

System Usability Scale for Cross-Platform Mobile Student's Task Management System

Omosebi Paul Adeoye*, Ogunsami Zainab Adekite and Omirinlewo Adeyinka Bimpe

Department of Computer and Information Sciences, Trinity University Yaba Lagos

*adesong2002@yahoo.com

ABSTRACT: Effective task management is crucial for students to achieve academic success, reduce stress, and increase productivity. Despite the availability of generic task management applications like Evernote and Google Tasks, these tools often lack features tailored specifically to the needs of students, such as task hierarchy and seamless collaboration for group projects. This study aims to address these gaps by introducing an innovative cross-platform mobile application specifically for student task management, designed and developed to meet the unique demands of students. Using the Dart programming language and Flutter framework, the proposed system offers a mobile solution that operates on both the iOS and Android platforms, leveraging the potential of ubiquitous computing. It provides a comprehensive set of functions that make collaborating easier, like communication tools, task prioritization, intuitive task structure, and task refinement. Students' collaborative experience is optimized by the application's support for file sharing and real-time communication. The development process involves analysis of related works and designing and implementing the system using Flutter and Google Firebase for real-time database support. The result of this study is a comprehensive mobile task management application that runs on both iOS and Android-based devices, providing extensive features that enable students to manage their academic responsibilities more effectively, thereby improving their productivity and academic performance. A System Usability Scale (SUS) survey was conducted to measure the perceived usability of the proposed system and obtained a good score of 71.83, which indicates the system is easy for users to interact with and understand the system's processes and interface.

KEYWORDS: Task Management, Mobile Application, Cross-Platform, Flutter Framework, Android, iOS, Google Firebase.

I. INTRODUCTION

Task Management is considered a significant tool in enabling students to achieve academic success. Tasks pertinent to students, such as projects, assignments, quizzes, tests, and personal obligations outside the standard student curriculum, must be executed against their specified deadlines as required. Effective task management, the ability to set goals and priorities, and monitoring the use of time can increase productivity and reduce stress, resulting in work efficiency and academic success [1]. Task management can help either individuals achieve goals, or groups of individuals collaborate and share knowledge for the accomplishment of collective goals. Effective practice of Task Management can enhance the rate at which students' tasks are executed resulting in a proportional increase in students' productivity. Computer Science can be used as a tool for analyzing task management and developing software that will aid student task management. Several generic applications already cater to this need, including Evernote, Notion, Todoist, Progress, and Google Tasks, among many others. Task Management is not only limited to students or the academic field but also in every other field, such as banking and finance, governance, medicine, and health care, among many others, as well as the simple household. Current generic task management software such as those earlier mentioned do not possess features that are specific to the needs of students or those in the field of academics such as enabling task refinement as well as seamless collaboration for group projects or tasks.

These generic tools hinder the completion of students' tasks due to several shortcomings observed during the review and analysis of existing systems. Many people face challenges when trying to organise their activities both at work and in their personal lives [2]. Hence why this study is carried out to design and implement a task management system that is specifically tailored to satisfy the needs of students as well as override common shortcomings in existing systems with features such as task hierarchy and prioritization, task refinement, and

easy joint student project collaboration to allow students work on joint projects seamlessly. In an academic context, group tasks such as assignments and presentations are commonly assigned to students. However, it is widely known that group assignments can be difficult to manage [3], hence why this study's proposed system aims to provide a collaborative interface for group tasks. This includes file transfer as well as communication within the platform. Due to the recent advancements in computing, specifically ubiquitous computing, the development of this system as a mobile application is highly beneficial to the student population as approximately 60% of the global population currently owns a smartphone, a significant increase from the 34% reported in 2019 [4]. This project is also developed with the two most common mobile operating systems in consideration, which are Android and iOS. Due to this, the system is implemented to run on both platforms using Dart programming language for development, and Flutter framework as it supports cross-platform execution of the same program.

Task management is based foundationally on planning, as it involves planning what activity or task should be executed at what time and of what importance. Planning a suitable and effective way to execute assigned tasks provides a clearer view and visualization of how to execute said tasks. Poor planning contributes largely to the failure of many students; hence, students must be equipped with tools that will enhance their productivity. The students' incapacity to manage themselves, such as controlling and managing their time effectively will lead them to find more difficulties in achieving a better academic achievement [5]. A very important factor that determines the academic success of a student involves their ability to execute tasks and their expected responsibilities at their assigned time. This includes turning in assignments, projects, and research by the specified deadline. Traditionally, these tasks would be handwritten or kept in memory, which is prone to loss, damage, and forgetfulness. In recent times where task management tools are in existence, bottlenecks such as a lack of task hierarchy/prioritization and poor task collaboration hinder students' task management.

This project falls into the category of personalized and adaptive learning tools. Adaptive learning technology provides personalized learning at scale by assessing learners' current skills/knowledge, providing feedback and content, and then constantly monitoring progress by utilizing learning algorithms that provide real-time updates and the necessary tools to improve student learning [6]. According to, [7] personalized and adaptive learning offers great opportunities to promote student learning. This is because the proposed system permits user personalization such that users can create their personalized workflow. As users interact with the system over a while, it can potentially be integrated into their work habits promoting improvements in time management, productivity and task completion rates. By considering all these problems, the design of a mobile task management system that tackles the previously expressed issues is an adequate and effective solution. This study aims to design and develop a cross-platform mobile student task management system. The proposed system solves the observed problems stated in the previous section by;

- a) Designing and developing an intuitive interface for students to add, organize, refine and prioritize tasks and assignments easily.
- b) Ensure the app works seamlessly on Android and iOS devices for a more comprehensive user reach.
- c) Allow students to collaborate on projects or share tasks with classmates, promoting teamwork and group study sessions.

II. RELATED WORKS

This section involves the review of related works to obtain a critical analysis of current academic literature in the study of task management system design and development. It is achieved by examining relevant academic contributions to derive in-depth knowledge of the study, identify running trends as well as highlight gaps for innovation in the proposed system. In this review, the set of criteria applied are largely relevant to task management systems and they include; platform type, collaborative user interface, task prioritization, task refinement and task sharing features.

Recent research in task management systems and their design and development has shown diverse approaches to improving productivity and organization. Reference [8] developed an Android-based application for departmental use in colleges, featuring real-time messaging and task allocation. While this system offered practical functionalities such as task assignment and user restriction/access control, it lacked a task refinement model and was limited to Android devices. Reference [9] addressed the issue of procrastination in their task management app design, incorporating gamification elements to enhance user engagement as well as intuitive user interface design to overcome the bottleneck that is the steep learning curve users often experience when using other task management applications. However, their system also lacked a task refinement model and real-time messaging capabilities.

Reference [10] developed an Android-based task management application, designed to enhance user productivity. This study illustrates the researcher’s objectives which include increasing goal success or task completion probabilities which is achieved by the incorporation of gamification techniques and a modern, simple user interface. While this provided statistical visualisation of task progress it lacked features such as task prioritization, real-time messaging functionality as well as a task refinement model. Reference [11] proposed a web-based student task management system with features like group collaboration and analytics, but it was heavily dependent on Internet connectivity and lacked mobile implementation. More recently, Reference [12] introduced an intelligent task management system utilizing AI and machine learning for task prioritization and suggestions. While innovative, this system was console-based and lacked a user-friendly interface and mobile implementation. Reference [13] opined that students need to predict their Cumulative Grade Point Average (CGPA) in order to improve their abilities in getting a better graduating CGPA. This can be done with proper task management. These studies collectively highlight the ongoing evolution of task management systems and the need for comprehensive solutions that address various user needs and technological advancements.

Table 1: Related Works

| S/N | Publication-Author(s) | Year | Findings | Identified Gap |
|-----|---|------|--|---|
| 1. | Online Task Management System (OTMS) – Hedaoo, G., Thoke, P., Tabhane, R., Meshram, S., Kumbhalkar, S., & Barapatre, M. | 2018 | <ul style="list-style-type: none"> • The proposed system of this study is an Android-based mobile application. • The scope of the study is reduced to the department of a college. • The system has two stakeholders; an admin user such as the Head of the Department who can assign tasks to staff and other users. • The proposed system permits real-time messaging, notifications, task allocation and complaint filing. | <ul style="list-style-type: none"> • Absence of a task refinement model • It is not student-centred as students cannot create or allot tasks. • Limited user reach as it is only accessible via Android-based devices. |
| 2. | Design of a Task-Management APP to Avoid Procrastination – Guimarães, C., & Lazaro, D. J. P. (| 2018 | <ul style="list-style-type: none"> • This study proposes a task management system that can aid users in avoiding procrastination. • The relationship between procrastination and task management is defined. • It provides empirical data on procrastination among students. • The proposed system employs gamification to facilitate user comfort. • Identified a common user problem via survey; learning curve of commonly used task management tools. | <ul style="list-style-type: none"> • Absence of a task refinement model • Absence of real-time messaging |
| 3. | Honeycomb: Android App Of Goal and Task Management for Increased Productivity – Arthur Knegetel | 2020 | <ul style="list-style-type: none"> • This study proposes an Android-based task management application. • Its core objective includes enhancing user productivity. • The proposed system incorporates gamification techniques as well as a modern user interface. • Features of the proposed system include goal setting, provision of statistics and recommendations to users. | <ul style="list-style-type: none"> • Absence of real-time messaging • Absence of task a task refinement model • Limited user reach as it is only accessible via Android-based platforms. |

| | | | | |
|----|---|------|---|---|
| 4. | Web-Based Student Task Management System – Nurzi, F. A., & Wahab, K. A. | 2022 | <ul style="list-style-type: none"> • This study proposes a web-based student task management. • Implementation of the proposed system is achieved using web technologies such as Next.js and TypeScript for the front-end as well as Node.js and GraphQL for the back-end. • The development methodology applied is Agile. • The proposed system is greatly student-centred as they can create tasks for themselves. • Group collaboration, real-time messaging, calendar view as well as analytics are provided features. | <ul style="list-style-type: none"> • Heavily dependent on internet connectivity. • Absence of task refinement model. • Absence of mobile computing implementation. |
| 5. | Intelligent Task Management System – Sravanth, M., & Dhanush, R. | 2023 | <ul style="list-style-type: none"> • The proposed system of this study is a Python-based Task Management system. • This study incorporates Artificial Intelligence and Machine Learning models to make accurate predictions based on task attributes entered by the user. • Natural Language Processing techniques are also applied to process task descriptions and enable the system to provide contextually relevant suggestions. • This study applies asynchronous processing to handle concurrent tasks. • This study identifies the relevance of advanced computing processes and techniques to expand/define the landscape of task management. • Enhanced task prioritization and handling via dynamic processes executed by A.I and M.L models. | <ul style="list-style-type: none"> • Absence of an intuitive user interface as the result is still console-based. • Absence of mobile computing implementation • Lack of real-time messaging and group collaboration interface. • Absence of task refinement model. |

The analysis of these related works and publications highlights observations such as common findings, trends over time as well as consistent gaps. Common findings derived from this review suggest that task management systems are being developed on various platforms (Android, web-based, Python-based). Common features include real-time messaging, notifications, task creation, prioritization and allocation, which are incorporated in some form. Trends observed over time suggest an increase in technological advancement such that Artificial Intelligence and Machine Learning are integrated into recent task management systems to enhance user productivity. More recent studies show a growing emphasis on heightening user experience by applying user-centric interface design principles such as gamification as well as identifying a need for the development of more intuitive interfaces to overcome a steep learning curve. Earlier systems displayed very limited user capabilities while later ones aim for a wider user reach. Consistent gaps identified in this review, suggested by a unanimous absence in task refinement features present an avenue in the current task management landscape as well as the limited platform availability whereby systems are commonly restricted to a single platform often which is highly dependent on internet connectivity. The absence of one or more comprehensive features such as real-time messaging, mobile access or group collaboration is also a gap to consider leveraging in the development of this study's proposed system. By identifying these findings and gaps, there exists an opportunity to enhance the utility and effectiveness of task management systems by developing a more robust system to enhance the utility and effectiveness of task management systems.

III. MATERIALS AND METHODS

This study applied several materials and methods to achieve its aim. The framework(s), methods and techniques utilised in this study include the following:

- i. Analysing and reviewing related works to obtain insight and gaps to leverage.
- ii. System Design and Analysis.
- iii. Implementation with Dart programming language and Flutter SDK.
- iv. Application of Firebase for authentication, database management and real-time messaging.

The review of related works has been earlier conducted in the previous section. Therefore, it will not be further discussed in this section. This study applies system design and analysis to ensure that the proposed system aligns with predefined objectives. Common findings, observed trends and consistent gaps for improvement observed from the analysis of related works are integrated into the development of the proposed system.

A) USE-CASE DIAGRAM

The diagram below illustrates the use cases of the proposed task management system, providing a comprehensive overview of its functionalities and user interactions. In this representation, two primary actors are identified: the "User" and the "Group Member," both of whom interact with the task management system for individual and collaborative tasks respectively. These actors symbolize the diverse user base that the system is designed to accommodate, ranging from individual users managing personal tasks to team members working on group projects as well as the ChatGPT API for task refinement. The "Task Management System" represents the proposed system to be developed in this study and within it is its functionalities. The functionalities also represent the use cases that are likely to occur as the user(s) operate the system. They include: Create Task, Assign Task, Set Priority, Collaborate on Task, Task Refinement etc.

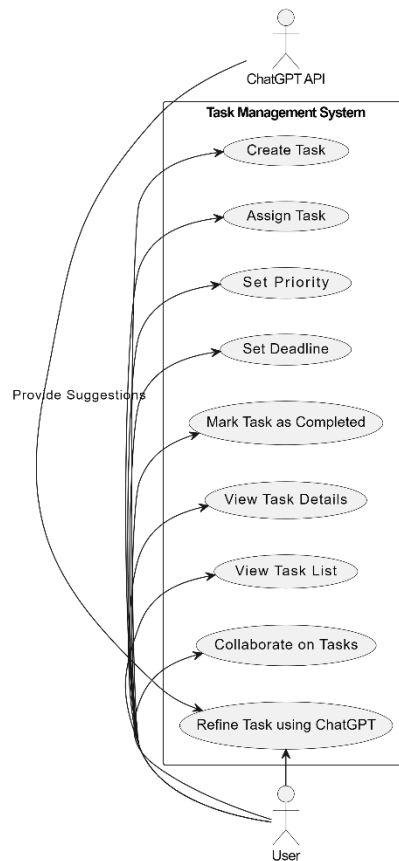


Figure 1: Use-Case Diagram of The Proposed System

B) ACTIVITY DIAGRAM

The activity diagram illustrates the behaviour of the proposed system such that when the task management application launches, the procedure starts. When a user starts a new task in the system, it asks if it's meant to be a group task or not. The system then allows users to assign group members who will work together

on the task if it is decided to make it a group task. When group members are assigned, the system allows them to work together on the task; however, if the task is not meant to be completed in a group, the system assigns it to a single user. The system then receives the user's request to create a task. The system verifies whether the task details have been confirmed and validates the information supplied for the task creation. the validation is successful, the system creates a new task and stores it in the database but in the case where the validation fails, the system displays an error message indicating the reason for the validation failure. The system then receives the request for task assignment and checks the permissions of the user to assign tasks. The system verifies whether the user has the necessary permissions to assign tasks. If the user has the required permissions, the system updates the task assignment in the database else if the user does not have the required permissions, the system displays a message indicating permission denial. The user sets the priority level for the task and the system receives the request to set the priority for the task. This request triggers the system to update the priority level of the task in the database. The user sets the deadline for completing the task. The system receives the request to set the deadline for the task and then updates the deadline of the task in the database. The user marks the task as completed after finishing it, and the system receives the request to mark the task as completed, which causes the system to update the status of the task as completed in the database.

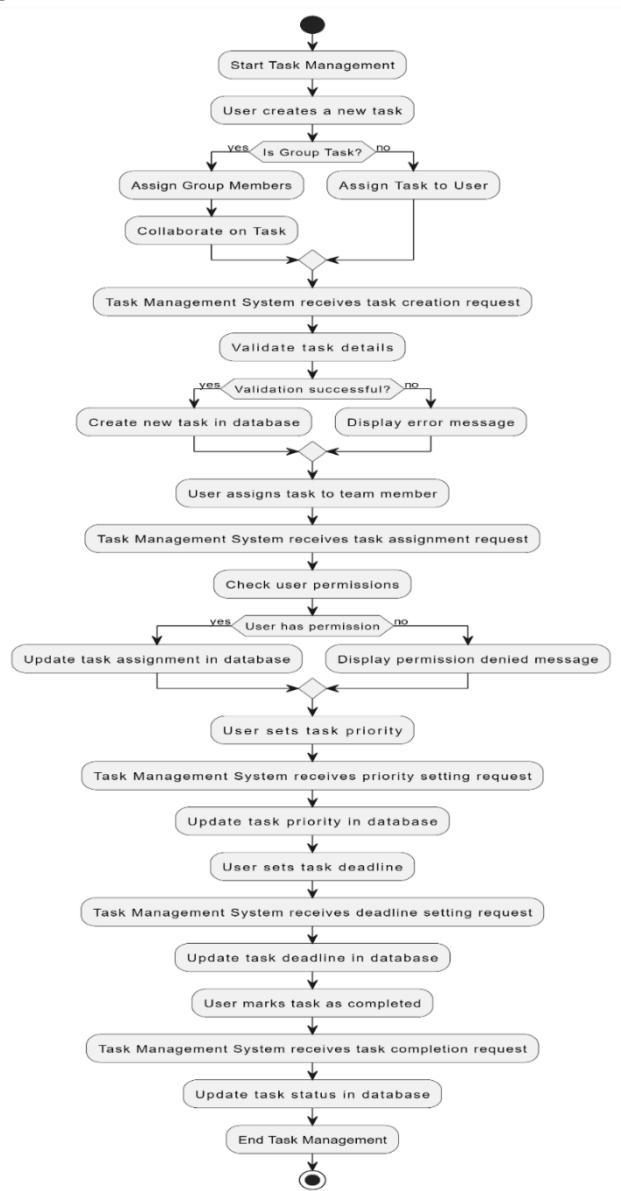


Figure 2: Activity Diagram of The Proposed System (Source: Author)

In this study, the development of the proposed system requires an array of tools to achieve the earlier proposed objectives. Some of these tools include programming languages for program writing, frameworks and other tools.

- a) The programming language used for this project is Dart. This programming language is used for writing programs for Flutter Applications. Commonly, Flutter is implied as the programming language in most instances, but rather, it is a Software Development Kit (SDK), a framework. Dart, as a programming language, is primarily used in Flutter for cross-platform mobile application development due to its versatile and high-performance features, which align with the objectives of the expected system. These include;
 - i. Cross-Platform Development: Dart as used in Flutter, allows developers to write code once and deploy it on multiple platforms, including iOS, Android, web, and desktop (Windows, macOS, Linux). This feature aligns with the objective of the expected system being run on different mobile platforms.
 - ii. Enhanced UI Development Library: Dart is particularly well-suited for building user interfaces (UIs). Its features contribute to a smooth and productive UI development experience which aligns with the objective of the expected system to have an intuitive user interface.
 - iii. Extensive Documentation: Dart is an extensively documented programming language with a wide array of libraries all of which permit the development of a robust and reliable system.
- b) Flutter is a software development kit (SDK) developed by Google for building natively compiled applications for mobile, web, and desktop from a single codebase. It supports both Android and iOS platforms, making it ideal for cross-platform development.
- c) Google Firebase is the framework, which is applied for backend and database functions, is a platform developed by Google for creating mobile and web applications. It provides a real-time NoSQL database, user authentication, push notifications, cloud storage, and hosting services, all of which provide real-time functionality to the application.
- d) ChatGPT 3.5-Turbo API is used to generate detailed subtasks for a given primary task, enhancing the overall task management experience for the user. It is implemented in the system's program as a function/method that constructs an HTTP POST request to the OpenAI API. This request includes a structured prompt instructing the ChatGPT 3.5-Turbo model to generate a specified number of subtasks based on the provided task details.

IV. RESULTS AND DISCUSSIONS

A. PROGRAM MODULES & INTERFACES

After implementation using the earlier discussed programming language, frameworks and other tools, the interface of the system appears as follows;

(a). Authentication Interface

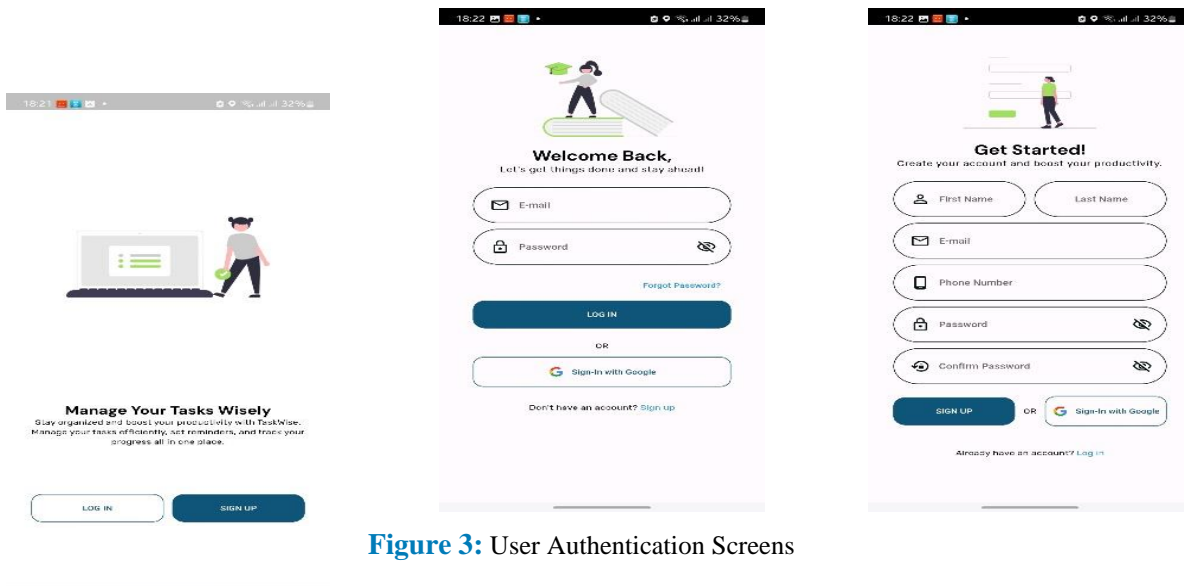


Figure 3: User Authentication Screens

In this segment, the user authentication feature provides users with the ability to create a new account or sign in using their existing account credentials upon opening the application for the first time. This functionality is made possible through the integration of Firebase authentication, which enhances security and user verification.

(b). Main Application Interface

This interface (Figure 4) serves as the home page of the entire application. Upon signing up or logging in, the user is redirected to this page to begin task creation and other related processes. It displays various functions such as a personalized welcome message to the user; “Hello, [First Name]”, using the first name value the user entered during sign-up which is stored in persistent memory; Firestore. A navigation bar is positioned below the screen to assist users navigate other functions or subsystems. In the centre of the navigation bar positioned at the bottom of the screen is an “Add” button, once clicked, redirects users to the task creation page. Figure 5 below displays the two options available for users regarding task creation, whereby they can select “Individual” for individual tasks or “Group” for group tasks. The selection of either of them redirects the user to the pages illustrated in Figures 6 and 7. Figure 6 displays the form required of the user to fill before creating or adding an individual task to their home page as in Figure 4. Text fields on this page include required input such as Task Name, Task Details, Priority, Status, Due Date and Set Time. The user also has the option to add subtasks if necessary. Upon filling in all necessary details, the user may add the task by clicking the “Add” button below or refine the task by clicking the “Refine” button. Figure 7 below is the task creation page for groups whereby a different approach to subtasks is employed whereby the user is requested to enter the number of group members for the task they are about to create. Upon entering the number of members, n , N number of fields dropdown for the user to create the tasks for each member. The user may also choose to refine the task but compared to the individual task page, generated subtasks are added as tasks for members that is; n , numbers of subtasks are generated and added as tasks for members.

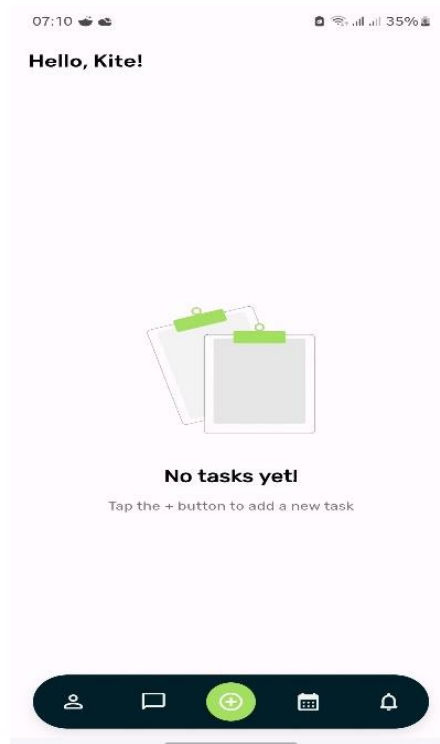


Figure 4: Main Homepage



Figure 5: Task Creation Page

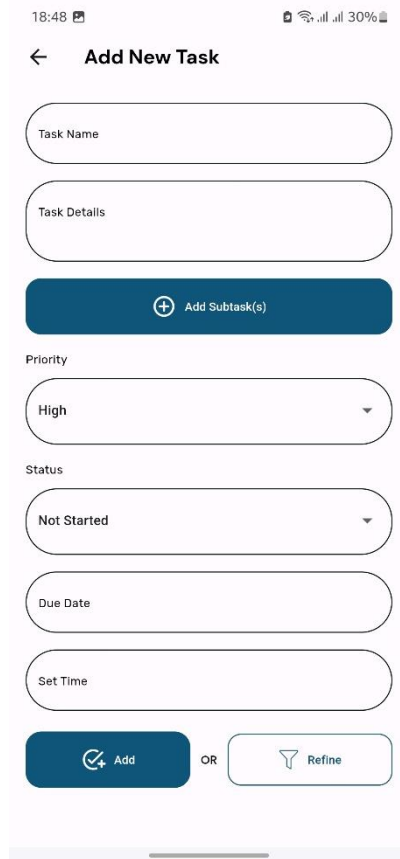


Figure 6: New Task Form

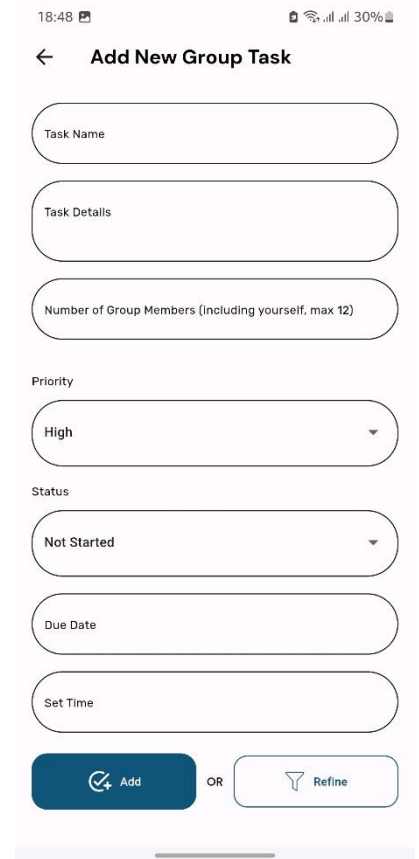


Figure 7: New Group Task Form

(c). Task Refinement

The user is redirected to the task refinement page as displayed in Figure 8, where the user can select a refinement level or specify a custom number of subtasks to which they want their task refined. After selection or custom number entry, upon clicking the “Generate Subtasks” button, the name of the task and details are sent as a POST request to the ChatGPT 3.5-Turbo API endpoint as it is used to dynamically generate subtasks based on the two specified parameters; Task Name and Task Details.

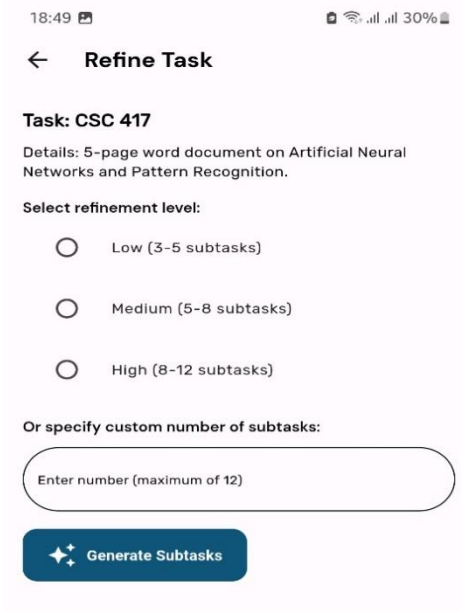


Figure 8: Task Refinement Screen

(d). Task Operations



Figure 9: Task View

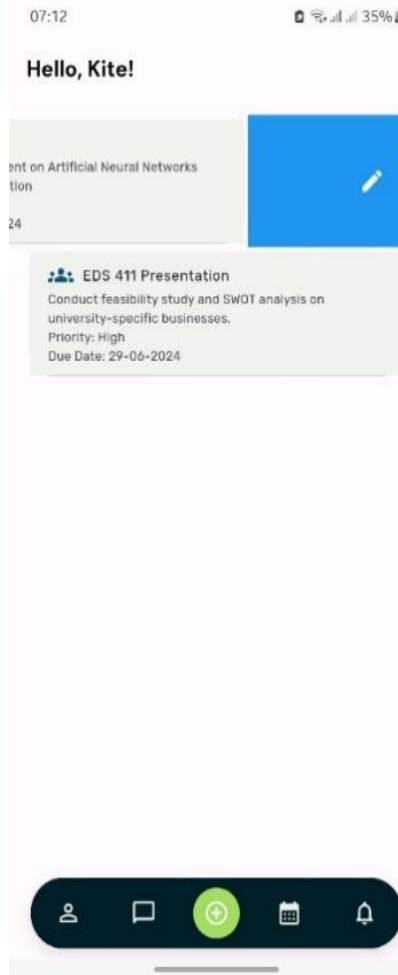


Figure 10: Edit Task

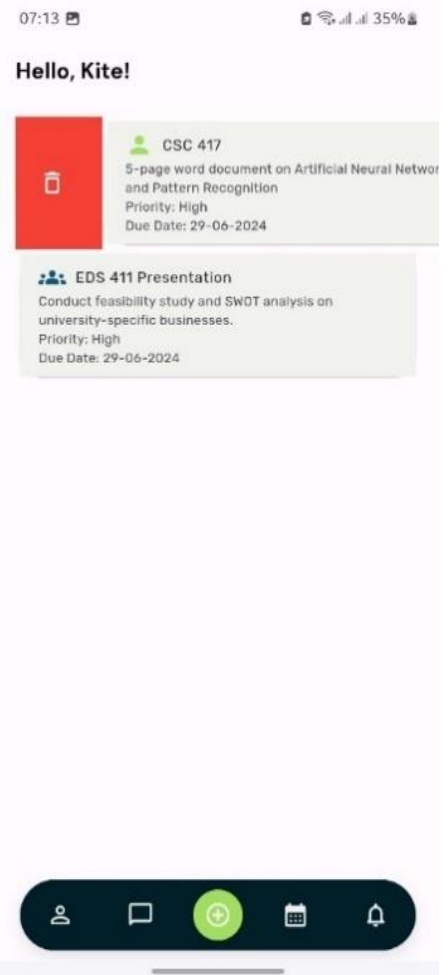


Figure 11: Delete Task

Figure 9 displays how tasks added from the task creation page are displayed on the home page. They appear as a list and each task is displayed as a tile containing selected details of the task added such as the name, details, priority and due date. Tasks created on either the Individual Task Page or Group Task Page are displayed on this list. They are differentiated by the use of an icon to indicate whether a task is an individual or group task. Figure 10 displays the implementation of the Dismiss widget to apply some functionality to the task entity in the list view. On swiping left, the user can edit or update an already existing task. Figure 11 displays the other functionality the Dismiss widget permits. On swiping right, the user can delete an already existing task.

(e). Subsystems

This includes added features that aid the intended functionality of the system and they include; Figure 12 displays the calendar view that appears when the user clicks the calendar icon in the bottom navigation bar on the home page. Here, the user can view their tasks through a calendar display where dates recorded under tasks have an indicator in the form of a red button to signify that it is a due date or deadline for a recorded task. Figure 13 illustrates the group chat interface which enables students to have real-time conversations where they can send messages or file attachments using Google Firestore. Figure 14 displays the user profile where several application settings are located for the user to access.

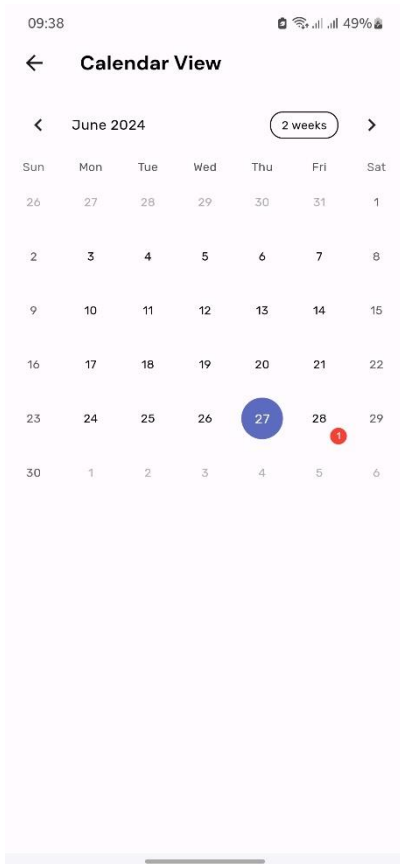


Figure 12: Calendar View

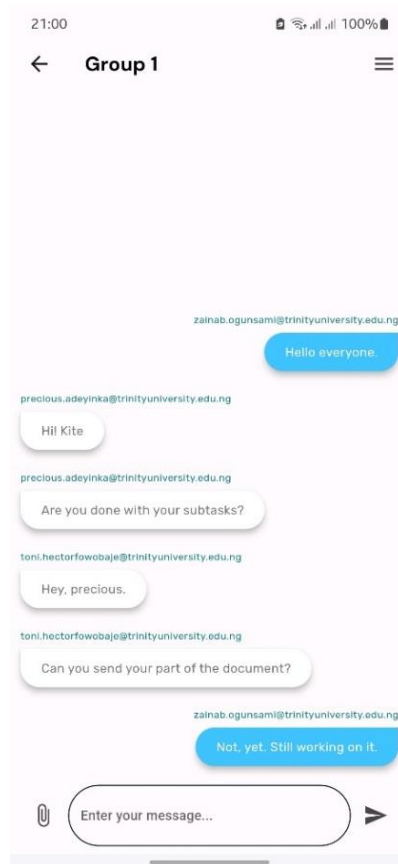


Figure 13: Chat Interface

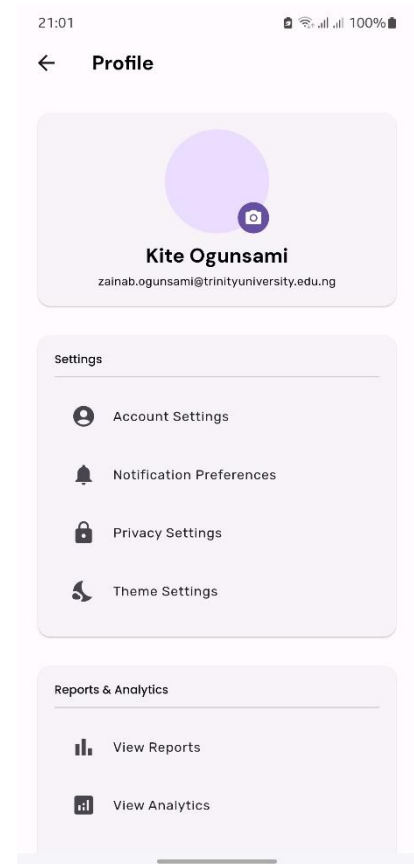


Figure 14: User Profile and Settings

B. SYSTEM USABILITY SCALE (SUS)

A System Usability Scale (SUS) survey was conducted to evaluate user feedback on the proposed system post-development. The survey consisted of 10 questions assessing the system's ease of use and user satisfaction. A total of 19 participants, all students, took part in the survey. The group comprised of 11 Android users and 8 iOS users.

Table 2: Sample of SUS Survey

| | Strongly | | | | Strongly |
|---|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 1 I think that I would like to use this task management system frequently. | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| 2 I found the system unnecessarily complex for managing tasks. | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| 3 I thought the task management system was easy to use across different devices. | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| 4 I think that I would need technical support to use all the features of this system. | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| 5 I found the various functions in this task management system were well integrated. | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| 6 I thought there was too much inconsistency in how tasks were displayed or managed. | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

| | | |
|----|---|-----------|
| | | ○ ○ ○ ○ ○ |
| 7 | I would imagine that most people would learn to use this task management system very quickly. | ○ ○ ○ ○ ○ |
| 8 | The app's interface looks visually appealing and consistent on my device (iOS/Android). | ○ ○ ○ ○ ○ |
| 9 | I felt very confident using the system to manage my tasks. | ○ ○ ○ ○ ○ |
| 10 | The overall user interface of the app enhanced my task management experience. | ○ ○ ○ ○ ○ |

Participants rated each SUS question on a scale from 1 to 5, where: 1 = Strongly Disagree and, 5 = Strongly Agree. However, according to Brooke's scoring, the outcome of these responses is rated on a scale of 0-4. The odd-numbered questions in the SUS have positive meanings, whereas the even-numbered questions have negative implications [14].

To calculate the SUS score:

(a). For odd-numbered questions – Positive Meanings (1, 3, 5, 7, 9): 1 is subtracted from the user's rating.

(b). For even-numbered questions – Negative Implications (2, 4, 6, 8, 10): 5 is subtracted from the user's rating.

(c). The sum of all these adjusted values is then calculated and multiplied by 2.5 to obtain the final score.

This final score typically ranges from 0 to 100. Higher scores indicate better usability; where a score of 70 is considered above average and acceptable, and scores over 85 are considered excellent suggesting a user-friendly system [15].

Table 3: SUS Score – Android Users

| Android- User Participants | Survey Scores | | | | | | | | | | SUS Score |
|----------------------------------|---------------|---|---|---|---|---|---|---|---|----|--------------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| 1 | 3 | 2 | 4 | 2 | 3 | 2 | 4 | 3 | 4 | 2 | 72.5 |
| 2 | 4 | 3 | 3 | 1 | 4 | 3 | 3 | 2 | 3 | 2 | 70.0 |
| 3 | 3 | 1 | 4 | 2 | 3 | 2 | 4 | 2 | 4 | 1 | 75.0 |
| 4 | 2 | 2 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 65.0 |
| 5 | 4 | 2 | 4 | 1 | 4 | 2 | 4 | 3 | 4 | 2 | 80.0 |
| 6 | 3 | 3 | 3 | 2 | 3 | 2 | 3 | 2 | 3 | 2 | 67.5 |
| 7 | 3 | 2 | 4 | 1 | 4 | 1 | 4 | 2 | 4 | 1 | 77.5 |
| 8 | 2 | 3 | 3 | 2 | 3 | 3 | 3 | 3 | 3 | 3 | 62.5 |
| 9 | 4 | 1 | 4 | 1 | 4 | 2 | 4 | 2 | 4 | 1 | 82.5 |
| 10 | 3 | 2 | 3 | 2 | 3 | 2 | 3 | 2 | 3 | 2 | 67.5 |
| 11 | 3 | 3 | 3 | 1 | 4 | 2 | 4 | 3 | 3 | 2 | 72.5 |
| Average SUS Score | | | | | | | | | | | 72.0 |

Table 4: SUS Score – iOS Users

| iOS-User Participants | Survey Scores | | | | | | | | | | SUS Score |
|--------------------------|---------------|---|---|---|---|---|---|---|---|----|--------------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| 1 | 4 | 1 | 5 | 2 | 4 | 2 | 5 | 1 | 4 | 2 | 85.0 |
| 2 | 2 | 3 | 3 | 4 | 2 | 3 | 3 | 2 | 2 | 3 | 57.5 |
| 3 | 3 | 2 | 4 | 1 | 3 | 2 | 4 | 3 | 3 | 1 | 70.0 |
| 4 | 5 | 1 | 4 | 2 | 5 | 1 | 4 | 2 | 5 | 1 | 90.0 |
| 5 | 1 | 4 | 2 | 3 | 2 | 4 | 3 | 3 | 2 | 4 | 45.0 |
| 6 | 3 | 2 | 3 | 2 | 4 | 1 | 4 | 2 | 4 | 2 | 72.5 |
| 7 | 4 | 1 | 5 | 1 | 4 | 2 | 5 | 1 | 4 | 1 | 90.0 |
| 8 | 2 | 3 | 3 | 3 | 2 | 3 | 3 | 3 | 2 | 3 | 55.0 |
| 9 | 3 | 2 | 4 | 1 | 3 | 2 | 4 | 2 | 3 | 2 | 80.0 |
| Average SUS Score | | | | | | | | | | | 71.67 |

In the System Usability Scale (SUS) survey above, the functionality and validation of the system is obtained. Android users demonstrate a higher average of 72.0, while iOS users report a mean SUS score of 71.67. The platform-specific scores, when summed, produce an overall SUS score of 71.83 for the system, which is above average and indicative of a good level of user satisfaction. The minimal difference, almost negligible between the platform-specific scores suggests that the system successfully provides a consistent user experience and system usability across the two platforms.

V. CONCLUSION

In this study, bottlenecks commonly found in generic task management tools are explored and analysed to develop solutions that can overcome them. This proposed system ensures that task management becomes easier and more comfortable to integrate into the daily activities of students as it was developed to address the need for students to organize their tasks more effectively. Among the many benefits of software development is the opportunity to improve and upgrade the software as technology advances. Therefore, recommendations for future readers of this work are as follows;

- a) Enable video and voice call functionalities to improve user collaboration.
- b) Integrate with popular third-party applications and services such as Learning Management Systems
- c) (LMS)
- d) Implement more robust analytics and reporting tools, preferably with machine learning, for personalized suggestions based on user behaviour.
- e) Expand the scope of the proposed software by developing web and desktop versions as it will be beneficial in reaching a wider group of users or simply enabling users to access the platform from any device of their choice.

REFERENCES

- [1] S. Meitarice, B. P. M. Mumtaz, and O. A. Dedek, "Automated Task Management System Using Analytical Hierarchy Process," *Jurnal Teknik Informatika*, vol. 13, no. 2, Oct. 2020.
- [2] M. S. Morales Sanchez, "Design of a tangible user interface for task management at home," M.S. thesis, Dept. of Design, Faculty of Architecture and Design, Norwegian Univ. of Science and Technology, 2019.
- [3] J. Xu, "A profile analysis of online assignment motivation: Combining achievement goal and expectancy-value perspectives," *Computers and Education*, vol. 177, 2022, doi: 10.1016/j.compedu.2021.104367.
- [4] A. Turner, "How Many People Have Smartphones Worldwide (Mar 2024)," BankMyCell, Mar. 13, 2024. [Online]. Available: <https://www.bankmycell.com/blog/how-many-phones-are-in-the-world>
- [5] N. M. S. Laraswati, K. Agustini, and N. Sugihartini, "The Relationship Both Time Management and Smartphone Using for Learning Source on Students Achievement," *International Journal of Instruction*, vol. 10, no. 1, pp. 1-15, 2017.
- [6] Educause Learning Initiative, "7 things you should know about Adaptive Learning," 2017. [Online]. Available: <https://library.educause.edu/~media/files/library/2017/1/eli7140.pdf>
- [7] D. L. Taylor, M. Yeung, and A. Z. Basset, "Personalized and adaptive learning," in *Innovative learning environments in STEM higher education*, J. Ryoo and K. Winkelmann, Eds. SpringerBriefs in Statistics, 2021, pp. 17-33. doi: 10.1007/978-3-030-58948-6_2
- [8] G. Hedao, P. Thoke, R. Tabhane, S. Meshram, S. Kumbhalkar, and M. Barapatre, "Online Task Management System (OTMS)," *IRE Journals*, vol. 2, no. 5, pp. 69-74, 2018.
- [9] C. Guimarães and D. J. P. Lazaro, "Design of a Task-Management APP to Avoid Procrastination," in *DUXU 2018, LNCS 10920*, A. Marcus and W. Wang, Eds. Springer International Publishing AG, 2018, pp. 16–27.
- [10] A. Knegetel, "Honeycomb: Android app of goal and task management for increased productivity," Bachelor's Degree Final Work, Universitat Jaume I, 2020.
- [11] F. A. Nurzi and K. A. Wahab, "Web-Based Student Task Management System," *Journal of UCYP Press*, vol. 1, no. 1, pp. 50-56, 2022. [Online]. Available: <https://doi.org/10.37698/eastj.v1i1.1192>
- [12] M. Sravanth and R. Dhanush, "Intelligent Task Management System," *International Journal of Creative Research Thoughts (IJCRT)*, vol. 11, no. 12, pp. e87-e108, 2023.
- [13] Azeez, T. O. Awe, A. C. and Omosebi, P.A. (2018). Predicting Students' Graduating Cumulative Grade Point Average Using Difference Level, Classification and Regression Tree and Linear Regression Algorithm, *University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR)*, Vol. 2, No. 1, pp. 40 - 47
- [14] A. Kaya, R. Ozturk, and C. Altin Gumussoy, "Usability Measurement of Mobile Applications with System Usability Scale (SUS)," in *Advances in Intelligent Systems and Computing*, 2019, pp. 389–400.
- [15] J. Brooke, "SUS: A 'Quick and Dirty' Usability Scale," in *Usability Evaluation In Industry*, P. W. Jordan, B. Thomas, B. A. Weerdmeester, and A. L. McClelland, Eds. 1996, pp. 189–194