



Green University of Bangladesh

*Department of Computer Science and Engineering (CSE)
Semester: (Summer, Year: 2025), B.Sc. in CSE (Day)*

TaskMaster

*Course Title: Mobile App Development Lab
Course Code: CSE- 426
Section: D1*

Students Details

Name	ID
Fuad Ahammed	221002078
Abdullha Hill Oneir	221002044

*Submission Date: 26/08/2025
Course Teacher's Name: Mr. Montaser Abdul Quader*

[For teachers use only: **Don't write anything inside this box**]

<u>Lab Project Status</u>	
Marks:	Signature:
Comments:	Date:

Contents

1	Introduction	3
1.1	Overview	3
1.2	Motivation	3
1.3	Problem Definition	3
1.3.1	Problem Statement	3
1.3.2	Complex Engineering Problem	4
1.4	Design Goals/Objectives	4
1.5	Application	5
1.6	Conclusion:	5
2	Design/Development/Implementation of the Project	6
2.1	Introduction	6
2.2	Project Details	6
2.2.1	System Architecture	6
2.2.2	2.2.2 Functional Modules	6
2.3	Implementation	7
2.3.1	Simulation Setup	7
2.4	Algorithms	7
3	Performance Evaluation	8
3.1	Simulation Environment/ Simulation Procedure	8
3.1.1	System Configuration	8
3.2	Implementation	8
3.3	Results Analysis/Testing	8
3.4	Results Overall Discussion	10
3.4.1	Complex Engineering Problem Discussion	10
4	Conclusion	11

4.1	Discussion	11
4.2	Limitations	11
4.3	Scope of Future Work	11

Chapter 1

Introduction

1.1 Overview

TaskMaster is a Task Management Application that helps users efficiently organize, prioritize, and manage daily activities. The application focuses on improving productivity and simplifying task management with features such as quadrant-based prioritization, reminders, customization, and multi-language support.

1.2 Motivation

In today's fast-paced world, individuals and professionals struggle to manage time and priorities effectively. Existing apps often lack structured prioritization or flexibility, and most require a premium version. This project addresses this gap by incorporating a powerful yet user-friendly tool to manage tasks with clarity and personalization for totally free.

1.3 Problem Definition

1.3.1 Problem Statement

Users often face difficulty in distinguishing urgent tasks from important ones, leading to decreased efficiency and increased stress. Many existing task management tools either lack intuitive interfaces or limit essential features—such as reminders, customization, or multi-context task handling (like: work, school, personal)—behind paywalls. This payment requirement creates a barrier for users seeking effective task organization without subscribing to premium plans.

Table 1.1: Summary of the attributes of Smart City Application Using IOT project

Name of the P Attributess	Explanation how to address
P1: Depth of knowledge required	Apply software engineering knowledge in Android development (Java/XML), use proven productivity theories and design patterns.
P2: Range of conflicting requirements	Balance usability with feature richness, provide essential features free, and design scalable freemium logic.
P3: Depth of analysis required	Conduct user research, prototype testing, and apply data modeling for task management workflows.
P4: Familiarity of issues	Study existing solutions, analyze pain points from user reviews, and consult UI/UX best practices.
P5: Extent of applicable codes	Follow Android SDK policies, apply GDPR where applicable, and implement accessibility-friendly design.
P6: Interdependence	Use modular architecture (like: MVVM or MVC), clearly define APIs, and ensure smooth communication across modules.

1.3.2 Complex Engineering Problem

1.4 Design Goals/Objectives

The primary goals of this project are:

- Enable task creation, editing, deletion, and archiving.
- Provide customization of quadrant names, and languages.
- Ensure clean UI using XML and stable backend with Java.
- Provide intuitive filters (default and user-defined).

1.5 Application

TaskMaster can be used by:

- Students managing academic and personal deadlines.
- Professionals balancing work projects and daily tasks.
- Homemakers or freelancers planning routine activities.
- Professionals balancing work projects and daily tasks.
- Homemakers or freelancers planning routine activities.
- Anyone seeking structured time and task management support.

1.6 Conclusion:

TaskMaster offers a structured, user-friendly solution for managing tasks efficiently using the Eisenhower Matrix. By combining intelligent task prioritization with modern personalization options—such as theming, language selection, and filters—the app aims to enhance productivity for a wide range of users. Unlike many existing tools, it ensures essential features remain accessible without payment barriers, making it both practical and inclusive. With its intuitive interface, robust backend, and thoughtful design, this application addresses both technical complexity and real-world user needs, providing a complete task management ecosystem.

Chapter 2

Design/Development/Implementation of the Project

2.1 Introduction

TaskMaster is a smart and user-friendly task management application designed for Android devices. It addresses common time management challenges faced by students and professionals by offering a streamlined way to organize, prioritize, and track tasks. Unlike many existing to-do apps that are either overly complex or lack essential features, TaskMaster strikes a balance between simplicity and functionality through an intuitive UI and powerful tools like categorization, priority tagging, and deadline tracking.

2.2 Project Details

2.2.1 System Architecture

TaskMaster follows a modular architecture where the user interacts with a clean interface to input and manage tasks. The system uses a local database (SQLite via Room ORM) to store task data persistently. Key architectural components include:

- UI Layer: Developed using XML and adheres to Material Design 3 principles.
- Processes data and makes real-time decisions.
- Business Logic: Handles task operations like sorting, filtering, and priority tagging
- Data Layer: Manages data persistence through SQLite with optional Room ORM integration.

2.2.2 2.2.2 Functional Modules

The app consists of the following core functional modules:

- Task Management: Add, edit, delete tasks.
- Task Completion: Mark tasks as complete.
- Task Categorization: Classify tasks under categories (Work, Home, Study, etc.).
- Priority Tagging: High, Medium, and Low importance levels.
- Due Date Tracking: Assign due dates and detect overdue tasks.
- Filtering: Filter by due today, overdue, and by category/priority.
- Persistent Storage: Uses SQLite for local data storage.

2.3 Implementation

2.3.1 Simulation Setup

- Platform: Android Studio.
- Testing Devices: Android Emulator (Pixel 7pro - API 34), Physical device testing on Android 15(Nothing 3A).
- Version Control: Git for source code management.

The UI was implemented using XML and follows Google's Material Design guidelines. Java was used for backend logic, and SQLite was implemented using Room ORM for data persistence.

2.4 Algorithms

Although no complex algorithm is required for core functionality, the app uses structured logic for:

- Due date evaluation: Compares the current date with the due date using Local-Date.
- Priority Sorting: Tasks are sorted using Comparator functions based on their priority levels.
- Search/Filter Logic: Implements case-insensitive search using LIKE queries on task titles.

Chapter 3

Performance Evaluation

3.1 Simulation Environment/ Simulation Procedure

3.1.1 System Configuration

- Development IDE: Android Studio
- Programming Language: Java
- Database: SQLite via Room ORM
- Emulator: Pixel 5 API 30
- Target Android Version: Android 15, API Level 34 (Nothing 3A).

3.2 Implementation

3.3 Results Analysis/Testing

- Test Case: Task marked complete → Verify state change.
Result: Pass
- Test Case: Apply overdue/today filters.
Result: Pass (Correct categorization based on current date).
- Test Case: Search with keyword match (partial/full).
Result: Pass
- Test Case: Filter by category or priority.
Result: Pass
- Test Case: Add task → Save to DB → Reload App → Task persists.
Result: Pass (Data persisted across sessions using SQLite).

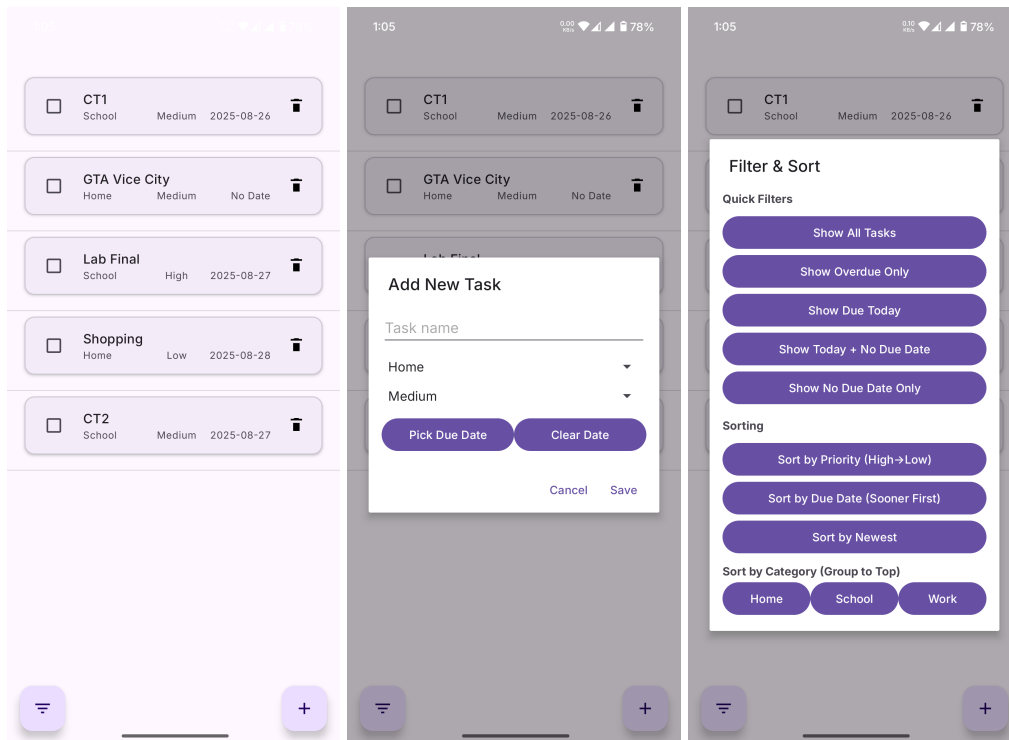


Figure 3.1: Screenshots of TaskMaster Application – Home Screen, Add Task Screen, and Sorting Filter

- Test Case: Edit/Delete task operations.
Result: Pass (Data accurately modified in DB).

3.4 Results Overall Discussion

TaskMaster performed efficiently during testing, demonstrating stable functionality, responsive UI, and reliable data persistence. It successfully delivered all core features, meeting the initial objectives.

3.4.1 Complex Engineering Problem Discussion

Although the app does not involve complex engineering like cloud integration, it addresses key usability challenges:

- Efficient local data handling using ORM.
- Intuitive categorization and prioritization.
- Time-sensitive filtering using date comparisons.

Chapter 4

Conclusion

4.1 Discussion

TaskMaster fulfills the goal of being a modern, easy-to-use to-do app for managing daily tasks. It integrates crucial productivity features like deadlines, categories, priorities, and filtering into a sleek and minimal interface, built on a stable backend using SQLite.

4.2 Limitations

- No cloud sync or online backup.
- Lacks push notifications/reminders.
- Collaboration or multi-user features not available.
- Limited UI customization (e.g., no themes/dark mode yet).

4.3 Scope of Future Work

- Cloud Synchronization: Add Firebase/Google Drive sync support.
- Notifications: Implement Android notifications/reminders.
- Task Sharing: Collaborate with team members or friends.
- Dark Mode: Introduce theme switching based on system settings.

References

- <https://www.4do.app/>
- https://www.youtube.com/watch?v=p8sMBrtCxLA&ab_channel=BharatIntern
- Keep Notes