ADAMA SCIENCE AND TECHNOLOGY UNIVERSITY

SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTING

DEPARTMENT OF SOFTWARE ENGINEERING

COURSE NAME: - MOBILE APPLICATION DESIGN AND DEVELOPMENT

PROJECT TITLE: TODO MOBILE APPLICATION

Section: 4

Name	ID.NO
Eyu Ashenafi	UGR/25481/14
Sumeya Adem	UGR /25313/14
Helen Tagay	UGR /25495/14
Abdi Dawud	UGR /25341/14
Yonas Bezawerk	UGR /25547/14
Natnael Chala	UGR /25371/14
Abyalew Lobe	UGR /25903/14

Table of Contents

Introduction

Welcome to the Todo Mobile App Documentation! This comprehensive guide is designed to help you navigate through the features, functionalities, and technical aspects of Todo app, your ultimate task management companion. Whether you're a user looking to maximize productivity, a developer seeking to integrate Todo with other services, or a technical support specialist, this documentation aims to provide you with all the information you need.

The purpose of this documentation is to serve as a one-stop resource for everything related to Todo. Here's what you can expect to find and how to use this documentation:

User Guides: Step-by-step instructions to help you get started and make the most out of Todo's features. We suggest you start from there

Technical Specifications: Detailed information about the app's architecture, system requirements, and performance optimizations.

Release Notes: About new features, bug fixes, and performance improvements to keep you informed about the latest changes.

Key Features of Todo

Task Addition:

- Add tasks with a title and description.
- Option to set a reminder time in 24-hour format using a virtual clock or keyboard input.
- Ensure a title is provided before allowing task addition.

Task Editing:

- Modify existing tasks using an edit button.
- Update the title, description, and reminder time.

Task Deletion:

- Remove tasks with a confirmation alert.

Dark/Light Mode:

- Toggle between dark mode and light mode for better user experience...

Reminders and Notifications:

- Set timely reminders to ensure you never miss a deadline.

Cross-Platform Availability:

- Todo is available on both iOS and Android, ensuring a consistent experience across different devices.

1. Project Overview

Todo is a powerful yet simple task management application tailored to help you organize your daily tasks, set reminders, and boost your productivity ultimately enhancing productivity and time management. The aim of Todo is to provide a user-friendly mobile application for efficient task management.

Todo is a mobile application designed to help users manage their daily tasks efficiently. The app provides basic CRUD (Create, Read, Update, Delete) operations for tasks, allows users to mark tasks as done, and offers notifications to remind users when it is time to begin a task based on its scheduled time.

Problem Domain

Many individuals struggle with task management and time management in their daily lives. Todo aims to address this issue by providing a simple yet effective tool for organizing tasks.

Current Market Issues and Problems

Fragmented Task Management: Many users rely on multiple tools and platforms to manage their tasks, leading to inefficiency and disorganization.

Lack of Timely Reminders: Existing apps often fail to provide timely and reliable reminders, causing users to miss crucial tasks.

Complex User Interfaces: Many task management apps are overloaded with features, making them difficult to use for those seeking simplicity.

Potential of Todo to Solve These Problems

Unified Task Management: By offering comprehensive task management functionalities within a single app, Todo reduces the need for multiple tools.

Simple and Intuitive Design: Focusing on a minimalist design, Todo aims to provide a user-friendly interface that is easy to navigate.

Feature Scope:

Create Tasks: Users can create new tasks with titles, descriptions, and due dates.

Edit Tasks: Users can modify existing tasks to update details or change due dates.

Delete Tasks: Users can remove tasks that are no longer needed.

Complete Tasks: Users can mark tasks as complete and optionally archive them.

Set Reminders: Users can set reminders for tasks to receive notifications at specified times.fi

Target Audience:

With its intuitive design and robust functionality, Todo caters to a wide range of users, the target audience includes students, professionals, and anyone seeking to organize their tasks and improve productivity By understanding their specific needs and preferences, Todo can be

designed and marketed effectively to cater to these diverse groups, ensuring a broad and satisfied user base.'

Here are the primary target user groups for Todo:

1. Students

Characteristics:

- Balancing academic responsibilities, extracurricular activities, and personal commitments.
- Tech-savvy and familiar with mobile apps.

Needs:

- Organize assignments, projects, and study schedules.
- Set reminders for deadlines and exam dates.

2. Professionals

Characteristics:

- Managing work responsibilities, career development, and possibly further education.
- Often juggling multiple projects and tight deadlines.

Needs:

- Track work tasks, meetings, and project deadlines.
- Set reminders for important work-related activities.
- Organize personal tasks alongside professional commitments.

3. Busy Parents

Characteristics:

- Managing household tasks, children's schedules, and possibly work responsibilities.

- Looking for ways to set daily routines and reduce stress.

Needs:

- Keep track of family activities, appointments, and tasks.
- Set reminders for school events, doctor's appointments, and family outings.
- Simplify the management of household chores and personal errands.

5. Productivity Enthusiasts

Characteristics:

- Passionate about personal development and productivity hacks.
- Regularly seeking tools to optimize their daily routines and habits.

Needs:

- Track personal goals, habits, and routines.
- Set reminders for daily, weekly, and monthly tasks.
- Organize various aspects of their lives to maximize efficiency and productivity.

2. User Requirements

The requirement used for the development of the app:

Task Management

- Create, edit, and delete tasks
- Assign tasks to user
- Set due title , description, and time for tasks
- Marking Tasks as Done: Users should be able to mark tasks as completed.
- View tasks in list
- Receive notifications on task updates and assignments

Scheduling & Calendars

- Set recurring tasks and reminders
- Visualize task deadlines and dependencies
- Mobile & Offline Access

Influence on Design and Functionality:

These requirements have led to a minimalist design, focusing on ease of use and quick access to essential features. The notification feature requires background services to ensure timely alerts.

3. Design Concepts

The Todo app was designed with the goal of providing users with a simple, yet powerful tool to manage their daily tasks and increase productivity. The core design concept behind the app are:

UI Design:

•Minimalist Interface: Clean and simple layout to avoid clutter.

•Intuitive Navigation: Easy-to-use for accessing different app sections so that user can use the application with no trouble.

•Consistent Visual Elements: we used consistent icons, colors, and typography for a cohesive look where we used the color blue mainly for it emits wisdom and stability that comes with time management

UX Considerations:

•User-Friendly: we ensure the app is easy to navigate and use, even for those who are not technical.

•Responsive Design: we tried to ensure the app works well on various screen sizes and orientations.

Navigation Flow:

Home Screen:

- Displays the list of tasks in a queue format.
- Contains an addition sign (+) button at the bottom for adding new tasks.

Add Task Screen:

- Appears when the addition sign button is clicked.
- Fields for title and description.
- Notify button to set reminder time.
- Options to set time via virtual clock or keyboard.
- Add Todo button (enabled only when the title is filled).

Edit Task Screen:

- Accessed by clicking the pencil button on any task.
- Similar to the Add Task Screen but with an Update Todo button.

Delete Task Confirmation:

- Triggered by clicking the bin button on any task.
- Shows an alert box asking for confirmation.

Settings Bar:

- Accessible by swiping from the left.
- Contains a switch to toggle between dark mode and light mode.

Application Flow

Start:

- Launch app, display Home Screen with task list.

Add Task:

- User clicks + button.
- Add Task Screen appears.
- User fills in title (mandatory) and description (optional).
- User sets reminder time using virtual clock or keyboard.
- User clicks Add Todo, task is added to the list in Home Screen.

Edit Task:

- User clicks pencil button on a task.
- Edit Task Screen appears.
- User updates title, description, or reminder time.
- User clicks Update Todo, updates are saved.

Delete Task:

- User clicks bin button on a task.
- Confirmation alert appears.
- User confirms deletion, task is removed from the list.

Toggle Dark/Light Mode:

- User swipes from the left to open settings bar.
- User toggles switch to change between dark and light mode.

End:

- User exits the app, changes and tasks are saved

.4. Development Approach

For the development of the Todo mobile application, we adopted the agile methodology. Agile is well-suited for projects that require flexibility, iterative progress ,continuous feedback and

improvements., and close collaboration among team members. Here's why we chose Agile and how it benefited our project:

Justification for Choosing Agile

Iterative Development: Agile allows for iterative development, which means we can build the app incrementally and improve it in successive cycles (sprints).This approach enabled us to deliver functional parts of the app early and gather user feedback continuously, ensuring that we stayed aligned with user needs and expectations.

Flexibility and Adaptability:Agile's flexibility is crucial for adjusting to changing requirements or unforeseen challenges. Given the fast-paced nature of mobile app development, Agile allowed us to adapt our plans quickly in response to new insights, technical hurdles, or market changes.

Continuous Feedback: Agile encourages regular feedback from stakeholders and users. By incorporating feedback in each iteration, we were able to refine features and improve the user experience continuously, leading to a more polished final product.

Enhanced Collaboration: Agile promotes close collaboration between cross-functional teams. Regular stand-ups, sprint reviews, and retrospectives helped ensure that all team members were on the same page, fostering a collaborative and communicative environment.

Incremental Releases: Agile supports incremental and iterative releases. This allowed us to deploy new features and improvements incrementally, reducing the risk of large-scale failures and enabling quicker time-to-market for essential functionalities.

Challenges Faced and How We Addressed Them

Notification Integration: Managing background services and ensuring timely notifications was challenging. We addressed this by using reliable scheduling libraries and testing across different devices.

Maintaining Consistent Communication: Ensuring effective communication among a distributed team. We utilized collaboration tools like Github for real-time communication,. Regular text messaging and meetings helped keep everyone synchronized.

Ensuring Quality: Maintaining high quality while iterating quickly. We implemented a robust testing strategy that included unit tests, integration tests, and end-to-end tests. Regular code reviews also helped maintain code quality.

5. Technological Stack

The Todo app has been built using a modern, robust, and scalable technological stack to ensure a high-performance, secure, and maintainable application. The key components of the technology stack are as follows:

1.React Native: While the core logic is written in vanilla JavaScript, React Native is employed for building the mobile application. This allows for a consistent experience across both iOS and Android platforms with a single codebase.

2.Expo CLI: Used to streamline the development process, Expo CLI provides a set of tools and services that simplify the creation and deployment of React Native applications.

4.Vanilla JavaScript: Todo is built using vanilla JavaScript, ensuring a lightweight and flexible codebase without the overhead of additional frameworks. This approach provides a clean, efficient, and easily maintainable front end.

Dependencies

a) **Expo Notifications:** To handle reminders and notifications, Expo Notifications is integrated, providing a robust solution for alerting users about their tasks.
b) **React Native Time Picker:** For setting reminder times, the app uses React Native Time Picker, offering a user-friendly interface for selecting times.
c) **Expo Permissions:** Ensures that the app requests and manages permissions efficiently, enhancing security and user trust.
d) **React Native Async-Storage:** For data persistence, Todo uses React Native's Async-Storage. This local storage solution is ideal for saving user data such as tasks and settings, ensuring that information is retained between app sessions without requiring a network connection.

6.Implementation Details

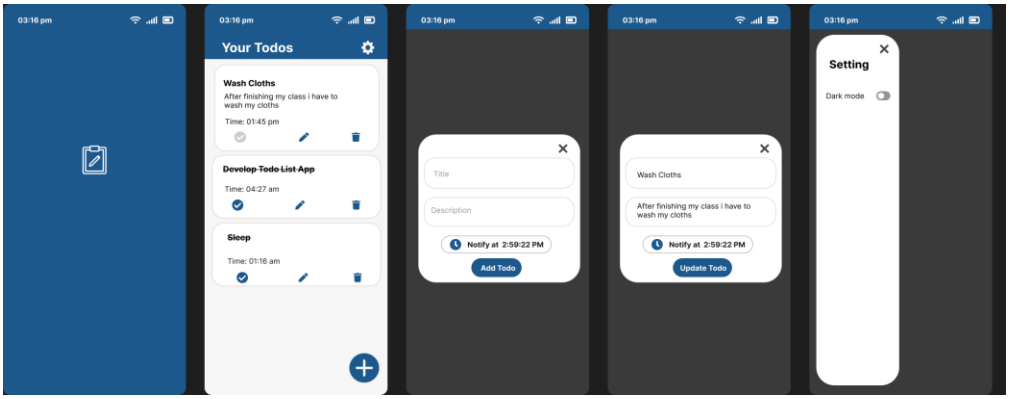
Key Features and Functionalities:

•Task CRUD Operations: Users can add new tasks, view task details, edit existing tasks, and delete tasks.

•Marking Tasks as Done: Users can mark tasks as completed, which updates their status in the database.

•Notifications: Users receive notifications based on the scheduled time for their tasks.

Screenshots/Mockups:



Functionality of the App

Create Tasks: Users can add new tasks with details such as title, description, and due date.

Read Tasks: View a list of tasks with essential details displayed.

Update Tasks: Edit task information, including status updates.

Delete Tasks: Remove tasks from the list.

Mark Tasks as Done: Mark tasks as completed to keep track of progress.

Receive Notifications: Get notified when it’s time to start a task based on its scheduled time.

Essential Features of the App

Task Management: Comprehensive CRUD operations for tasks.

Task Status Updates: Ability to mark tasks as done.

Timely Notifications: Push notifications for task reminders.

User-Friendly Interface: Simple and intuitive UI design.

7. Testing and Quality Assurance

Ensuring the functionality, performance, and reliability of the Todo mobile application is crucial for delivering a seamless and efficient user experience. Here, we outline the testing strategies employed and reflect on their effectiveness.

1. **Unit Testing**: Ensured individual components work as expected. We wrote unit tests for all critical functions and methods using testing frameworks

Effectiveness: It has proven effective in catching bugs early in the development process. By isolating each function, we can ensure that the core logic of each functionality works as intended. However, unit tests alone cannot guarantee the integration of these units in the larger system.

2**. Integration Testing :** It verified that different components of the app work together seamlessly. We developed integration tests to validate the interaction between different components, such as the frontend and backend services. We Used tools like React Native Testing Library to simulate user interactions

Effectiveness: Integration testing has helped identify issues that arise from the interaction between different components. It ensures that the app's modules work together seamlessly, although it can be challenging to simulate all possible interactions.

3. **End-to-End (E2E) Testing**: To simulate real-world scenarios and validate the app's functionality from the user's perspective. We created test scenarios that mimic typical user

workflows, such as task creation, editing, and deletion. We also executed E2E tests on both iOS and Android devices to ensure cross-platform consistency.

Effectiveness:E2E testing has been highly effective in ensuring that the app functions as expected in real-world scenarios. It helps catch issues that might not be evident in unit or integration tests. However, E2E tests can be time-consuming and require more resources to maintain.

4. Performance Testing: To evaluate the app's responsiveness, stability, and resource usage under various conditions. For this we tested the app under different network conditions and on various devices. It measure key performance metrics such as load times, memory usage, and CPU usage.

Effectiveness: Performance testing has been instrumental in identifying bottlenecks and optimizing the app for better performance. By monitoring performance metrics, we can ensure a smooth user experience across different devices and conditions.

Reflection on Testing Effectiveness

The testing strategies employed for Todo have significantly contributed to its functionality, performance, and reliability. The combination of unit, integration, E2E, and performance testing has provided a robust framework for quality assurance.

Strengths:

Comprehensive coverage of different testing aspects.

Early detection of bugs and performance issues.

Continuous improvement through user feedback and performance monitoring.

Challenges:

Maintaining and updating test suites as the app evolves.

Balancing the depth of testing with resource constraints.

Ensuring that automated tests accurately reflect real-world usage.

Overall, the testing approach for Todo has been effective in delivering a high-quality mobile application. Continuous evaluation and refinement of our testing strategies will further enhance the app's reliability and user satisfaction.

8.Restrictions and limitation of the app

While Todo aims to provide a seamless task management experience, there are several potential restrictions and limitations that users and developers should be aware of. These restrictions can stem from technical, operational, and strategic considerations.

8.1 Technical Restrictions

A. Platform Limitations:

• Cross-Platform Compatibility: Although react native is used for cross-platform development, there may be subtle differences in how the app performs on iOS vs. Android devices especially in the appearance.

• Device-Specific Issues: : Differences in device hardware (e.g., screen sizes, processing power) can affect the app's performance and user experience. Certain functionalities, especially those related to notifications and background services, may behave differently across various devices and operating system versions.

B. Notification Limitations:

• Battery Optimization: On some devices, aggressive battery optimization settings may prevent the app from delivering timely notifications.

• Background Execution: Restrictions on background execution by certain operating systems might affect the app's ability to trigger notifications accurately.

C. Performance Constraints:

• Large Task Lists: Handling a large number of tasks might lead to performance issues, such as slow loading times and lagging interfaces.

8.2 Operational Restrictions

Notification Permissions: Notifications and reminders depend on user-granted permissions. If permissions are not granted, the app will not be able to send reminders.

Time Format: The app exclusively uses a 24-hour format for setting times to avoid confusion. Users accustomed to a 12-hour format (AM/PM) may need to adjust.

Platform Compatibility: While React Native ensures compatibility across iOS and Android, certain features may behave differently on different devices due to variations in operating systems and hardware.

User Input Validation: Users must enter a title for tasks; without it, the task cannot be added. This ensures that all tasks have a minimum level of detail but may inconvenience users who prefer minimal input.

8.3 Strategic Restrictions

A. Feature Scope:

• Recurring Tasks: Implementing recurring tasks requires careful consideration of user interface design and backend logic to ensure it does not overwhelm users with notifications.

Market Competition:

Differentiation: Standing out in a crowded market of task management apps requires continuous innovation and user-centric enhancements.

9. Future Enhancements Potential Enhancements:-

-Recurring Tasks: Allow users to set recurring tasks (daily, weekly, etc.).

- Task Prioritization: Enable users to prioritize tasks and sort them accordingly.

- Collaboration Features: Allow multiple users to share tasks and collaborate.

-Accessibility: Ensuring the app is fully accessible to users with disabilities (e.g., visually impaired users) requires additional design and development efforts, which might not be fully realized in initial versions.

Some of Future Integrations:

Communication/collaboration tools (e.g. Slack, Microsoft Teams, Zoom)

Project management tools (e.g. Trello, Asana, Jira)

CRM/sales tools (e.g. Salesforce, HubSpot, Zoho)

Accounting/invoicing software (e.g. QuickBooks, FreshBooks, Xero)

Alignment with Project Objectives and User Needs

These enhancements aim to improve user engagement and productivity, aligning with the core objective of helping users manage their tasks effectively.

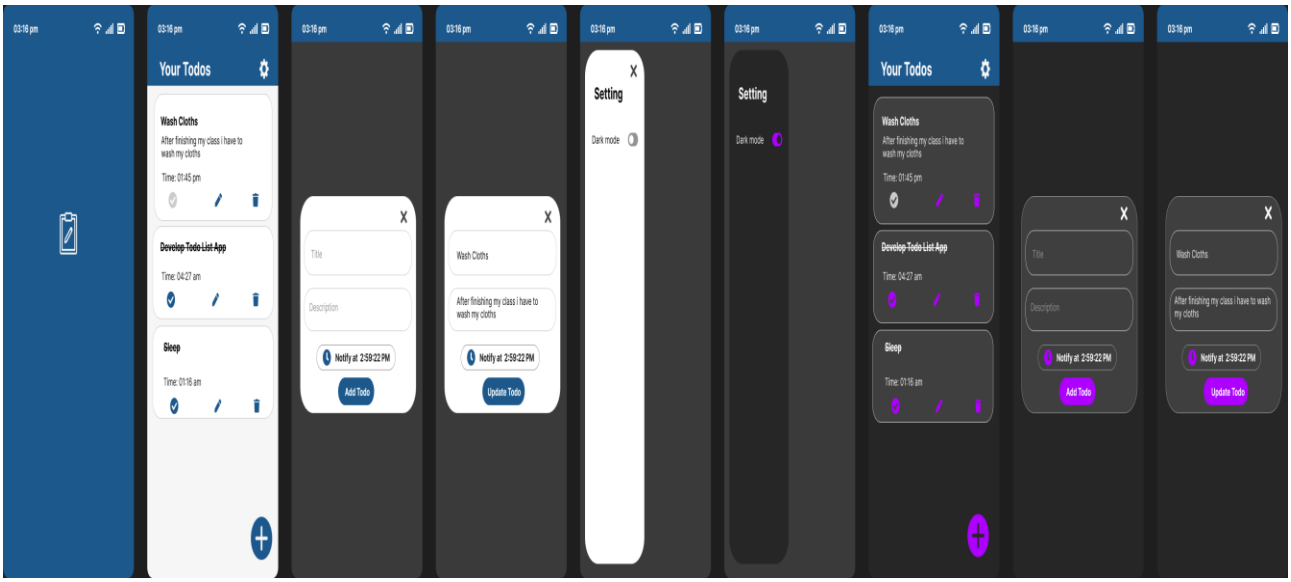
Essential New Aspects for Project Enhancement

Recurring Tasks: Implement functionality for setting recurring tasks (daily, weekly, etc.).

Task Prioritization: Introduce a system to prioritize tasks and sort them by importance.

Collaboration Features: Develop features that allow multiple users to share and collaborate on tasks.

wireframe for the app



Conclusion

Thank you for choosing Todo as your task management solution. Let’s get started on your journey to increased productivity and organization!

We are excited to have you on board with Todo and are committed to providing you with the best possible experience. This documentation is designed to be a valuable resource as you explore and utilize Todo’s features. Should you have any questions or need further assistance, please do not hesitate to reach out to our support team..

The technological stack for Todo combines the simplicity and performance of vanilla JavaScript with the cross-platform capabilities of React Native. This setup, complemented by React Native and Async-Storage for local data persistence, provides a robust foundation for a reliable and user-friendly task management application