



TODO APP

Task - 1



Money Transfer Application

- A to-do app is a software application designed to help users manage tasks, assignments, and deadlines. It allows users to create, organize, and prioritize tasks and track their progress. With the help of a to-do app, users can increase their productivity, stay organized, and keep track of their daily activities. These apps can be used for personal or professional purposes, and they are available on various platforms, including desktop and mobile devices.

LMS Username	Name	Batch
2113a495	Adelin Grace Tanya S	A49



Table of Contents

Introduction

- To-do app
- Software application
- Manage tasks
- Mobile devices.

Overall Description

Task management, productivity, organization, time management, project management, deadline tracking, reminders, notes, prioritization, collaboration, mobile app, web app, cross-platform, intuitive interface, customizable, user-friendly.

Diagrammatic Representations

1. Use case diagram
2. Class diagram
3. Data flow diagram
4. Sequence diagram

Step-Wise Description

Introduction:

A to-do app is a software application designed to help users manage tasks, assignments, and deadlines. It allows users to create, organize, and prioritize tasks and track their progress. With the help of a to-do app, users can increase their productivity, stay organized, and keep track of their daily activities. These apps can be used for personal or professional purposes, and they are available on various platforms, including desktop and mobile devices.

Purpose:

The purpose of a to-do app is to help users manage their tasks, prioritize their work, and increase their productivity by providing a centralized location to organize and track their responsibilities. By using a to-do app, users can improve their time management, reduce stress, and stay focused on their goals.

Scope:

The scope of a to-do app typically includes features such as creating and organizing tasks, setting due dates and priorities, receiving reminders and notifications, and tracking progress. The app may also include features such as collaboration, syncing across devices, and integration with other apps or platforms.

Glossary:

React -> JavaScript library

Json -> JavaScript Object Notation

DFD -> Data Flow Diagram

FD -> Context Flow Diagram

ER -> Entity Relationship

IDE -> Integrated Development Environment

SRS -> Software Requirement Specification

Product Perspective:

A to-do app is a software application that provides users with a centralized platform to manage and organize their tasks and activities. It aims to increase productivity, efficiency, and time management skills by allowing users to prioritize and track their daily, weekly, and long-term goals. Additionally, the app may offer features such as collaboration, reminders, and progress tracking to enhance user experience and satisfaction.

Hardware Requirements:

A modern processor (such as an Intel Core i5 or i7) or a modern mobile processor (such as a Qualcomm Snapdragon or Samsung Exynos).

Sufficient RAM and storage to ensure smooth performance and efficient data management. At a minimum, the device should have 2 GB of RAM and 16 GB of storage.

Software Requirements:

The app must be able to store and manage task data, including task descriptions, due dates, and priority levels.

The app must be able to send reminders or notifications to the user regarding upcoming or overdue tasks.

The app must have a user-friendly interface that allows users to easily add, edit, and delete tasks.

The app must be able to sync data across different devices and platforms to ensure that users have access to their to-do lists no matter where they are.

Functional Requirements

1.Task Management

Users should be able to create, edit, and delete tasks.

Each task should have a title, description, due date, and priority level.

Users should be able to assign tasks to different categories or projects, and they should be able to move tasks between categories as needed.

The system should allow for collaboration and delegation of tasks, so users can work together on projects and share progress updates.

2.Reminders and Notifications

The system should provide users with the ability to set reminders and alerts for tasks to ensure timely completion.

Notifications should be customizable, allowing users to choose how they receive them (e.g. email, push notification, etc.).

Users should be able to snooze or dismiss reminders as needed.

3.Notes and Attachments

Users should be able to add notes and attachments to tasks for additional context or reference.

The system should allow for different file types to be attached, such as documents, images, or links.

Notes and attachments should be searchable, making it easy for users to find what they need.

4.User Interface and User Experience

The app should have an intuitive and user-friendly interface, making it easy for users to navigate and use.

The app should be customizable, allowing users to personalize their experience (e.g. change the color scheme, font, etc.).

The app should be accessible, ensuring that it is usable for users with disabilities

5.Cross-Platform Compatibility

The app should be available on multiple platforms, such as desktop, mobile devices, and web browsers.

The app should synchronize seamlessly across platforms, so users can access their tasks and data from any device.

The app should be compatible with different operating systems, such as iOS, Android, Windows, and macOS.

Non-Functional Requirements

1. Performance

The app should have fast and responsive performance, allowing users to quickly add, edit, and complete tasks.

The app should load quickly, minimizing wait times and increasing user satisfaction.

The app should be able to handle large amounts of data without slowing down.

2. Security

The app should have robust security measures to protect user data and privacy.

Users' login credentials should be securely stored and transmitted.

The app should use encryption to protect user data both in transit and at rest.

3. Scalability

The app should be designed to handle an increasing number of users and data.

The app should be able to handle a large number of concurrent users without slowing down or crashing.

The app's infrastructure should be scalable, allowing for easy upgrades and expansions.

4. Compatibility

The app should be compatible with different devices and browsers, ensuring that it works seamlessly on different platforms.

The app should be compatible with different screen sizes, allowing users to use the app on different devices, from mobile phones to large desktop monitors.

The app should be compatible with different versions of operating systems, ensuring that it works on older as well as newer devices.

5. Accessibility

The app should be accessible to users with disabilities, including those with visual, auditory, or motor impairments.

The app should be designed with accessibility in mind, using colors, fonts, and other design elements that are easy to read and distinguish.

The app should provide alternative ways of interacting with the interface, such as keyboard shortcuts or voice commands.

Constraints

1. Compatibility constraints:

The app must be compatible with a specific minimum version of an operating system or web browser.

2. Resource constraints:

The app must have a maximum file size limit for notes and attachments.

The app must be designed to minimize resource usage (e.g. memory, CPU) to ensure optimal performance.

3. Legal constraints:

The app must comply with data protection and privacy laws in the jurisdictions where it is used.

The app must not allow users to create or share content that violates laws or infringes on the rights of others.

4. Time constraints:

The app must be designed to perform tasks within a certain time limit, such as loading tasks or sending reminders.

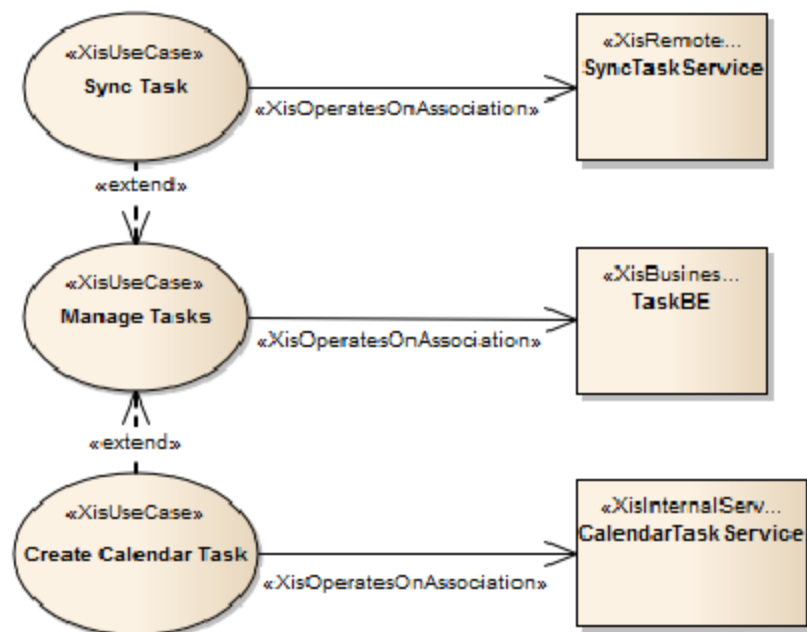
The app must be designed to perform tasks even in situations of slow internet connectivity.

5. User interface constraints:

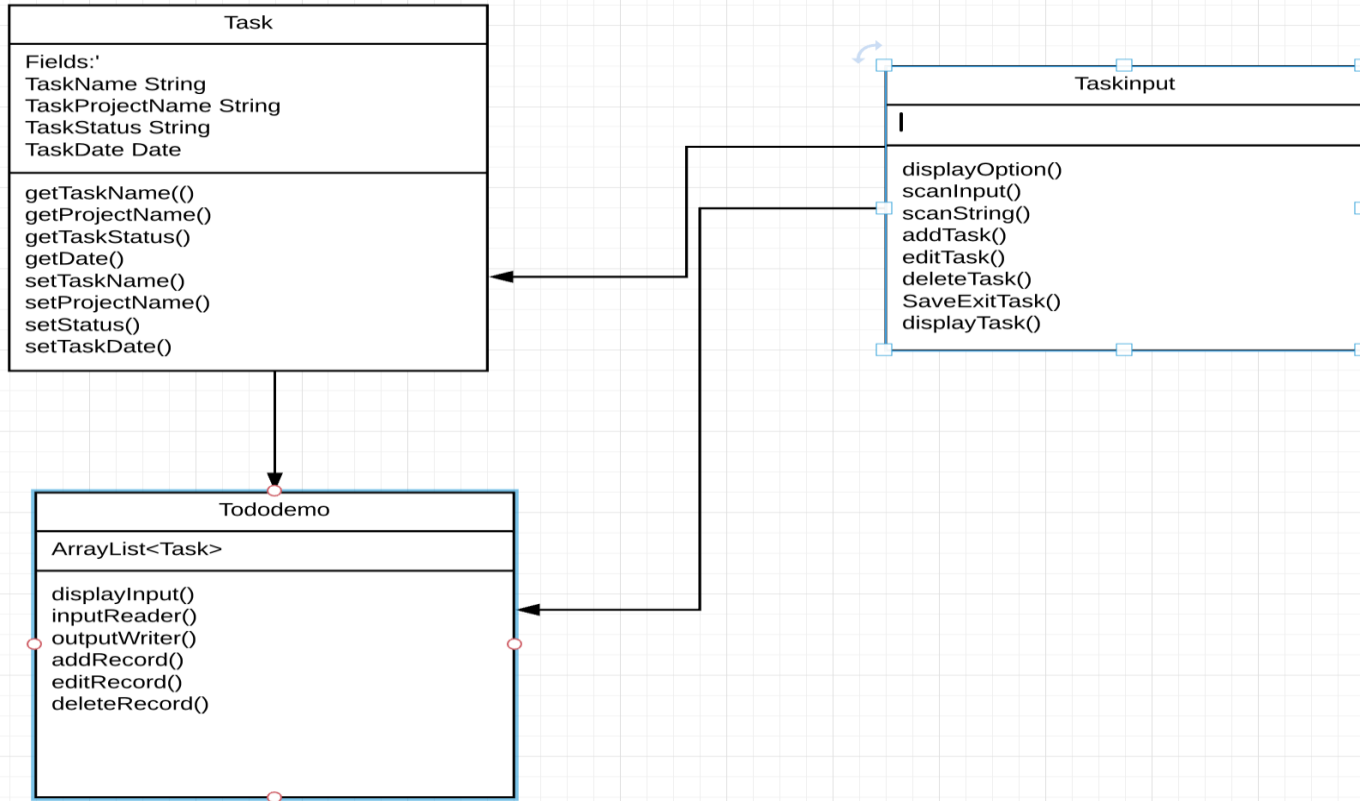
The app must have a user interface that is intuitive, easy to navigate and understand, and aesthetically pleasing.

The app must have controls and feedback mechanisms that are clearly visible and easy to use.

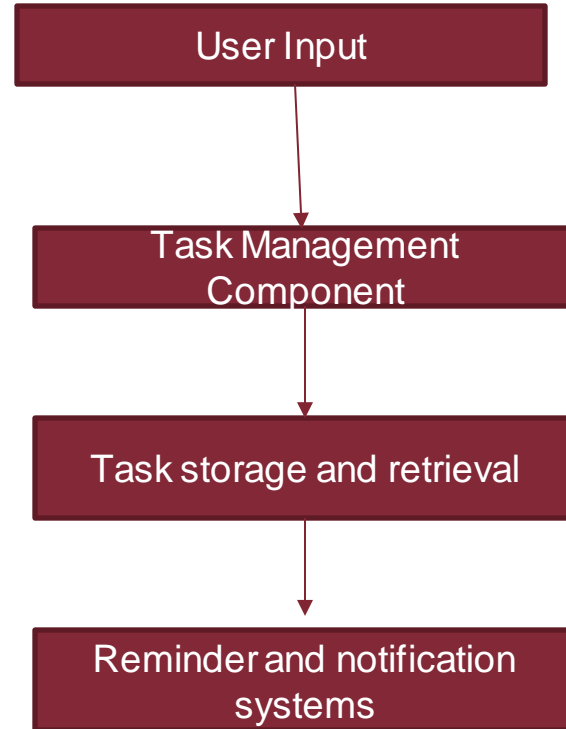
Use case diagram



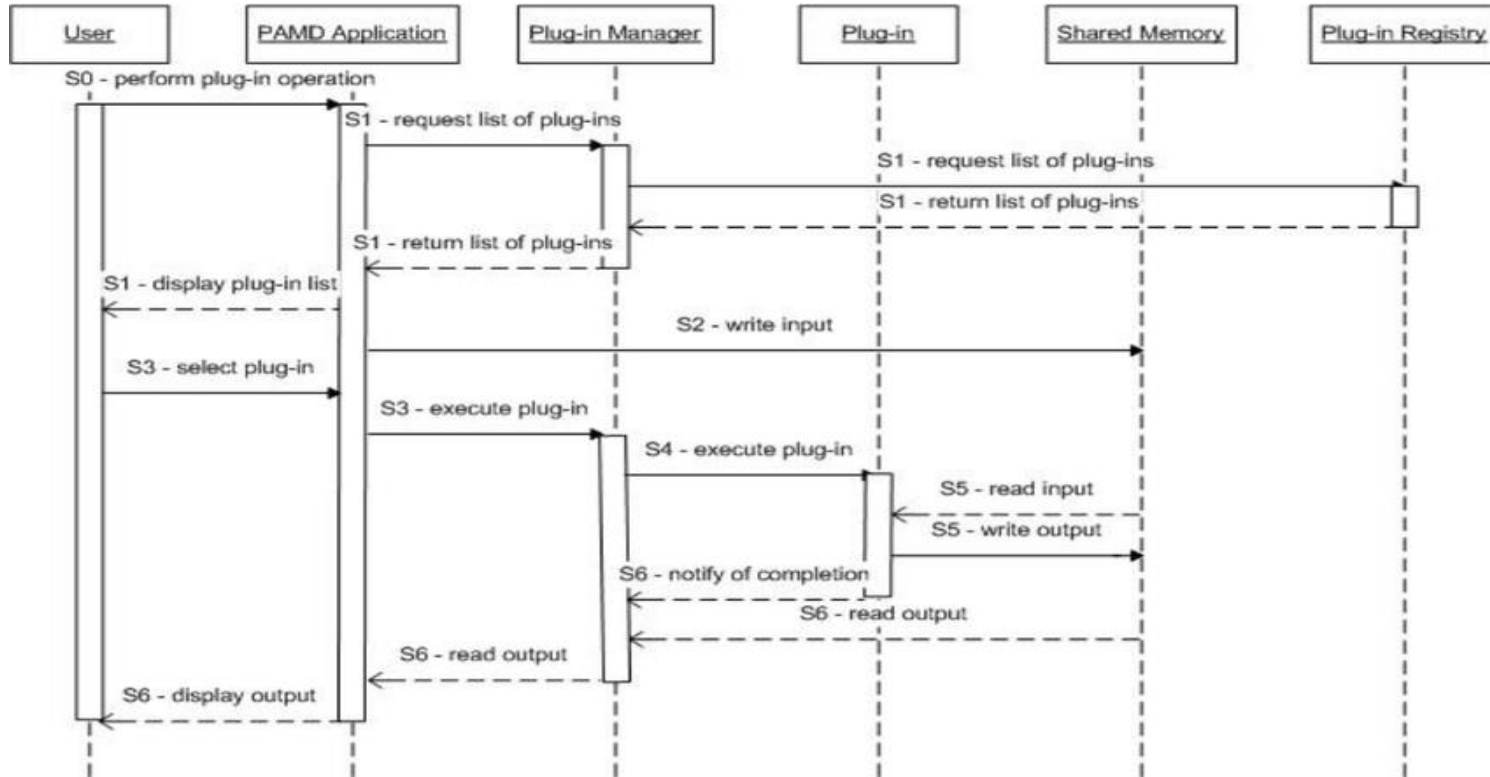
Class Diagram



Data Flow Diagram



Sequence Diagram



Submission Github



[https://github.com/Adelingrace/
NM](https://github.com/Adelingrace/NM)

Thank you!

