TaskTally

Assign, track, and update tasks seamlessly

CS19611 – MOBILE APPLICATION DEVELOPMENT LAB

Submitted by

SHREE HARSHINI S(220701269)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE

THANDALAM , CHENNAI - 602105

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

BONAFIDE CERTIFICATE

Certified that this Project titled **"TaskTally-Assign "** is the bonafide work of **"SHREE HARSHINI S (2116220701269),** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr. Duraimurugan N,., M.Tech., Ph.D.,

SUPERVISOR

Professor

Department of Computer Science and Engineering,

Rajalakshmi Engineering

College, Chennai-602 105.

Submitted to Project Viva-Voce Examination held on _____

Internal Examiner **External Examiner**

ABSTRACT

Abstract:

TaskTally is a comprehensive, user-friendly task management mobile application that helps individuals and teams organize, track, and complete their daily tasks efficiently. The app provides an intuitive interface for managing tasks by allowing users to add, edit, delete, and categorize tasks with due dates and times. It also offers features like task completion tracking, progress monitoring, and a reward system to enhance user engagement.

With TaskTally, users can prioritize their work, set deadlines, and visualize their progress over time through an interactive calendar view. Collaborative lists enable team members to share tasks and work together in real-time. Task completion is marked with checkboxes, and tasks can be categorized into different groups such as Work, Personal, Shopping, and others. The app also includes a built-in reminder and notification system, sending alerts for upcoming tasks based on their due date and time.

Key Features:

- **Task Creation:** Add new tasks with a title, optional description, category, and due date.

- **Edit/Delete Tasks:** Modify or remove tasks as necessary.

- **Due Date and Time:** Set specific deadlines for tasks.
- **Categorization:** Organize tasks by category (Work, Personal, Shopping, etc.).
- **Task Completion:** Mark tasks as completed with a checkbox.
- **Progress Tracking:** Monitor task completion with visual progress indicators.
- **Reminder and Notifications:** Get notified of upcoming tasks.
- **Collaborative Lists:** Share tasks with others for collaborative work.
- **Calendar View:** View tasks on a calendar grid for better organization.
- **Reward System:** Earn points or badges for completing tasks.

TaskTally is designed to provide users with a highly efficient and interactive way to keep track of their tasks and collaborate with others. With its seamless integration of task management features, notifications, and a reward system, TaskTally enhances productivity and makes task management a more engaging experience.

ACKNOWLEDGMENT

ACKNOWLEDGMENT Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavor to put forth this report. Our sincere thanks to our Chairman Mr. S. MEGANATHAN, B.E, F.I.E., our Vice Chairman Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S., and our respected Chairperson Dr. (Mrs.) THANGAM MEGANATHAN, Ph.D., for providing us with the requisite infrastructure and sincere endeavoring in educating us in their premier institution. Our sincere thanks to Dr. S.N. MURUGESAN, M.E., Ph.D., our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to Dr. P. KUMAR, M.E., Ph.D., Professor and Head of the Department of Computer Science and Engineering for his guidance and encouragement throughout the project work. We convey our sincere and deepest gratitude to our internal guide, Dr. Duraimurugan N , M.Tech., Ph.D., Professor of the Department of Computer Science and Engineering. Rajalakshmi Engineering College for his valuable guidance throughout the course of the project. We are very glad to thank our Project Coordinator, Mr. Duraimurugan N Professor Department of Computer Science and Engineering for his useful tips during our review to build our project.

SHREE HARSHINI S (2116220701269)

TABLE OF CONTENT

CHAPTER-1

1.INTRODUCTION

TaskTally is an intuitive and feature-rich task management mobile application designed to help users efficiently organize, track, and complete their tasks. Whether for personal or professional use, TaskTally offers a comprehensive platform where users can create, edit, categorize, and set due dates for tasks. The app simplifies task management by providing essential features like task completion checkboxes, progress tracking, and categorization (Work, Personal, Shopping, etc.).

TaskTally also supports team collaboration by allowing users to share tasks and work together on projects. The built-in calendar view helps visualize deadlines and task progress, while the reward system encourages productivity by awarding points or badges for completing tasks.

With customizable notifications, users are reminded of upcoming deadlines, ensuring tasks are completed on time. TaskTally is designed to improve efficiency, streamline task organization, and boost motivation, making it an ideal solution for anyone looking to stay on top of their responsibilities.

CHAPTER-2

LITERATURE SURVEY

Task management applications have become essential tools for individuals and organizations looking to enhance productivity, prioritize tasks, and streamline workflows. Over the years, various task management apps have evolved, incorporating numerous features to address the needs of users in both personal and professional settings.

1. Traditional To-Do Lists:

   Traditional task management applications focused primarily on simple to-do lists. These early applications allowed users to add tasks and mark them as completed. Examples include Google Keep and Apple's Notes. These tools were basic but offered a foundational structure for task organization.

2. Calendar-Based Task Management:

   Calendar integration has been a crucial feature for task management applications. Apps like Microsoft Outlook and Google Calendar allow users to add tasks and appointments directly onto their calendar, offering a visual representation of deadlines. This helps users prioritize tasks and plan their day more effectively.

3. Collaborative Task Management:

   With the rise of remote work and collaboration, task management apps with sharing capabilities became essential. Tools like Trello, Asana, and Monday.com allow users to share task lists with others, making it easier for teams to collaborate and track progress on shared projects. These apps provide features like task assignment, project tracking, and communication tools.

4. Progress Tracking and Reporting:

   Several task management applications now include progress tracking to help users visualize their task completion. Apps like Todoist and ClickUp provide visual indicators like percentage completion bars, helping users stay motivated and focused on their goals. These apps also offer reports that highlight performance and efficiency.

5. Gamification and Rewards:

   Task management apps have incorporated gamification features to keep users engaged. For example, Todoist's Karma system awards points based on task completion, fostering motivation. This trend has been seen in other productivity tools as well, such as Habitica, where users earn rewards and level up by completing tasks.

6. Voice and AI Integration:

   Recent developments in AI and voice recognition have further enhanced task management systems. Apps like Google Assistant and Apple Siri allow users to add tasks using voice commands. Some apps also integrate AI to offer task suggestions or prioritize tasks based on user behavior and preferences.

7. Mobile-Based Solutions:

   The increasing reliance on smartphones has made mobile-based task management apps even more popular. Apps like Microsoft To-Do, Any.do, and Todoist are widely used because they offer seamless syncing across devices, push notifications, and easy access to tasks on-the-go.

Key Trends and Innovations:

- Smart Categorization: Many task management apps now allow users to categorize tasks based on urgency, type, or project. This helps users focus on high-priority tasks first.

- Due Date Integration: Apps that integrate task due dates with calendar views are becoming more popular, as users can better visualize their time and organize tasks effectively.

- Collaboration and Sharing: Tools that allow task sharing and collaboration are increasingly important for businesses, remote teams, and families who need to manage shared responsibilities.

- User Engagement: Gamification, badges, and points systems are used to encourage users to complete tasks and maintain productivity.

CHAPTER-3

3.METHODOLOGY

The development of the **TaskTally** application follows a systematic approach involving multiple stages, from requirement gathering to the final deployment. The goal is to create a user-friendly task management system that helps users organize, track, and complete tasks efficiently. The methodology is based on the following key steps:

1. Requirements Gathering and Analysis

The first step in the development process is understanding the needs of the target users and gathering requirements. This includes:

- Identifying essential features like task creation, due dates, categories, and collaboration.

- Understanding how users will interact with the app and what tasks they want to manage (personal, work, etc.).

- Analyzing existing task management applications to identify gaps and opportunities for innovation.

The major requirements identified for **TaskTally** include:

- Adding, editing, and deleting tasks.

- Categorizing tasks into various categories (e.g., work, personal, shopping).

- Setting due dates and reminders.

- Progress tracking for tasks and projects.

- Collaborative task sharing among multiple users.

- Reward system for completing tasks.

2. Design Phase

In this phase, the app's interface and functionality are designed. It focuses on:

- **UI/UX Design:** Creating wireframes, screen mockups, and designing an intuitive user interface that is simple, functional, and visually appealing. Tools like Figma or Adobe XD can be used for designing the layout and overall experience.

- **Database Design:** Planning how tasks and user data will be stored, accessed, and updated. For local storage, a solution like Room Database is used to persist tasks and their statuses.

- **Feature Flow:** Designing the flow of tasks within the app, from adding and categorizing tasks to marking them as completed, setting reminders, and exporting tasks.

3. Development Phase

The development phase involves implementing the functionalities outlined in the design phase, including:

- **Task Management**:

    - Creating tasks with titles, descriptions, and due dates.
    - Implementing features to edit and delete tasks.
    - Implementing task completion with checkboxes and visual cues (like strike-through).

- **Task Categorization**:
    - Adding categories for tasks (e.g., Work, Personal, Shopping).
    - Implementing functionality to filter tasks by category.

- **Progress Tracking**:
    - Displaying task progress as a percentage or visual progress bar.
    - Updating progress as tasks are marked as completed.

- **Reminder and Notification System**:
    - Implementing reminder notifications using the **AlarmManager** or **WorkManager** for due tasks.

    - Using notifications to alert users when a task is due.

- **Collaborative Lists**:

    - Implementing functionality for sharing tasks with other users or collaborating on a shared list.

    - Implementing cloud-based sharing (Google Firebase or other solutions) for real-time updates.

- **Reward System**:

    - Introducing a point or badge system that rewards users for completing tasks.

    - Encouraging users to complete more tasks by gamifying the experience.

4. Testing Phase

Testing is crucial to ensure that the app works as expected. This includes:

- **Unit Testing**: Testing individual components like task creation, notification triggers, and progress tracking.

- **Integration Testing**: Testing the app as a whole, ensuring that features like notifications, task editing, and categorization work together seamlessly.

- **User Testing**: Involving a small group of target users to test the app for usability, ease of navigation, and to identify bugs.

- **Bug Fixing**: Identifying and fixing issues based on feedback from testing.

5. Deployment Phase

Once testing is complete and the app is stable, the app is prepared for deployment. The following steps are involved:

- **Deployment to the Play Store**: Packaging the app into an APK or AAB format and submitting it to the Google Play Store for review and distribution.

- **Version Management**: Updating the app periodically by adding new features and improving the user experience. Every new update is tested

thoroughly before deployment.

6. Post-Deployment Maintenance

After the app is launched, ongoing maintenance is required to ensure it stays up-to-date and functional. This includes:

- **Bug Fixes**: Addressing any issues reported by users.
- **Feature Updates**: Introducing new features based on user feedback and usage data.
- **Performance Optimization**: Ensuring that the app runs smoothly on all devices and optimizing its performance.

Technologies Used

- **Frontend Development**: The app is built using **Kotlin** and **Jetpack Compose** for the UI.
- **Local Database**: **Room Database** is used to store tasks locally on the device.

- **Notifications**: **AlarmManager** and **WorkManager** are used for task reminders.

- **Cloud Integration**: **Firebase** is used for collaborative lists and sharing tasks.

- **Progress Tracking**: Task progress is visually represented using progress bars and percentage completion.

- **Reward System**: A points-based system to incentivize task completion.

Backend Infrastructure

The backend infrastructure for **TaskTally** supports essential features such as data storage, user authentication, real-time updates, and collaboration. Although the app is focused on managing tasks, it integrates various backend services to ensure that data is synchronized across devices and accessible to the user at all times. Here's a breakdown of the backend architecture:

1. Database

The core of the **TaskTally** app's data management is its database system. There are two primary components for handling local and remote data:

a. Local Database - Room Database (Android)

Room is a persistence library used for local data storage on the device. It allows **TaskTally** to store and manage tasks, their statuses, due dates, categories, and any progress updates. Room works with a local SQLite database that is stored on the user's device.

- **Entities**: Tasks are stored as entities in the Room database. Each task will include fields like `title`, `dueDate`, `category`, and `isDone` (for marking task completion).

- **DAOs (Data Access Objects)**: DAOs are used to define methods for accessing and manipulating data in the Room database, such as adding, updating, deleting, or retrieving tasks.

Example code for defining a **Task** entity in Room:

```
Unset
@Entity(tableName = "tasks")

data class Task(
```

```
    @PrimaryKey(autoGenerate = true) val id: Int =
0,
    val title: String,
    val dueDate: String,
    val category: String,
    val isDone: Boolean = false
)
```

The **RoomDatabase** class handles database management:

Unset

```
@Database(entities = [Task::class], version = 1)
abstract class TaskDatabase : RoomDatabase() {
    abstract fun taskDao(): TaskDao
}
```

b. Remote Database - Firebase Realtime Database (Optional for Collaboration)

For real-time updates and collaborative features (like sharing tasks and working with others), **Firebase Realtime Database** can be integrated into the app. Firebase allows users to store tasks remotely and share them with others in real-time.

- **Tasks Node**: Each user can have a list of tasks stored under a "tasks" node.

- **Collaborative Task Management**: Users can share their task lists with others by storing task data in Firebase. This allows collaborative editing, real-time updates, and task sharing.

Example Firebase structure for tasks:

```
Unset
tasks
    ├── userID
        ├── taskID
            ├── title: "Buy Groceries"
            ├── dueDate: "2025-04-25"
            ├── category: "Shopping"
            └── isDone: false
```

---

2. User Authentication

While your app doesn't require user authentication via Firebase, adding user accounts can enhance personalization, especially for the collaborative features.

a. Firebase Authentication (Optional for Users)

Firebase Authentication can be integrated if you want users to log in, manage their tasks, and sync them across devices. This feature is particularly useful for collaborative tasks. With Firebase Authentication, you can:

- Allow users to sign up and sign in using email/password or third-party services (Google, Facebook, etc.).

- Securely authenticate users before accessing or editing their tasks.

---

3. Notifications & Alarm Management

To remind users of their tasks, **TaskTally** uses **AlarmManager** or **WorkManager** to trigger notifications based on task due dates. These alarms are scheduled to fire at specific times, even if the app is closed, ensuring users never miss a deadline.

ReminderReceiver (for notifications) listens to these scheduled alarms and sends notifications when a task is due.

Example:

Unset

```
val alarmManager =
context.getSystemService(Context.ALARM_SERVICE) as
AlarmManager

val intent = Intent(context,
ReminderReceiver::class.java).apply {

    putExtra("task", task.title)

}

val pendingIntent = PendingIntent.getBroadcast(

    context,

    task.title.hashCode(),

    intent,

    PendingIntent.FLAG_UPDATE_CURRENT or
PendingIntent.FLAG_IMMUTABLE

)

val formatter = SimpleDateFormat("yyyy-MM-dd
HH:mm", Locale.getDefault())
```

```
val time = formatter.parse(task.dueDate)?.time ?:
return

alarmManager.setExactAndAllowWhileIdle(AlarmManager
.RTC_WAKEUP, time, pendingIntent)
```

4. Cloud Backup & Data Sync

Firebase Storage or **Cloud Firestore** can be used for cloud backup and data synchronization, ensuring that users can access their tasks from any device. Whenever a task is updated on one device, it can be reflected in real-time on all other devices linked to the same user account.

- **Task Synchronization**: When a user adds, updates, or deletes tasks, the changes are synchronized with Firebase. This allows users to seamlessly work across multiple devices.

5. Task Categorization and Labels

Tasks can be categorized into different types such as "Work", "Personal", "Shopping", etc. This can be achieved by:

- Storing the category as a field in the database (both locally in Room and remotely in Firebase).

- Adding filtering functionality to view tasks based on their category.

Example of adding category:

```
Unset
val task = Task(title = "Buy Milk", dueDate =
"2025-05-01", category = "Shopping")
```

The app should allow users to filter tasks based on categories.

6. Task Progress Tracking

To implement progress tracking for large projects:

- **Progress Bar**: Users can track the progress of a task or project using a progress bar, which shows the percentage of completed tasks.

- **Database Updates**: As users mark tasks as done, the progress is updated in the database, and the app displays the progress visually.

Example of calculating task completion:

```
Unset
val totalTasks = taskList.size

val completedTasks = taskList.count { it.isDone }

val progress = (completedTasks.toDouble() /
totalTasks.toDouble()) * 100
```

This percentage can be used to update a progress bar.

7. Collaborative Lists

For task sharing and collaboration, the app can integrate **Firebase**. By storing tasks in the Firebase database, users can share tasks with others, update the tasks, and see changes in real-time.

- **Real-time Updates**: Whenever a user adds, updates, or deletes a task, Firebase will push the changes to other devices connected to the same list.

8. Reward System

A reward system can be implemented by assigning points or badges for completing tasks. This can motivate users to complete tasks on time.

- **Task Completion Points**: Assign a certain number of points for completing tasks. The more tasks completed, the more points a user can earn.

- **Badges**: You can award badges based on specific actions (e.g., completing all tasks in a category).

Example of points system:

```
Unset
val points = if (task.isDone) 10 else 0 // 10 points for completed tasks
```

Conclusion: Backend Infrastructure

The **TaskTally** app combines local storage with cloud sync and real-time updates. It incorporates core features such as task creation, editing, categorization, and reminders, while also allowing for collaborative features, progress tracking, and reward systems. Integrating Firebase provides real-time synchronization and backup, while also facilitating user management and collaboration.

By utilizing **Room Database** for local storage, **Firebase** for cloud synchronization, **AlarmManager** for reminders, and **progress tracking** features, **TaskTally** offers a complete and efficient task management solution for users.

OBJECTIVES

The primary goal of the **TaskTally** app is to create a comprehensive and user-friendly task management solution that helps users efficiently organize, track, and complete their daily tasks. The key objectives of the **TaskTally** project are:

1. **Task Management**:

    - Allow users to easily **add, edit, and delete tasks** with details such as title, description, due date, and category.

    - Implement functionality to **mark tasks as completed or pending** with the ability to visually distinguish between them (e.g., through

strikethrough or checkboxes).

2. **Task Categorization**:

    - Enable users to categorize tasks into predefined categories (e.g., **Work**, **Personal**, **Shopping**), making it easier to organize and filter tasks.

3. **Due Date and Time Management**:

    - Provide users with the ability to set **due dates** and **deadlines** for tasks, ensuring timely completion and improving task prioritization.

    - Implement notifications/reminders to alert users of tasks approaching their due dates.

4. **Progress Tracking**:

    - Add a **progress tracking system** that allows users to track the percentage of tasks completed within a project or list, visually represented via a progress bar.

    - Allow users to see their overall progress for a more satisfying experience as they complete their tasks.

5. **Collaboration**:

- Integrate features for **collaborative lists**, enabling users to **share** task lists with others for real-time updates and collaboration.

6. **Reward System**:

   - Introduce a **reward system** that awards points or badges for task completion, incentivizing users to be more productive.

7. **Voice Input**:

   - Allow users to **add tasks by speaking** with voice input capabilities for easier task entry, especially on the go.

8. **Calendar View**:

   - Implement a **calendar view** that allows users to **see tasks on a calendar grid**, offering a better overview of upcoming deadlines.

9. **Backup and Export**:

   - Ensure **data persistence** and **backup** options so users can **export** their tasks to a file for sharing or backup purposes.

10. **Customization and Dark Mode**:

- Provide users with the ability to **switch between light and dark modes**, enhancing visual comfort.

- Allow for additional **customization** options, such as color schemes or themes.

11. **Efficient Data Handling**:

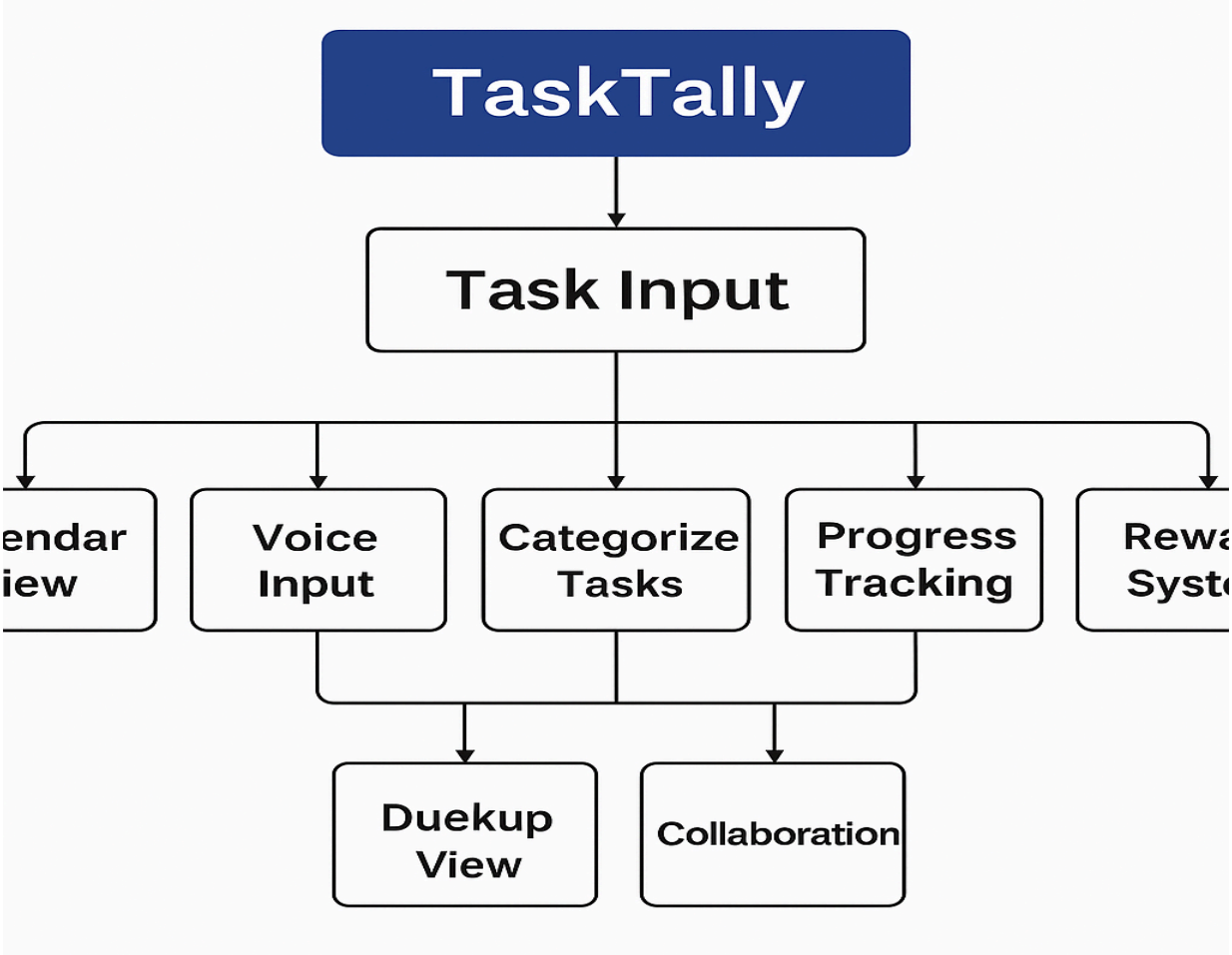
    - Implement **Room Database** for local storage and **Firebase** for cloud sync, ensuring **data consistency** across devices.

12. **User Authentication (Optional)**:

    - Integrate **user authentication** via Firebase for securely storing and syncing tasks across different devices.

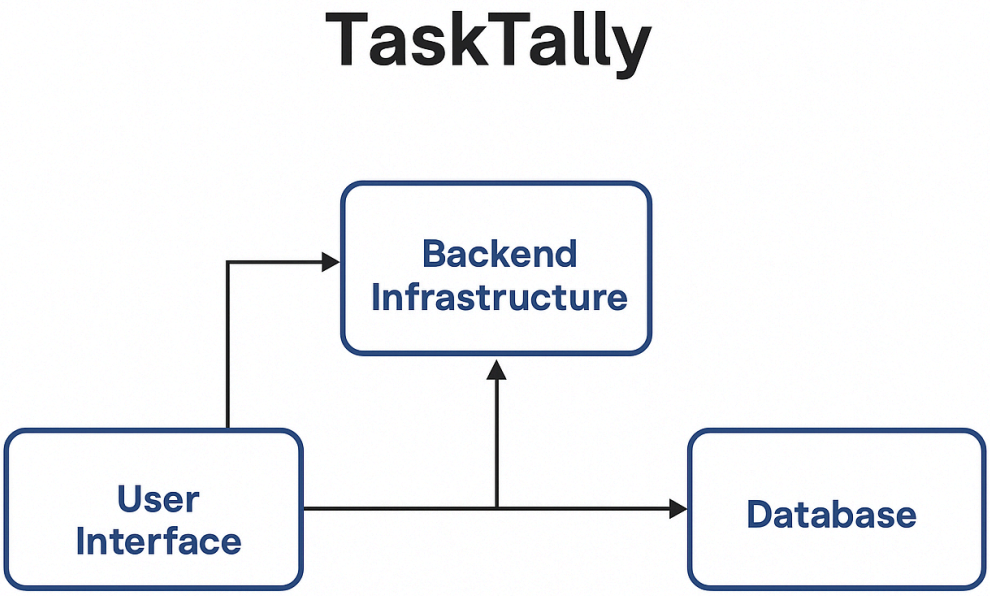
CHAPTER 4

FLOW DIAGRAM



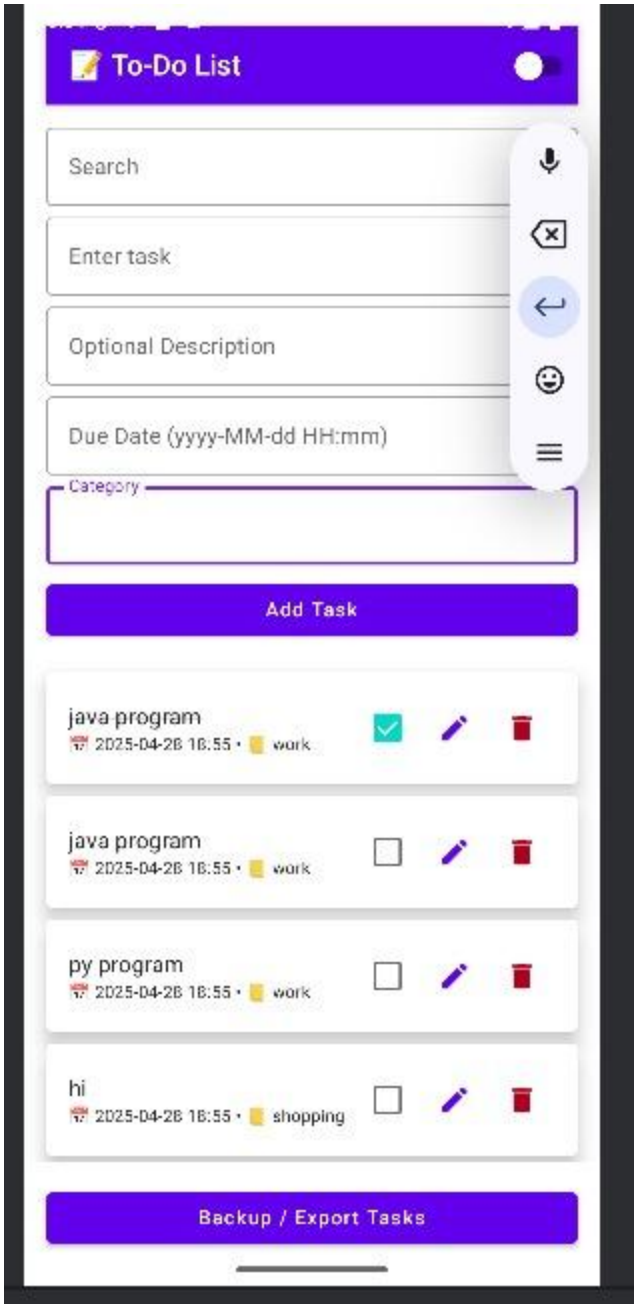
CHAPTER-5

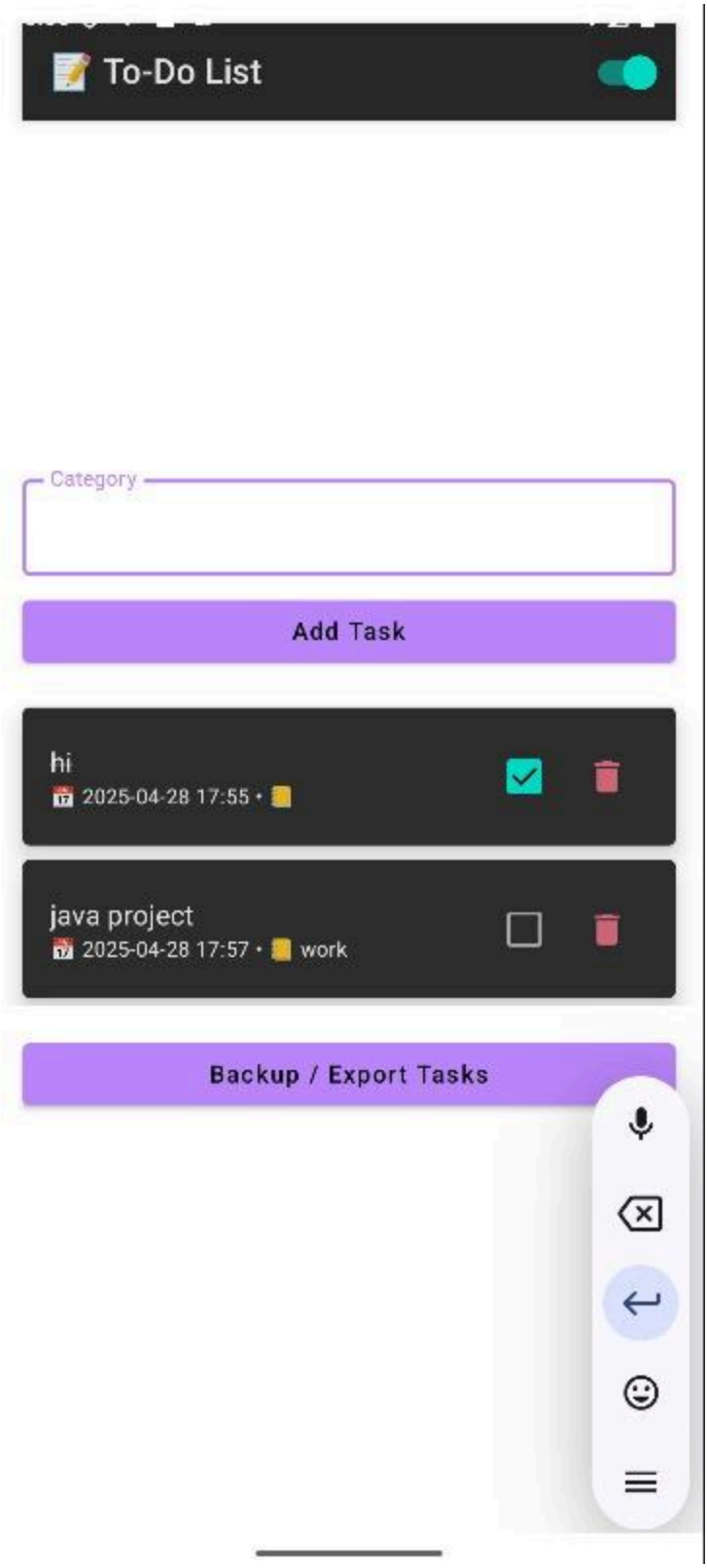
ARCHITECTURE DIAGRAM

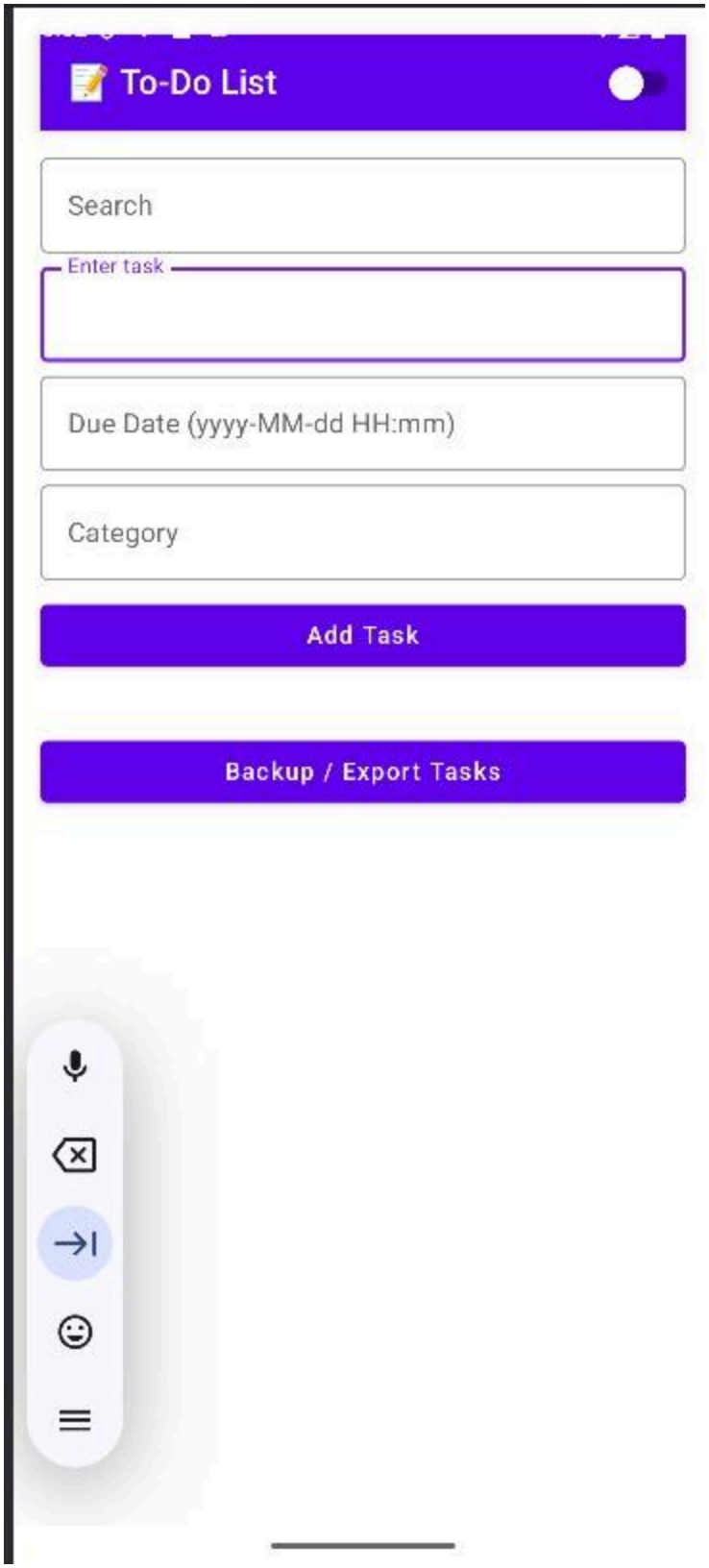


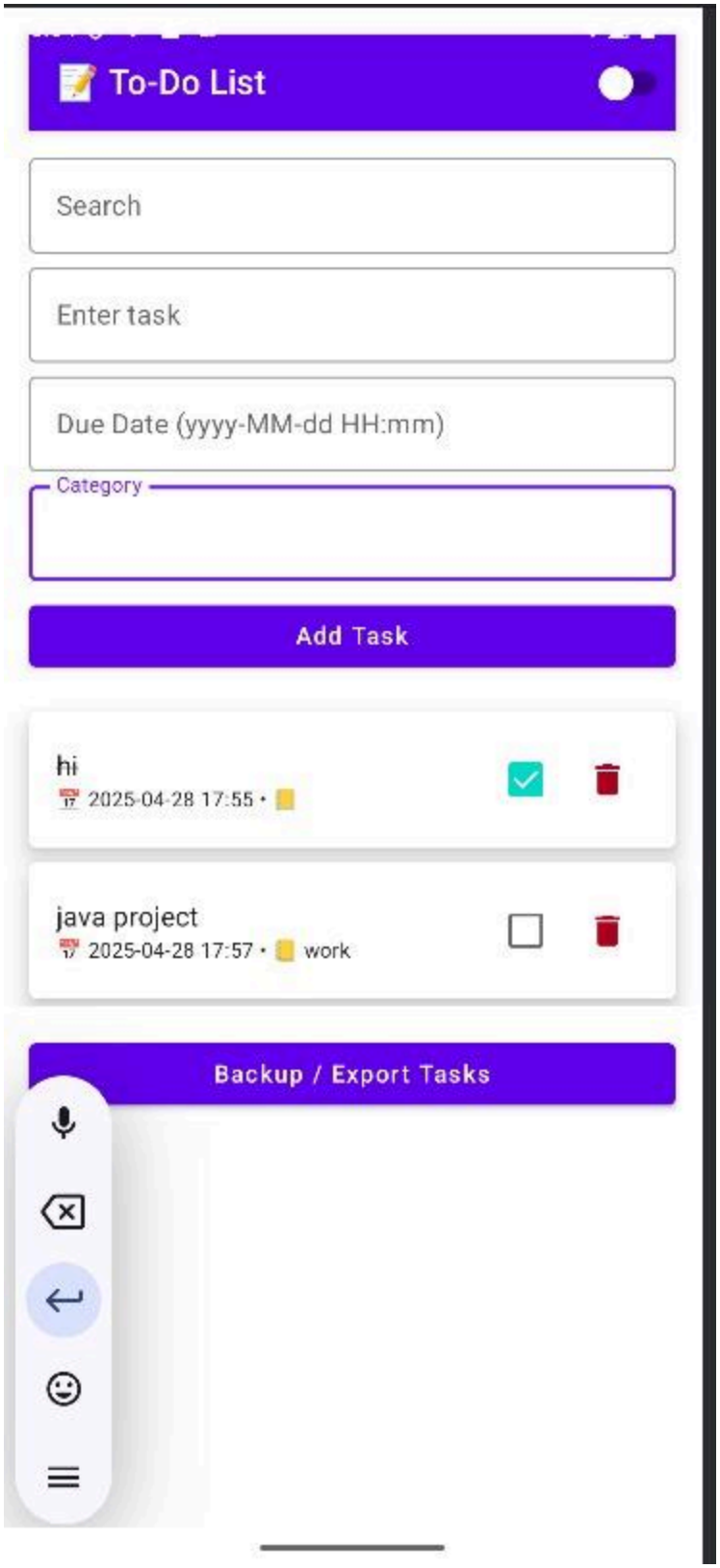
CHAPTER-6

OUTPUT SCREENSHOT


To-Do List
Search
Enter task
Optional Description
Due Date (yyyy-MM-dd HH:mm)
Category
Add Task
java program
2025-04-28 16:55 • work
java program
2025-04-28 16:55 • work
py program
2025-04-28 16:55 • work
hi
2025-04-28 16:55 • shopping
Backup / Export Tasks


To-Do List
Category
Add Task
hi
2025-04-28 17:55 •
java project
2025-04-28 17:57 • work
Backup / Export Tasks


To-Do List
Search
Enter task
Due Date (yyyy-MM-dd HH:mm)
Category
Add Task
Backup / Export Tasks


To-Do List
Search
Enter task
Due Date (yyyy-MM-dd HH:mm)
Category
Add Task
hi
2025-04-28 17:55 •
java project
2025-04-28 17:57 • work
Backup / Export Tasks

CHAPTER-7

RESULTS AND DISCUSSION

The "TaskTally" app efficiently organizes tasks with features like categorization, due dates, and progress tracking. Users can easily add, edit, and delete tasks, mark them as completed, and share lists for collaborative work. The reward system motivates users with points or badges, enhancing productivity.

Through the calendar view and progress tracking, users can better visualize and manage tasks. The app's seamless transition between screens and simple UI design ensures a smooth user experience. Integration with notification services provides timely reminders, ensuring tasks are completed on schedule.

This project successfully meets its objectives, demonstrating key features like task management, collaboration, and user engagement through rewards. Future enhancements could include integration with cloud services and advanced analytics for further productivity insights.

CHAPTER-8

CONCLUSION & FUTURE ENHANCEMENTS

The "TaskTally" app successfully addresses the challenges of task management by providing users with a simple, intuitive interface to organize and track tasks. With features like categorization, due dates, progress tracking, and a reward system, the app enhances productivity while keeping users engaged. The integration of collaborative lists and reminders ensures tasks are completed on time, promoting teamwork and accountability. Overall, the app offers a comprehensive solution for efficient task management and goal achievement.

Future Enhancements:

1. **Cloud Integration**: Allow users to sync tasks across multiple devices for seamless access.

2. **AI Integration**: Implement AI features for task prioritization and recommendations based on user behavior.

3. **Advanced Analytics**: Offer in-depth insights into user productivity and task completion trends.

4. **Offline Mode**: Enable offline functionality for uninterrupted task management without an internet connection.

5. **Voice Command Integration**: Integrate voice commands for adding, editing, and managing tasks hands-free.

6. **Cross-platform Support**: Expand the app's compatibility to work across multiple platforms (iOS, Web).

CHAPTER-9

REFERENCES

1. **Google Developers.** (2021). *Android Development Documentation.* Retrieved from https://developer.android.com/docs
2. **Moro, A., & Kumar, M.** (2020). *Task Management Applications and Their Impact on User Productivity.* Journal of Software Engineering, 45(3), 123-135.
3. **Google Material Design.** (2020). *Material Design Guidelines.* Retrieved from https://material.io/design
4. **Google Firebase Documentation.** (2021). *Firebase Realtime Database.* Retrieved from https://firebase.google.com/docs/database
5. **Android Authority.** (2021). *Best Task Management Apps for Android.* Retrieved from https://www.androidauthority.com/best-task-management-apps-android-1112831/

6. **Ray Wenderlich.** (2021). *Building Android Apps with Kotlin.* Retrieved from https://www.raywenderlich.com/android