which display the productivity levels of the group. They can also highlight areas that may require attention. Collaboration tools allow teams to share plans and documents easily, improving cooperation and efficiency within the organization. Customizable templates help maintain brand consistency and standardize processes across different departments. In educational settings, teachers can use the app to assign homework and plan classes more efficiently, improving communication and engagement with students.

2.4.3 Developers

Developers benefit from the system by receiving and analyzing user feedback, which they can use to continually improve the app's functionality and user experience. They are responsible for fixing bugs and implementing updates, ensuring that the app runs smoothly and remains reliable for all users. By focusing on enhancing the user interface and overall experience, developers contribute to increased user satisfaction and retention. Working on a diverse range of features, such as integration with other platforms, collaboration tools, and game development, offers developers opportunities for professional growth and skill enhancement. Maintaining a high-quality app not only boosts the product's competitiveness in the market but also contributes to the developers' reputation and potential career advancement.

2.4.4 Administrator

One of the primary advantages is the ability to manage system security effectively; administrators can enforce strict access controls, ensuring that only authorized users can access sensitive data and functionalities. Additionally, administrators uphold the app's credibility by ensuring compliance with industry standards, legal regulations, and organizational policies. The system also facilitates effective support and supervision by allowing administrators to oversee user feedback, coordinate with developers on resolving bugs, and manage communications related to system updates or maintenance schedules.

2.5 Functional Requirements

2.5.1 General Users

- Users can create new accounts using a valid email address and password. Each email address can only be used to create one account.
- The system must require users to authenticate using an email and password.
- A user can reset their password if they forget via the registered email.
- Users can schedule tasks and appointments, receive notifications and reminders.
- Users can choose how soon from the task or appointment expected starting time they will be sent notifications and reminders.
- Users can customize notification and reminder settings for tasks and appointments, including the frequency and time before the event starts when notifications should be sent. Users can also turn off and turn on the reminders at any time.
- Users can search for notes, tasks, and events using keywords or tags.
- Users can share their notes, plans, and documents with other users, granting them specific permissions to view, edit, or copy the shared content. Users can control who has access to their shared items and what actions they are allowed to perform.



2.5.2 Single Users

- Users can create, edit, and delete personal notes and documents.
- Users can prioritize tasks using a system such as labels, colors, or numerical values.
- The platform will allow users to mark tasks as "Completed," "On-going," or "Not Started."
- At the end of the day, the platform will calculate the user's productivity based on the ratio of daily completed tasks to total daily tasks. This daily productivity score will be saved for historical reference.

2.5.3 Organizations

- The website must enable organizations to create tasks for employees, ensuring clarity in responsibilities.
- Organizations can send notifications regarding assigned tasks, deadlines, and important updates to employees.
- The organization administrator must have access to an overview dashboard that displays the progress of tasks assigned to employees including completion rates, deadlines, and any pending tasks.
- The system must send alerts to organization administrators or project owners if tasks are not progressing according to plan.

2.5.4 Developer

- The developer team must use version control systems to manage code changes and collaboration among developers.
- The system must notify the developer team in real-time when there is a malfunction or maintenance requirement of the system.
- The developer team must provide comprehensive API documentation for developers to integrate with the platform.
- Whenever there is a bug or error reported by a user, the developer team must ask for any additional details that might help in reproducing the bug.
- The developer team must use an issue tracking system to log, track, and manage bugs, feature requests, and other tasks.
- The developer team must ensure that there are regular backups of the system and a recovery plan in place to restore data and functionality in case of a failure.
- When fixing errors or updating the system, the developer team must deploy the fix or the changes to a staging environment first to ensure it works in a production-like setting.
- The system must send alerts to the security team for suspicious activities (e.g., multiple failed login attempts).
- The login system must have measures in place to prevent brute-force attacks and password guessing. The system will limit the number of incorrect password attempts for a given account within a specific time frame and reset after 10 minutes.
- The login system must enforce strict password policies, including password complexity requirements.



2.5.5 Administrator

- The system must allow administrators to grant or revoke user permissions dynamically.
- The administrator must ensure that only authorized users can access the system, managing user authentication and permissions.
- The administrator must be able to extract detailed reports from the app, including usage statistics and user activity.
- The system must provide the administrator with real-time alerts about any issues, such as system malfunctions or security breaches.
- The administrator must be responsible for maintaining the credibility and reliability of the system, ensuring consistent performance.
- The administrator must manage backups and data recovery processes to prevent data loss and ensure the system's integrity.
- The administrator must oversee the app's security and monitoring access control.
- The system must provide the administrator with real-time alerts about any issues, such as system malfunctions or security breaches.

2.6 Non- Functional Requirements

2.6.1 Performance

- **Response Time:** The system should respond to user inputs within 1 second for most operations.
- **Data Sync Speed:** Synchronization of data between devices should happen within 1-2 minutes.
- **Scalability:** The platform should be able to handle an increasing number of users, pages, and content without a significant decrease in performance (100 users at the same time).

2.6.2 Reliability

- **Data Integrity:** The system should ensure that no data is lost or corrupted during operations like saving or syncing.
- **Backup and Recovery:** The system should support disaster recovery plans, allowing for rapid recovery in the event of system failure.

2.6.3 Usability and Interface

- **Responsiveness:** The UI should be responsive across different devices (desktop, mobile, tablet) and screen sizes.
- **Ease of Use:** New users should be able to understand and use the core features within 30 minutes of onboarding.
- **Cross-Platform Support:** The system must be compatible with popular operating systems (Windows, macOS, Linux, iOS, Android).
- **Personalization Options:** The UI should allow users to customize the layout, theme (e.g., dark-/light mode), and other visual settings to match their preferences.



- **Languages:** The system should support multiple languages; currently, English and Vietnamese, with the potential to add more languages in the future.
- **User Profiles:** Allow users to save their personalized settings (colors, fonts, background) within their profile, set their avatar so their preferences are applied across devices.
- **Animation:** When users complete tasks daily, weekly, or monthly, a small fireworks display could be triggered in the background to celebrate their accomplishment.

2.6.4 Maintainability

- **Code Modularity:** The system architecture should allow for easy updates and bug fixes without disrupting the user experience.
- **Documentation:** The system should have comprehensive documentation to support developers, users, and administrators.

2.6.5 Scalability

- **Database Scaling:** The database must be capable of handling hundreds of users and data with minimal performance degradation.
- **Horizontal Scaling:** The system should be capable of distributing workloads across multiple servers to handle increased traffic or data load.

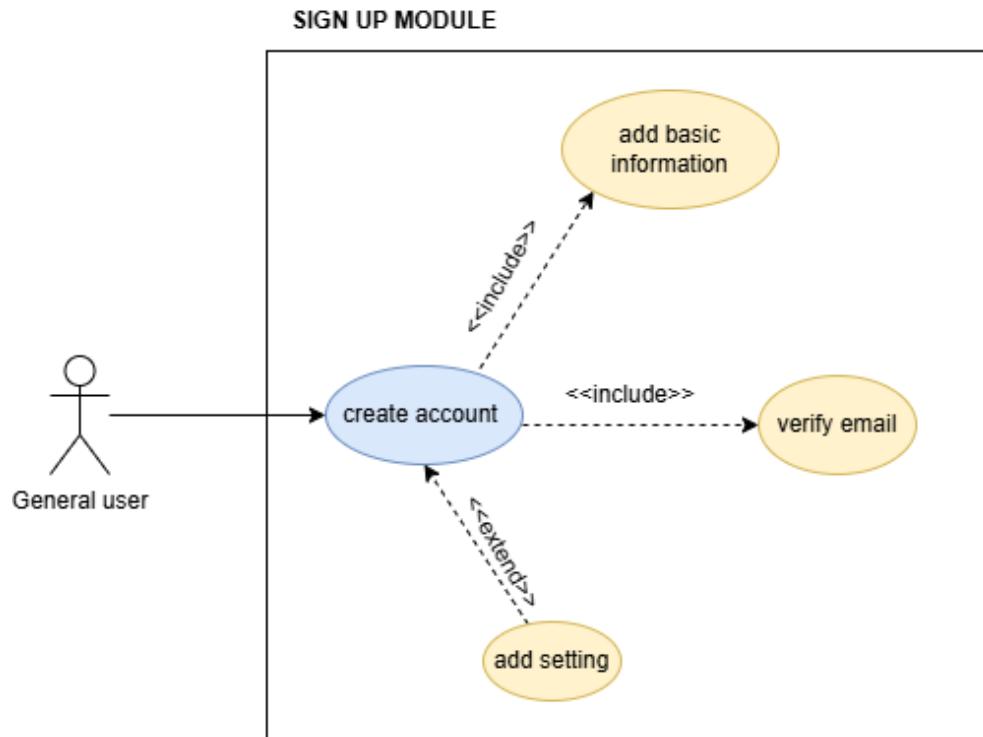
2.6.6 Security

- The platform must encrypt user account passwords using a strong hashing function before storing them in the database to protect against unauthorized access and data breaches.

3 Use case

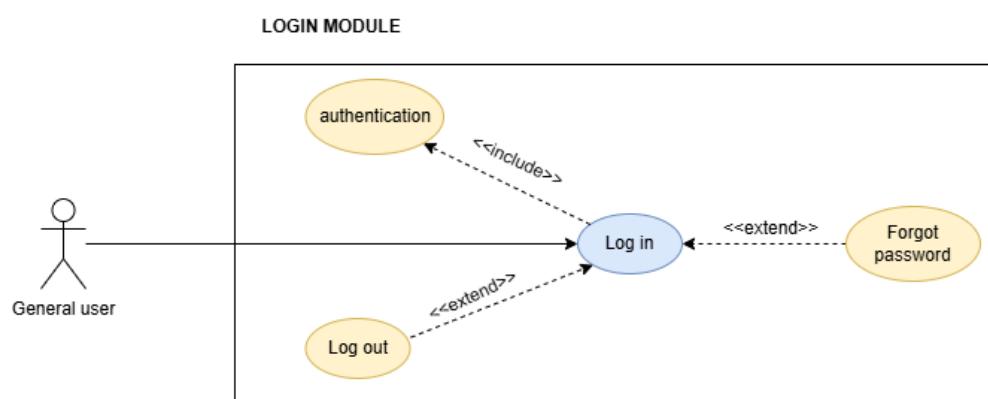
3.1 Use case diagram for the whole system

3.2 Use case: Login & Sign-up module



Use-case Diagram for Sign-up Module

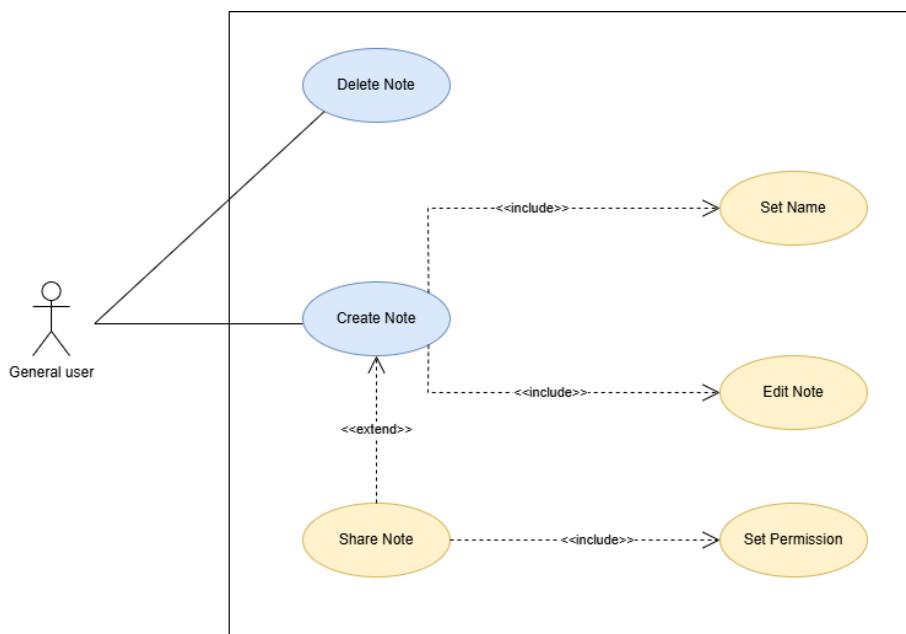
Use case name	Create account
Created by	Phương Thảo
Date created	17.10.2024
Description	This use case describes the process of a general user creating a new account by providing a username and password, along with setting up a personal workspace in the system.
Trigger	The user initiates the sign-up process by selecting the "Sign Up" option on the login page.
Primary actor	General user
Pre-condition	The user must have access to the sign-up page and must not already have an existing account with the same username.
Post-condition	A new account is created, and the user has access to their personal workspace in the system.
Normal flow	<ol style="list-style-type: none"> 1. User selects the "Sign Up" option. 2. User enters a desired username and password. 3. User submits the sign-up form. 4. The system verifies the information. 5. A confirmation message is displayed, and the user is redirected to their personal workspace.
Alternative flow	(None provided)
Exception flow	If the username is already taken: <ol style="list-style-type: none"> 1. The system displays an error message. 2. The user is prompted to choose a different username and can retry.



Use-case Diagram for Login Module

Use case name	Login
Created by	Phương Thảo
Date created	17.10.2024
Description	The user will enter the NoteFlow main page by logging in with the registered account and password. The main screen will display the user's workspace homepage interface.
Trigger	User clicks on the "Login" button.
Primary actor	General users
Pre-condition	The user has successfully registered using an email and password.
Post-condition	The system will display the main screen with options on the left, each leading to a different workspace.
Normal flow	<ol style="list-style-type: none"> 1. The user clicks on the login button. 2. The user enters the username and password.
Alternative flow	(None provided)
Exception flow	If the user logs in with the wrong password: <ol style="list-style-type: none"> 1. The user can click on 'Forgot your Password' to receive assistance. 2. The system will send assistance to the registered email.

3.3 Use case: Take Note module



Use-case Diagram for Take Note Module

3.3.1 Create Note

Name	Create Note
ID	(ID not provided)
Actor	General User
Description	The User creates a new note.
Trigger	User clicks on the button "Add a note".
Precondition	User logged in successfully.
Postcondition	A new note is created and saved in the system.
Main Flow	<ol style="list-style-type: none">1. User selects "Add a note".2. The system displays the create note interface.3. User enters the note content.4. The system confirms that the action has been carried out successfully.
Alternative Flow	(None provided)
Exception Flow	<p>At step 4: The system fails to save the note due to a connection error.</p> <p>4.1. The system displays an error message.</p>

3.3.2 Edit Note

Name	Edit Note
ID	(ID not provided)
Actor	General User
Description	The User edits an existing note.
Trigger	User selects the note to edit.
Precondition	User logged in successfully and the note exists.
Postcondition	A note is edited and saved automatically in the system.
Main Flow	<ol style="list-style-type: none">1. User selects the note to edit.2. The system displays the note content (aka create note interface).3. User edits the note content (modify text, format content, insert or delete images/files, rename the note, rearrange sections, etc.).4. The system confirms that the action has been carried out successfully.
Alternative Flow	(None provided)
Exception Flow	<p>At step 4: The system fails to save the note due to a connection error.</p> <p>4.1. The system displays an error message.</p>



3.3.3 Delete Note

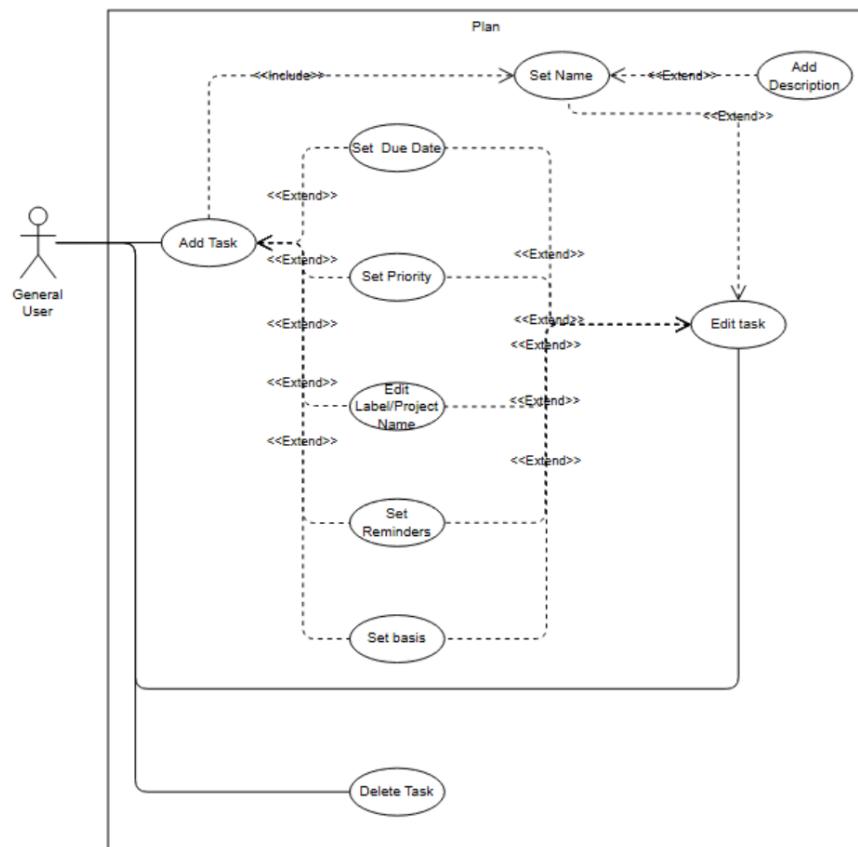
Name	Delete Note
ID	(ID not provided)
Actor	General User
Description	The User deletes existing notes.
Trigger	User selects the notes to delete.
Precondition	User logged in successfully and the notes exist.
Postcondition	The notes are deleted from the system.
Main Flow	<ol style="list-style-type: none">1. User selects the notes to delete.2. The system prompts for deletion confirmation.3. User confirms deletion.4. The system deletes the notes and moves them to the recycle bin.
Alternative Flow	<p>At step 3: If the user cancels the deletion, the notes are not deleted.</p> <p>3.1. The notes remain unchanged in the system.</p> <p>At step 4: If the user wants to recover the deleted notes, go to the recycle bin. Only notes deleted within 30 days can be recovered.</p>
Exception Flow	<p>At step 4: The system fails to delete the notes due to a connection error.</p> <p>4.1. The system displays an error message.</p>



3.3.4 Share Note

Name	Share Note
ID	(ID not provided)
Actor	General User
Description	The User shares notes with others.
Trigger	User selects the note to share.
Precondition	User logged in successfully and the notes exist.
Postcondition	The notes are shared with the recipient and permissions are set.
Main Flow	<ol style="list-style-type: none">1. User selects the notes to share.2. User has 2 options: share the link of the notes publicly or to a designated recipient.3. User sets permissions (view, edit, comment, etc.) for the designated recipient.4. User clicks "Share".5. The system shares the note with the set permissions.
Alternative Flow	<p>At step 3: If the user does not select a recipient, the system prompts the user to choose one.</p> <ol style="list-style-type: none">2.1. The user selects a recipient and sets permissions.
Exception Flow	<p>At step 4: The system fails to share the notes due to a connection error.</p> <ol style="list-style-type: none">4.1. The system displays an error message.

3.4 Use case: Plan module

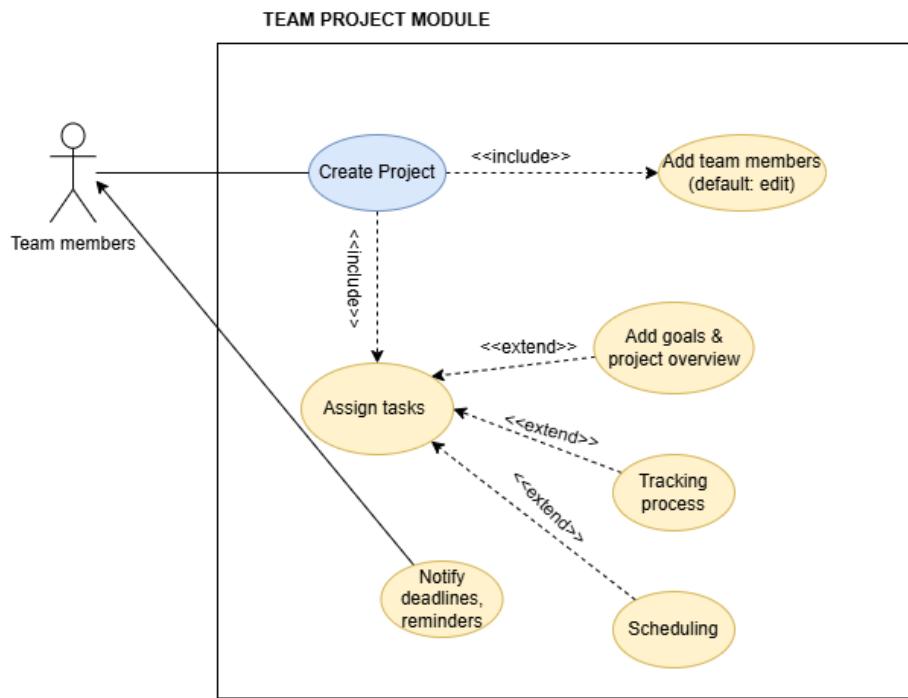


Use-case Diagram for Plan Module



Name ID	Add Task
Actor	General Users
Description	Users add tasks.
Trigger	Users click on the button “Add task.”
Precondition	Users logged in successfully.
Postcondition	Users finish adding tasks without errors.
Main Flow	<ol style="list-style-type: none">1. The system displays the “Main tasks” page.2. User clicks on “Add task.”3. The system displays the “Edit Task” page with features that users can edit:<ul style="list-style-type: none">• Set Due Date• Set Priority• Edit Label/Project Name• Set reminders• Set occurrence4. Users click on “Confirm” to carry out the action.5. The system confirms that the action has been carried out successfully.
Alternative Flow	<ol style="list-style-type: none">4a. If “Cancel” is selected instead of confirming the action, the system returns to step 1 (the “Main tasks” page).
Exception Flow	<ol style="list-style-type: none">5. If the “Name” is left empty or just contains blank characters:<ol style="list-style-type: none">5.1. The system sets default “untitled.”5.2. The system returns to step 3 for the users to reattempt the action.

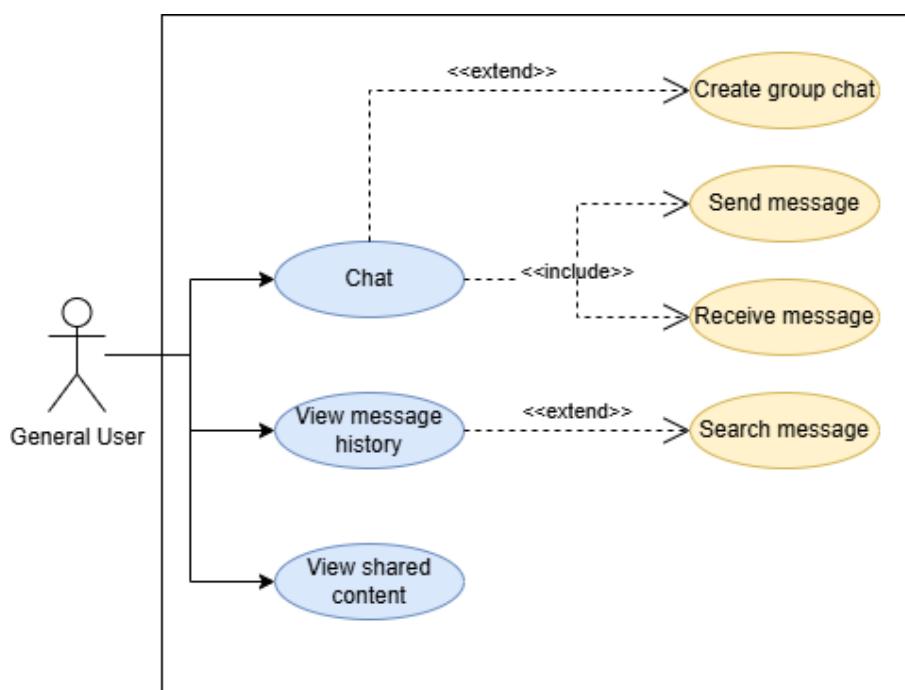
3.5 Use case: Team project



Use-case Diagram for Team-Project Module

Use case name	Create Project
Created by	Phương Thảo
Date created	17.10.2024
Description	This use case describes the process of general users collaborating on a project within the system, allowing them to manage tasks, share files, and communicate.
Trigger	One user initiates one team project by clicking on "Create Project" and "Add team members" to the co-workspace.
Primary actor	General users
Pre-condition	Users must be logged into the system and must have the necessary permissions to create or join a project.
Post-condition	A team project is created or accessed, users can collaborate on tasks and manage project resources.
Normal flow	<ol style="list-style-type: none"> 1. User selects "Create Project" or "Join Project". 2. User enters project details (name, description, etc.). 3. Users add team members and provide permission. 4. The system creates the project and redirects the user to the project dashboard. 5. Users can now manage tasks, share files, and communicate within the project.
Alternative flow	If the user tries to access a project they don't have permission or a project that doesn't exist, the system displays an error message.
Exception flow	If the system fails to add a member or create projects for the team, it will display an error message to notify the user to try again.

3.6 Use case: Chat module





Use-case Diagram for Chat Module

3.6.1 Chat

Use Case	Chat
Primary Actor	General User
Secondary Actor	None
Description	Allows the user to engage in a one-on-one or group chat by sending and receiving messages.
Trigger	User clicks on the chat symbol.
Preconditions	User is logged into the app.
Postconditions	The message is successfully sent, received, and displayed in the chat interface.
Main Flow	<ol style="list-style-type: none">1. User opens the chat feature.2. User types a message.3. User clicks "Send".4. System delivers the message.5. Recipient receives and views the message.
Alternative Flow	<ol style="list-style-type: none">3a. Message is saved as a draft if not sent.5a. If the recipient is offline, the message is queued until they are online.
Exceptions	Message delivery fails due to network issues. System notifies the user of the failure.

3.6.2 Send Message

Use Case	Send Message
Primary Actor	General User
Secondary Actor	None
Description	Enables a user to send a message in the chat.
Trigger	User clicks the "Send" button after typing a message.
Preconditions	User is engaged in a chat session.
Postconditions	Message is successfully delivered to the system for processing.
Main Flow	<ol style="list-style-type: none">1. User types the message.2. User clicks "Send".3. System processes and delivers the message.
Alternative Flow	<ol style="list-style-type: none">2a. If the user navigates away before sending, the message is saved as a draft.
Exceptions	Message delivery fails due to a system error or network issue.

3.6.3 Receive Message

Use Case	Receive Message
Primary Actor	General User
Secondary Actor	None
Description	Allows the user to receive and view a message in the chat.
Trigger	A new message is sent to the user.
Preconditions	The user is connected to the system (online).
Postconditions	The message is displayed to the user.
Main Flow	<ol style="list-style-type: none">1. System detects a new message.2. System sends a notification to the user.3. User opens and views the message.
Alternative Flow	<ol style="list-style-type: none">3a. If the user is offline, the system stores the message and delivers it when the user reconnects.
Exceptions	Message not received due to system failure. System attempts re-delivery.



3.6.4 Search Message

Use Case	Search Message
Primary Actor	General User
Secondary Actor	None
Description	Allows the user to search through past messages based on keywords, dates, or filters.
Trigger	User initiates a search within the message history.
Preconditions	User has access to the chat history.
Postconditions	System returns search results based on the provided criteria.
Main Flow	<ol style="list-style-type: none">1. User enters search criteria (keywords, date, etc.).2. System searches through chat history.3. System displays search results.
Alternative Flow	<ol style="list-style-type: none">3a. System displays "No results" if no matching messages are found.
Exceptions	Search fails due to system or network issues.

3.6.5 View Message History

Use Case	View Message History
Primary Actor	General User
Secondary Actor	None
Description	Allows the user to view previously sent and received messages in the chat.
Trigger	User selects the option to view chat history.
Preconditions	User is logged into the app and has participated in a chat session.
Postconditions	Chat history is displayed to the user.
Main Flow	<ol style="list-style-type: none">1. User selects the "View Message History" option.2. System retrieves the message history.3. User views the list of past messages.
Alternative Flow	<ol style="list-style-type: none">3a. User uses the search function to filter messages.
Exceptions	No message history available due to system error or empty history.

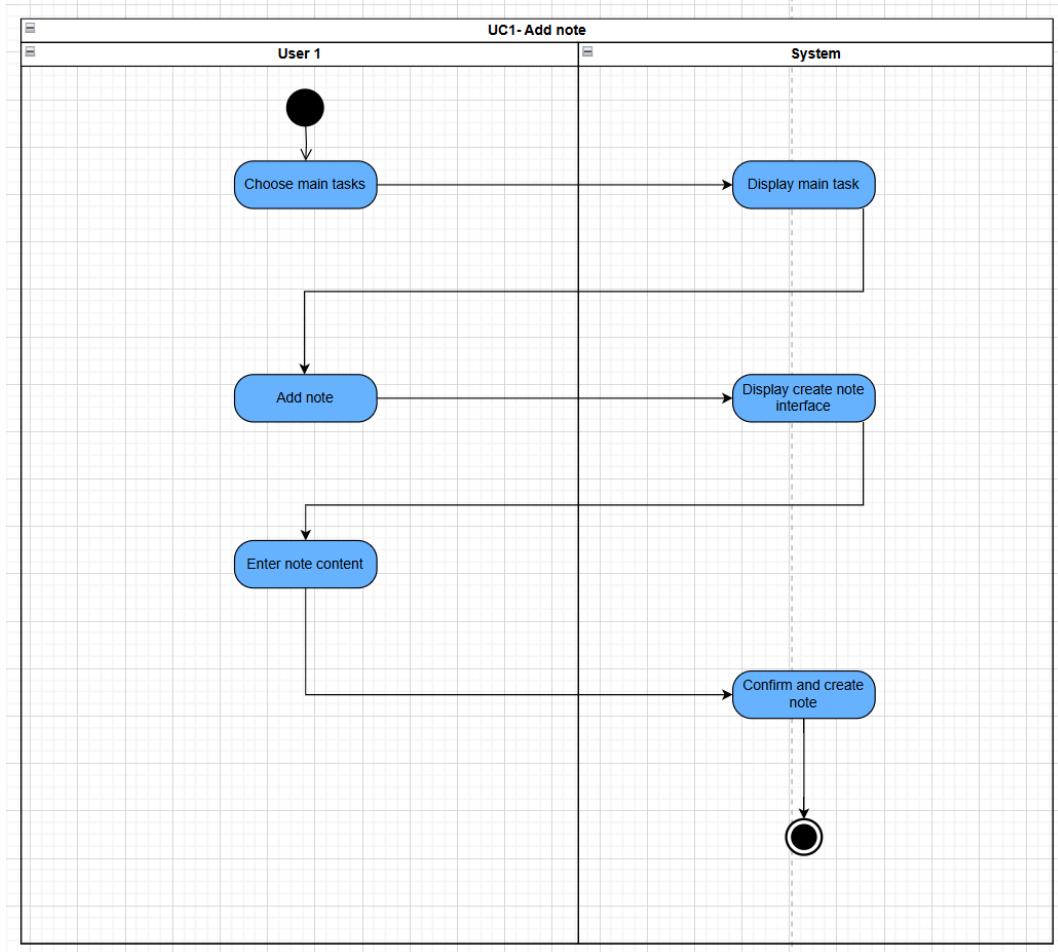


3.6.6 View Shared Content

Use Case	View Shared Content
Primary Actor	General User
Secondary Actor	None
Description	Allows the user to view files, media, or other content shared within the chat.
Trigger	User selects the option to view shared content.
Preconditions	There is shared content in the chat.
Postconditions	The system displays the shared content.
Main Flow	<ol style="list-style-type: none">1. User selects "View Shared Content".2. System retrieves and displays the shared content.
Alternative Flow	<ol style="list-style-type: none">2a. User filters shared content by type (e.g., files, media).
Exceptions	No shared content is available in the chat, or retrieval fails due to system error.

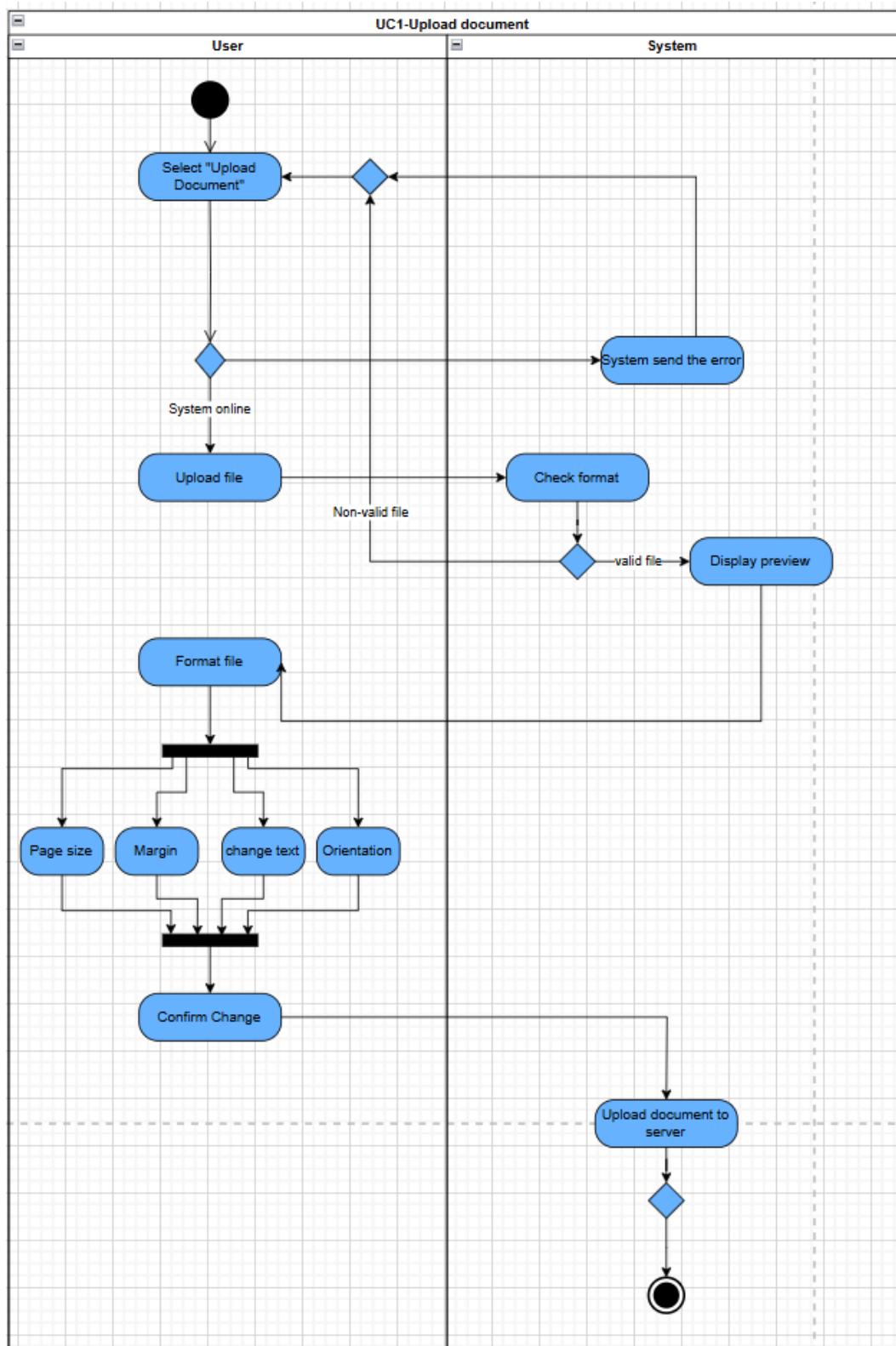
4 System modeling

4.1 Activity Diagram



Add Note - Activity Diagram

The activity diagram illustrates the process of adding a note in the system. It involves two participants: "**User 1**" and the "**System**". The flow begins with **User 1** selecting the main tasks, prompting the system to display the main tasks. Then, **User 1** chooses to add a note, and the system displays the "create note" interface. Afterward, User 1 enters the note content, and the system confirms and creates the note, completing the process. The diagram emphasizes the interaction between the user and the system in a step-by-step manner.

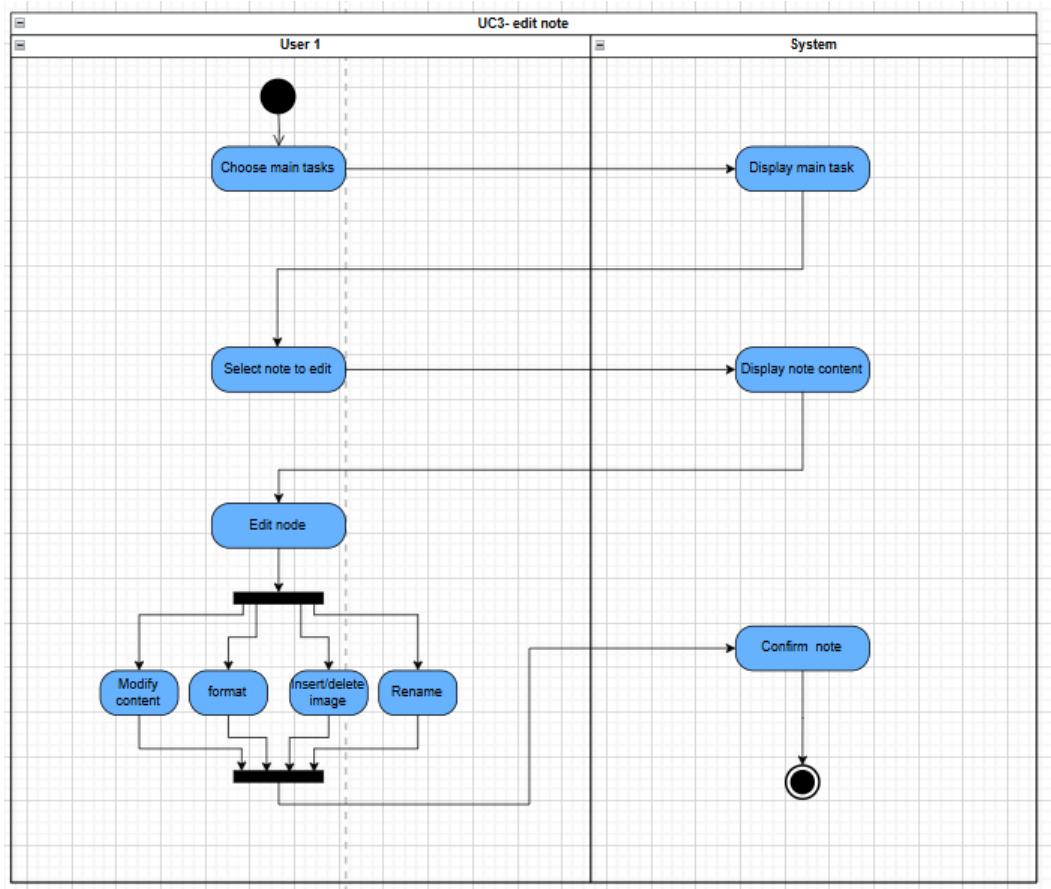


Upload Document - Activity Diagram

The activity diagram depicts the process of uploading a document in the system. The interaction begins with the user selecting the **"Upload Document"** option. If the system is online, the user uploads the file, and the system checks its format. For non-valid files, the system sends an error message. If the file is valid, the system displays a preview.

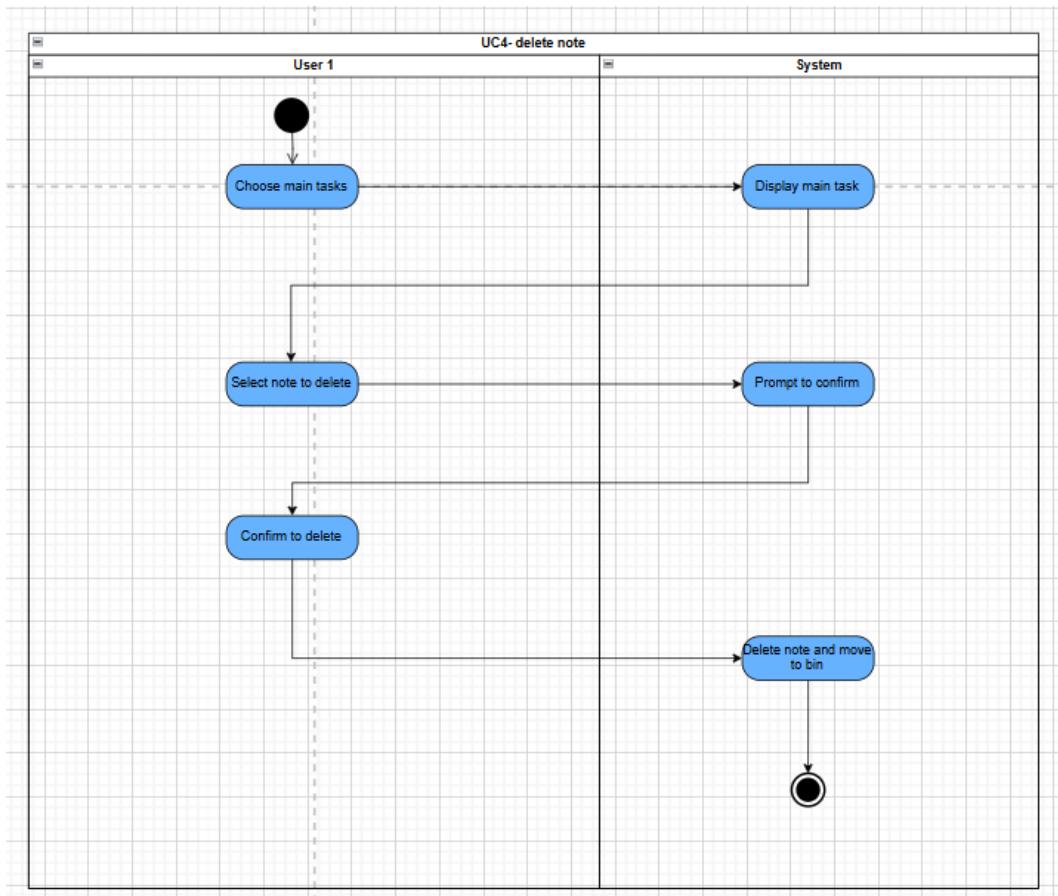
The user then proceeds to format the file, which includes options like adjusting page size, margins, text,

and orientation. After confirming the changes, the system uploads the document to the server, marking the completion of the process. The diagram highlights both error handling and the flexibility to format the document before final submission.



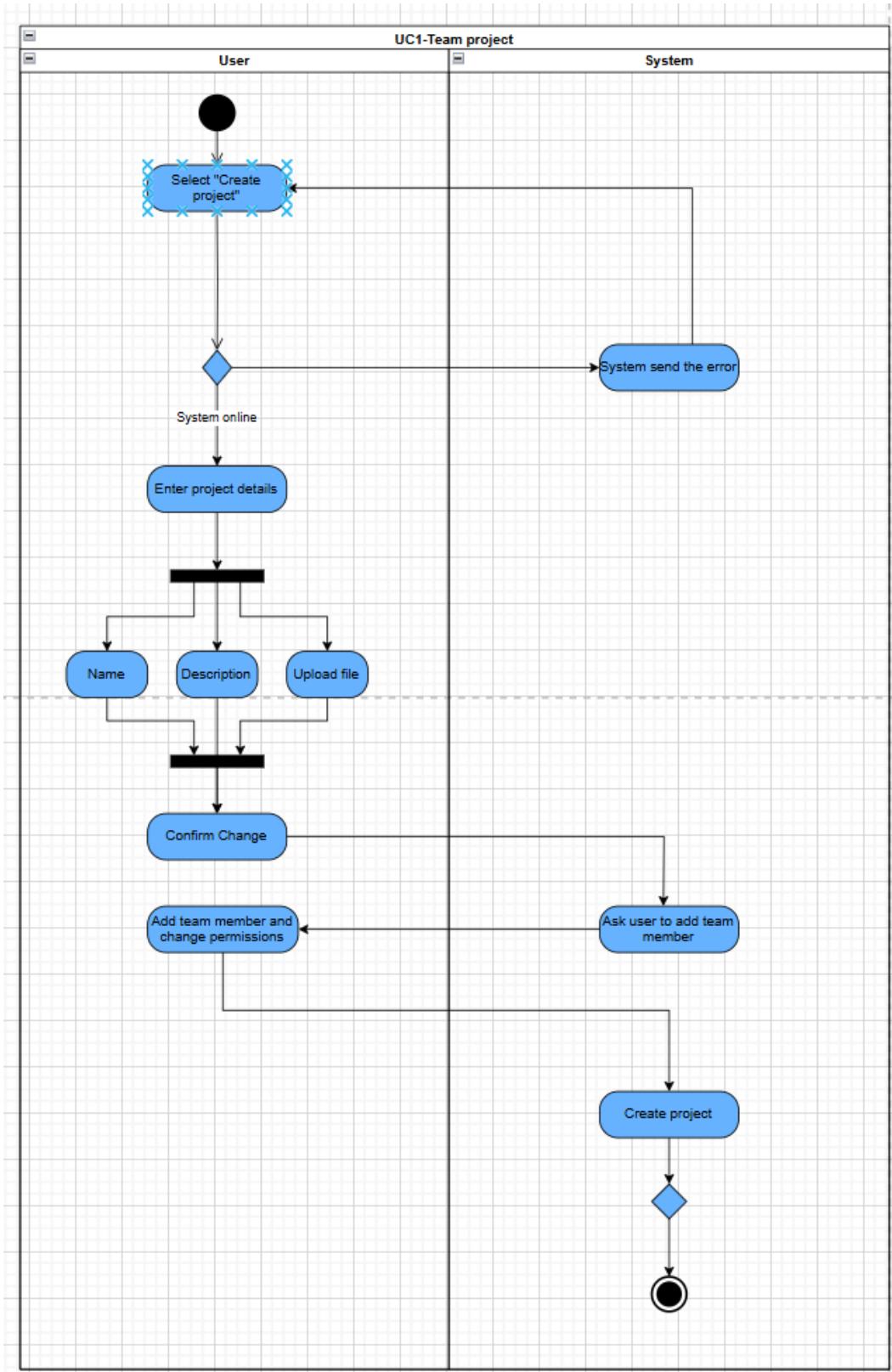
Edit Note - Activity Diagram

The activity diagram illustrates the process of editing a note (UC3 - Edit Note), where the User selects a main task, chooses a note to edit, and enters the editing mode. The System displays the note content, allowing the user to modify content, format text, insert/delete images, or rename the note. Once the edits are complete, the user confirms the changes, and the system updates the note, concluding the process.



Delete Note - Activity Diagram

The activity diagram for "UC4 - Delete Note" shows the process where the User selects a main task and chooses a note to delete. The System displays the main task and prompts the user to confirm the deletion. After the User confirms, the System deletes the note and moves it to the recycle bin, completing the process.



Team Project - Activity Diagram

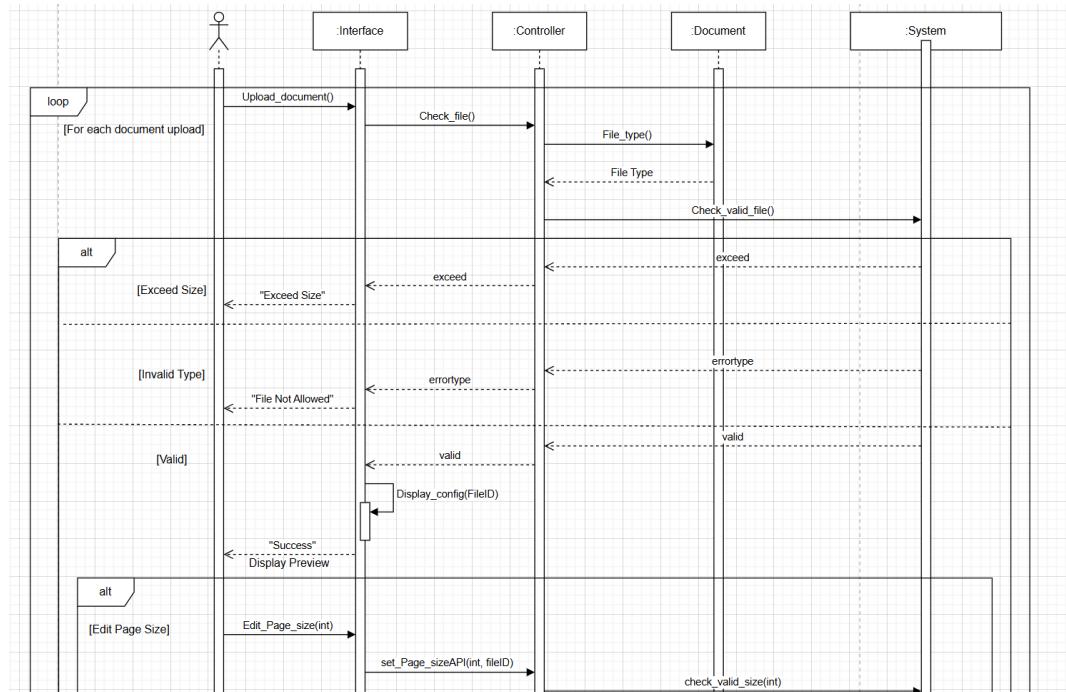
The activity diagram illustrates the process of creating a team project in a system. It involves interactions between the user and the system. Below is a step-by-step description of the workflow:

1. The user begins by selecting the "**Create Project**" option.
2. The system checks if it is online:

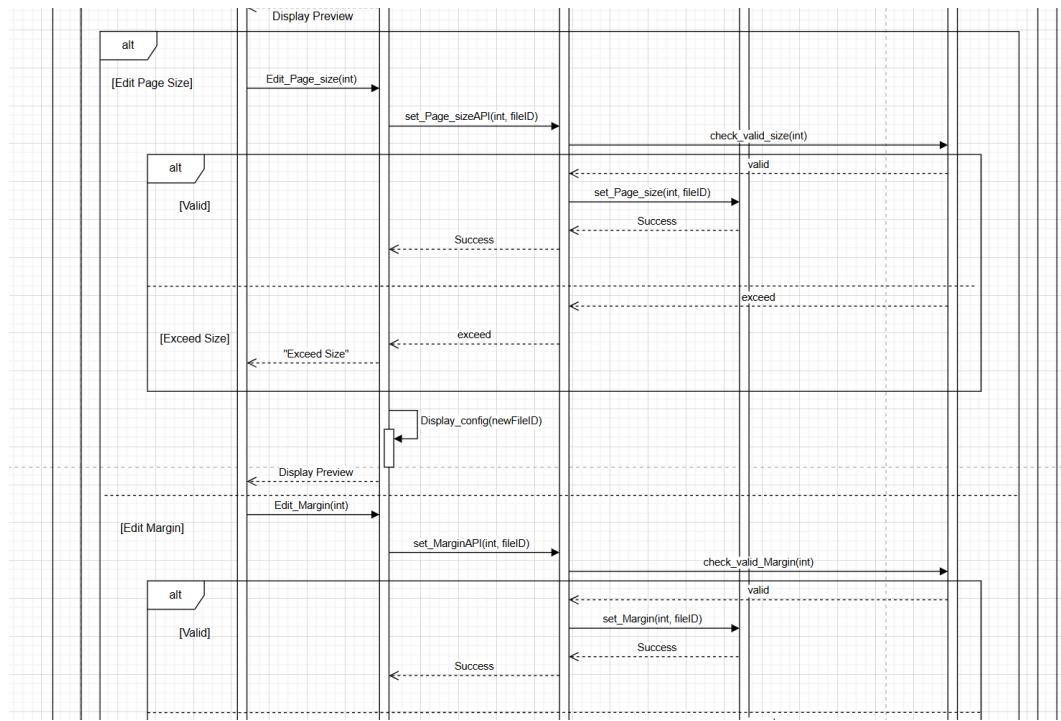
- If offline, the system sends an error message to the user.
 - If online, the user proceeds to the next step.
3. The user enters project details, which include:
- Project **Name**.
 - Project **Description**.
 - **Upload file(s)** related to the project.
4. After entering the details, the user selects "**Confirm Change**" to proceed.
5. The user can then **add team members** and **assign/change permissions**.
6. If no team members are added, the system prompts the user to add team members.
7. Finally, the project is created, and the system completes the process.

4.2 Sequence Diagram

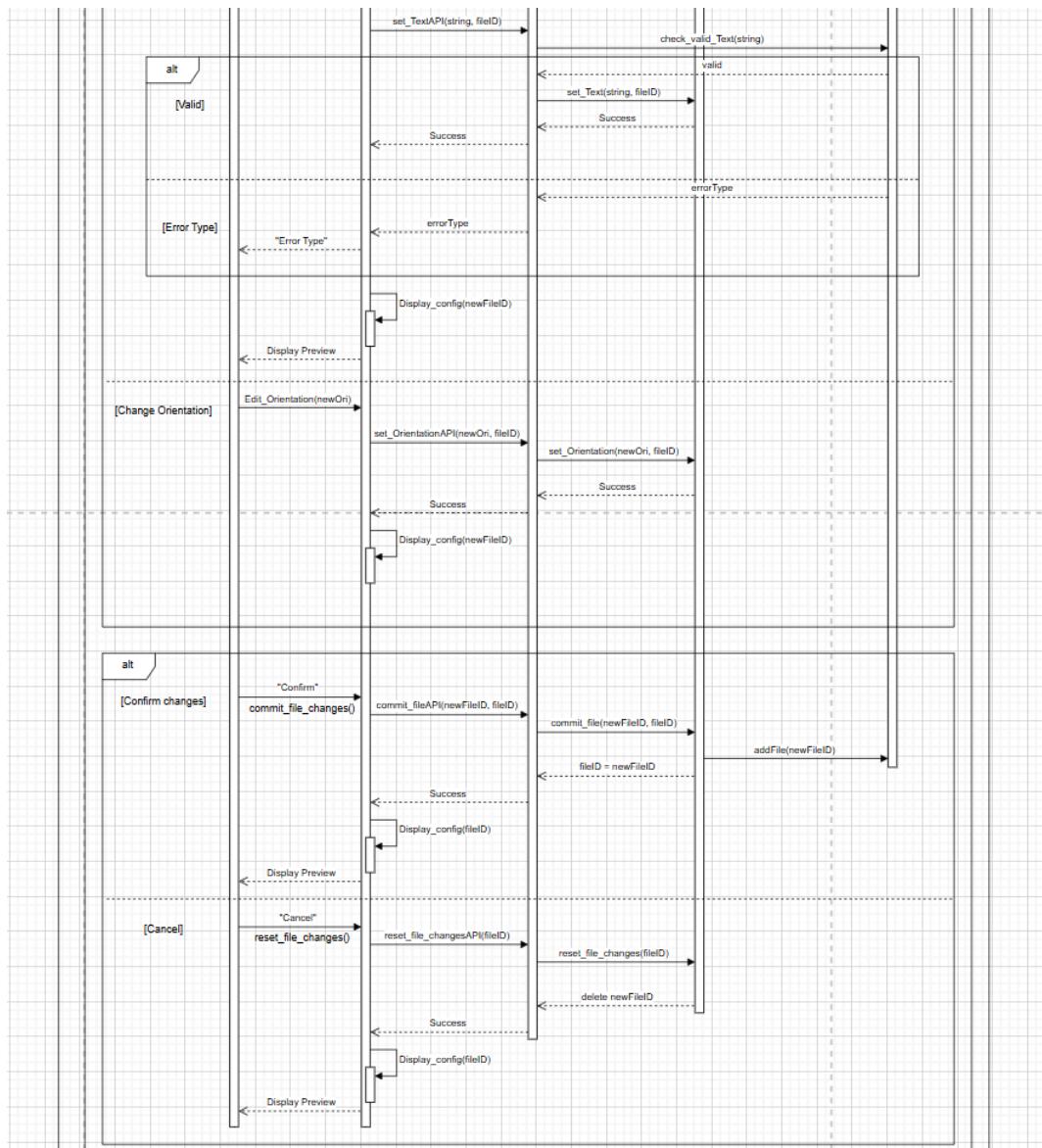
4.2.1 Upload Document



Upload Document - Sequence Diagram



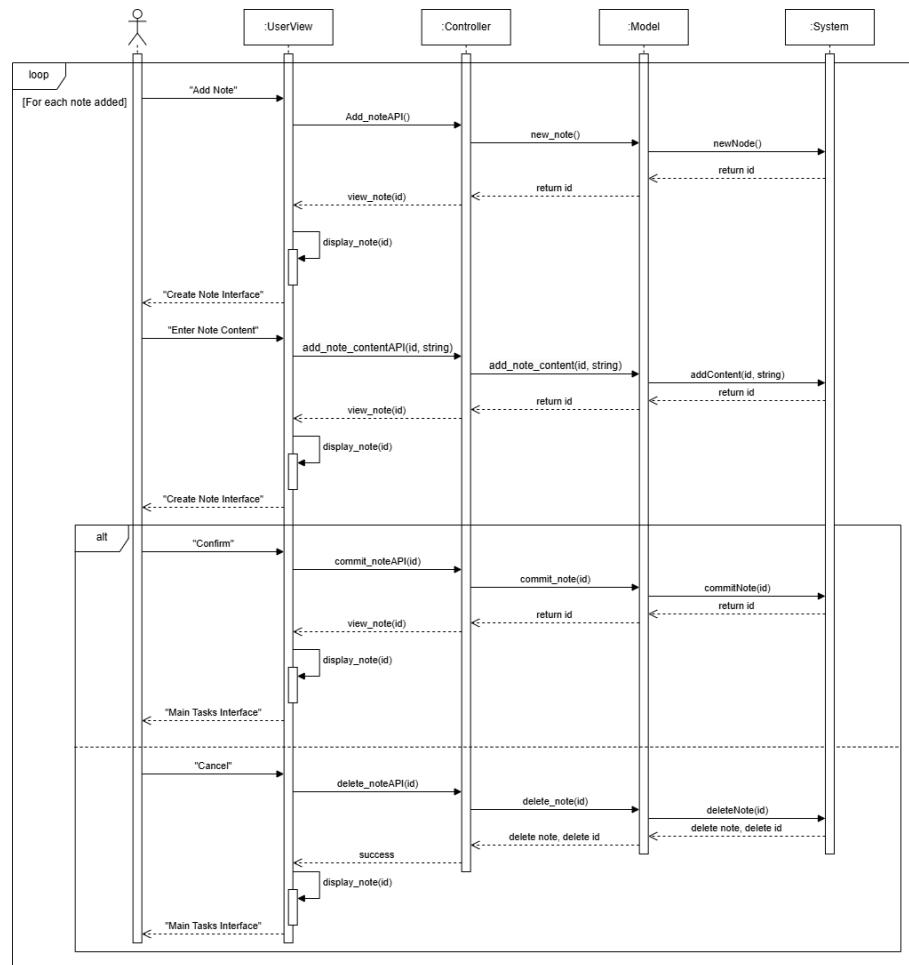
Upload Document - Sequence Diagram



Upload Document - Sequence Diagram

This sequence diagram illustrates the interactions between the components **Admin**, **Back-end**, **Controller**, **Database**, and **Admin UI** within the system. The Admin initiates requests such as **Login**, **createPolicy**, and **updatePolicy**, which are sent to the Back-end. The Back-end forwards these requests to the Controller, where the business logic is processed, and operations like **findPolicy** or **updatePolicy** are performed on the Database. The results from the Database are returned through the Controller and Back-end, ultimately displayed to the Admin via the Admin UI.

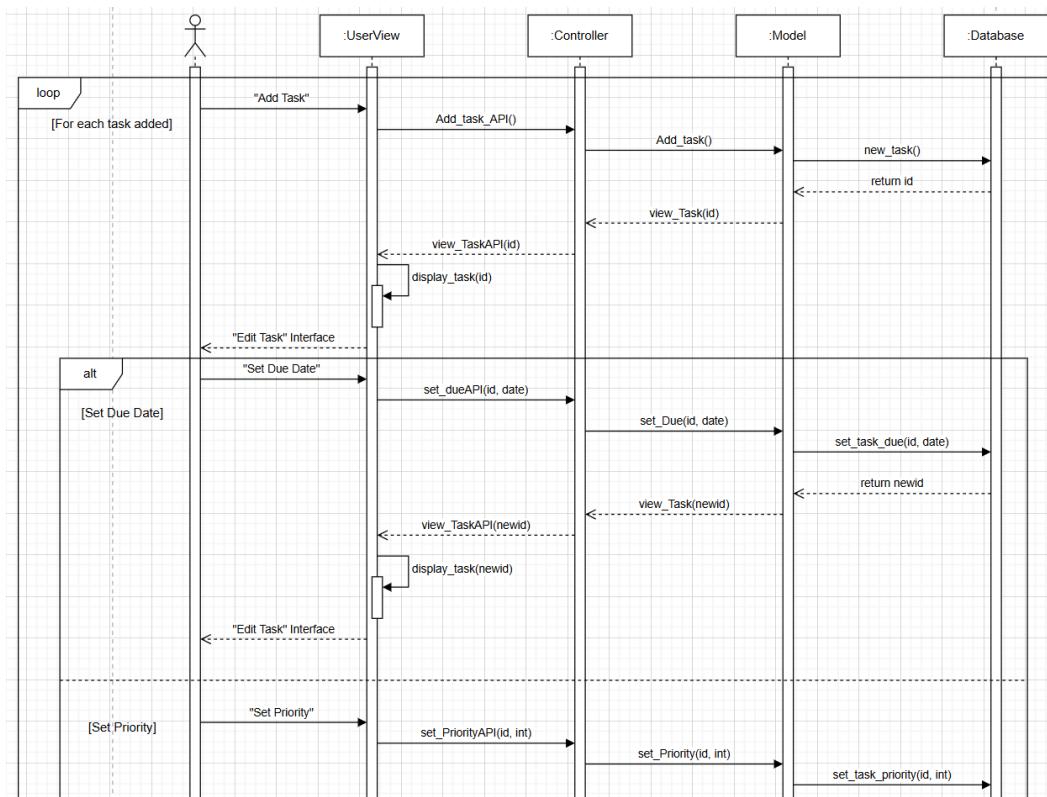
4.2.2 Take Note



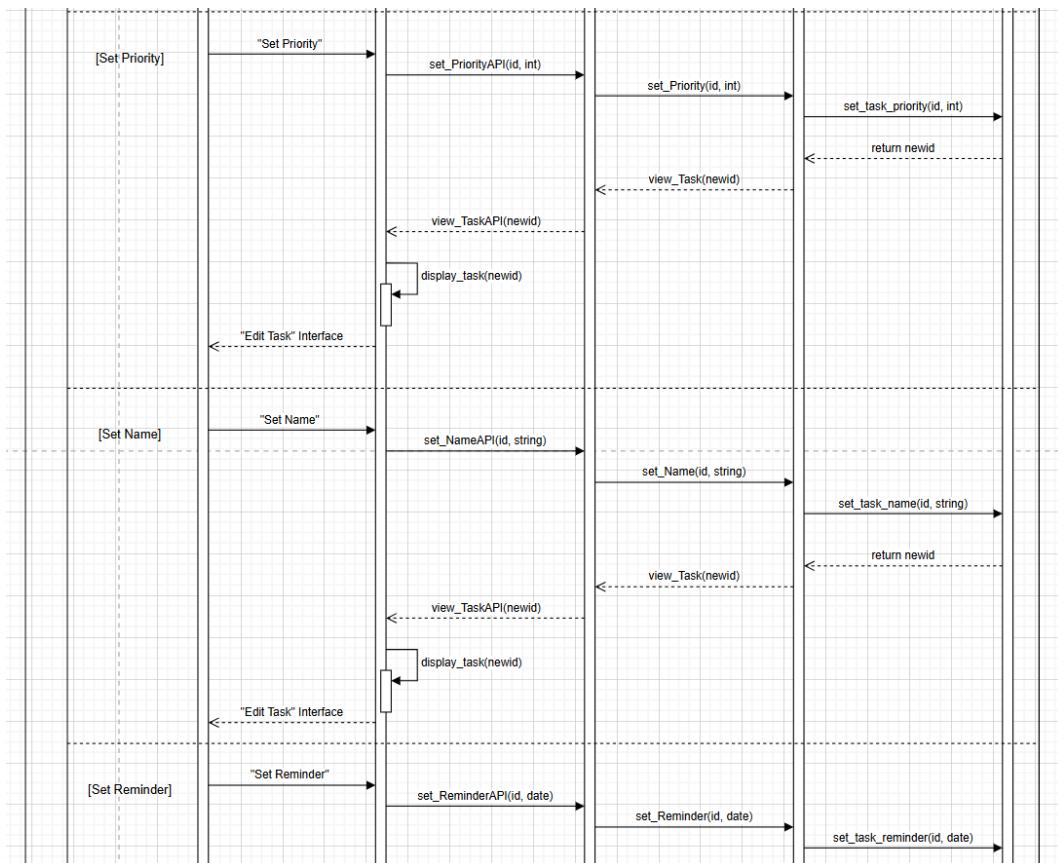
Take Note - Sequence Diagram

This diagram illustrates the interactions within a note management system across four components: **UserView**, **Controller**, **Model**, and **System**. The process begins with the user adding a note through the **UserView**, which triggers the **Add_noteAPI()** in the **Controller**. The **Controller** calls **new_note()** in the **Model**, which interacts with the **System** to create the note, returning its ID back to the **UserView** for display. Users can then add content to the note via **add_note_contentAPI(id, string)**, which follows a similar flow to store the content in the **System**. The note can be committed or deleted based on the user's choice, using **commit_noteAPI(id)** or **delete_noteAPI(id)** respectively. Each action propagates through the **Controller** and **Model**, with the **System** handling the underlying operations. The **UserView** is updated after every operation to reflect the current state of the notes, ensuring seamless user interaction.

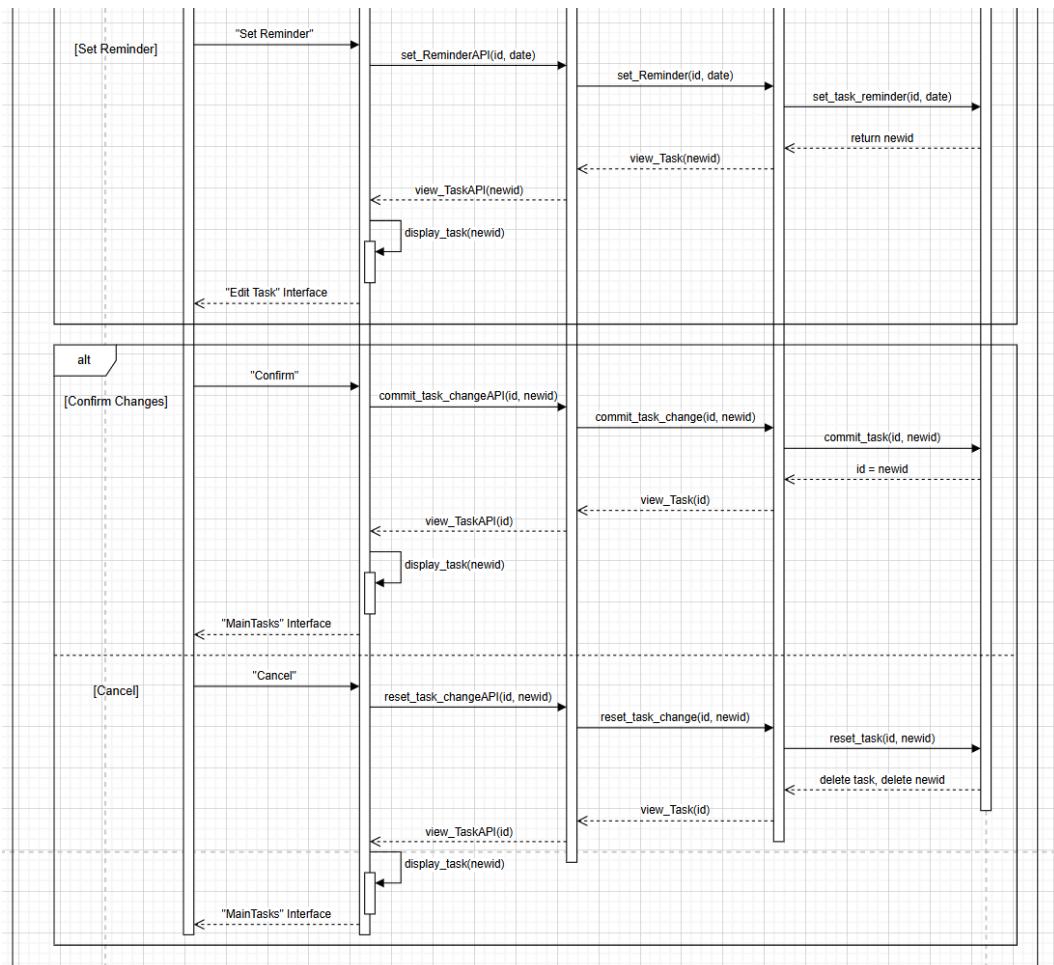
4.2.3 Plan



Plan - Sequence Diagram



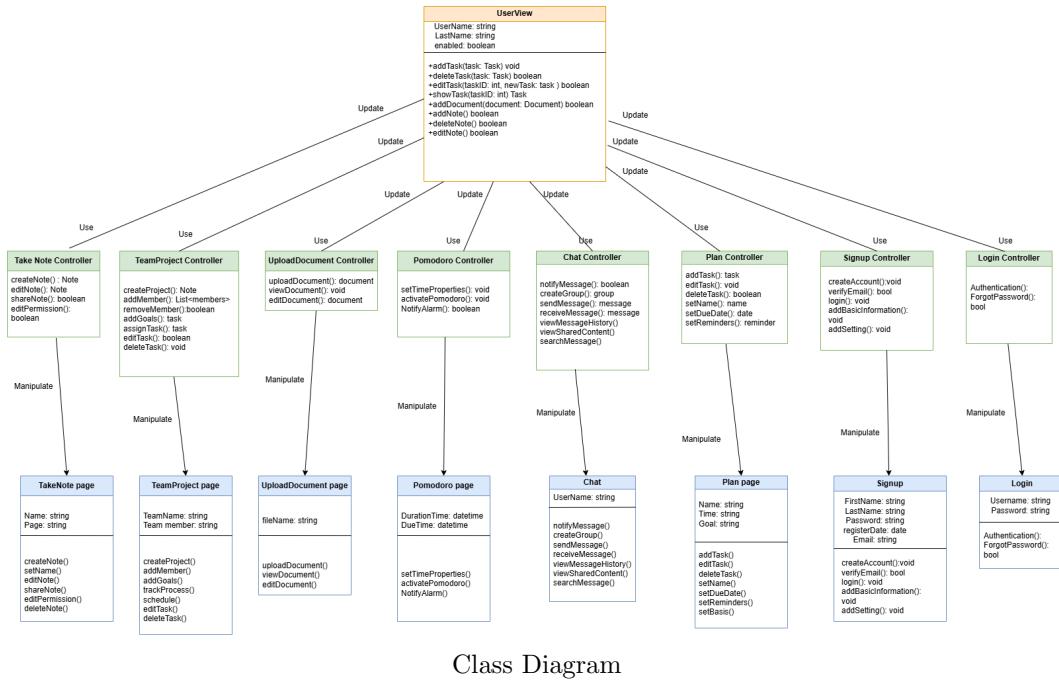
Plan - Sequence Diagram



Plan - Sequence Diagram

This sequence diagram illustrates a task management system where users interact with a User View to perform actions such as adding tasks, editing task attributes, confirming changes, and closing tasks. When a user adds a task, the Controller invokes the `Add_Task()` method in the Model, which saves the task details to the Database. Editing involves modifying attributes like due date (`set_DueDate()`), priority (`set_Priority()`), task name (`set_Name()`), and reminders (`set_Reminder()`), with each change flowing from the Controller to the Model and then updating the Database. Users confirm edits via the `commit_task_changes()` method, which validates and saves changes. Tasks can also be marked as completed using the `close_task()` function. Loops handle multiple tasks, and alternative paths handle conditional operations, ensuring seamless task updates and database synchronization.

4.3 Class Diagram



Class Diagram

1. Model

- **UserView**: The central class that manages user data. It contains:

- **Attributes**:

- * **UserName**: The user's first name.
- * **LastName**: The user's last name.
- * **enabled**: The account activation status.

- **Methods**:

- * Functions for adding, deleting, and editing objects such as **Task**, **Document**, and **Note**.
- * Other utility functions, such as **addTask()**, **deleteTask()**, **addNote()**, **editNote()**, etc.

2. View

The user-facing pages that display data and allow interactions:

- **TakeNote page**: Displays and manages notes with functions like create, edit, share, and delete.
- **TeamProject page**: Displays team information, adds/removes members, and tracks project progress.
- **UploadDocument page**: Allows users to upload, view, and edit documents.
- **Pomodoro page**: Provides a timer interface for Pomodoro task management.
- **Chat page**: Handles chat functionalities like sending messages, creating groups, and viewing history.
- **Plan page**: Displays task planning details and allows users to manage tasks.
- **Signup page**: Provides account creation functionality.
- **Login page**: Handles user login and password recovery.

3. Controller

The controllers handle business logic and mediate between the **Model** and **View**:

- **Take Note Controller:** Manages note-related requests, such as creating, editing, and sharing notes.
- **TeamProject Controller:** Handles operations for team projects, such as adding/removing members and assigning tasks.
- **UploadDocument Controller:** Manages document upload and editing.
- **Pomodoro Controller:** Manages Pomodoro timer properties and notifications.
- **Chat Controller:** Handles messaging functions, including sending/receiving messages and group chat creation.
- **Plan Controller:** Manages tasks, including adding, editing, and deleting them.
- **Signup Controller:** Manages account creation, email verification, and user information setup.
- **Login Controller:** Handles user authentication and password recovery.

Workflow

- **View:** Users interact with the interface, such as the `TakeNote` page, `Plan` page, or `Chat` page.
- **Controller:** The user's requests are processed by the respective controller.
- **Model:** The controllers interact with the `UserView` model to perform data operations.
- **Result:** After processing, the updated data is displayed on the respective view.

4.4 User Interface

The user interface of this web-based note-taking application is thoughtfully designed to enhance productivity and collaboration. From the moment users land on the Welcome Page to navigating through powerful features like notes, planning tools, and team projects, every page is tailored for an intuitive and seamless experience. Below is a detailed description of each interface, showcasing the functionality and user-friendly design that sets this platform apart. Figma Link: [Figma NoteFlow Link](#)

4.5 Welcome Page

The screenshot shows the homepage of NoteFlow, a note-taking website. At the top, there's a navigation bar with 'Welcome Page', 'NoteFlow', 'Sign Up', and 'Log in'. The main heading 'Note taking made simple' is displayed with a subtext 'Passionately made by students. Noted, the all in one note taking website.' To the right is an illustration of a person holding a large pencil, standing next to a clipboard with a checklist. Below this, a callout box contains sections for 'Web design', 'Goals', and 'What to do?'. A curved arrow points from this box to a calendar entry for 'Monday, May, 3rd'. The 'Plan your day' section below the calendar lists various daily tasks in two columns.

Note taking made simple

Passionately made by students.
Noted, the all in one note taking website.

Web design
Web design is a process of making a website for the user.

Goals
The goal is to make the website easy to use for the user and drive the necessary growth.

What to do?

- Conduct Research
- Develop wireframes

Write Notes
Write any notes you want

Plan your day
Make sure your day is well planned

17

Monday

May, 3rd

- Do laundry
- Morning run
- Call mom
- Go to work
- Daily meeting
- Buy dinner
- Go to school
- Study
- Do homework
- Cooking
- Play sport
- Code

The collage illustrates various digital tools and concepts:

- Pomodoro:** A timer application interface with a tomato icon and the text "POMODORO". It shows a timer set for 24:52 minutes. Below it is the slogan "It keeps your mind sharp".
- Work with your team:** A section about project management featuring a "Project Management Template" screenshot and an illustration of four people working together.
- Chatting:** A section about real-time communication featuring a speech bubble icon with a question mark and two user icons with exclamation marks.



(a,b)a ; a=b ... [a]b } >>> } a
//_abcd\\ _a=b; >>> \$ { } a=
:b //ab d ... [a]b }_>>
a;a = b >>> [a]b }>
= //ab c - ac> a (a,b)...
a=b ... [>b]>=\$
= b } } a=b //ab cc) [a]

Ready to take your notes to the next level?

[Try Now](#)

We are always ready to help you and answer your question

Email
syahrulmitahfario@gmail.com

Social Network

X

Get in Touch

Tell us your goals and what note taking means to you

Name

E-Mail

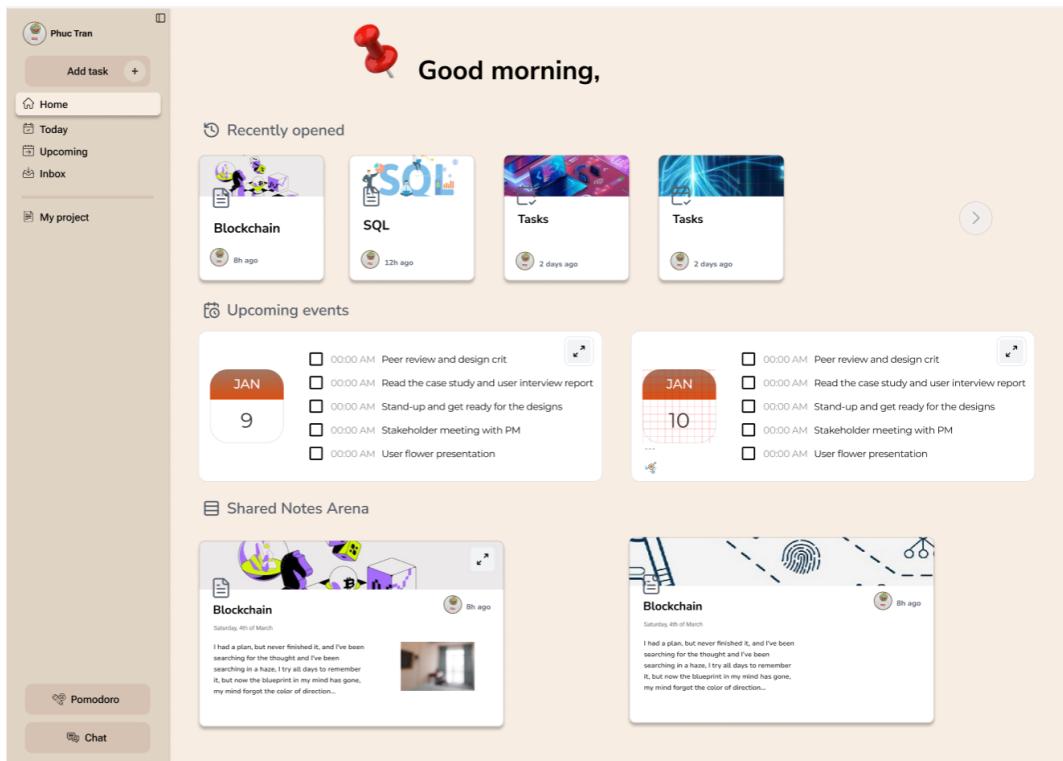
Message

[Submit](#)

Welcome Page

The Welcome Page greets users with an elegant and intuitive design, setting the tone for a seamless experience. At the top, a navigation bar prominently features options to **Sign Up** and **Log In**, inviting users to get started or access their accounts effortlessly. Below, a concise overview introduces the platform's main features: **Write Notes**, **Plan Your Day**, **Pomodoro Timer**, **Team Projects**, and **Chatting**, giving users a glimpse of the platform's versatile capabilities. The page concludes with contact information and the platform's address, fostering a sense of reliability and accessibility. A modern, visually appealing layout ensures first-time visitors feel intrigued and welcomed.

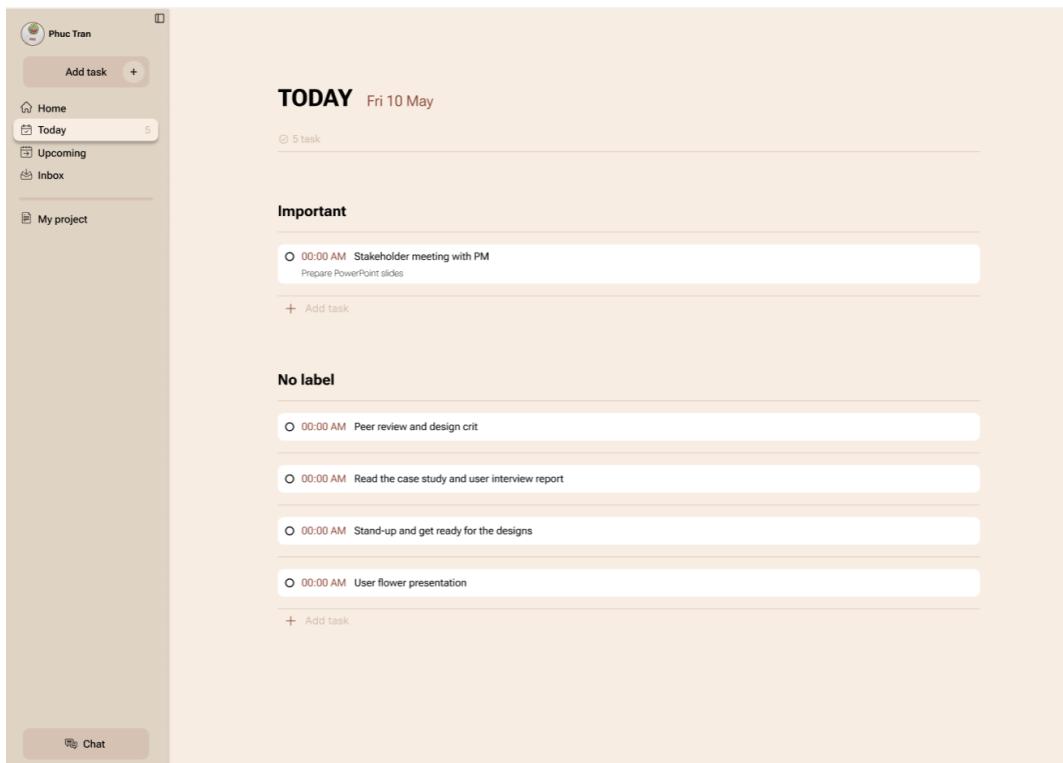
4.6 Home Page



Home Page

Once logged in, the Home Page acts as a personalized dashboard, designed for maximum productivity. Featured prominently are Recently Opened Notes, Upcoming Events, and a curated list of user-selected items Pinned for Quick Access, ensuring important tasks are always within reach. On the left side, a sleek vertical navigation bar offers access to various features, making transitions between sections smooth and intuitive. The layout is clean yet dynamic, providing an organized overview while leaving room for customization.

4.7 Take Notes



Take Notes

The Take Notes page mirrors the versatility and simplicity of platforms like Notion, offering a minimalist yet powerful space for capturing ideas. Users can create, organize, and format notes using an array of intuitive tools, including headings, bullet points, code blocks, and media embeds. The page encourages creativity and productivity, with drag-and-drop functionality and a collapsible sidebar for effortless navigation. Each note can be categorized, shared, or pinned, empowering users to manage their ideas effectively in a distraction-free environment.

4.8 Plan & Add Tasks

The screenshot shows a task management interface for 'Phuc Tran'. On the left, a sidebar includes 'Home', 'Today', 'Upcoming' (which is selected), and 'Inbox'. Below these are 'My project' and 'Chat' buttons. The main area is titled 'UPCOMING' and shows tasks for May 2024. The date 'May 2024' is at the top, followed by a weekly calendar from Monday to Sunday. The 'Upcoming' section lists tasks for Friday, May 10:

- 00:00 AM Stakeholder meeting with PM: Prepare PowerPoint slides
- 00:00 AM Peer review and design crit
- 00:00 AM Read the case study and user interview report
- 00:00 AM Stand-up and get ready for the designs
- 00:00 AM User flower presentation

A '+ Add task' button is located at the bottom of the list.

Plan

The screenshot shows a detailed calendar view for May 2024, similar to the one above but with more specific task details. The days of the week are labeled: Mon, Tue, Wed, Thu, Fri, Sat, Sun. Each day has a grid of four boxes, each containing a list of tasks. The tasks are identical to those listed in the 'Upcoming' view above, repeated for each hour slot. The sidebar on the left is identical to the one in the first screenshot.

Calendar Upcoming Tasks

This function takes inspiration from Google Calendar, offering an intuitive way to schedule and organize tasks. A user-friendly form allows users to input task names, set deadlines, add descriptions, and assign categories. The interface features a calendar view, enabling users to see their tasks in a broader context and manage overlapping schedules. With a clean design and seamless integration into other platform features, this page makes task management effortless.



5 Architecture Design

5.1 Presentation Strategy

This is the first layer in our architecture. We will adopt a strategy that focuses on simplicity, ease of use, and user experience. To achieve this, we will utilize several modern and specific technologies:

5.1.1 Front-end Library and Framework

We use front-end development technologies such as React.js and Next.js 3. React.js allows us to create a flexible and efficient user interface, while Next.js 3 enhances performance with server-side rendering and static site generation.

5.1.2 Responsive Design

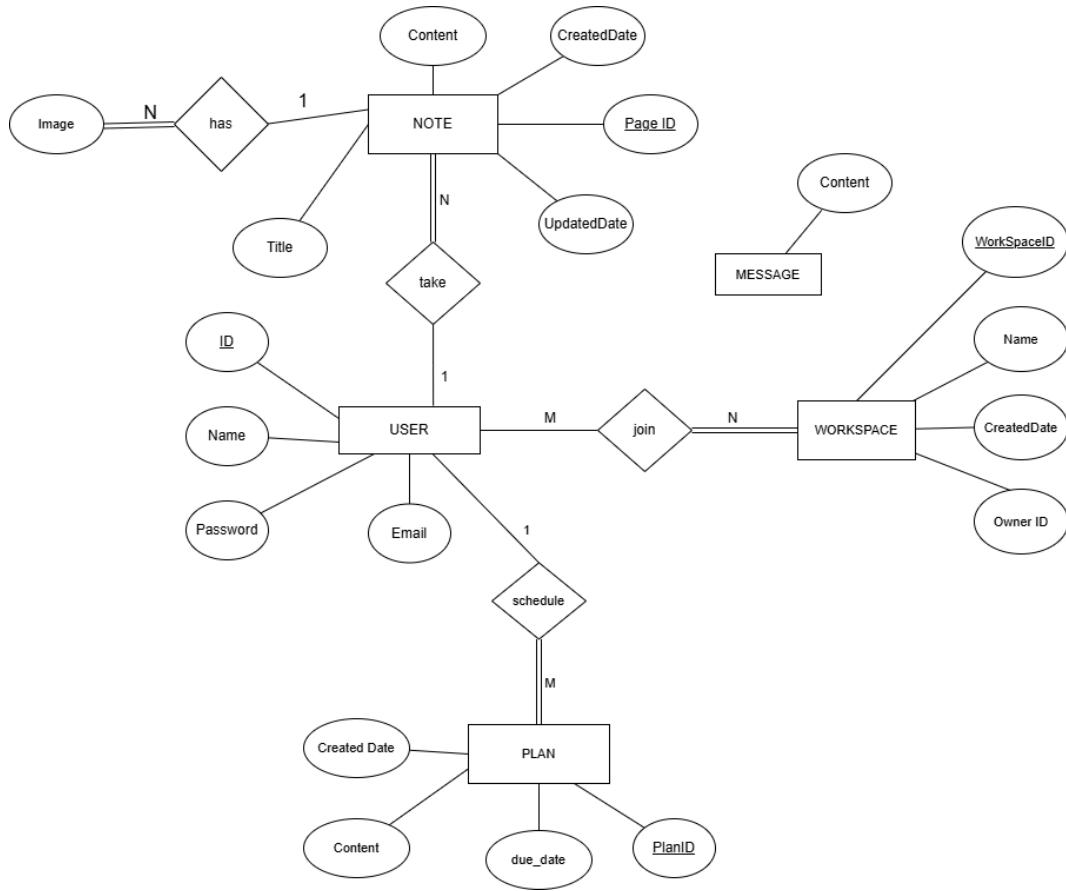
Even though our initial focus is on the PC website version, we ensure that our design is responsive to provide a seamless experience on different screen sizes and resolutions, accommodating various PC monitors and browsers.

5.1.3 User-Friendly Features

We consider intuitive elements such as buttons and easy-to-use menus to make it easy for first-time users to access and use the app. Features such as drag-and-drop functionality for task management, customizable templates for note-taking, and an integrated calendar for scheduling will enhance usability. Furthermore, users can collaborate on notes and tasks with team members or colleagues, making it easier to share information and work together on projects. Also, NoteFlow integrates with other popular applications (e.g., Google Calendar) to streamline workflows and enhance productivity.

By using React.js and Next.js 3 for the user interface, along with flexible design and a focus on user experience, we will create a satisfactory user-oriented interface for NoteFlow while ensuring that the system integrates well with the overall system architecture, providing a seamless and efficient user experience.

5.2 Data Storage Approach



The ERD consists of the following entities, attributes, and relationships:

Entity-Relationship Diagram Description

The ERD consists of the following entities, attributes, and relationships:

Entities and Attributes

- USER** - ID: Unique identifier for the user. - Name: The name of the user. - Password: User's password for authentication. - Email: Email address of the user.
- NOTE** - Title: Title of the note. - Content: The main content of the note. - CreatedDate: Date when the note was created. - UpdatedDate: Date when the note was last updated. - Page_ID: Identifier for the page associated with the note.
- PLAN** - PlanID: Unique identifier for the plan. - Content: Description or content of the plan. - CreatedDate: Date when the plan was created. - due_date: Deadline or due date for the plan.
- WORKSPACE** - WorkSpaceID: Unique identifier for the workspace. - Name: Name of the workspace. - CreatedDate: Date when the workspace was created. - Owner_ID: Identifier of the user who owns the workspace.
- MESSAGE** - Content: Content of the message.
- Image (Weak Entity)** - No specific attributes mentioned.

Relationships

- **User-Workspace (join):** A user can join multiple workspaces, and a workspace can have multiple users. (Cardinality: M:N)
- **User-Note (take):** A user can take multiple notes, but a note belongs to a single user. (Cardinality: 1:N)
- **User-Plan (schedule):** A user can schedule multiple plans, but each plan is associated with a single user. (Cardinality: 1:M)
- **Workspace-Message:** A workspace can contain multiple messages, but each message belongs to a single workspace. (Cardinality: 1:N)
- **Note-Image (has):** A note can have multiple images, but an image belongs to a single note. (Cardinality: 1:N)

5.3 API Management

We will follow the REST (Representational State Transfer) APIs principle, which allows us to take advantage of HTTP caching mechanisms, reducing server load and improving response time for requests from students or SPSO. Due to its flexibility and usability, we use standard HTTP methods (GET, POST, PUT, DELETE) to support a variety of functionalities, including:

5.3.1 Authorization and Authentication APIs

These APIs will be used when users of NoteFlow send a login form to the server. We will integrate Clerk authentication service into our application to manage user authentication. This helps to determine the role of users (Students, Administrators, etc.) and redirect them to their corresponding interface, ensuring they can only see and interact with features relevant to their roles.

5.3.2 Task Management APIs

These APIs will allow users to create, update, prioritize, and track tasks. API endpoints will include adding new tasks, assigning deadlines, and updating statuses. Additionally, users can retrieve lists of their own tasks based on various filters such as due dates, etc.

5.3.3 Note-taking and Collaboration APIs

Users can use these APIs to create, edit, share their notes, and upload documents (with limited storage space). Real-time editing and notification systems will be available for team-based work.

5.3.4 Calendar and Scheduling APIs

These APIs will provide users with the ability to create, view, and manage events, appointments, and deadlines.

5.3.5 Pomodoro Timer and Mini-Games APIs

Users can request the Pomodoro timer functionality to improve focus. The timer will track sessions, alert users when breaks are due, and log the total time spent on focused work. Some mini-games will also be designed to provide users relaxation to maintain productivity.

5.3.6 System Control APIs

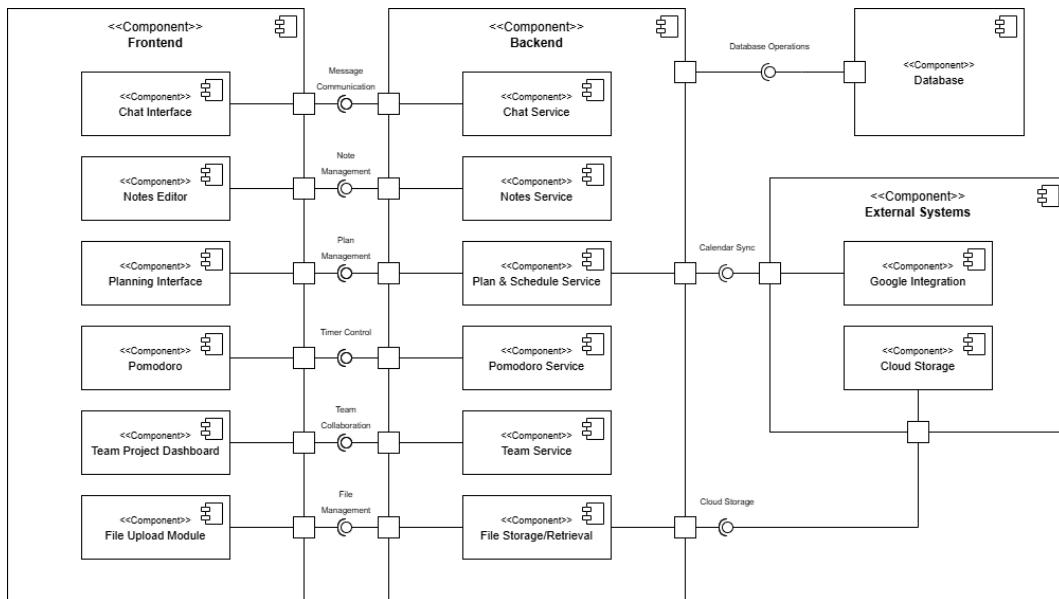
These administrative APIs will allow platform administrators to configure and customize system settings, such as adjusting notification preferences, defining user roles, and setting up integration options.

5.3.7 Integration with External Services

We will integrate Clerk's SDK into the NoteFlow application to manage user authentication and authorization. This will include setting up the necessary API keys and configuring the frontend and backend to handle authentication flows. We can define roles (General Users, Administrators, etc.) within Clerk to manage access permissions and ensure that users only see and interact with features relevant to their roles. By using Clerk's OAuth 2.0 service, users can authenticate with their social media accounts, simplifying the sign-up and login process.

By effectively utilizing HTTP methods for CRUD operations, NoteFlow can empower users to efficiently manage their tasks, notes, and calendar events, enhancing their overall productivity and organization. This capability ensures a responsive and user-friendly application that meets diverse user needs. Additionally, integrating Clerk as an external service streamlines the authentication process, allowing developers to focus on other aspects of the application while maintaining robust security and effective user management. Together, these elements create a seamless experience that fosters control and efficiency for all users.

5.4 Component Diagram



The component diagram describes a system with the following elements:

Frontend

The frontend contains the following components, each representing user-facing modules:

- **Chat Interface:** Allows users to send and receive messages.
- **Notes Editor:** Provides functionality to create, edit, and manage notes.
- **Planning Interface:** Enables users to plan and schedule tasks or events.
- **Pomodoro:** Implements a timer-based focus mechanism for productivity.
- **Team Project Dashboard:** Offers collaborative tools for managing team projects.
- **File Upload Module:** Allows users to upload and manage files.



Backend

The backend supports the functionalities of the frontend through the following services:

- **Chat Service:** Handles message communication between users.
- **Notes Service:** Manages note-related operations.
- **Plan & Schedule Service:** Processes planning and scheduling tasks.
- **Pomodoro Service:** Manages Pomodoro timer sessions.
- **Team Service:** Coordinates team collaboration efforts.
- **File Storage/Retrieval:** Provides storage and retrieval capabilities for uploaded files.

Database

A database is used to store and manage persistent data for the system. It interacts directly with back-end services for database operations.

External Systems

The system integrates with external services to provide additional functionality:

- **Google Integration:** Synchronizes calendar data with Google services.
- **Cloud Storage:** Stores user files in the cloud for access and management.

Communication Flow

The communication flow between components is as follows:

- Front-end components communicate with their corresponding back-end services via defined interfaces.
- The back-end interacts with the database for data persistence and retrieval.
- External systems such as Google Integration and Cloud Storage are accessed by the back-end for calendar synchronization and file storage.



6 Implementation - Sprint 1 (4)

6.1 Set up repository on Github

Link Github: [Github Link!](#)

6.2 Steps to Run the Project

To run the `appv1.2` branch of this project, follow these steps:

1. **Clone the repository:** Use the following command to clone the repository:

```
git clone https://github.com/your-username/integration-project-noteflow.git
```

2. **Navigate to the project directory:** Change to the project directory with:

```
cd integration-project-noteflow
```

3. **Switch to the appv1.2 branch:** Use the following command:

```
git checkout appv1.2
```

4. **Install the necessary dependencies:** Run the appropriate command based on your environment:

- For Node.js:

```
npm install
```

- For Python:

```
pip install -r requirements.txt
```

5. **Set up environment variables (if required):** Create a `.env` file and add the necessary configuration as per the project's documentation.

6. **Start the application:** Use the following command to run the application:

```
npm start
```

Access the application in your browser at <http://localhost:3000> or the specified port.

6.3 Usability test with the User Interface

Usability testing is a method to evaluate how easy and user-friendly a product design is for its target audience. It involves researchers, also called "facilitators" or "moderators," asking participants to complete specific tasks using the product. By observing their interactions, developers can identify existing issues or errors in the software and uncover opportunities for improvement. Key aspects of the testing include ease of navigation, clarity of design, and responsiveness. The primary goal is not only to ensure the product meets user expectations but also to enhance the user experience and identify future development opportunities.



6.3.1 Recruit participants/ testers

At this stage, our group will select seven participants to test their experience with the front-end interface in MVP1 that the team has developed. The reason for selecting only seven participants is that too few testers might result in undetected issues or tasks that may not challenge some students but could be difficult for others. On the other hand, involving too many users is not feasible due to time and budget constraints.

Our group will have these seven participants complete tasks using the software twice. After the first round of testing and receiving feedback, the team will revise MVP1 and conduct a second round of testing. Following this, the feedback from the second test will be documented, and the web application will be finalized with the necessary adjustments based on MVP1.

6.3.2 Define tasks

To ensure test cases accurately reflect the tester's objectives and effectively cover all software use cases, the team has established the following guidelines:

- Avoid UI-specific terms: Descriptions should focus on the user's goal rather than the specific interface elements.
- Keep descriptions concise: Test case descriptions should not be step-by-step instructions. The focus should be on the overall goal, not the exact process to achieve it.
- Keep descriptions concise: Test case descriptions should not be step-by-step instructions. The focus should be on the overall goal, not the exact process to achieve it.
- Ensure test cases are realistic: Avoid creating scenarios that are impossible or haven't occurred yet.
- Balance simplicity and complexity: Test cases should not be overly simple or complex.
- Provide context with scenarios: A brief scenario can help testers better understand the purpose of a test case.

From these guidelines, our team has proceeded to describe several tasks based on the system's critical use cases.



Use Case	Task Description
Access NoteFlow, Login - Sign Up	Access website, log in and sign up. While testing, tester can check if the welcome page exists properly, or if there are boxes for filling out username and password in Login - Signup page.
Access Home Page	Access homepage. While testing, tester can check if the Pinned tasks, Recently opened tasks and Upcoming events exists according to description and whether they are interactive.
Take Note	Taking new note. While testing, tester can check if the tools are available.
Chat	Chatting. While testing, tester can check if the chatting interface is available as well as the search bar for messages.
Use Pomodoro Timer	Use pomodoro timer. While testing, tester can check if the timer has sessions for session length and break length as it should have.
Check Today's Plan(s)	Access the Today-plan page. While testing, tester can check if the to-do list for today has sufficient information.
Check Upcoming Event(s)	Access the Upcoming-event page. While testing, tester can check if they can switch between the weekly view and the monthly view.
Check Team Project(s)	Access the Team-project page. While testing, tester can check if the tasks in that page is easy to track.
Check Notifications	Access the Plan Notifications. While testing, tester can check if the notifications are on time and clear to see.

6.3.3 Define test strategy (qualitative vs. quantitative, remote vs. in person)

Based on the project's nature, as well as budgetary and time constraints, our team has decided to adopt the following testing approach:

- Remote moderated testing: The moderator will interact with the test participant and request tasks be completed. There are 2 ways: Supervised and Unsupervised. The supervised testing is quite similar to in person, but the participant and the facilitator are able to be in different locations and use software like Google Meet, Zoom, Microsoft Teams,.. to communicate. The unsupervised one involves using software that can create tasks and guide participants through them, acting as a virtual facilitator. Our team has decided to use Supervised method. This decision was made based on the following reasons:
 - Speed and cost-effectiveness: This method eliminates the need for participants and moderators to travel to a physical location. Additionally, it allows for greater flexibility in scheduling.
 - Expanded participant pool: In-person testing requires participants to be in close proximity to the testing location. Remote testing allows the team to recruit participants from a wider geographic area.
- Furthermore, the team will employ qualitative research methods. This approach focuses on gathering detailed insights, observations, and understanding how individual users interact with the product or service. The team has chosen this method because it aligns with the primary goal of understanding user behavior and experiences.

6.3.4 Conduct the test

Link Figma: [NoteFlow Prototype Link](#)

Link Google Form: [Google Form](#)



Link Google Sheet (Response): [Feedback Form \(Responses\)](#)

Step 1: Select a Testing Tool

- Google Form: Chosen for its simplicity, ease of use, and effectiveness in collecting structured user feedback.
- Introduction: Briefly explain the purpose of the test and provide the Figma prototype link.
- Participants know their feedback is anonymous.

Step 2: Select Questions and Provide Use Cases for Testing

- Questions: Prepare questions based on tasks. For instance: Taking note, planning tasks, managing team projects and chatting.
- Design tasks that reflect real-world scenarios, such as:
 - Take a new note for a task due tomorrow.
 - Tracking tasks in a team project.
 - Chatting with a team member.

Step 3: Choose participants

- Target Audience:
 - Include group members actively involved in the design process for early feedback.
 - Invite real-world participants who are potential users of the website, such as students and printing service operators.

Step 3: Conducting the Test

- Test Interface: Use a shared Figma Prototype Link was used as the testing interface.
- Participant Observation: A moderator observes participant actions and behaviors during the test to capture usability issues or confusion points.
- Post-Task Questions: After each task, the moderator asked 1-2 questions to gain deeper insights into the participant's experience.

Step 4: Test Conclusion

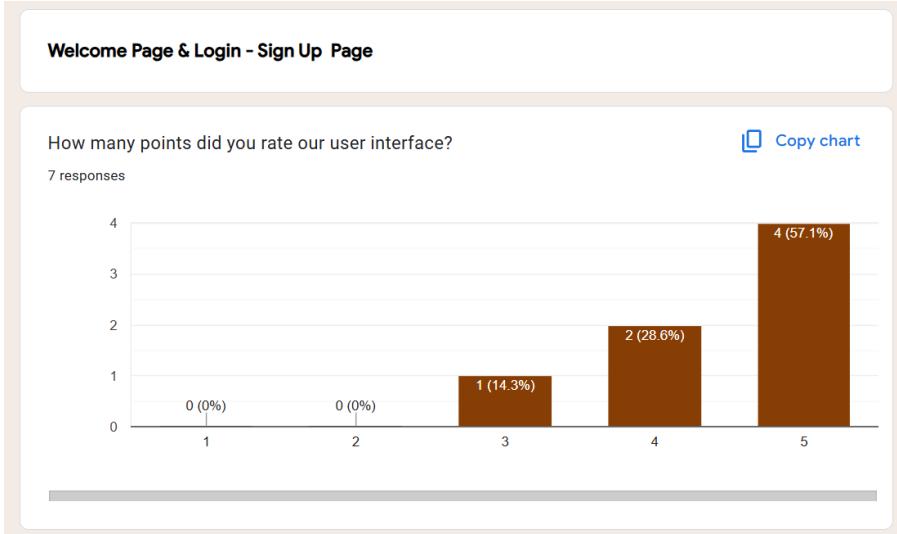
- Recording Feedback: Save the feedback as google sheet, determine the strengths and weaknesses to improve
- Team Discussion: Conduct a collaborative review with the team to analyze recorded feedback and discuss potential improvement

Step 5: Analyze Feedback

- Feedback Analysis:
 - Export responses from Google Forms into a spreadsheet for organization and sorting
 - Prioritize design adjustments based on feedback, focusing on tasks where participants faced difficulties or suggested enhancements
- Action Plan: Implement the changes and schedule follow-up tests to ensure improvements meet user expectations.

6.3.5 Document the feedback from testers

- **Welcome Page & Login - Sign Up Page:** The chart below summarizes the feedback on the user interface of the home page and login page, based on the survey question: "*How many points did you rate our user interface?*":



- **Survey Results** A total of 7 responses were collected, and the distribution of ratings is as follows:

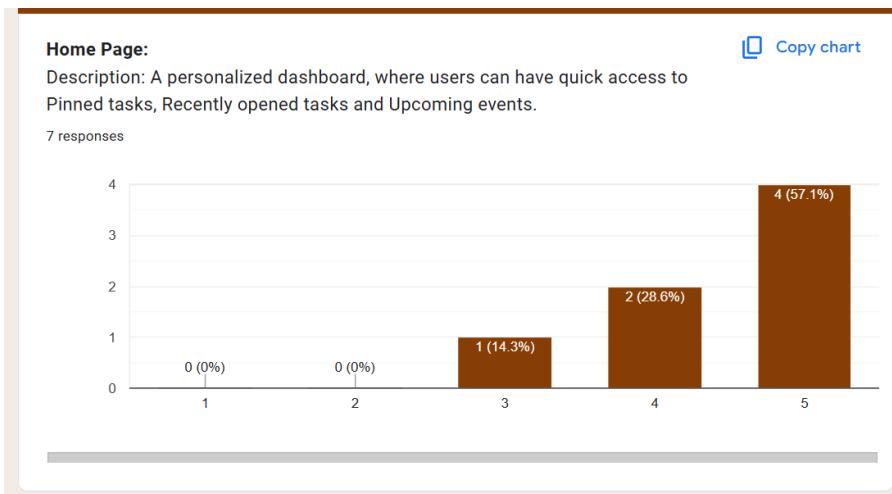
- * **1 point:** 0 response (0%)
- * **2 points:** 0 responses (0%)
- * **3 points:** 1 responses (14.3%)
- * **4 points:** 2 response (28.6%)
- * **5 points:** 4 responses (57.1%)

- **Analysis**

- * Positive Feedback: A significant majority, 57.1%, rated the user interface as very good. Combined with the 28.6% who rated it as good, this indicates that most users are very satisfied with the interface.
- * Moderate Feedback: One respondent rated the interface as moderate, suggesting there may be minor areas for improvement.
- * No Negative Feedback: There were no ratings indicating that the interface needs significant improvement (ratings of 1 or 2).

- **Conclusion:** Overall, the feedback is highly positive, with most users finding the interface either excellent or very good. This is a strong indicator that the design and functionality of the Welcome Page and Login - Sign Up Page are well-received by users.

- **Home Page:** The chart below summarizes the feedback on the user interface of the home page, based on the description: "*A personalized dashboard, where users can have quick access to Pinned tasks, Recently opened tasks and Upcoming events.*":



– **Survey Results** A total of 7 responses were collected, and the distribution of ratings is as follows:

- * **1 point:** 0 response (0%)
- * **2 points:** 0 responses (0%)
- * **3 points:** 1 responses (14.3%)
- * **4 points:** 2 response (28.6%)
- * **5 points:** 4 responses (57.1%)

– **Analysis**

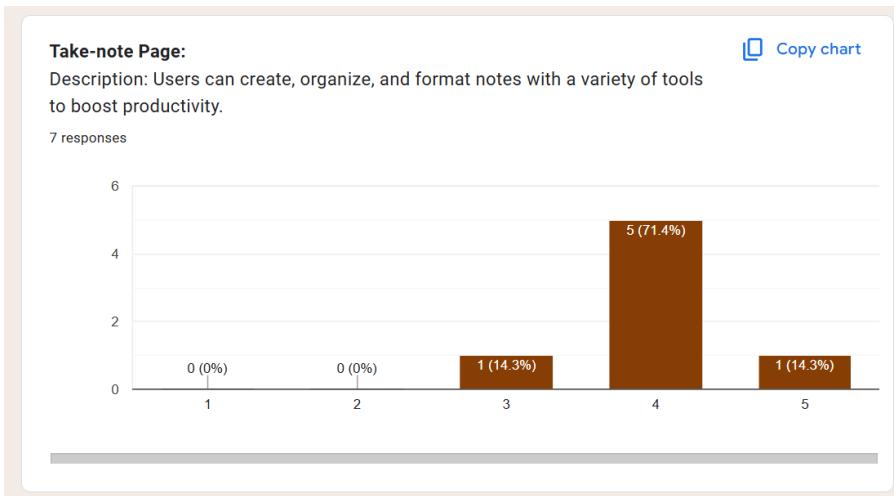
- * Positive Feedback: The Home Page received a "very good" rating from a notable 57.1% of respondents. Combined with the 28.6% who rated it as "good," this demonstrates a high level of user satisfaction with the personalized dashboard.

- * Moderate Feedback: One respondent rated the Home Page as moderate, indicating potential areas for minor improvements or adjustments.

- * No Negative Feedback: There were no responses highlighting minor problems or significant improvement needs (ratings of 1 or 2).

– **Conclusion:** The feedback for the Home Page remains overwhelmingly positive with the new rating scale. The majority of users find it very good or good, showing strong satisfaction with the personalized dashboard. With one user rating it as moderate, it's worth considering gathering more detailed feedback to identify specific areas for potential improvement.

- **Take-note Page:** The chart below summarizes the feedback on the user interface of the take-note page, based on the description: "*Users can create, organize, and format notes with a variety of tools to boost productivity.*":



– **Survey Results** A total of 7 responses were collected, and the distribution of ratings is as follows:

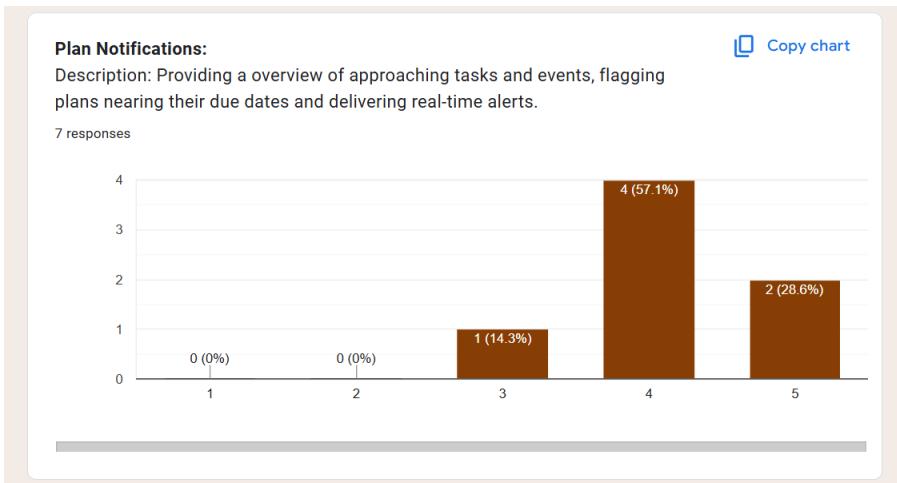
- * **1 point:** 0 response (0%)
- * **2 points:** 0 responses (0%)
- * **3 points:** 1 responses (14.3%)
- * **4 points:** 5 response (71.4%)
- * **5 points:** 1 responses (14.3%)

– **Analysis**

- * Positive Feedback: The majority of respondents (71.4%) rated the Take-note Page as good, while one respondent (14.3%) rated it as very good.
- * Moderate Feedback: One respondent (14.3%) rated it as moderate, suggesting some areas where improvements could be made
- * No Negative Feedback: There were no ratings indicating minor problems or significant needs for improvement (ratings of 1 or 2).

– **Conclusion:** The feedback for the Take-note Page is largely positive. The majority of users find it good, showing that the feature is effective in creating, organizing, and formatting notes with various tools to boost productivity. The very good rating adds a layer of satisfaction, while the moderate rating suggests there might be some room for minor enhancements. Overall, the feature performs well and meets user expectations effectively.

- **Plan Notifications:** The chart below summarizes the feedback on the user interface of the plan notifications, based on the description: *"Providing a overview of approaching tasks and events, flagging plans nearing their due dates and delivering real-time alerts."*:



- **Survey Results** A total of 7 responses were collected, and the distribution of ratings is as follows:

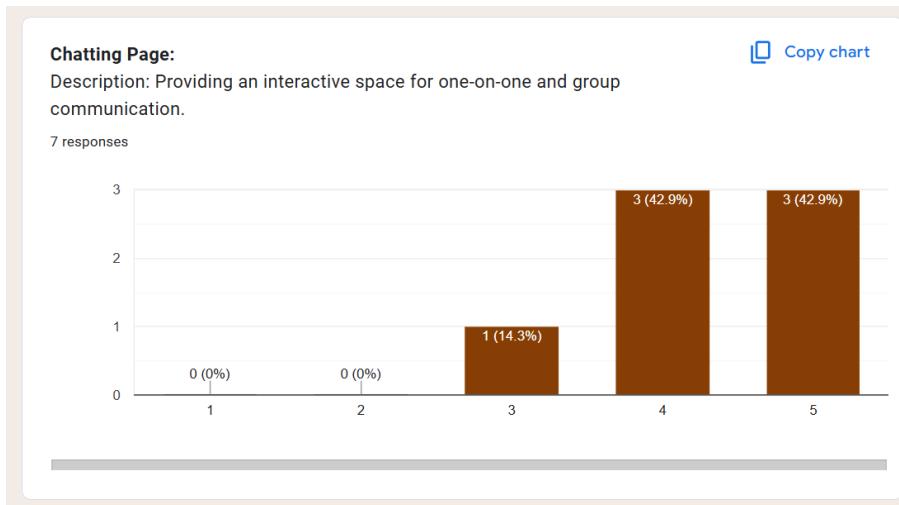
- * **1 point:** 0 response (0%)
- * **2 points:** 0 responses (0%)
- * **3 points:** 1 responses (14.3%)
- * **4 points:** 4 response (57.1%)
- * **5 points:** 2 responses (28.6%)

- **Analysis**

- * Positive Feedback: Only 28.6% of respondents rated the feature as "very good," indicating that a noteworthy portion of users are highly pleased with it compared to the additional 57.1% rating the notification as simply "good".
- * Moderate Feedback: One respondent (14.3%) provided a "moderate" rating, pointing to possible areas for minor refinements.
- * No Negative Feedback: There were no responses suggesting minor problems or a need for significant changes (ratings of 1 or 2).

- **Conclusion:** The feedback for the Plan Notifications feature is predominantly positive. The majority of users find it only good rather than very good, showing that while the system is well-received and effective, there is still room for excellency. The single moderate rating suggests there might be some room for minor enhancements, but overall, the feature performs well and meets user expectations.

- **Chatting Page:** The chart below summarizes the feedback on the user interface of the chatting page, based on the description: "*Providing an interactive space for one-on-one and group communication.*":



– **Survey Results** A total of 7 responses were collected, and the distribution of ratings is as follows:

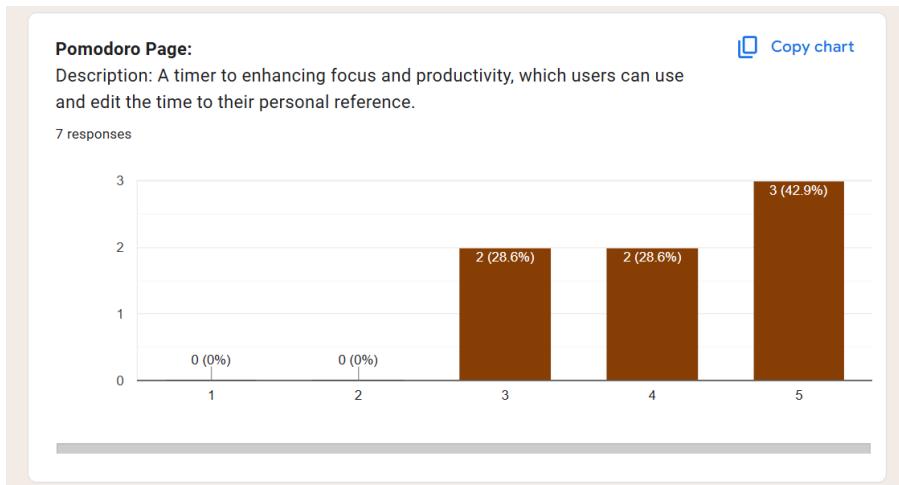
- * **1 point:** 0 response (0%)
- * **2 points:** 0 responses (0%)
- * **3 points:** 1 responses (14.3%)
- * **4 points:** 3 response (42.9%)
- * **5 points:** 3 responses (42.9%)

– **Analysis**

- * Positive Feedback: Three respondents (42.9%) rated the Chatting Page as "very good," indicating strong approval of its design and functionality while another three respondents (42.9%) rated it as "good," showing general satisfaction with the feature.
- * Moderate Feedback: One respondent (14.3%) rated it as moderate, suggesting there may be areas for minor refinement.
- * No Negative Feedback: None of the respondents rated the Chatting Page below three, which indicates no significant dissatisfaction with the feature. (ratings of 1 or 2).

– **Conclusion:** The feedback for the Chatting Page demonstrates that the majority of users are satisfied with its design and functionality. While no major concerns were raised, there are minor areas that could be improved to enhance the user experience further as suggested by the one moderate response.

- **Pomodoro Page:** The chart below summarizes the feedback on the user interface of the pomodoro page, based on the description: "*A timer to enhancing focus and productivity, which users can use and edit the time to their personal reference.*" :



- **Survey Results** A total of 7 responses were collected, and the distribution of ratings is as follows:

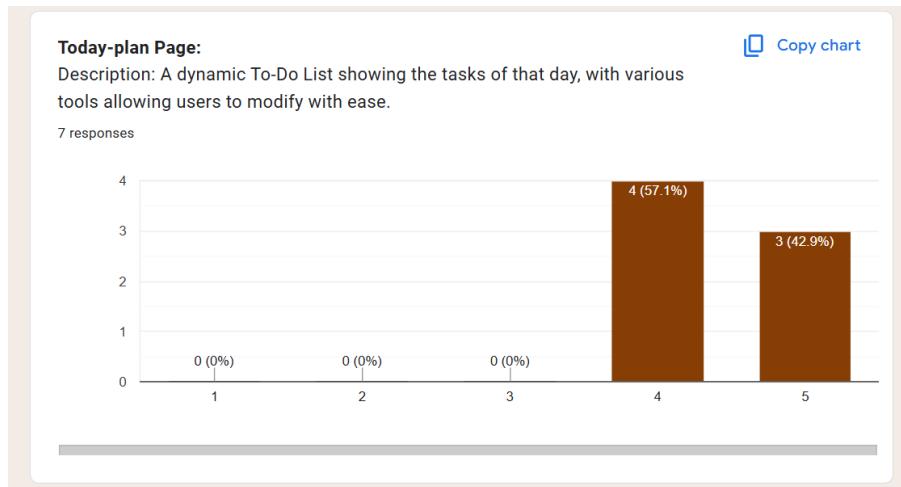
- * **1 point:** 0 response (0%)
- * **2 points:** 0 responses (0%)
- * **3 points:** 2 responses (28.6%)
- * **4 points:** 2 response (28.6%)
- * **5 points:** 3 responses (42.9%)

- **Analysis**

- * Positive Feedback: A large portion of respondents (42.9%) rated the Pomodoro Page as very good, indicating strong satisfaction with its functionality, while another 28.6% of respondents agreed but rated it as good, suggesting that they believe the page has yet to reach excellency.
- * Moderate Feedback: 28.6% of respondents rated it as moderate, indicating that there can be some areas for enhancement.
- * No Negative Feedback: No respondents rated the page as having minor problems or needing significant improvements (ratings of 1 or 2).

- **Conclusion:** The feedback for the Pomodoro Page is predominantly positive. The majority of users find it very good or good, showing satisfaction with its ability to enhance focus and productivity as described. Nonetheless, the moderate responses suggest that there might be some areas for minor improvements, but overall, the feature is effective and meets user expectations.

- **Today-plan Page:** The chart below summarizes the feedback on the user interface of the today-plan page, based on the description: "*A dynamic To-Do List showing the tasks of that day, with various tools allowing users to modify with ease.*"



– **Survey Results** A total of 7 responses were collected, and the distribution of ratings is as follows:

- * **1 point:** 0 response (0%)
- * **2 points:** 0 responses (0%)
- * **3 points:** 0 responses (0%)
- * **4 points:** 4 response (57.1%)
- * **5 points:** 3 responses (42.9%)

– **Analysis**

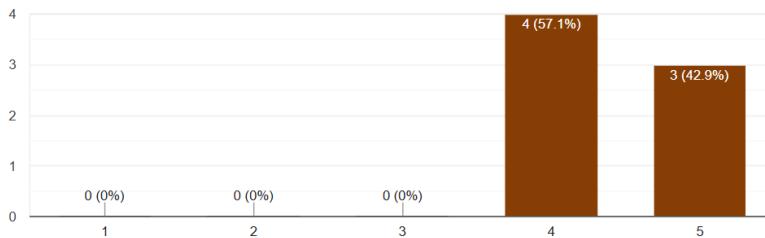
- * Positive Feedback: The majority of respondents (57.1%) rated the Today-plan Page as good, and a significant portion (42.9%) rated it as very good, indicating that the page is generally well-received and meets user expectations, with most users finding it effective or satisfactory.
- * No Moderate or Negative Feedback: There were no ratings indicating moderate satisfaction, minor problems, or significant needs for improvement, which is a very positive sign.
- **Conclusion:** The feedback for the Today-plan Page is overwhelmingly positive, and the absence of moderate or negative feedback suggests that the feature is well-received and meets user expectations effectively. However, the slightly higher percentage for "good" suggests there may still be room for minor improvements to make it even better for users.
- **Upcoming-event Page:** The chart below summarizes the feedback on the user interface of the upcoming-event page, based on the description: "*A dynamic To-Do List showing the upcoming tasks of, where users can switch between viewing their tasks for the week or for the month.*" :

Upcoming-event Page:

Description: A dynamic To-Do List showing the upcoming tasks of, where users can switch between viewing their tasks for the week or for the month.

[Copy chart](#)

7 responses



- **Survey Results** A total of 7 responses were collected, and the distribution of ratings is as follows:

- * **1 point:** 0 response (0%)
- * **2 points:** 0 responses (0%)
- * **3 points:** 0 responses (0%)
- * **4 points:** 4 response (57.1%)
- * **5 points:** 3 responses (42.9%)

- **Analysis**

- * Positive Feedback: The feedback for the Today-plan Page is overwhelmingly positive, with 100% of respondents rating it as either "Very Good" or "Good." Specifically, 42.9% rated it as "Very Good" and 57.1% as "Good".
- * No Moderate or Negative Feedback: There were no ratings lower than 4, suggesting that no significant issues were identified.

- **Conclusion:** All respondents either find the Upcoming Page either good or very good in allowing them to view the upcoming to-do list. Overall, the page appears to meet user expectations and performs effectively, with only a slight opportunity for further enhancement based on the slightly higher number of "Good" ratings.

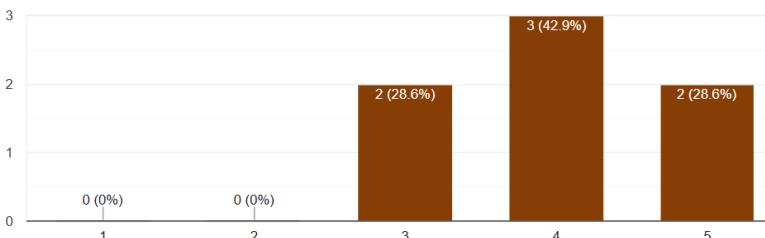
- **Team-project Page:** The chart below summarizes the feedback on the user interface of the team-project page, based on the description: "*A collaborative workspace connecting team members to view, assign and track tasks collectively.*" :

Team-project Page:

Description: A collaborative workspace connecting team members to view, assign and track tasks collectively.

[Copy chart](#)

7 responses



- **Survey Results** A total of 7 responses were collected, and the distribution of ratings is as follows:

- * **1 point:** 0 response (0%)
- * **2 points:** 0 responses (0%)
- * **3 points:** 2 responses (28.6%)
- * **4 points:** 3 response (42.9%)
- * **5 points:** 2 responses (28.6%)

- **Analysis**

- * Generally Positive Feedback: The majority of respondents rated the Team-project Page positively, with 42.9% giving it a rating of 4 and 28.6% giving it a rating of 5.
- * Moderate Feedback: 28.6% of respondents rated the page as moderate, which suggests there are some areas where improvements could be made.
- * No Negative Feedback: There were no ratings indicating that the interface needs significant improvement (ratings of 1 or 2).

- **Conclusion:** The feedback for the Team-project Page is overall positive. The majority of users find it good or very good, indicating that the collaborative workspace is effective in helping team members view, assign, and track tasks collectively. However, the moderate feedback suggests there might be some areas for enhancement to boost user satisfaction further.

- **General Conclusion:** With the help of our respondents' general feedback:

Your general feedback on Noteflow user interface?

5 responses

user-friendly interface, theme bring relax feeling
Không có
Has good designs with great colors but some pages have small text, small button to click
Pretty good, easy to use
good

we have reached the conclusion that overall, NoteFlow has a user-friendly user interface with good design, inferring from the majority of respondents rating each pages positive and the lack of negative responses. However, the unexpectedly high amount of moderate and good ratings shows that there is still room for improvement in our user interface, one of them, as a feedback suggests, is the sizes of the text and some components are quite small, making the user experience unsatisfactory. We are truly grateful for these feedback will take them into consideration for our front-end implementation.

7 Task 5: Implementation - Sprint 2

7.1 Set Up

All the necessary applications, environment details, and instructions for running the demo application are included in the `README.med` file on our GitHub repository. Additionally, we have provided account credentials for both user roles, allowing everyone to try out the app.

7.2 Welcome Page



Figure 1: Welcome Page.

The homepage is designed with a clean and intuitive interface to enhance user experience. It features a welcoming banner with the tagline "*Note taking made simple*" and highlights the purpose of the platform. Key sections on the homepage include:

- A visually appealing description of features such as goal-setting, task planning, and note-taking. Each section uses icons and short explanations to guide users through the platform's functionalities.
- A dedicated section for organizing tasks and writing notes, showcasing options to structure daily activities effectively.
- An engaging contact form at the bottom to allow users to easily reach out for support or inquiries.

7.3 Login and Sign up

7.4 Welcome Page

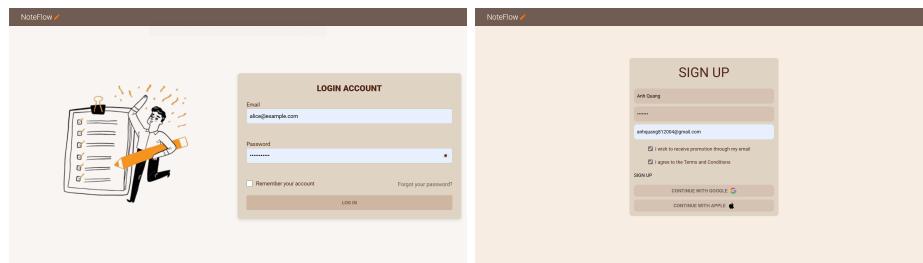


Figure 2: Login and Sign up page.

The application provides a streamlined user authentication process through its *Login* and *Signup* functionalities. These interfaces are designed with simplicity and user-friendliness in mind, ensuring a seamless experience for users.

7.4.1 Signup

The *Signup* interface allows new users to register for the application by providing essential details such as their name, email, and password. Additional options include the ability to opt-in for promotional emails and the mandatory agreement to the application's Terms and Conditions. Users can also choose to sign up using third-party platforms like Google or Apple, offering convenience and flexibility.

7.4.2 Login

The *Login* interface is tailored for returning users. They can access their account by entering their registered email and password. Features such as a *Remember Me* option enhance the user experience by enabling quicker access on future visits. For users who forget their password, a *Forgot Password* link is provided to assist in recovering account access.

Both interfaces emphasize a clean and intuitive design, promoting ease of use and efficiency in account management.

7.5 Homepage

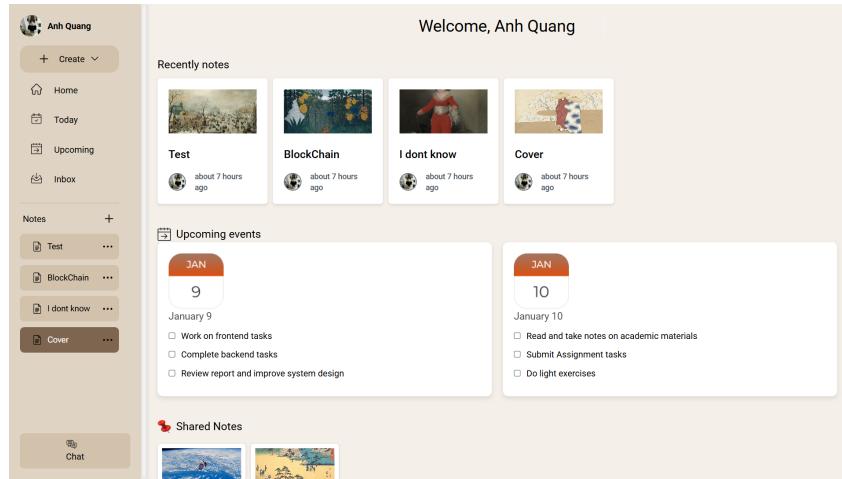


Figure 3: Homepage

The *Homepage* serves as the central hub for user interaction, providing an overview of recent activities and quick access to essential features. It includes the following elements:

- **Recently Notes:** Displays a grid of recently created or modified notes, each showing a preview and the time of its last update.
- **Upcoming Events:** Highlights scheduled tasks and events for the current and following days in a calendar-style layout.
- **Shared Notes:** Allows users to view notes shared with them by collaborators.

This layout ensures users have easy access to important information and can efficiently manage their tasks and notes.

7.6 Plan Component

The *Plan Component* organizes the user's tasks and events into manageable views, facilitating streamlined planning and tracking.

7.6.1 Make Plan

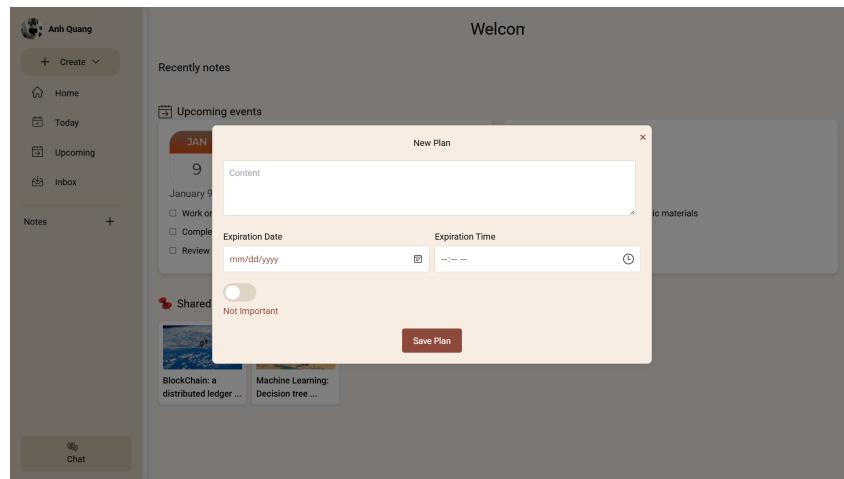


Figure 4: Make Plan

The *Make Plan* feature allows users to create new tasks or events. Users can:

- Enter the task content in a designated text box.
- Specify an expiration date and time using a date-time picker.
- Mark the task as *Important* or *Not Important*.
- Save the plan for integration into the task list.

7.6.2 Today Plan



Figure 5: Today Plan

The *Today Plan* section lists all tasks and notes scheduled for the current day. Tasks are categorized into:

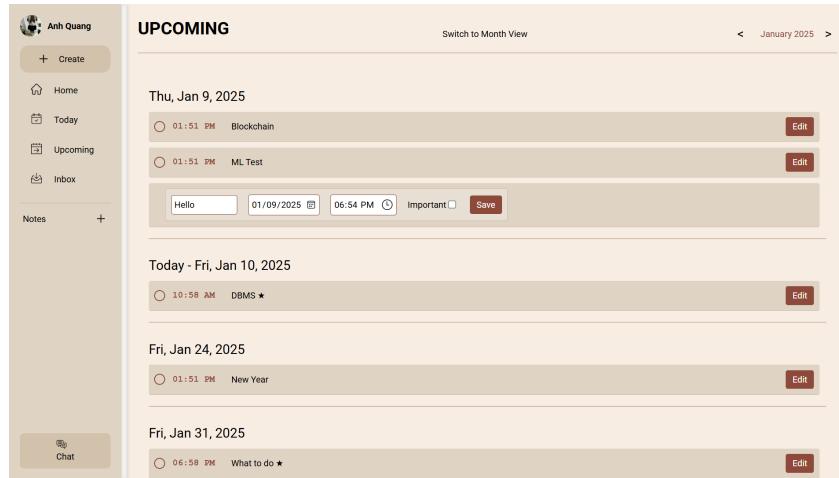
- **Important:** High-priority tasks displayed at the top.
- **Not So Important:** Tasks of lower priority shown separately for easy distinction.

Each task includes an *Edit* button for quick updates.

7.6.3 Upcoming Plan

The *Upcoming Plan* organizes tasks by date, offering detailed and structured views:

- **Day View:** Displays tasks for a specific day in chronological order.

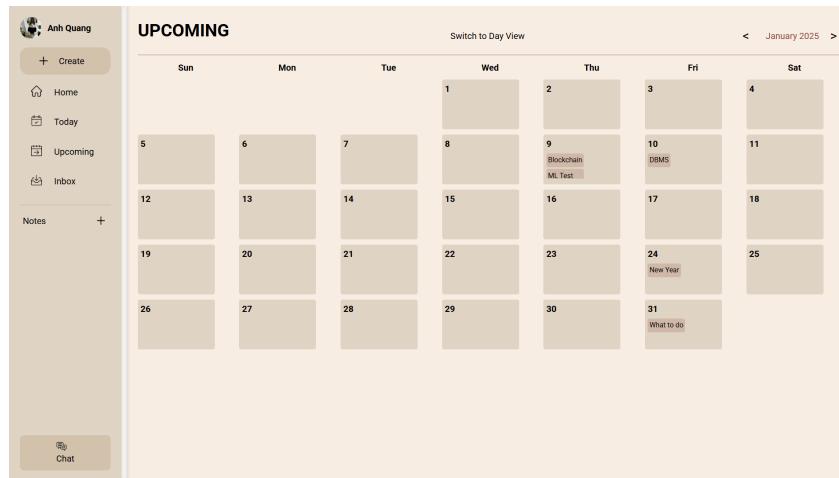


The screenshot shows the 'UPCOMING' view for January 2025. On the left is a sidebar with user profile, navigation links (Home, Today, Upcoming, Inbox), Notes, and Chat. The main area has a header 'UPCOMING' with 'Switch to Month View' and date navigation. It lists tasks for specific dates:

- Thu, Jan 9, 2025:** 01:51 PM Blockchain, 01:51 PM ML Test. Buttons: Edit, Save.
- Today - Fri, Jan 10, 2025:** 10:58 AM DBMS ★. Button: Edit.
- Fri, Jan 24, 2025:** 01:51 PM New Year. Button: Edit.
- Fri, Jan 31, 2025:** 06:58 PM What to do ★. Button: Edit.

Figure 6: Day View

- **Month View:** Presents an overview of tasks distributed across the current month.



The screenshot shows the 'UPCOMING' view in Month View mode for January 2025. The layout is a grid where each cell represents a day of the week. Days with tasks are highlighted in light brown. Tasks from Figure 6 are visible in their respective calendar slots:

Sun	Mon	Tue	Wed	Thu	Fri	Sat
			1	2	3	4
5	6	7	8	9 Blockchain ML Test	10 DBMS	11
12	13	14	15	16	17	18
19	20	21	22	23	24 New Year	25
26	27	28	29	30	31 What to do	

Figure 7: Month View

Tasks can be marked as *Important* and edited directly from this view, ensuring flexibility in task management.

Month View: Users can see all tasks and events at a glance in a calendar format, making it easy to plan ahead.

Day View: Focuses on the tasks scheduled for a single day, offering a concise and clear presentation of activities.

The *Plan Component* is designed to provide users with a dynamic and efficient planning experience, tailored to their preferences and organizational needs.

7.7 Note Component

The *Note Component* is designed to provide users with a versatile and user-friendly interface for creating, managing, and sharing notes. It includes the following features:

7.7.1 Note Interface

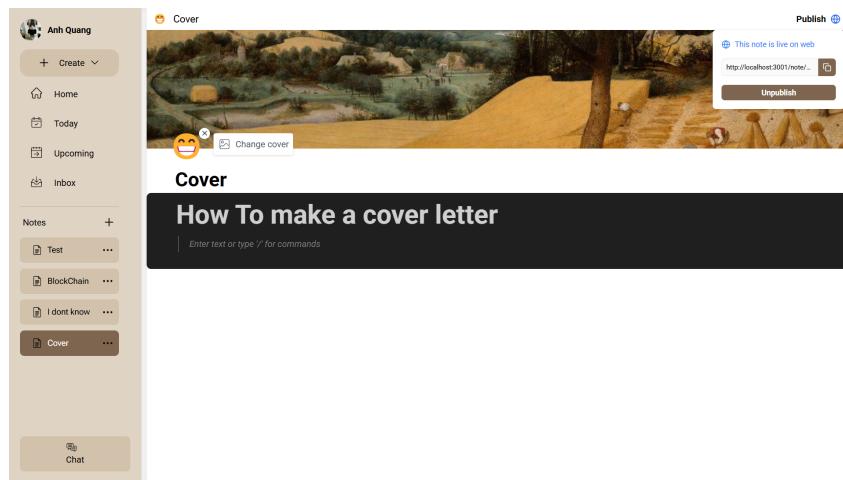


Figure 8: Note Interface

The *Note Interface* provides a comprehensive environment for note-taking and customization. Users can:

- **Create New Notes:** Use the *Create* button to add a new note to the list. Each note is displayed with a unique title and timestamp.
- **Edit Content:** Modify the note title and body, enabling users to update or enhance their notes as needed.
- **Customize Appearance:**
 - Change the cover image to suit the note's context.
 - Add an emoji to personalize the note.
- **Publish Notes:** Users can publish their notes to share with others via a unique link. Published notes include metadata such as the creation and last updated timestamps.

7.7.2 Share Note

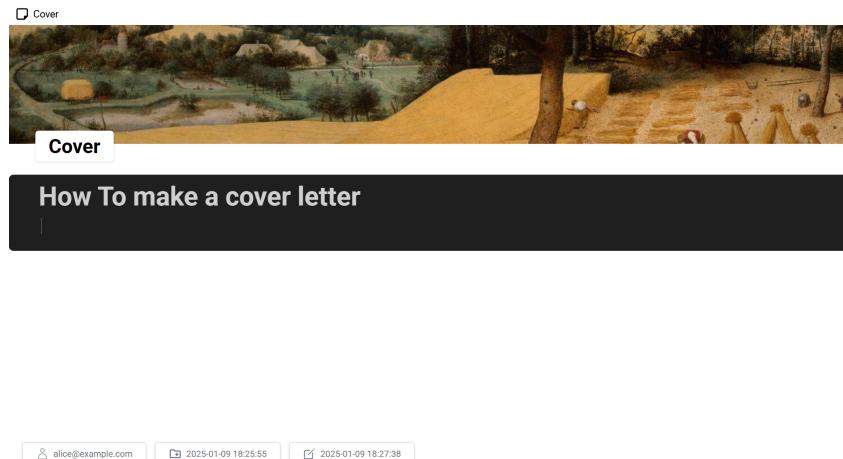


Figure 9: A shared note

The *Share Note* functionality allows users to collaborate by sharing their notes with others. Once a note is published, users can:

- Share the note using the generated link.
- Unpublish the note if collaboration is no longer required.

This feature is particularly useful for collaborative projects, allowing seamless sharing and editing among multiple users.

7.8 Settings

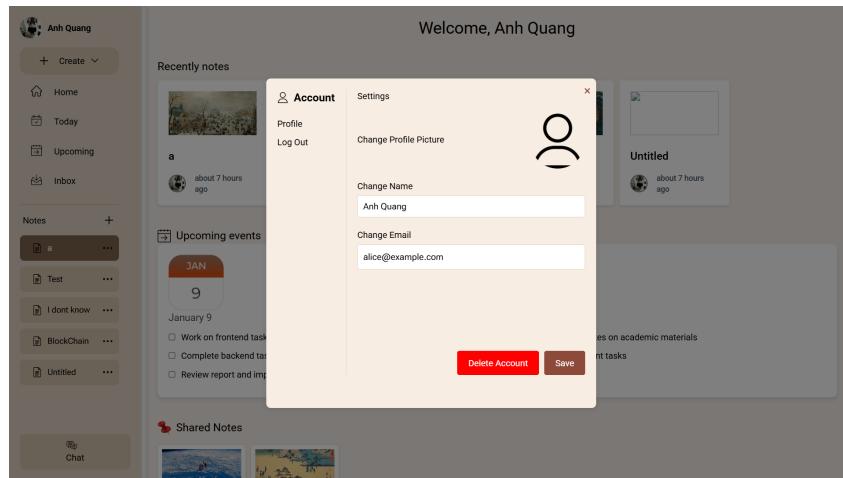


Figure 10: Setting interface

The *Settings* section provides users with options to manage their account and personalize their experience. Key features include:

- **Profile Management:** Users can update their profile information such as username, email, or avatar.
- **Preferences:** Adjust application preferences, including themes, notifications, and display options.
- **Log out and deletion:** Users can also log out or delete their account if they want to.

The *Settings* menu ensures users have full control over their account and can tailor the application to their needs.

7.9 Lesson Learned

The implementation of the NoteFlow project provided us with valuable insights and experiences that enhanced our technical and personal skills. Below are the key lessons we learned during the project:

- **Figma Design:** One of the initial and crucial stages of the project was designing a mock-up UI using Figma. This tool allowed us to create detailed and interactive prototypes of the user interface. By doing so, we were able to visualize the application structure and workflows before diving into the development process. This approach significantly saved time during the implementation phase and ensured clarity in aligning team expectations with the final product design.
- **API Design:** Designing and classifying APIs effectively was another important lesson. We learned the importance of structuring APIs in a modular, consistent, and secure manner. By adhering to RESTful principles and using FastAPI for the backend, we were able to create efficient and scalable APIs. This experience emphasized the role of proper documentation and testing to ensure smooth integration between the frontend and backend components.



• **Time Management:** Time management played a critical role throughout the development process. While we made substantial progress in implementing core features, some functionalities of the system remained unfinished due to challenges in effectively allocating time. This experience taught us the importance of prioritizing tasks, setting realistic deadlines, and maintaining a balance between quality and speed. We view this as a valuable lesson to improve our project management skills in future endeavors.

• **Use of New Technology:** The NoteFlow project provided us with the opportunity to explore and utilize new technologies, which enhanced our understanding of modern software development practices. We used:

- **React for Frontend:** React allowed us to build a dynamic and responsive user interface with reusable components, improving both development efficiency and user experience.
- **FastAPI for Backend:** FastAPI enabled us to create a high-performance and modern backend, supporting asynchronous programming and simplifying API creation.
- **PostgreSQL for Database:** PostgreSQL served as a reliable and robust relational database for storing and managing user data, ensuring data consistency and scalability.

Working with these technologies not only strengthened our technical expertise but also helped us understand how to integrate multiple tools into a cohesive system to deliver a standard software project.

- **Collaboration and Teamwork:** Throughout the project, collaboration and teamwork were essential. Communicating effectively, assigning tasks, and resolving conflicts in a timely manner were pivotal in maintaining project momentum. We learned the importance of fostering a supportive team environment and leveraging each team member's strengths.
- **Problem Solving and Adaptability:** Finally, the project challenged us to solve complex problems and adapt to unforeseen circumstances. Whether it was debugging code, handling integration issues, or managing scope creep, we gained resilience and adaptability, which are essential skills for our future careers.

The lessons learned from the NoteFlow project have provided us with a strong foundation to approach future projects with greater confidence and competence, ensuring continuous growth as software developers.