Software Product Release

PlanBee

Prepared by

Student	ID
Joud Almutairi	443200544
Sara Alzayed	443203107
Sereen Alhmoud	443200463
Alhanouf Aldakel Allah	443200999

Supervised by
L. Ghaida AlFayez

Table of Contents

1 Chapter 1: Introduction

With more responsibilities and projects to handle, effective task management is critical for both individuals and teams. Whether in academic, professional, or personal contexts, staying organized while collaborating efficiently can be challenging. The inability to manage tasks effectively often leads to missed deadlines, decreased productivity, and miscommunication. This project addresses these pain points by focusing on the field of task and project management.

PlanBee is designed as a solution to these challenges. It's a task and project management application that allows users to add tasks, track their completion, and collaborate on shared projects with team members. The application supports the creation of personal task lists, group projects, and real-time collaboration, which enhances both productivity and teamwork. By following Agile methodology, this application aims to provide a user-centered experience that adapts to the needs of dynamic, fast-paced work environments.

The significance of developing such an application lies in two main aspects: on a local level, it assists students and professionals in managing their projects more efficiently; globally, it contributes to improving team collaboration and project success across industries. With remote and hybrid work becoming more prevalent, the need for effective digital collaboration tools has never been greater. PlanBee addresses this need by offering a streamlined solution for task management.

This proposal will first outline the problem, highlighting the challenges of inefficient task and project management. It will then present PlanBee as the solution, emphasizing its ability to streamline task tracking and team collaboration. The product vision will explain how PlanBee supports users in staying organized and productive. Additionally, the product roadmap will detail the development stages using Agile methodology, and the objectives will define the goals for product functionality, project execution, and team learning. The scope will clarify the app's features and limitations, ensuring a focused and realistic development approach. We'll introduce the Scrum Team involved, including the Product Owner, Developers, Scrum Master, and Stakeholders. Lastly, we'll list the references.

1.1 The Problem

Managing multiple tasks and projects, whether academic or professional, is often overwhelming, leading to missed deadlines and decreased productivity. For example, students and professionals struggle to organize tasks, assign responsibilities, and track project progress efficiently, resulting in wasted time and miscommunication.

1.2 The Solution

PlanBee simplifies task and project management by allowing users to organize daily tasks and collaborate on group projects in one place. Users can create project-specific workspaces, assign tasks to team members, and track progress, enhancing productivity and ensuring accountability.

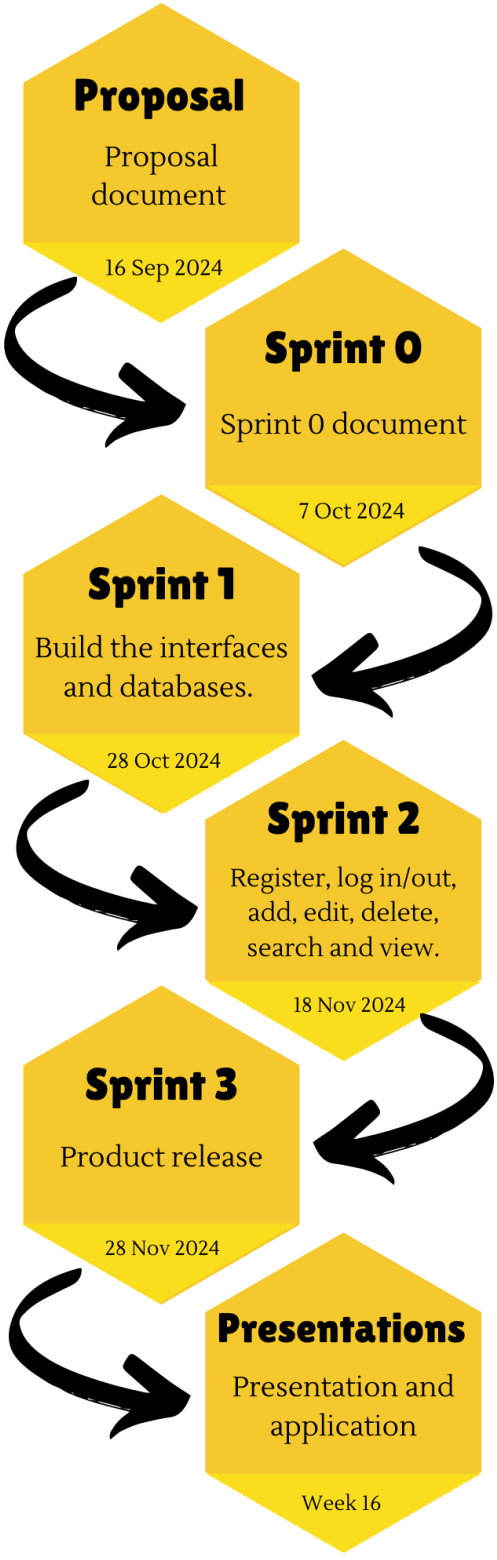
1.3 The Product

1.3.1 Product Vision

For **individuals and teams** who **need to efficiently manage tasks and collaborate on projects with friends or colleagues** the **PlanBee** app is a **task management and collaboration app** that **helps users stay organized, track progress, and work together seamlessly in real-time, making project collaboration easy and fun.** Unlike **other task management apps that focus solely on individual tasks, such as 'ToDo List'**[1]. Our product **allows easy real-time collaboration for tasks between a working team.**

1.3.2 Product Roadmap

Figure 1 - Project Roadmap



1.3.3 Objectives

- **Product (customer focus-value)**:

PlanBee is designed to simplify how individuals and teams organize their tasks and projects. Managing tasks can often be a challenge, whether on a personal level or within a team.

With PlanBee, users can effortlessly create, organize, and manage their tasks with ease. Whether you're working alone or collaborating with a team, the app enhances productivity and streamlines project management.

It addresses the needs of both individuals and teams and includes the following key features:

1. **Easily Create Tasks**:
   Allow users to quickly and easily create personal tasks to improve their organization.

2. **Manage and Edit Tasks:**
   Enable users to update and delete tasks easily, keeping their tasks organized and up to date.

3. **Collaborative Features:**
   Allow users to form project groups and work on shared tasks, enhancing teamwork.

4. **Account Creation:**
   Provide a feature for users to create personal accounts, manage their tasks, and log out as needed

.

5. **Task Search Capability**:
   Offer a quick search feature for tasks to make them easily accessible and manageable.

6. **User-Friendly Interface:**
   Provide an intuitive and easy-to-navigate user interface to enhance the user experience.

7. **Task Notifications:**
   Provide notifications to alert users about upcoming deadlines or important updates on their tasks.

- **Project (solution focus-plan):**[2]

1. **Domain Analysis**:
   Review task and project management practices to define PlanBee's core features.

2. **Requirements Gathering and Analysis**:
   Identify and document user needs and challenges; prioritize requirements.

3. **Interface Design:**
   Create initial designs and wireframes for the user interface.

4. **Iterative Development** :
   Develop core features incrementally, testing and refining based on feedback.

5. **Testing and Feedback Integration**:
   Test features thoroughly, collect user feedback, and make necessary improvements.

6. **Advanced Feature Integration:**
   Add advanced features like real-time collaboration and task notifications after core functionality is stable.

7. **Documentation and Final Reporting**:
   Document the development process, testing results, and feedback; prepare a final report with achievements and recommendations.

- **Learning (student focus):**
  - **Use new IDE**: Gain hands-on experience with **Flutter Flow** for developing and managing the PlanBee app.
  - **Use new tools**: Work with tools like **GitHub** for project management, enhancing team workflow and collaboration.
  - **Learn** how to test the application.
  - **Learn** and implement Agile app development methodology.
  - **Learn** how users are stored within an application through database.
  - **Learn** to use UI and UX principles.

1.3.4 Scope

PlanBee will be available on all iOS mobile devices versions 12 or higher [3]. Each user will be able to create their own account and log in or out as needed. The app will provide a task overview through a dashboard on the user's homepage, making it easy to manage and track personal tasks. Users will have the ability to add new tasks, either keeping them private or choosing to collaborate by sharing tasks with colleagues in a shared workspace where everyone can see and track the tasks in real-time. The app will also include a search function to help users quickly locate specific tasks.

On the contrary, PlanBee will not support any language other than English, limiting its use to English-speaking users. It will only be available on iOS devices, excluding other platforms such as Android or web-based versions. For the task collaboration feature, the app users will need internet connection to access and collaborate on tasks. No advanced project management features, such as Gantt charts, resource management, or third-party integrations, are within the current scope of development.

1.4 The Scrum Team

Table 1 - Scrum Team

Scrum Team	
Product Owner:	Joud Almutairi
Developers:	Alhanouf Aldakel Allah Joud Almutairi Sara Alzayed Sereen Alhmoud
Scrum Master (SM):	L. Ghaida Alfayez
Stakeholders:	L. Ghaida Alfayez

2 Chapter 2: Domain Analysis

The domain of this project is task and project management, specifically focusing on a mobile application for task tracking and collaboration called PlanBee. Task management has become a critical aspect of both personal and professional life, particularly with the rise of remote and hybrid work environments.

To gather information for this domain, a combination of different resources was utilized. This includes:

• **Domain experts**: Professionals who regularly use project management tools, like students, office workers, and project managers, provided insights into the daily challenges of task tracking and team collaboration.
• **Books and Online Resources:** Documentation of existing project management systems such as 'Trello'[4] and 'TickTick'[6] helped in understanding their functionality and user interface.
• **Existing software:** We analyzed several similar apps like 'Trello'[4], 'Microsoft to Do'[5] , and 'TickTick'[6] to gather knowledge about current market offerings and user expectations.

2.1 Terminology

1. **Task Dashboard:** A central area within the app where users can view all their tasks, including personal and collaborative ones, in an organized manner.
2. **Workspace:** A virtual space where users can collaborate on tasks with others. Each workspace is project-specific and allows members to share, assign, and track tasks in real-time.
3. **Real-Time Collaboration:** A feature that allows users to update and view changes made to tasks instantly. It ensures that all users in a workspace see updates as they happen without any delays.
4. **Task Assignment:** The process of assigning a specific task to a user or team member within a workspace, indicating who is responsible for completing it.
5. **Deadline Notification:** An alert that notifies users about upcoming deadlines or due dates for their tasks, helping them stay on track and manage their time effectively.**Personal Task:** A task that is visible only to the user who created it, ensuring privacy for personal activities or responsibilities that are not part of a collaborative project.
6. **User Account:** The profile created by each user when they sign up for the app, which stores personal information, task history, and preferences.
7. **Task Prioritization:** The process of assigning priority levels to tasks, helping users focus on the most critical activities first.
8. **Task collaboration:** The process where multiple individuals work together to plan, assign, and complete tasks within a shared project. It involves organizing tasks, communicating progress, and ensuring that everyone involved has access to the necessary information and tools to complete their responsibilities efficiently.
9. **Milestones:** Key markers or goals within a project timeline that represents significant achievement.

10. **User Roles**: Different permissions and access levels assigned to users within PlanBee Member Leader, Team members, and User to manage collaboration effectively
11. **Progress Tracking**: The functionality that enables users to monitor the completion status of tasks within PlanBee

2.2 General Domain Knowledge

In the domain of task and project management software, certain general knowledge is essential. For example, features like task creation, assignment, and collaboration. Tracking in real-time refers to the ability to monitor the progress of tasks or activities as they happen, without any delay. In the context of an app like PlanBee, real-time tracking means that when a user updates or modifies a task (such as marking it as completed, or changing its details) these changes are instantly visible to all collaborators.

Familiarity with task collaboration processes, user account management, and the importance of notifications is fundamental to creating an efficient tool. Task collaboration processes refer to the structured methods by which individuals work together to complete tasks within a project. These processes involve assigning, sharing, and coordinating tasks among team members to ensure

efficient completion of goals. These processes help teams stay aligned, reduce miscommunication, and ensure that projects are completed on time and within scope.

2.3 Customers and Users

The primary users of PlanBee include:

• **Students:** Groups working on academic projects can benefit from PlanBee's real-time collaboration features.
• **Professionals:** Individuals managing multiple tasks within their jobs can use the app to organize and track work.
• **Freelancers:** Users who need to manage their independent work while collaborating with clients will find the collaborative workspaces helpful.

Potential users may also include small teams and organizations that prefer a simpler, user-friendly task management tool over complex systems. The attitude of these users tends to prioritize simplicity, efficiency, and mobile accessibility.

2.4 The Environment

PlanBee is a mobile application that operates on the iOS platform and is available for download from the Apple App Store. Users will need an iPhone or iPad running iOS version 12.0 or higher to install and use the app.

2.5 Tasks and Procedures

This section described how users currently manage personal and collaborative tasks, highlighting common practices and shortcuts that people use to stay organized.

Personal Task Management:

Users generally keep track of their personal tasks through simple to-do lists, often using note-taking apps, physical notebooks, or phone reminders. Some individuals prefer more informal methods like sticky notes or setting alarms. Users may skip updating their lists regularly or forget to mark tasks as complete, leading to outdated information. This often results in missing deadlines or forgetting critical tasks.

Collaborative Task Management:

Teams often manage project-based tasks by holding in-person meetings or using spreadsheets or shared documents to outline and track progress. Communication tools like email or messaging platforms are commonly used for updates and coordination. Team members might find difficulties arranging meetings, sometimes they rely on informal direct messages or phone calls to quickly address tasks, causing communication gaps and task misalignment.

Task Assignment:

Users often assign tasks manually by emailing or messaging team members. Tools like shared cloud documents may also be used, where team members self-assign or comment on tasks. Team members may fail to assign tasks explicitly, assuming others know what needs to be done. This can lead to confusion about who is responsible for specific tasks.

Tracking Progress:

Teams attempt to monitor progress using shared files or software tools that require manual updates. However, this process is often inconsistent, with some updates being missed or incomplete. Team members may neglect to track smaller updates, waiting until larger milestones are completed, which reduces overall visibility into the project's progress.

This section outlines how individuals and teams currently approach task management, revealing inefficiencies that PlanBee aims to streamline and automate for better productivity and collaboration.

2.6 Competing Software

The following table provides a comparative analysis of several task management applications: **Trello**[4], **Microsoft To Do**[5], **TickTick**[6], and **PlanBee**. Each application has unique features and capabilities designed to enhance productivity and collaboration among users. The table highlights key functionalities, such as task management, project management, real-time collaboration, task search, notifications, and whether all features are available for free. By examining these features, potential users can better understand how each application meets their needs and how **PlanBee** positions itself within the competitive landscape, especially in facilitating daily task management and project collaboration.

Table 2 - Competing software and their features comparing with PlanBee.

Feature	Trello	Microsoft To Do	TickTick	PlanBee
Task Management	✓	✓	✓	✓
Project Management	✓		✓	✓
Real-time Collaboration	✓		✓	✓
Task Search	✓	✓	✓	✓
Notifications	✓	✓	✓	✓
All Features Available for Free				✓
Supports multiple languages	✓	✓	✓	

2.7 Similarities Across Domains and Organizations

Understanding Generic vs. Specific Features in the Domain:

In the field of task and project management, there are several generic features shared among many applications, including **Trello**[4], **Microsoft To Do**[5], **TickTick**[6], and **PlanBee**. These features form the foundation of any task or project management tool and include:

1. **Task Management:** Most applications provide basic functionality to create, assign, and track tasks. This feature is essential and common across all similar systems, making it a generic feature in the domain.
2. **Collaboration:** Many systems allow multiple users to collaborate on projects or tasks, share information, and provide progress updates. This collaborative aspect is also widely available and can be considered a generic feature.

Specific Features and What Distinguishes the Domain:

While generic features form the foundation, specific features help distinguish systems from one another. In the case of PlanBee, several aspects set it apart from other task management tools:

1. **Intuitive User Interface:** PlanBee offers a user-friendly interface designed for smooth navigation, making it accessible to users of all skill levels.
2. **Customizable Task Views:** Users can personalize their task views according to their preferences, enhancing productivity and task visibility.
3. **Collaboration Features:** PlanBee includes advanced collaboration features, such as real-time editing and feedback options, fostering teamwork and communication.

By analyzing the comparison table, it becomes clear that while many tools share common (generic) features such as task management and collaboration, PlanBee's specific features, like its intuitive interface and customizable task views, differentiate it from other tools in the market. This distinction is crucial for creating software that meets the specific needs of the target audience while remaining marketable to a broader user base.

3 Chapter 3: Requirements Engineering

To gather the requirements for the “PlanBee” application, we used a variety of methods, including user interviews and surveys. Our requirement-gathering process began with structured interviews with three stakeholders, during which we asked five specific questions aimed at understanding their expectations regarding the application’s functionalities, ease of use, and features. The interviews provided qualitative insights into user needs, allowing us to delve deeper into their experiences and preferences.

After the interviews, we distributed a carefully designed survey containing six questions to a broader audience of participants, collecting a total of 40 responses. This approach helped us gain quantitative insights into user preferences and the challenges they face in task management. The surveys were designed to cover various aspects, including desired features, ease of use, and any obstacles users encounter in managing tasks.

The results obtained from both the interviews and the survey reveal important insights into user preferences and the challenges they face in task management, highlighting the strong demand for a professional task management application. The interviews with the three participants provided key insights into their experiences and preferences. One participant mentioned relying on basic tools like note-taking apps and messaging platforms, underscoring the need for improved task clarity and organization. Another participant, a college student, identified poor communication and task overlap as key issues, emphasizing that a real-time collaboration tool would enhance workflow and help maintain focus on task completion. The third participant discussed difficulties in tracking task progress and clarifying responsibilities among team members, highlighting the importance of instant updates and deadline notifications to boost productivity and reduce misunderstandings.

In addition to these qualitative insights, the survey results showed that 45% of participants regularly experience difficulties managing individual and group tasks, while 20% do so occasionally, and 5% reported no challenges, emphasizing the urgent need to improve task management skills among the majority.

Regarding the use of task management tools, 42.5% of participants indicated that they use specialized apps or tools to organize their daily tasks and group projects, while 57.5% confirmed they do not rely on any tools. This suggests a gap between the need for task management tools and the current reality of their usage, necessitating the development of innovative applications that meet users’ needs.

When asked about the challenges they face in organizing tasks and collaborating with their teams, the results showed that “distributing tasks among team members” was one of the biggest difficulties, with 72.5% of participants indicating they struggle with task distribution. This was followed by “tracking progress” and “effective communication,” where 67.5% of participants reported difficulties in these areas. From these results, it can be concluded that there is an urgent need to develop effective mechanisms for task distribution and ensure follow-up, facilitating smooth and effective collaboration among teams.

Furthermore, when participants were asked to assess the importance of real-time collaboration in managing group tasks, 75% expressed that this feature is essential. Regarding the features they prefer in task management tools, 87.5% placed importance on task assignment and distinguishing tasks nearing deadlines, while 72.5% valued progress tracking, 65% valued task distribution, 55% valued instant

updates, and 35% valued notifications and alerts. Finally, 97.5% of participants expressed interest in using an application that facilitates task management and group projects, reflecting a strong desire to improve work methods and collaboration.

Finally, in the requirements engineering section, we established a strong foundation for the "PlanBee" application by identifying user needs and expectations. We defined the key users of the system, including general users, team members, and member leaders, which allowed us to understand their specific requirements. A use case diagram was created to visually represent the interactions between users and the application, illustrating essential functionalities such as task assignment, progress tracking, and deadline management.

Additionally, we compiled a product backlog containing user stories and prioritized features that will guide the development process. The requirements engineering phase was crucial in shaping the development of the "PlanBee" application. By clarifying user roles, visualizing interactions, and organizing key features, we established a solid foundation that will guide the next steps in creating a powerful task management tool aimed at enhancing user productivity and collaboration.

3.1 System Users

The user base of PlanBee consists of three main types: general users, team members, and Members' Leader. Each type has different characteristics in terms of educational level, experience, and technical skills, with the app designed to be user-friendly for all users.

1. **General User:**

• Educational Level: At least a middle school education. They should be able to read and write and recognize words and letters in English.

• Experience: May be limited to moderate experience using task management tools. They use PlanBee for managing personal tasks or working on small group projects.

• Technical Skills: They have basic knowledge of using applications and smartphones.

• Gender: Includes both males and females.

2. **Team Member:**

• Educational Level: At least a middle school education. They should be able to read and write and recognize words and letters in English.

• Experience: Have experience working within teams and group projects. They use PlanBee to organize projects and assign tasks.

• Technical Skills: They have basic knowledge of using applications and smartphones.

• Gender: Includes both males and females.

3. Member Leader:

• Educational Level: Expected to have at least a middle school education, with some knowledge of team and project management. They should be able to read and write and recognize words and letters in English.

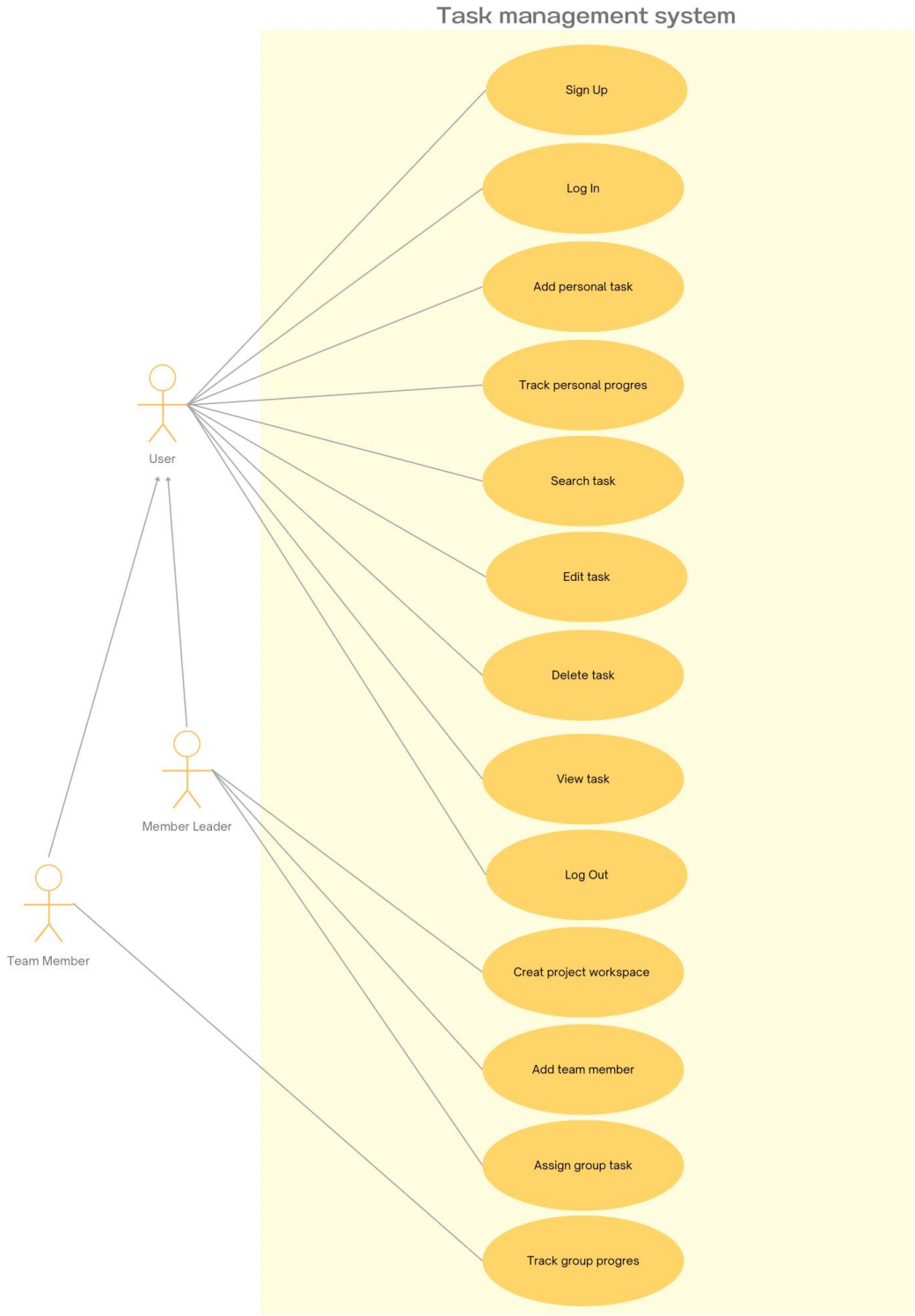
• Experience: Have some experience in coordinating projects and managing teams, but the app is designed to simplify task organization and management, even for those with minimal experience.

• Technical Skills: They have basic knowledge of using applications and smartphones.

• Gender: Includes both males and females.

3.2 Use Case Diagram

Figure 2 - Use Case Diagram Of PlanBee



3.3 Product Backlog

The product backlog is a prioritized list of features and tasks that define the product's functionality and release goals. It serves as a clear and shared guide for the development team, outlining what needs to be built and the sequence in which it should be developed. The product backlog is an essential component of the Scrum framework, visible to all project stakeholders.

Backlog items (PBIs), typically written as user stories, represent features or changes that deliver value to the user. These can range from new functionalities to improvements and defect fixes. Non-functional requirements, such as performance and security, are also included in the backlog.

For this project, we utilized Jira to manage and organize the backlog items, ensuring a structured approach to tracking and prioritizing the work.

3.4 Definition of Done

Table 3 - Definition of Ready for PlanBee user stories

Example Definition of Ready	
✓	Business value is clearly articulated
✓	Details are sufficiently understood
✓	Dependencies are identified; no blocking dependencies exist
✓	Team is appropriately staffed relative to the PBI
✓	Estimated and small enough to be completed during sprint
✓	Acceptance criteria are clear and testable
✓	Performance criteria, if any, are defined and testable
✓	Team understands how to demo the completed PBI

Functional Requirements:

Table 4 - Product backlog of functional requirements of PlanBee.

PBI	Size (Story Points)	Type	Acceptance Criteria
As a user, I want to **sign up** for a new account so that I can create and manage my tasks.	3	Feature	- As a nonregistered user who wants to sign-up, if I go to signup page and enter my name, username, email, password, and confirm password and click on sign up button, then I should be able to access the application. - As a nonregistered user who wants to sign-up, if I go to signup page and enter my name, username, email that already exists, password, and confirm password and click on sign up button, then I should receive an error message. - As a nonregistered user who wants to sign-up, if I leave any required fields (name, username, email that already exists, password, confirm password) and click on sign up button, then I should receive an error message. - As a nonregistered user who wants to sign-up, if I go to signup page and enter my name, username, email, password, and a different confirm password and click on sign up button, then I should receive an error message.
As a user, I want to **log in** to the system securely so that I can access my tasks and projects.	2	Feature	- If I go to the log-in page and enter my email and password and click on Log in, then the data associated to my user should be accessible. - If I go to the log in page and enter an incorrect email and a correct password and click on Log in, then log in fails with an error message that the email is wrong. - If I go to the log in page and enter my email and an incorrect password and click on Log in, then log in fails with an error message that the password is wrong.
As a user, I want to **log out** of the system so that my session is closed securely.	2	Feature	As a user, if I log out then my session should end immediately, and I should be redirected to the login page.
As a user, I want to **create** personal tasks so that I can track my individual progress.	3	Feature	As a user, if I create a task then it should appear in my task list immediately.
As a user, I want to **set deadlines** for my tasks so that I can stay on track.	2	Feature	As a user, if I set a deadline then I should receive a notification one day before the deadline.

As a user, I want to **edit** my tasks so that I can update them as needed.	2	Feature	As a user, if I edit a task then the updated task should appear in my task list immediately.
As a user, I want to **Track** my personal tasks so that I can monitor my progress efficiently.	3	Feature	As a user, if I add a new task to my personal task list, then the task should appear in my dashboard, and the task count should update accordingly.
As a user, I want to **delete** my tasks so that I can remove tasks that are no longer needed.	2	Feature	As a user, if I delete a task then the system should confirm the deletion before removing the task.
As a member leader, I want to **create** project workspaces so that I can organize group tasks.	3	Feature	As a member leader, if I create a workspace then it should be visible to assigned team members and track task progress.
As a members' leader, I want to **add** team members to a project workspace, so that I can ensure all team are included and can collaborate on the project tasks	3	Feature	As a member leader, if I enter the email of a team member and click "Add," then the system should send an invitation and the added member should appear in the project's team list.
As a members' leader, I want to **assign** tasks to specific team members so that everyone knows their responsibilities.	3	Feature	As a member leader, if I assign tasks then the assigned team members should be notified immediately.
As a user, I want to **receive notifications** for task updates so that I can stay informed.	3	Feature	As a user, if a task is updated and I don't have the app open then I should receive a notification banner.
As a team member, I want to **track** the progress of each task so that I can monitor the project's status.	3	Feature	As a team member, if I track a task then the progress should update in real-time on the dashboard.

Non-Functional Requirements:

Table 5 - Product backlog of non-functional requirements of PlanBee.

PBI (Non-functional Requirement)	Size (Story Points)	Type	Acceptance Criteria
As a user, I want the user interface to be intuitive and easy to use so that I can efficiently navigate the app.	3	Feature	As a user, if I use the app for the first time then user tests should show a 90% success rate for navigation, with no more than 3 clicks to access main features and quick iterations based on usability feedback.
As a user, I want the app to load within 3 seconds so that I can quickly access my tasks.	3	Feature	As a user, if I open the app then the dashboard should load in under 3 seconds in 95% of cases, tested with varying amounts of data and no noticeable lag during task loading.

4 Chapter 4: System Design

The System Design section consists of five key components, each playing an essential role in shaping the overall design of our system.

First, we present the selected **System Architecture** and provide a detailed diagram to explain it. Second, we include a comprehensive **Class Diagram** that highlights the structure of the system's classes, their relationships, and their methods. Third, we illustrate the **Data Design** using an ER Diagram, a database schema, and a detailed data dictionary. Fourth, we showcase the **Component Design** with pseudocode and effort estimates for functions like adding, editing, and deleting tasks. Finally, we demonstrate the **Interface Design** through a navigation diagram, annotated screen designs, and five UX enhancements that were integrated into the application.

These components, which include PlanBee's architecture, class design, data management strategy, component-level functionality, and user interface improvements, come together to deliver a comprehensive, efficient, and user-friendly solution tailored to the needs of our users.

4.1 System Architecture

The system architecture of an application plays a significant role in determining its performance, scalability, and maintainability. In this section, we will discuss the system architecture chosen for our application **PlanBee**.

The system architecture serves as the foundation of the entire application, defining how various components interact and communicate with one another to deliver the desired functionality and user experience. After a comprehensive evaluation of multiple design patterns, we decided on the **Client-Server Architecture** for **PlanBee** *(Figure 3)*. This architecture divides the system into two primary components: the client-side and the server-side.

The client-side component is responsible for delivering a user-friendly interface where users can create, edit, delete, and manage tasks and projects. It ensures an intuitive experience by efficiently capturing user inputs, displaying real-time updates, and organizing information in a clear and accessible manner.

On the other hand, the server-side component forms the backbone of the application by acting as a centralized system that manages all critical data. It includes the business logic and is responsible for responding to user requests, processing data, retrieving or updating records from the database, and maintaining the overall consistency of the application. This is achieved through Firebase services, which ensure secure and efficient data handling, including real-time synchronization across devices.

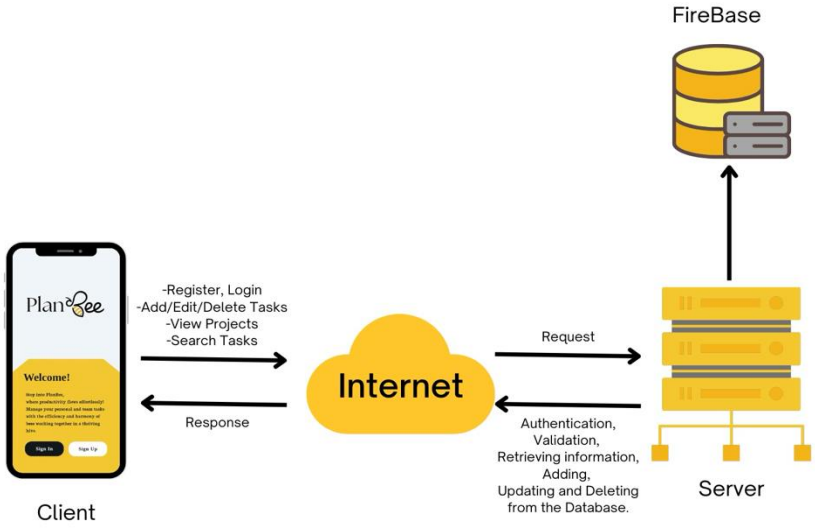
We chose the **Client-Server Architecture** for several reasons. First, it provides a clear separation between the **presentation layer (client-side)** and the **business logic and data management layer (server-side)**. This separation improves modularity and maintainability, making it easier to update or enhance individual components without impacting the overall system. Additionally, this architecture supports scalability by allowing multiple users to connect to the server simultaneously, efficiently handling large volumes of requests and data. The centralized nature of the server also enhances security by simplifying the implementation of authentication and access control measures, ensuring that only authorized users can access or modify sensitive information.

While selecting the appropriate architecture, we evaluated several alternatives but ultimately excluded them for various reasons. For example, the **Monolithic Architecture**, though simple to implement, was dismissed due to its lack of scalability and the challenges it poses in updating or modifying specific parts of the system without affecting the entire application. Similarly, the **Microservices Architecture** was excluded because its complexity exceeds the current needs of PlanBee, as it requires independent

management of multiple services. The **Peer-to-Peer Architecture**, commonly used in decentralized systems, was deemed unsuitable due to its lack of centralized data management and potential synchronization issues. Lastly, the **Pipe and Filter Architecture**, often utilized in linear data processing systems, was not a viable option since it lacks support for user interaction, a critical feature of PlanBee.

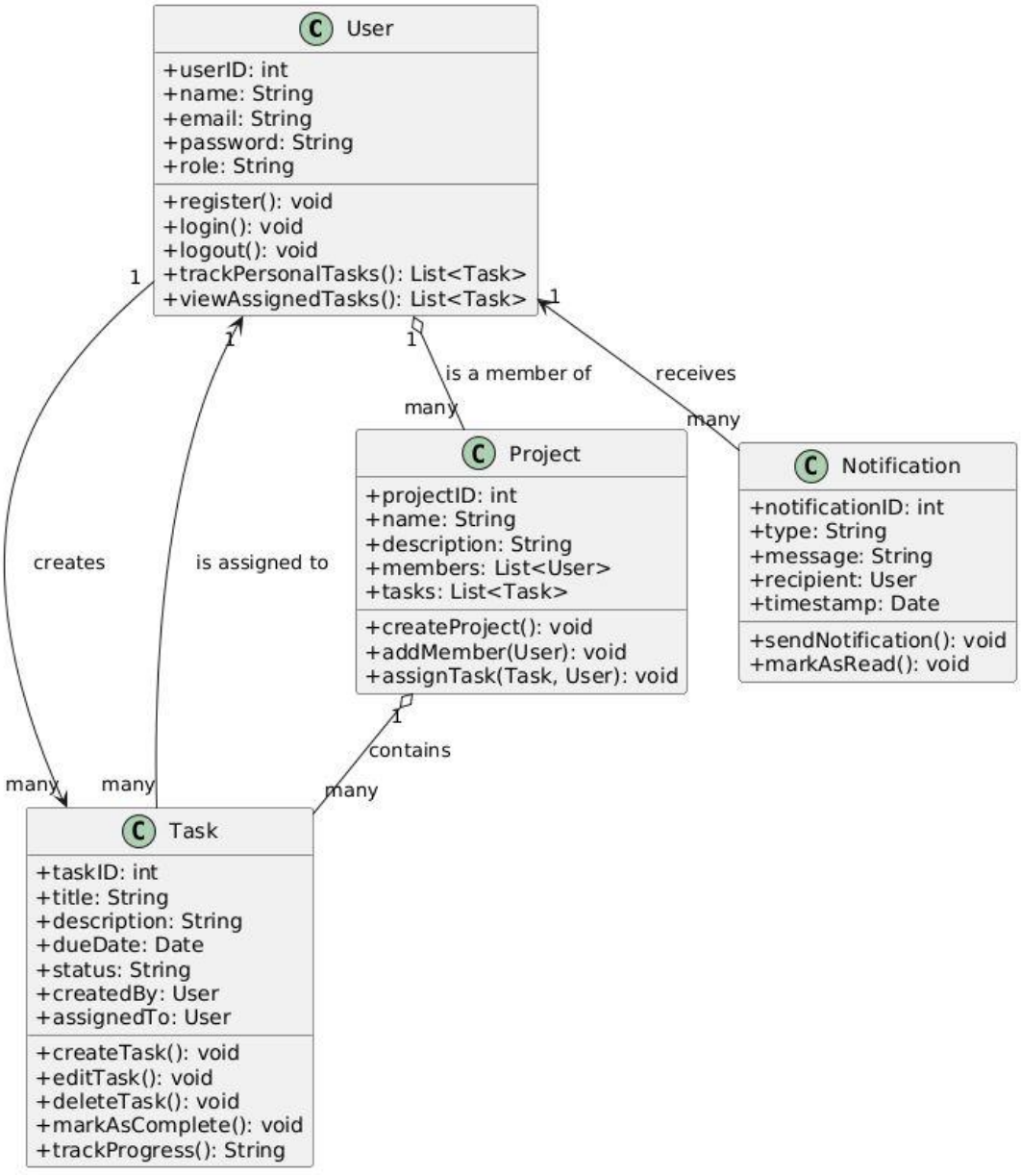
In conclusion, after carefully analyzing the specific requirements of **PlanBee**, we determined that the **Client-Server Architecture** is the most suitable choice. It offers the essential separation of concerns, scalability, and security required for the effective development and deployment of the application. This decision ensures that PlanBee can deliver a seamless and reliable user experience while maintaining flexibility for future enhancements.

Figure 3 - System Architecture of PlanBee



4.2 Class Diagram

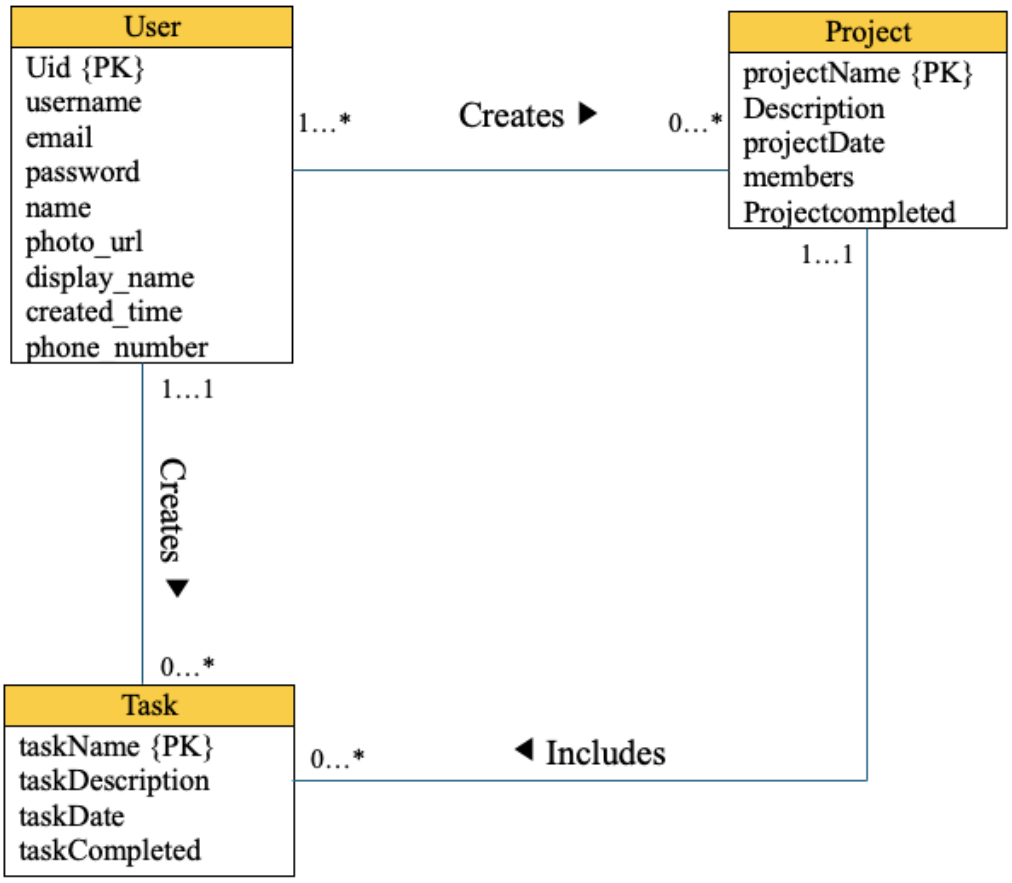
Figure 4 - Class Diagram Of PlanBee



4.3 Data Design

- **EER Diagram**

Figure 5 - EER Diagram of PlanBee



- **Schema**

User (Uid, username, email, password, name, photo_url, display_name, created_time, phone_number)

Primary Key: Uid (FlutterFlow requires the Uid to be the primary key by default)

Project (projectName, Description, projectDate, members, Projectcompleted, pUser)

Primary Key : projectName

Foreign Key : pUser References User (Uid)

Task (taskName, taskDescription, taskDate, taskCompleted,User)

Primary Key: taskName

Foreign Key: User References User (Uid)

Data Dictionary

- Data Dictionary showing description of all entities

Table 6 - Data Dictionary of all PlanBee entities

Entity Name	Description	Aliases	Occurrence
User	Represents a user of the application.	-	A user can create one,none or many projects and can create one, none or many tasks.
Project	Represents a project created by the user.	-	A project includes zero or many tasks.
Task	Represents an individual task.	-	One or many tasks can be included in one and only one project and is created by one and only one user.

- **Data Dictionary showing description of all relationships**

Table 7 - Data Dictionary of all PlanBee relationships

Entity Name	Multiplicity	Relationship	Entity Name	Multiplicity
User	1..1 1..*	Creates Creates	Task Project	0..* 0..*
Project	1..1	Includes	Task	0..*

Creates relationship between User and Task represents the Add/Delete Task function.
Creates relationship between User and Project represents the Add/Delete Project function.

- **Data Dictionary showing description of all attributes**

Table 8 - Data Dictionary of all of PlanBee's attributes

Entity Name	Attribute	Description	Data Type	Length	Nulls	Multi-Valued	Default Value	Range	PK
User	username	Unique username of the user.	VARCHAR	50	No	No			
	email	Email address of the user.	VARCHAR	255	No	No			
	password	User's password.	VARCHAR	100	No	No			
	name	First name of the user.	VARCHAR	50	No	No			
	photo_url	URL of the user's profile picture.	VARCHAR	255	Yes	No			
	Uid	Unique identifier for the user.	VARCHAR	50	No	No			Y
	display_name	Display name of the user.	VARCHAR	50	Yes	No			
	created_time	Timestamp when the user was created.	DATETIME		No	No			
	phone_number	Phone number of the user.	VARCHAR	15	Yes	No			
Project	projectName	Unique name of the project.	VARCHAR	255	No	No			Y
	Description	Description of the project.	VARCHAR	255	Yes	No			
	projectDate	Timestamp of when the project was created or updated.	DATETIME		No	No			
	members	Names of the project members.	VARCHAR	255	Yes	No			
	Projectcompleted	Boolean indicating if the project is completed.	BOOLEAN		No	No			
Tasks	taskName	Unique name of the task.	VARCHAR	255	No	No			Y
	taskDescription	Details about the task.	VARCHAR	255	Yes	No			
	taskDate	Timestamp when the task was created or edited.	DATETIME		No	No			
	taskCompleted	Boolean indicating if the task is completed.	BOOLEAN		No	No			

4.4 Component Design

Table 9 - Component Design Table

Sprint number	PBI (user story)	User story tasks with effort estimates in hours				Sum of effort
1	As a user, I want to create personal tasks so that I can track my individual progress.	Design the "Create Task" interface and navigation.	Build the form with fields (task name, description, deadline, priority)	Implement backend validation and store task in database.	Display the newly created task in the "Tasks" list.	9
		Hours: 2	Hours: 3	Hours: 2	Hours: 2	
1	As a user, I want to edit my tasks so that I can update them as needed.	Add an "Edit Task" button in the task details page.	Load the task details in an editable form.	Implement the update function in the backend to save changes.		6
		Hours: 1	Hours: 2	Hours: 3		
1	As a user, I want to delete my tasks so that I can remove tasks that are no longer needed.	Add a "Delete Task" button in the task details page.	Implement a confirmation dialog for deletion.	Remove the task from the database and update the task list.		5
		Hours: 1	Hours: 1	Hours: 3		
1	As a member leader, I want to create project workspaces so that I can organize group tasks.	Design the "Create Project" interface and navigation.	Build the form with fields (task name, description, deadline, members)	Implement backend validation and store task in database.	Display the newly created project in the "Projects" list.	9
		Hours: 2	Hours: 3	Hours: 2	Hours: 2	
1	As a members' leader, I want to add team members to a project workspace, so that I can ensure all team are included and can collaborate on the project tasks	Add an "Add Members" button in the project details page.	Apply a search functionality to add members.	Implement the update function in the backend to save changes.		6
		Hours: 1	Hours: 3	Hours: 2		
1	As a members' leader, I want to assign tasks to specific team members so that everyone knows their responsibilities.	Add an "Assign Task" button in the project details page.	Implement the logic for assigning the task.	Display the added task in the project details page		7
		Hours: 1	Hours: 4	Hours: 2		

Pseudocode:

- Add Task

Classification: Function
Definition: The user should be able to create a new personal task to track their progress.
Construction:
Inputs: task name, task description, task date.
Precondition: The user must be logged in.
Postcondition: The task is added to the database and displayed in the task list.

IF user clicks "Add Task" button in tasks page **THEN**
  DISPLAY "Create Task" page
  READ task name, task description, task date
  WHILE any field is empty **DO**
    PROMPT user to fill the missing field
  END WHILE
  SAVE task details to the database
  DISPLAY success message
  UPDATE "Tasks" list with the new task
END IF

- Edit Task

Classification: Function
Definition: The user should be able to update an existing task to reflect changes in requirements.
Construction:
Inputs: Updated task name, task description, task date .
Precondition: The task must already exist, and the user must be logged in.
Postcondition: The task is updated in the database, and the updated details are displayed in the task list.

IF user clicks "Edit Task" button **THEN**
  DISPLAY editable task form
  READ task name, task description, task date
  VALIDATE inputs
  IF inputs are valid **THEN**
    SAVE updated task details to the database
    DISPLAY success message
    UPDATE "Tasks" list with the edited task
  ELSE
    PROMPT user to fix invalid inputs
  END IF
END IF

- Delete Task

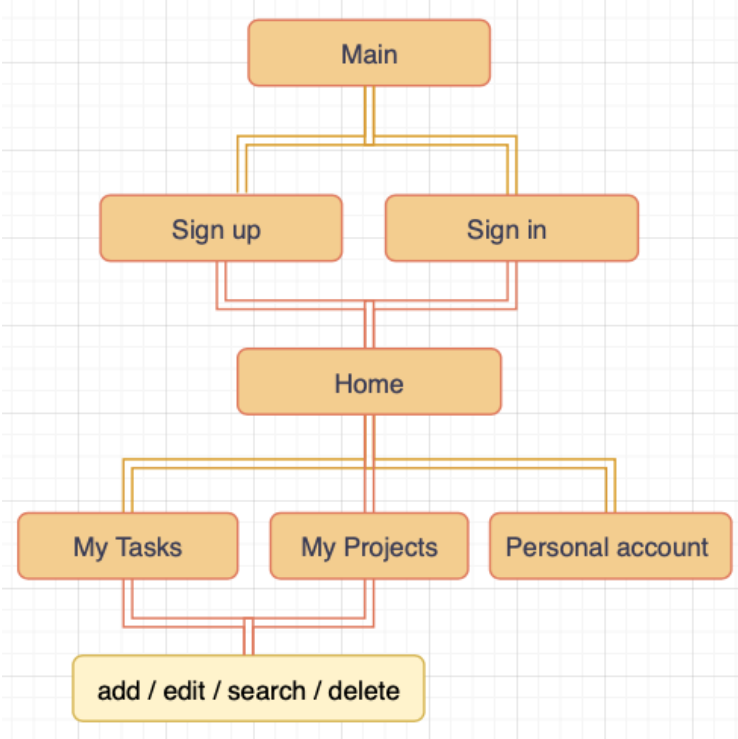
Classification: Function
Definition: The user should be able to delete a task that is no longer required.
Construction:
Precondition: The task must exist, and the user must be logged in.
Postcondition: The task is removed from the database and no longer appears in the task list.

IF user clicks "Delete Task" button **THEN**
  DISPLAY confirmation dialog
  IF user confirms deletion **THEN**
    REMOVE task from the database
    DISPLAY success message
    UPDATE "Tasks" list without the deleted task
  ELSE
    DO nothing
  END IF
END IF

4.5 Interface Design

- **User Navigation Hierarchy**

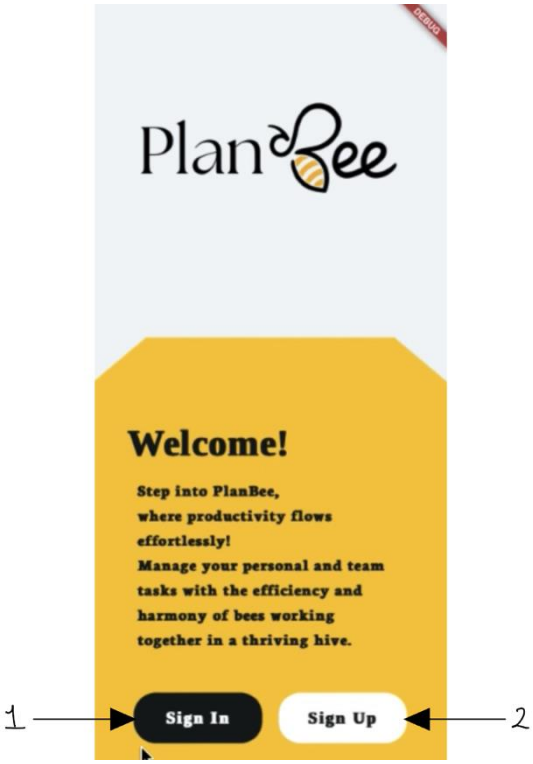
Figure 6 - User Navigation Hierarchy of PlanBee



- **UX Guidelines implemented for each interface:**
  1. Provide Feedback.
  2. Error prevention.
  3. Recognition than Recall.
  4. Consistency and standards.
  5. Flexibility and Efficiency of use.

- **User interfaces**

Figure 7 - Main Page



#	Description
1	Sign in button that directs the user to the sign in page
2	Sign up button that directs the user to the sign-up page

#	Description
3	User’s first name input field (mandatory)
4	User’s Username input field (mandatory)
5	User’s email input field (mandatory)
6	User’s conformation password input field (mandatory)
7	User’s password input field (mandatory)
8	Sign up button that transfers the new user to user’s homepage
9	User’s email input field (mandatory) for signed up users
10	User’s password input field (mandatory) for signed up users
11	Sign in button that transfers the user to user’s home page
12	Register button that transfers the new user to sign up page
13	Home button to direct the user to user’s homepage
14	Tasks button to direct the user to task page
15	Projects button to direct the user to projects page
16	Account button to direct the user to personal account page

Figure 9 - Sign up page (if user click on #2)

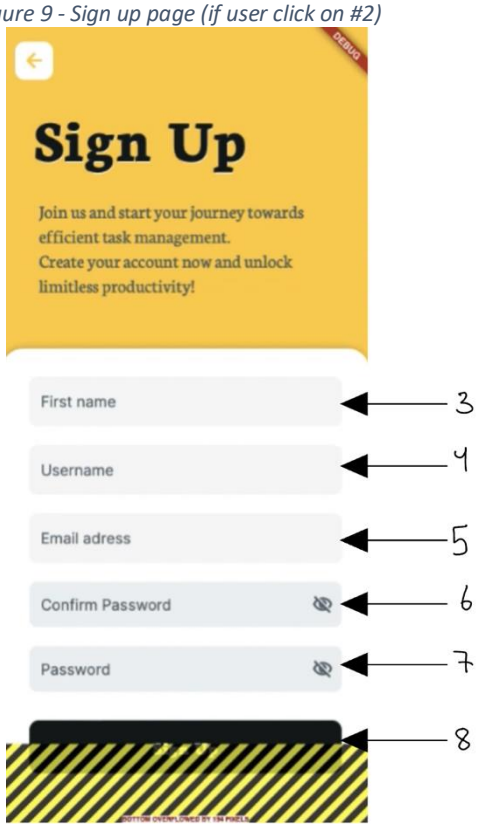


Figure 9 - Sign in page (if user click on #1)

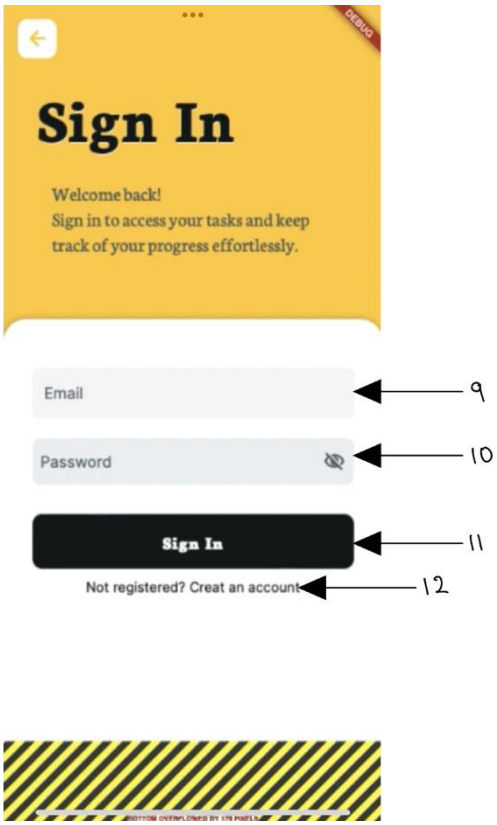


Figure 8 - Home page (if user login successfully)

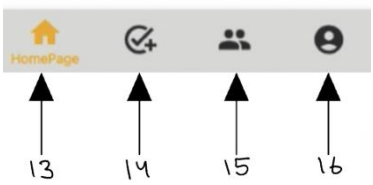
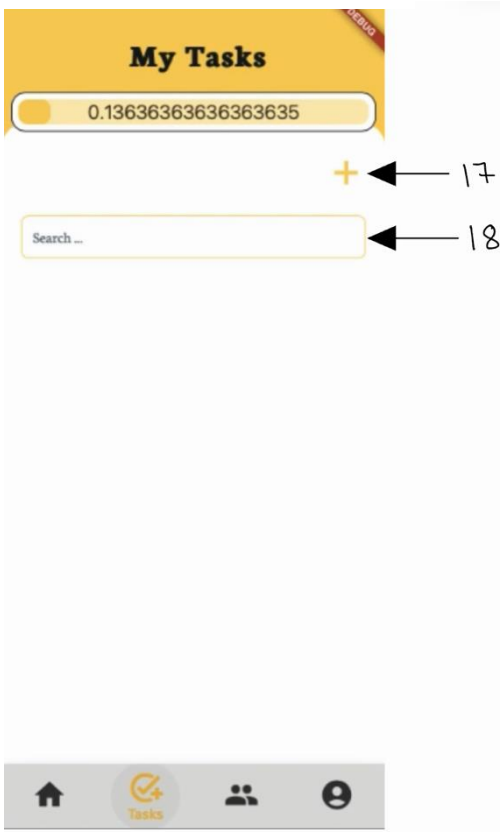


Figure 10 - tasks page (if user click on #14)



#	Description
17	add task button transfers the user to add tasks page
18	Search bar that automatically shows the results of the searched task name as user types
19	Task name input field
20	Task description input field
21	Task date selection
22	After filling inputs ,create task button creates the user's task and adds it to the tasks page
23	Display message to notify the user that the task has been successfully added

Figure 12 - create tasks page (if user click on #17)

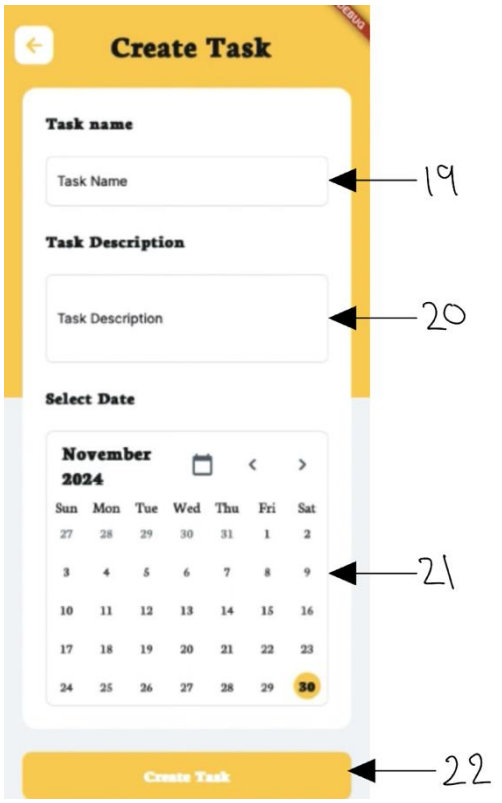


Figure 11 - (if user click on #22)

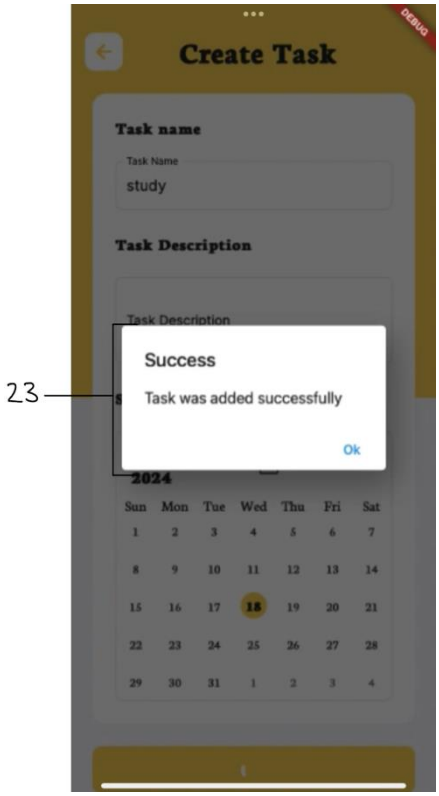
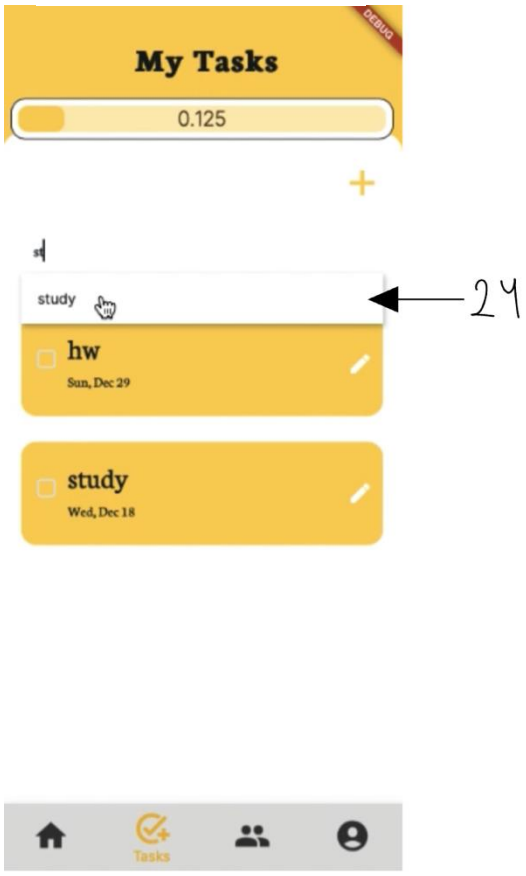


Figure 13 - Search Functionality



#	Description
24	The corresponding searched task name shows up as the user is typing, pressing it will show filtered list of tsks containing tasks with that name
25	Delete search button that cancels the search and returns to the original full list of tasks
26	Edit task button that transfers the user to edit task page

Figure 14 - Search Functionality result

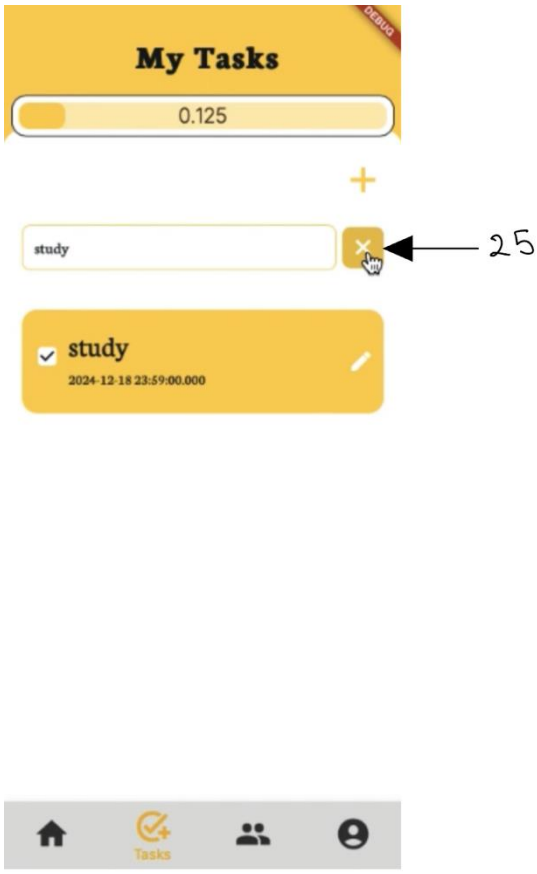


Figure 15 – Edit task button

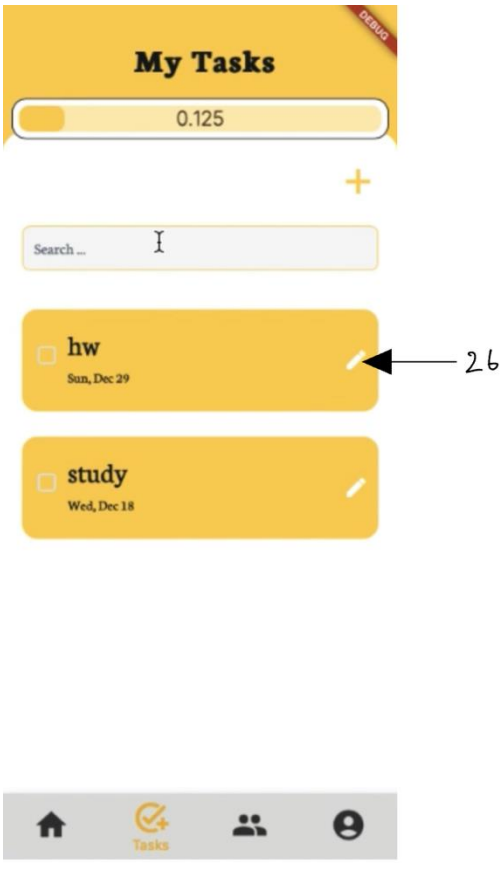
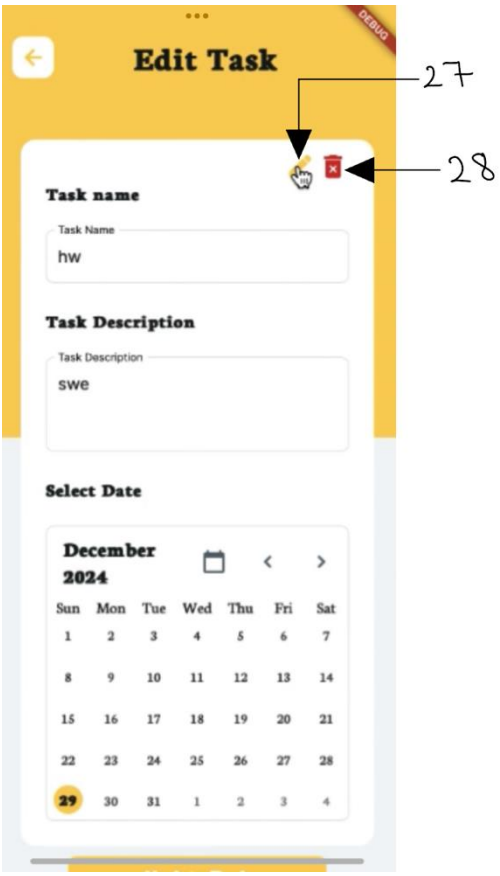


Figure 16 - edit task page (if you click on #26)



#	Description
27	Tap on the edit icon to start editing task details
28	Delete icon that deletes the user's task
29	Display a message to notify the user that the task has been successfully updated

Figure 17 – Confirmation message

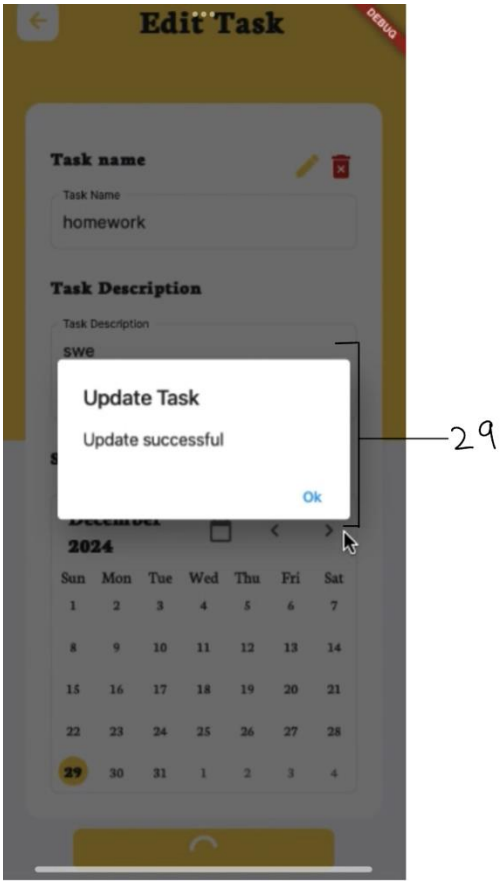
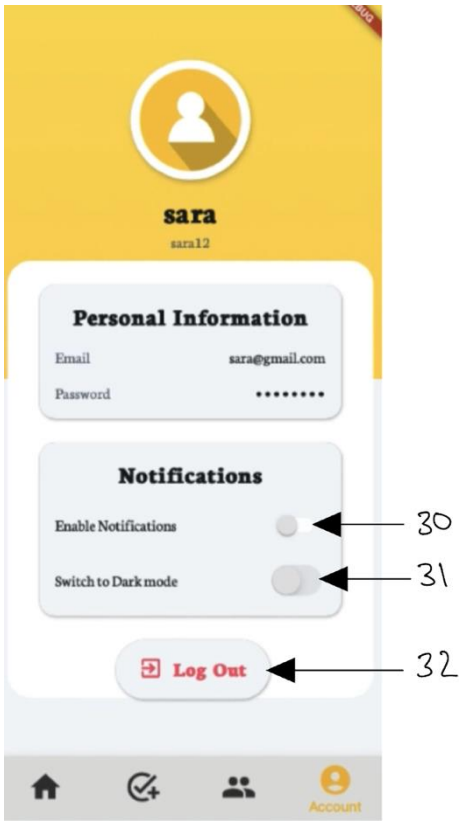
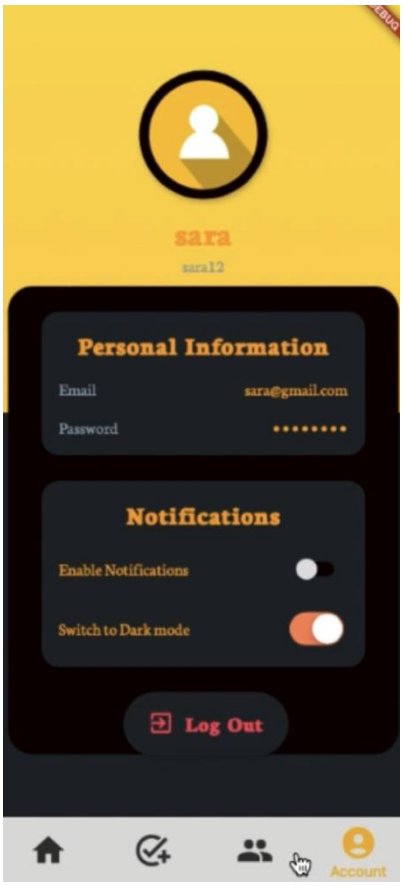


Figure 18- personal account page (if you click on #16)



#	Description
30	Enable notification toggle button to turn on the task deadline notifications
31	Dark mode toggle button to turn on the Dark mode
32	Log out button that logs the user out of his account and redirects him to main page

Figure 19 - personal account page (after you click on #31)



- **Error Message**

Sign up without filling all the required fields

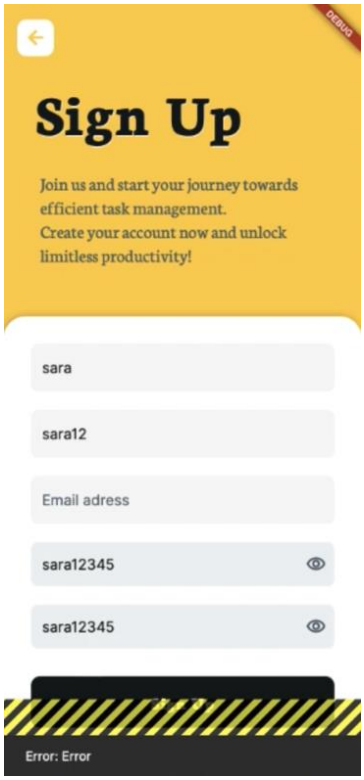


Figure 20 - Error message in sign up (empty field)

Sign in without filling the required fields

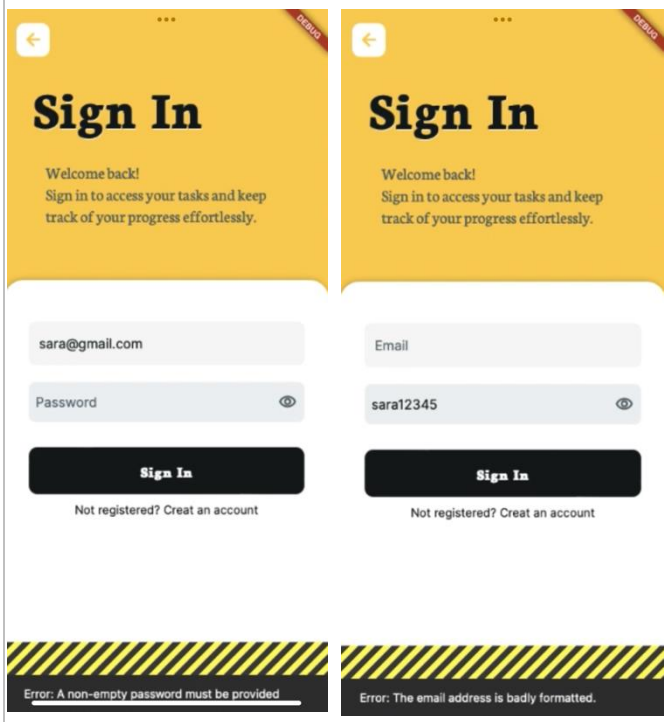


Figure 21 - Error message in sign in (empty field)

Sign up with credentials already used

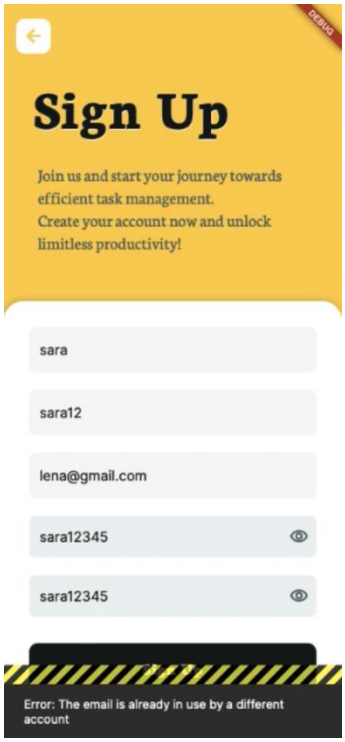


Figure 22 - Error message in sign up (used credentials)

Sign in with wrong credentials

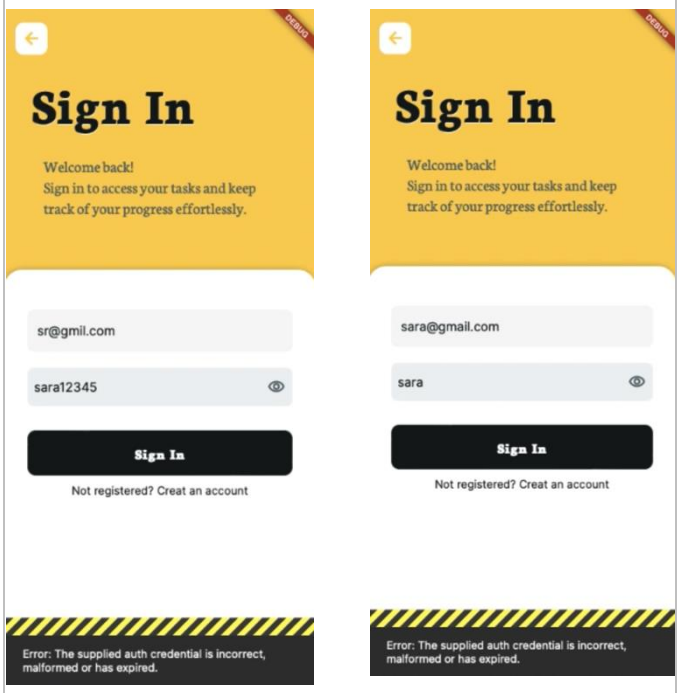
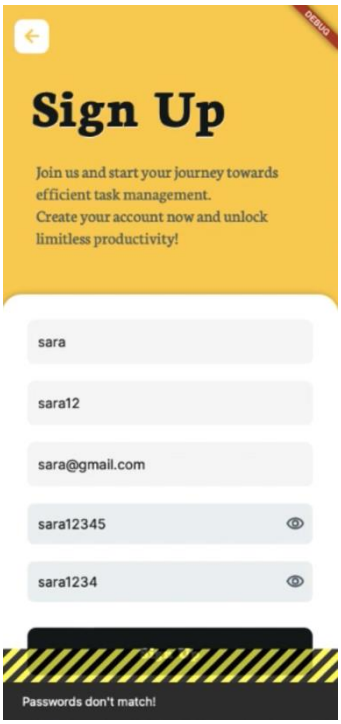
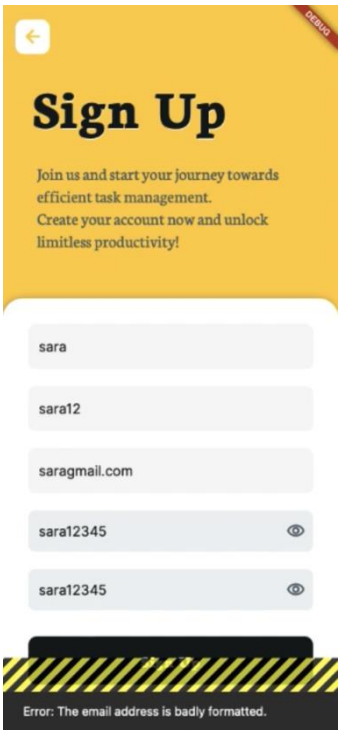


Figure 23 - Error message in sign in (wrong credentials)

Sign up with conformation password that doesn't match the password  *Figure 24 - Error message in sign in (mismatched passwords)*	
Sign up with wrong credentials  *Figure 25 - Error message in sign in (wrong credentials)*	

5 Chapter 5: Implementation

In this part, we will explain the implementation steps of our **PlanBee** app. It involves creating and building the app, which includes designing the database, implementing its functionalities, linking pages together, and ensuring smooth application flow.

For development, we utilized **FlutterFlow**, a powerful platform for building visually appealing and fully functional apps. FlutterFlow allowed us to efficiently design the user interface and implement the backend logic, providing a streamlined development experience with minimal coding. This choice enabled us to focus on enhancing the app's features and optimizing its functionality.

The implementation of **PlanBee** involved several essential steps to ensure a smooth and efficient development process:

1. **Requirements gathering:** We started by collecting and analyzing the necessary requirements to lay the foundation for development. This step focused on understanding the business objectives, user expectations, and technical needs of the app.

2. **Building User Interfaces:** The next step was developing the user interfaces. This involved designing wireframes to outline the structure and layout of each screen, ensuring an intuitive and user-friendly experience for PlanBee users.

3. **The main development process:** which involves building the application, including the front-end and back-end component.

    - **Front-end:** we built the Front-End of the application using Dart programming language in FlutterFlow environment, which included building user interfaces.

    - **Back-end:** after building the front-end, we moved on to developing the Back-end of the application using Dart programming language and FireBase to build the database. The back-end development included:
        - Creating the database using FireBase website, then connect it to our project in FlutterFlow.
        - Connect the pages together to build the system flow.
        - Implementing the functions of the app (Register, Log-in, Log-out, View tasks, Add a task, Edit a task, Delete a task, View task details and user details, and Search tasks).

4. **Testing and debugging:** during the development process, we performed testing and debugging on both the front-end and back-end, including user acceptance testing, user story acceptance testing, and integration testing, to verify that the application's functions work properly and that they fulfill user requirements.

Our application is made up of several important software components, including:

1. **Front-end**: The front-end of the app was created using the Dart programming language in FlutterFlow. It includes the user interface and elements such as the splash screen that users interact with directly.

2. **Back-end**: The back-end was also built using Dart and Firebase. It handles the implementation of all the app's essential functions and processes.

3. **Database**: The app uses a non-relational database to store user data, task data, and project data. It is connected to the Firebase API, which manages and links the database with the application.

As for the challenges, the primary hurdle we encountered during development was implementing the **search** functionality. Despite following all the required steps, the search feature initially did not work as expected. After troubleshooting, we discovered that the issue stemmed from the task component setup. To resolve this, we removed the task component from the tasks page and restructured it as a standalone element. This adjustment allowed the search functionality to work correctly and integrate seamlessly with the application.

Jira Project

6 Chapter 6: Testing

6.1 User Story Acceptance Testing

User story acceptance testing determines whether a user story fulfills its purpose from the user's perspective. It answers the question: will *acceptance criteria* be met upon execution of the user story? Use the following table to document user acceptance testing.

Table 10 - User Story Acceptance Testing

Sprint Number	User story	Acceptance criteria	Test action(s)	Pass ?	Comments
2	As a user, I want to **sign up** for a new account so that I can create and manage my tasks.	- As a nonregistered user who wants to sign-up, if I go to signup page and enter my name, username, email, password, and confirm password and click on sign up button, then I should be able to access the application. - As a nonregistered user who wants to sign-up, if I go to signup page and enter my name, username, email that already exists, password, and confirm password and click on sign up button, then I should receive an error message. - As a nonregistered user who wants to sign-up, if I leave any required fields (name, username, email that already exists, password, confirm password) and click on sign up button, then I should receive an error message. - As a nonregistered user who wants to sign-up, if I go to signup page and enter my name, username, email, password, and a different confirm password and click on sign up button, then I should receive an error message.	1 - Click on sign up button from the welcome page. 2 - Fill the signup fields, with invalid email, mismatched passwords and a blank field. 3- Click on (sign up) button. 4 - Verify that an error message is shown. 5 - Fill all signup fields properly (Email, Name, username, Password, and confirm password) 6 - Verify that the account is created, and that the user is directed to the Homepage.	Yes	-
2	As a user, I want to **log in** to the system securely so that I can access my tasks and projects.	- If I go to the log-in page and enter my email and password and click on Log in, then the data associated to my user should be accessible. - If I go to the log in page and enter an incorrect email and a correct password and click on Log in, then log in fails with an error message that the email is wrong. - If I go to the log in page and enter my email and an incorrect password and click on Log in, then log in fails with an error	1 - Click on log in option from the welcome page. 2 - Fill the log in fields with invalid email, and a correct password. 3 - Verify that an error message is shown indicating wrong email. 4 - Fill the log in	Yes	-

		message that the password is wrong.	fields with a valid email, and an incorrect password. 5 - Verify that an error message is shown indicating wrong password. 6 - Fill the log in fields with correct email and password. 7 - Verify that the user is authenticated and that the user is directed to the Homepage.		
2	As a user, I want to **log out** of the system so that my session is closed securely.	As a user, if I log out then my session should end immediately, and I should be redirected to the login page.	1 - Navigate to "profile" page. 2 - Click on logout button, then verify that the user is logged out of their account and no longer has access to the application. 3 - Try to access a page within the application. 4 - Verify that the session is expired and the user must log back in.	Yes	-
2	As a user, I want to **create personal tasks** so that I can track my individual progress.	As a user, if I create a task then it should appear in my task list immediately.	1 - Navigate to "tasks" page. 2 - Click on the "+" button. 3 - Enter the task's information (name. description, and date). 4 - Press on "add task" button. 5 - Verify that the task has been added successfully.	Yes	-
2	As a user, I want to **edit** my tasks so that I can update them as needed.	As a user, if I edit a task then the updated task should appear in my task list immediately.	1 - Navigate to "tasks" page.	Yes	-

			2 - Click on the pencil button. 3 – Click on the pencil button again in task details page, then edit the information (task name, or description, or date). 4 - Press on “edit task” button. 5 - Verify that the task has been edited successfully.		
2	As a user, I want to **delete** my tasks so that I can remove tasks that are no longer needed.	As a user, if I delete a task then the system should confirm the deletion before removing the task.	1 - Navigate to “tasks” page. 2 - Click on the pencil button. 3 – Click on the trash can button in task details page. 4 - Verify that the task has been deleted successfully.	Yes	-
2	As a user, I want to **search** my tasks so that I can find what I want in a fast way.	- As a user, when I access the tasks search section in my tasks page in the app, I should be presented with a search bar. - If I submit the search query, then the system should process the request and display the search results, focusing on tasks that match the entered name. - If I submit the search, then the search results should prioritize exact matches or closely related matches to the entered name, ensuring that the preferred tasks are displayed.	1 - Click on the search bar in tasks page. 2 - Type the required task name. 3 - Verify that the results appear in real time with exact matching tasks. 4 - Verify that the desired task is shown.	Yes	-
2	As a user, I want to **view** my task’s information, so that I can remember all the datils.	- As a user, if I access the tasks list from the app, then I should be able to click on a specific task to view its detailed information. - If I go to the detailed information then the system should display relevant details about the task, including its name, description, and date.	1 - Open the tasks page in the app. 2 - Verify that all tasks in the system are shown in the tasks page. 3 - Click on a specific task.	Yes	-

			4 - Verify that all the task's information are shown (name, description, and date).		
2	As a member leader, I want to **create project** workspaces so that I can organize group tasks.	As a member leader, if I create a workspace then it should be visible to assigned team members and track task progress.	1 - Navigate to "projects" page. 2 - Click on the "+" button. 3 - Enter the project's information (name. description, members, and date). 4 - Press on "add project" button. 5 - Verify that the project has been added successfully.	Yes	-
2	As a user, I want the user interface to be intuitive and easy to use so that I can efficiently navigate the app.	As a user, if I use the app for the first time then user tests should show a 90% success rate for navigation, with no more than 3 clicks to access main features and quick iterations based on usability feedback.	1 - Open the app for the first time. 2 - Attempt to navigate to the main features (create a task, edit a task, view task) without prior instructions. 3 - Count the number of clicks it takes to access each main feature. 4 - Record success or failure in accessing features within three clicks. 5 - Collect user feedback on navigation ease and iterate on usability improvements based on responses.	Yes	-
2	As a user, I want the app to load within 3 seconds so that I can quickly access my tasks.	As a user, if I open the app then the dashboard should load in under 3 seconds in 95% of cases, tested with varying amounts of data and no noticeable lag during task loading.	1 - Open the app and measure the time taken for the dashboard to load. 2 - Test with different datasets: - Minimal data (no tasks or projects).	Yes	-

			- Moderate data (50 tasks and 10 projects). - High data load (150+ tasks and 30+ projects). 3 - Repeat the test multiple times to ensure consistency. 4 - Verify that the dashboard loads within 3 seconds in 95% of cases across all datasets. 5 - Perform actions such as viewing or updating tasks and observe for any noticeable lag. 6 - Record results and identify areas for optimization if criteria are not met.		

6.2 Integration Testing

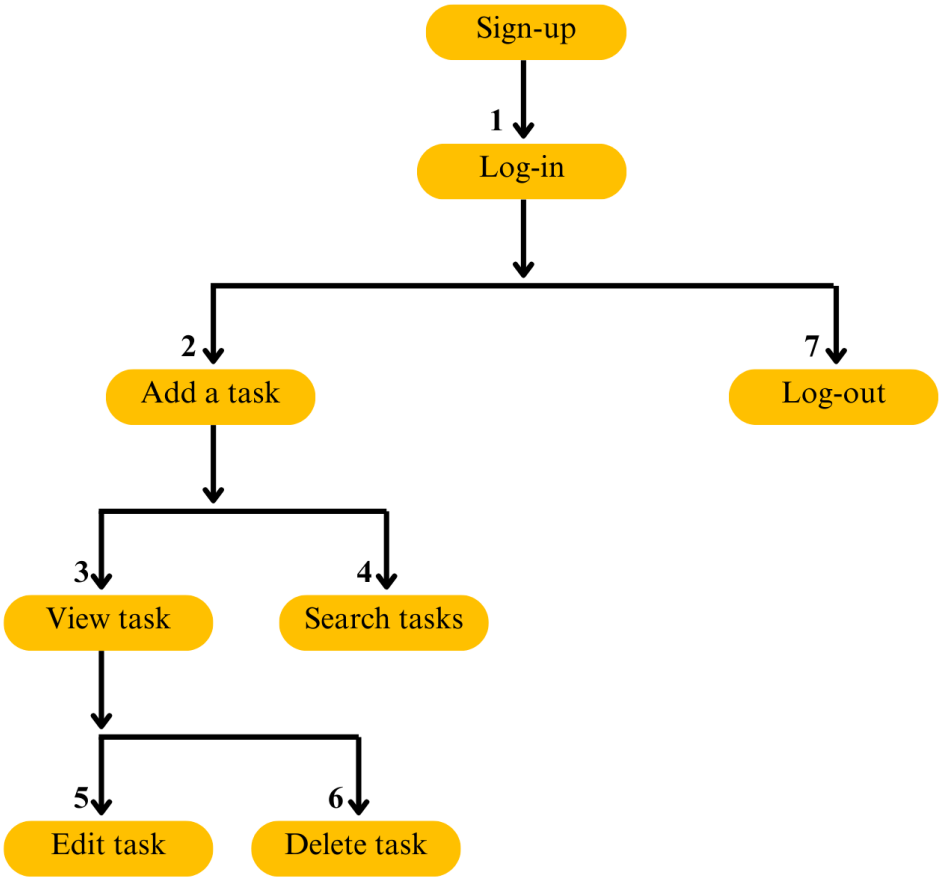
In this section, we will use integration testing to examine the interactions that occur when components are combined into a system.

The next testing we will perform is the integration testing. Integration testing checks how individual components interact when integrated into a system. It is conducted each time a new component is added to the system, and it focuses on the interface between the system components, to verify that different components of the application work seamlessly together, ensuring a smooth, integrated user experience.

6.2.1 Integration Hierarchy

In this sub-section, we will present **PlanBee's** integration hierarchy graph, which is designed to show the structure and the sequence in which various components of the system are integrated and tested to form **PlanBee** application.

Figure 26 - Integration Hierarchy Of PlanBee



6.2.2 Integration Testing Plan

In this sub-section, we will present the integration testing plan table that is introduced to ensure that the system components interact with the new components properly by examining some test cases related to them, with an indication whether or not these components pass the test cases.

Table 11 - Integration Testing Plan

System components	New component	Test case	Pass?	Comments
- Sign up to the system	Log in to the system	1. A user signed up into the system successfully. 2. The signed-up user logged in to the system.	Yes	-
- Sign up to the system - Log in to the system	Add a task	1. A user signed up into the system successfully. 2. The signed-up user logged in to the system. 3. The logged in user added a task in tasks page.	Yes	-
- Sign up to the system - Log in to the system - Add a task	View task	1. A user signed up into the system successfully. 2. The signed-up user logged in to the system. 3. The logged in user added a task in tasks page. 4. The logged in user viewed the details of a specific task.	Yes	-
- Sign up to the system - Log in to the system - Add a task - View task	Search task	1. A user signed up into the system successfully. 2. The signed-up user logged in to the system. 3. The logged in user added a task in tasks page. 4. The logged in user viewed the details of a specific task. 5. The logged in user searched his tasks by task name in tasks page.	Yes	-
- Sign up to the system - Log in to the system - Add a task - View task - Search task	Edit task	1. A user signed up into the system successfully. 2. The signed-up user logged in to the system. 3. The logged in user added a task in tasks page.	Yes	-

		4. The logged in user viewed the details of a specific task. 5. The logged in user searched his tasks by task name in tasks page. 6. The logged in user edited a task that he made previously by modifying the name / description / date.		
- Sign up to the system - Log in to the system - Add a task - View task - Search task - Edit task	Log out	1. A user signed up into the system successfully. 2. The signed-up user logged in to the system. 3. The logged in user added a task in tasks page. 4. The logged in user viewed the details of a specific task. 5. The logged in user searched his tasks by task name in tasks page. 6. The logged in user edited a task that he made previously by modifying the name / description / date. 7. The logged-in user logs out of the system and the session is ended.	Yes	-

6.3 User Acceptance Testing

User Acceptance Testing (UAT) is a critical phase aimed at ensuring that the system meets business requirements and functions effectively for end users. To conduct UAT, a group of six participants from the target audience, selected based on their domain knowledge and familiarity with task management applications, was invited to test the system. The testing environment was set up to simulate real-world scenarios, allowing participants to independently use the application and complete tasks related to adding, editing, deleting, and searching for tasks. A questionnaire consisting of ten unbiased questions was prepared to assess the usability, functionality, and overall performance of the application. Participants provided feedback through the questionnaire, and their responses were collected and analyzed to evaluate the system's effectiveness. The analysis highlighted areas of strength, as well as opportunities for improvement, ensuring that the application is ready to meet user expectations

6.3.1 Demographics of participants

In this section, we present the demographic information of the participants who took part in the User Acceptance Testing (UAT) for the application. A total of 6 participants were selected based on specific criteria to ensure a representative sample. The participants were from diverse age groups and genders, and they spend varying amounts of time daily on task management. Below is a summary of the demographic details:

1. **Gender Distribution**: The participants included both male and female users.
2. **Age Groups**: Participants ranged in age from 20 to 30 years, with the majority being in the 20-25 age group.
3. **Time Spent on Task Management**: The participants' responses were evenly distributed spending of time on task management.

Figure 27 - Question 1

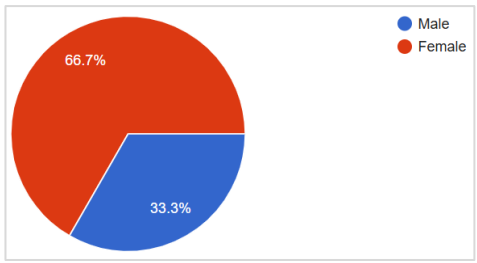


Figure 28 - Question 2

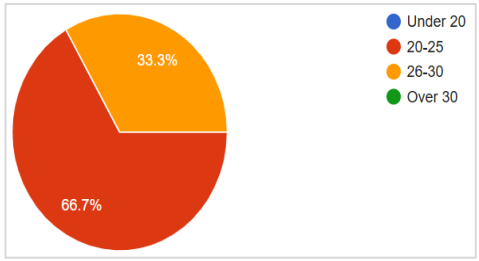
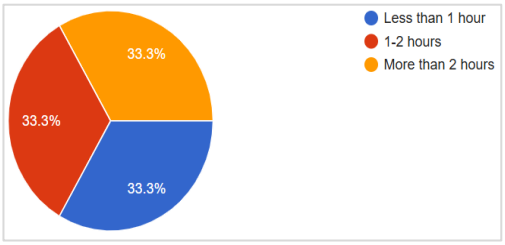


Figure 29 - Question 3



6.3.2 Questionnaire

In this section, we present the questionnaire designed to evaluate the functionality, usability, and performance of the application. The questions were carefully crafted to ensure clarity and minimize bias, allowing participants to provide honest and accurate feedback about their experience with the system. The questionnaire covers various aspects, including task management features, ease of use, and overall satisfaction with the application

Demographic Questions:

1. **Gender:**
    - Male
    - Female
    - Prefer not to say
2. **Age:**
    - Under 20
    - 20-25
    - 26-30
    - Over 30
3. **How many hours do you spend daily on task management?**
    - Less than 1 hour
    - 1-2 hours
    - More than 2 hours

Functional and Usability Questions:

4. **Did you encounter any challenges while adding a new task to the app?**
    - Yes
    - No
    - A little bit
5. **Did you experience any difficulties while editing an existing task?**
    - Yes
    - No
    - A little bit
6. **Is the search feature working correctly and showing results related to tasks?**
    - Yes
    - No
    - Maybe
7. **How easy was it to use the app's interface?**
    - Excellent
    - Very Good
    - Good
    - Poor
8. **Did you experience any difficulties while logging in or out?**
    - Yes
    - No

- Maybe

9. **What feature did you like the most in the app?**
    - User Interface
    - Search speed
    - Task management functionality
10. **How would you rate the speed of the application when adding, editing, or deleting tasks?**
    - Excellent
    - Good
    - Fair
    - Poor

Figure 30 - Question 4

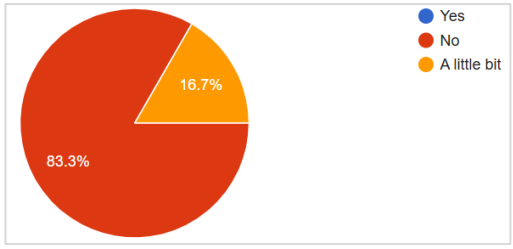


Figure 31 - Question 5

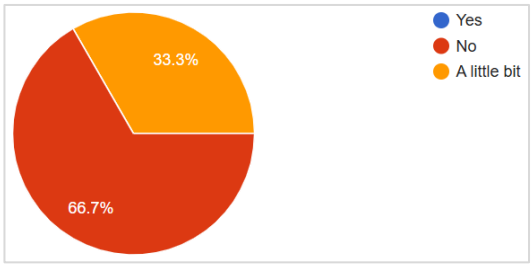


Figure 32 - Question 6

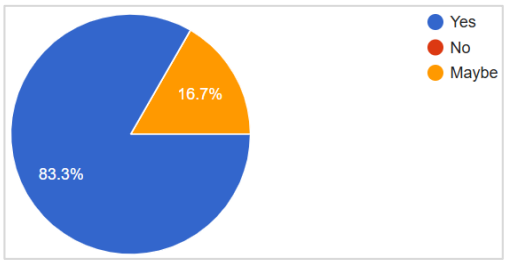


Figure 33 - Question 7

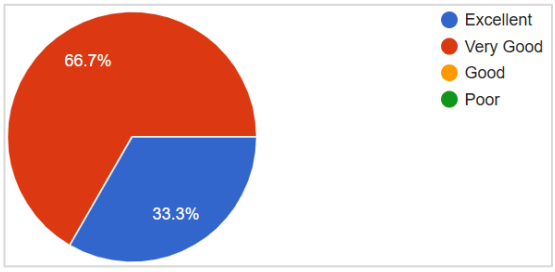


Figure 34 - Question 8

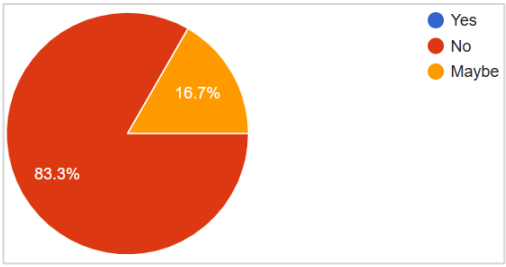


Figure 35 - Question 9

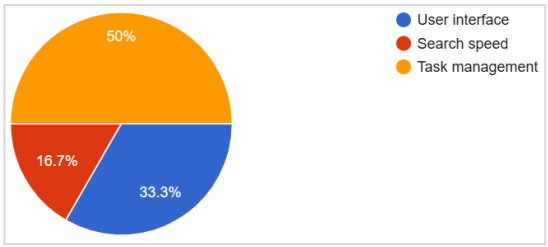
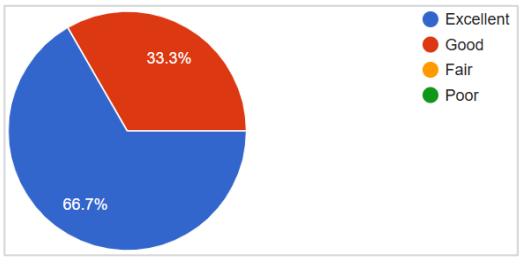


Figure 36 - Question 10



- Did you face any challenges while adding a new task to the app?
The responses were divided between "No" and "A little bit" indicating that some participants faced minor difficulties or did not experience any issues while adding tasks.

-Did you face any difficulties while editing an existing task?
The responses were similar to the first question, with a balanced distribution of answers. Some participants indicated they did not face significant difficulty in editing tasks.

-Does the search feature work correctly and display relevant results for tasks?
Most participants answered positively, while 16.7% expressed slight dissatisfaction with the performance of the search feature.

- How easy was it to use the app's interface?
Participants expressed overall satisfaction with the user interface, with responses favoring good to excellent performance of the interface.
- Did you encounter any issues during login or logout?
The majority of participants did not face any difficulties when logging in or out of the app.

- Which feature did you like the most in the app?
The "Task Management" feature was the most liked among participants, followed by the user interface and search speed.

- How would you rate the speed of the app when adding, editing, or deleting tasks?
Most responses indicated satisfaction with the app's speed, noting good performance in various tasks such as adding, editing, and deleting tasks.

7 Chapter 7: Conclusion and Future Work

7.1 Conclusion

The project successfully achieved its primary goals by delivering a task management application that effectively meets users' needs. Through a combination of thoughtful design and targeted features, users can efficiently add, edit, delete, and search for tasks. The user interface has been designed to be intuitive, and the overall performance has been optimized to ensure smooth workflows.

User acceptance testing was conducted to gather feedback from the target audience, providing valuable insights that will guide future improvements. The feedback highlighted some areas that require enhancement, particularly in terms of the performance of certain features and response times, which will inform the next steps for refining the application.

Overall, the project successfully met user needs and business requirements. While there is room for future improvements, particularly in expanding features and optimizing performance, the application provides a strong foundation for task management. Future work will focus on enhancing the user experience and ensuring the app can scale to meet the needs of a larger user base.

7.2 Future Work

While the project has successfully met its main objectives and provided an effective task management system, there are still opportunities for future development to enhance performance and improve the user experience. The areas of future work include the following:

- The user interface will be redesigned based on the feedback gathered during the user acceptance testing. The focus will be on making the interface more user-friendly and intuitive to meet the diverse needs of users.
- There may be the addition of a feature for generating detailed reports on completed and upcoming tasks. This feature will help improve users' productivity and time management.
- Work will be done to expand the application to support multiple languages. This will cater to users from different cultural and geographical backgrounds.
- Improving the application’s performance and reducing response time will be a priority, especially as the number of users increases. Efforts will be made to ensure the system can handle a larger number of users and tasks in the future.

8 References

[1] Better App Tech, "To Do List: Task Manager & Reminder App," Better App Tech, 2024. [Online]. Available: https://www.betterapptech.com/products/todolist/.

[2] The Align approach was followed, see: Al Bassam, H. (2024). IT320 Practical Software Engineering course slides. First Semester 1446 H.

[3] Apple. (n.d.). About iOS 12 updates. Apple Support. Retrieved September 15, 2024, from https://support.apple.com/en-sa/118387

[4] Atlassian. (n.d.). *Trello*. Trello by Atlassian. Retrieved from https://trello.com

[5] Microsoft Corporation. (n.d.). *Microsoft To Do*. Retrieved from https://to-do.microsoft.com

[6] Appest Inc. (n.d.). *TickTick: To-do List & Task Manager*. Retrieved from https://www.ticktick.com