Bucket List Mobile App

# Project Overview

The **Bucket List Mobile App** is designed to help users create, manage, and accomplish their personal goals and experiences. Whether it's traveling to a dream destination, learning a new skill, or achieving a personal milestone, the app provides an intuitive platform to organize and track progress.

This mobile application will be available on both **iOS and Android** platforms, ensuring a seamless experience across devices. The backend will be powered by **AWS**, providing robust cloud-based functionalities, including authentication, data storage, and notifications.

# Key Functionalities

The system provides the following main functionalities:

1. User Authentication
   * Users can register and log in using email and password authentication.
   * Authentication is securely handled via **Clerk**, ensuring user data protection and secure sessions.
2. Bucket List Management
   * Users can create, edit, and delete tasks on their bucket list.
   * Tasks are categorized (e.g., Adventure, Personal Growth, Travel).
   * Tasks can be marked as “Completed” or “In Progress.”
3. Task Generation System
   * The app suggests tasks if users need inspiration.
   * Task suggestions are based on user preferences or a random selection.
4. Progress Tracking
   * Users can track completed and pending tasks.
   * Visual indicators (e.g., progress bars, checkmarks) help users see their progress.
5. Push Notifications
   * Users receive reminders to work on their bucket list.
   * Notifications can be scheduled based on user preferences.

# System Workflow

1. **User Registration/Login →** Authentication via Clerk.
2. **User Creates a Bucket List →** Adds, edits, or deletes tasks in their list.
3. **Task Generation System →** Suggests new tasks based on user interests or a random algorithm.
4. **Task Progress Tracking →** Updates task status and syncs with the backend.
5. **Notifications →** Sends reminders for pending tasks.

# UML Diagrams

## Sequence Diagram

Bucket app sequence diagram

Mermaid code: 
sequenceDiagram
  actor User
  participant Clerk
  participant System
  participant Database

  User->>Clerk: Register/Login
  Clerk->>System: Authenticate User
  System->>Database: Store/Retrieve User Data
  Database-->>System: User Data
  System-->>Clerk: Authentication Result
  Clerk-->>User: Authentication Success

  User->>System: Create Task
  System->>Database: Store Task Data
  Database-->>System: Task Saved
  System-->>User: Task Created Confirmation

  User->>System: Generate Task
  System->>Database: Retrieve User Tasks
  Database-->>System: Task List
  System-->>User: Suggest Task

  User->>System: Update Task Status
  System->>Database: Update Task Progress
  Database-->>System: Status Updated
  System-->>User: Progress Saved

  System->>User: Send Notifications (Reminder)


## Class Diagram (Simplified)

Bucket app class diagram

Mermaid code:
classDiagram
  class User {
    -String userID
    -String email
    -String password
    -List~Task~ bucketList
  }
  
  class Task {
    -String taskID
    -String userID
    -String description
    -String category
    -String status
  }
  
  class Notification {
    -String notifID
    -String userID
    -String taskID
    -Date timestamp
  }

  User "1" -- "0..*" Task
  User "1" -- "0..*" Notification
  Task "1" -- "1" Notification



## Activity Diagram (Task Creation Process)

Bucket app activity diagram

Mermaid code:
stateDiagram
  direction LR
  [*] --> Login
  Login --> NavigateTaskPage
  NavigateTaskPage --> AddNewTask
  AddNewTask --> SaveTask
  SaveTask --> TaskListUpdated
  TaskListUpdated --> [*]

# User Stories

The system provides the following main functionalities:

## **User Authentication**

As a user, I want to securely create an account with my email so that I can access the app and save my progress.

* **How It Works:**

Users can sign up by providing their email and password. The authentication process is managed securely via **Clerk**, which handles user data protection and secure sessions. After registering, users receive a confirmation email to verify their account. Once verified, they can log in anytime to access their bucket list and progress.

* **Acceptance Criteria:**
  + Users must provide a **valid email** and a **strong password** to register.
  + Users should receive a confirmation email and successfully verify it to activate their account.
  + Upon successful login, users should be redirected to their personalized dashboard.
  + Unverified users should not be able to log in until they confirm their email.

## **Bucket List Management**

As a user, I want to create, edit, or delete tasks on my bucket list so that I can manage my goals efficiently.

* **How It Works:**

After logging in, users can add new tasks by filling out a simple form with a title, description, and category. They can also edit tasks if their plans change or delete tasks they no longer want to pursue. Each task can be marked as “Completed” or “In Progress” based on the user’s progress.

* **Acceptance Criteria:**
  + Users must be able to add new tasks with a title, description, and category.
  + Users should be able to edit or delete tasks without any errors.
  + Task status should update accurately when marked as “Completed” or “In Progress.”
  + Tasks should be saved to the backend.

## **Task Generation System**

As a user, I want the app to suggest tasks when I run out of ideas so that I can keep adding exciting goals to my bucket list.

* **How It Works:**

When users need inspiration, they can tap a "Suggest Task" button. The app will either randomly suggest a task from a pre-defined list or recommend tasks based on the user’s past interests and categories. This feature ensures users always have fresh ideas to add to their bucket list.

* **Acceptance Criteria:**
  + Users must be able to click a button to get a task suggestion.
  + Suggested tasks should either be random or based on user preferences.
  + Suggested tasks should be easily addable to the user’s bucket list with one tap.
  + **No duplicate** tasks should be suggested if they already exist in the list.

## **Progress Tracking**

As a user, I want to track which tasks I’ve completed and what’s pending so that I can see my progress easily.

* **How It Works:**

Users can view all their tasks in a list format with indicators like progress bars or checkmarks. When a task is marked as "Completed," it visually changes to show progress. A progress summary may also show how many tasks are completed versus pending.

* **Acceptance Criteria:**
  + Users must be able to mark tasks as “Completed” or “In Progress.”
  + Visual indicators (like progress bars and checkmarks) must update accurately.
  + Users should see a summary of completed and pending tasks.
  + Progress data should sync with the backend.

## **Push Notifications**

As a user, I want to receive reminders to work on my bucket list so that I stay motivated to complete my tasks.

* **How It Works:**

Users can enable notifications in the app settings to get reminders about pending tasks. Notifications can be customized based on user preferences, like daily, weekly, or for specific tasks that have been pending for too long.

* **Acceptance Criteria:**
  + Users must be able to enable or disable notifications.
  + Notifications should be customizable by frequency and task.
  + Users should receive notifications even when the app is not open.
  + Notifications must link directly to the relevant task in the app.