

Description

RemindMeNow is a user-friendly mobile app designed to keep you on top of your daily tasks, appointments, and important dates. This intuitive application is your go-to solution for staying organized and never missing an important moment. Whether you're a busy professional, a student juggling assignments, or someone who simply values efficiency, RemindMeNow is your trusted companion.

Domain

Entity: Reminder

Title: A brief and recognizable name or label for the reminder. This helps users quickly identify what the reminder is for.

Description: Additional notes or information related to the reminder. Users can include any details or instructions they find relevant.

Category: A categorization label for organizing reminders. This field allows users to sort their reminders into different groups, like "Work," "Personal," or "School."

Due Date: The date when the reminder should be triggered. It specifies the day on which the reminder is relevant.

Due Time: The specific time of day when the reminder should appear. This ensures that users are alerted at the exact moment they need to remember something.

Priority: An indication of how important or urgent the reminder is. Users can assign priority levels such as "High," "Medium," or "Low" to help them prioritize their tasks effectively.

CRUD

Create (C)

Operation: Create a new reminder.

Details: Users can add a new reminder by providing a title, description, selecting a category, specifying the due date, due time, and assigning a priority level. Once all the required information is entered, they can save the reminder.

Read (R)

Operation: View existing reminders.

Details: Users can access their list of reminders, displaying all saved reminders with their details. They can sort and filter reminders to quickly find the one they're looking for. Clicking on a reminder will allow users to view its full details.

Update (U)

Operation: Modify an existing reminder.

Details: Users can select a reminder they wish to update. They can then edit the title, description, category, due date, due time, and priority as needed. After making changes, they save the updated information, which overwrites the previous data for that reminder.

Delete (D)

Operation: Remove a reminder from the list.

Details: Users can select a reminder they want to delete. They will have the option to swipe left on a reminder in order to delete it. Confirming this action will permanently remove the reminder from their list, and it will no longer be available.

Storing data

Dual Persistence for Vital Data: Data persistence is crucial for a seamless user experience. Therefore, "RemindMeNow" has been designed to be connected to both a local database and a server, allowing users to access their reminders both online and offline. This ensures that your reminders are always available when you need them.

Bi-Directional Data Changes: When users create, update, or delete reminders, these changes are stored both on the server and in the local database. This means that the information is always synchronized. Even if you make changes while offline, they will be pushed to the server once the internet connection is re-established.

Online Updates: When users are online, the app fetches reminder data directly from the server. This ensures that any updates or modifications to reminders are reflected immediately, providing a real-time experience.

Offline Accessibility: When users are offline, "RemindMeNow" still provides access to their reminders. The app retrieves temporary data from the local database, allowing users to view and modify their reminders. Any changes made while offline are saved locally and then synchronized with the server when an internet connection becomes available.

Offline mode

Creation of Reminders (C):

When offline, any newly created reminders are saved locally in the device's database.

These reminders are accessible for viewing and editing while offline. However, they are not immediately synchronized with the server.

Reading Reminders (R):

When offline, the app retrieves and displays reminders from the local database.

Users can view and work with their existing reminders, and any changes are reflected based on the locally saved data.

Updating Reminders (U):

While offline, when a user edits an existing reminder, the app updates the reminder's information in the local database.

These changes are stored locally, allowing users to see the modified data when there's no internet connection. However, they are not yet transmitted to the server.

Deletion of Reminders (D):

When the user decides to delete a reminder while offline, the app removes the reminder from the local database.

The deleted reminder is no longer visible to the user while offline. However, this deletion is pending synchronization with the server.

When the application goes back online, all the changes on the local database are transmitted to the server.