

Recipe Manager App - Lab Tasks

Overview

Students will build a comprehensive recipe manager app using React Native with the following technologies:

- React Native
 - FlatList for recipe display
 - TextInput for recipe input and search
 - Tab Navigation and Stack Navigation
 - AsyncStorage for user preferences
 - SecureStore for user profile data
 - SQLite for recipe data storage
-

Lab Task 1: Navigation Setup & Basic UI Structure

Objectives:

- Set up React Navigation with Tab and Stack navigators
- Create basic screen components
- Implement a modern, food-focused UI structure

Requirements:

1. Install Required Dependencies:

2. @react-navigation/native
3. @react-navigation/bottom-tabs
4. @react-navigation/stack
5. react-native-screens
6. react-native-safe-area-context

7. Create Navigation Structure:

- Bottom Tab Navigator with 4 tabs:
 - **Recipes** (Recipe List)
 - **Add Recipe**
 - **Favorites**
 - **Profile**
- Stack Navigator for:
 - Recipe Details screen (accessible from Recipes/Favorites)
 - Edit Recipe screen
 - Recipe Steps screen (cooking mode)

8. Screen Components to Create:

- RecipesScreen.js - Main recipe list with search
- AddRecipeScreen.js - Add new recipe form
- FavoritesScreen.js - Saved favorite recipes
- ProfileScreen.js - User profile and app settings

- RecipeDetailsScreen.js - Individual recipe details
- EditRecipeScreen.js - Edit existing recipe
- CookingModeScreen.js - Step-by-step cooking instructions

9. UI Requirements:

- Use food-themed tab icons (🔍, +, ❤️, 🧑)
- Implement proper header titles with search functionality
- Add modern card-based layouts
- Include recipe placeholder content for each screen

Deliverables:

- Working navigation between all screens
 - Food-themed UI layout for each screen
 - Clean, modern styling with recipe cards
-

Lab Task 2: Data Management & Core Recipe Functionality

Objectives:

- Implement SQLite database for recipe storage
- Create CRUD operations for recipes
- Build the Add Recipe functionality with image support and validation

Requirements:

1. Database Setup:

- Install expo-sqlite or react-native-sqlite-storage
- Create database schema:


```
CREATE TABLE recipes (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        title TEXT NOT NULL,
        description TEXT,
        category TEXT NOT NULL,
        difficulty TEXT NOT NULL,
        prep_time INTEGER NOT NULL,
        cook_time INTEGER NOT NULL,
        servings INTEGER NOT NULL,
        ingredients TEXT NOT NULL,
        instructions TEXT NOT NULL,
        image_url TEXT,
        is_favorite INTEGER DEFAULT 0,
        created_at DATETIME DEFAULT CURRENT_TIMESTAMP);
```

2. Add Recipe Screen Implementation:

- Form fields:
 - Recipe Title (TextInput)
 - Description (Multiline TextInput)
 - Category (Picker/Dropdown)
 - Difficulty Level (Easy/Medium/Hard)
 - Prep Time & Cook Time (Numeric inputs)
 - Servings (Numeric input)
 - Ingredients List (Dynamic TextInput list)
 - Instructions (Step-by-step TextInput list)
 - Photo (Image picker - optional)
- Form validation:

- Required fields validation
 - Time values must be positive
 - At least one ingredient and instruction required
 - Save to SQLite database
- 3. Recipes Screen Implementation:**
- Fetch recipes from SQLite
 - Display recipes using FlatList in card format
 - Show: title, category, difficulty, prep time, thumbnail
 - Implement search functionality (by title/ingredients)
 - Add filter by category and difficulty
 - Implement pull-to-refresh
 - Recipe card onPress navigation to details
- 4. Categories & Difficulty Levels:**
- Categories: Breakfast, Lunch, Dinner, Dessert, Snacks, Beverages, Appetizers
 - Difficulty: Easy, Medium, Hard

Deliverables:

- Functional Add Recipe form with validation
 - Recipes screen displaying recipes from database with search/filter
 - Working database CRUD operations for recipes
-

Lab Task 3: Advanced Features & User Experience

Objectives:

- Implement recipe editing, favorites, and cooking mode
- Add user profile and preferences
- Implement secure data storage
- Add advanced UI features and animations

Requirements:

- 1. Recipe Details & Management:**
 - Recipe Details Screen:
 - Display full recipe information
 - Ingredient list with checkboxes
 - Step-by-step instructions
 - Cooking timer integration
 - Edit, Delete, and Favorite buttons
 - Share recipe functionality
 - Edit Recipe Screen:
 - Pre-populate form with existing data
 - Update database record
 - Delete functionality with confirmation dialog
 - Favorite/unfavorite toggle
- 2. Cooking Mode Screen:**

- Step-by-step cooking instructions
- Navigation between recipe steps
- Built-in timers for each step
- Ingredients checklist
- Keep screen awake during cooking
- Large, easy-to-read text for cooking

3. Favorites Screen:

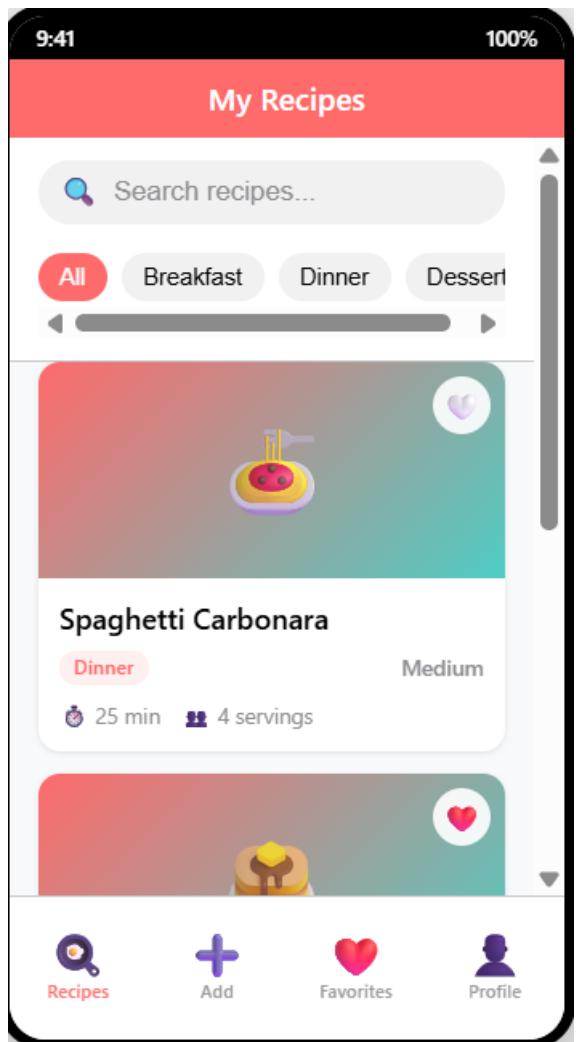
- Display favorited recipes using FlatList
- Same card layout as main recipes screen
- Quick unfavorite option
- Search within favorites

4. Profile Screen with AsyncStorage:

- User preferences:
 - Default serving size
 - Preferred measurement units (Metric/Imperial)
 - Dietary restrictions tags
 - Theme preference (Light/Dark)
- Recipe statistics:
 - Total recipes created
 - Favorite categories
 - Cooking frequency
- Data management:
 - Export recipes
 - Import recipes from JSON
 - Clear all data option

5. Security & Advanced Features:

- Use SecureStore for:
 - User profile information
 - Personal recipe notes
 - Shopping list data
- Enhanced UI/UX:
 - Recipe image gallery
 - Ingredient quantity calculator
 - Shopping list generation from recipes
 - Recipe rating system
 - Recent recipes section
 - Offline recipe access



Add Recipe

9:41 100%

RECIPE TITLE

Enter recipe name

CATEGORY Breakfast **DIFFICULTY** Easy

PREP TIME Minutes **COOK TIME** Minutes

SERVINGS

Number of servings

DESCRIPTION

Brief description...

Add Recipe

Recipes Add Favorites Profile

Add Recipe

9:41 100%

Breakfast Easy

PREP TIME Minutes **COOK TIME** Minutes

SERVINGS

Number of servings

DESCRIPTION

Brief description...

Add Recipe

Recipes Add Favorites Profile

The image displays two side-by-side screenshots of a mobile application interface for a recipe. Both screens show the same recipe details for "Spaghetti Carbonara".

Top Screen (Left):

- Header:** "9:41" and "100%" at the top corners.
- Back Button:** "← Back" in the top left.
- Title:** "Recipe Details" in the top center.
- Image:** A small icon of a pan with spaghetti and a heart icon in the top right corner.
- Section Header:** "Spaghetti Carbonara" in bold black text.
- Cooking Information:** "Dinner" (highlighted in pink), "Medium • 25 min • 4 servings".
- Ingredients Section:** "Ingredients" with a list:
 - 400g Spaghetti pasta
 - 200g Pancetta or bacon
 - 3 large eggs
 - 100g Parmesan cheese
 - 2 garlic cloves
- Buttons:** "Start Cooking" (red background) and "Edit Recipe" (gray background).
- Bottom Navigation:** Icons for "Recipes" (magnifying glass), "Add" (plus sign), "Favorites" (heart), and "Profile" (person).

Bottom Screen (Right):

- Header:** "9:41" and "100%" at the top corners.
- Back Button:** "← Back" in the top left.
- Title:** "Recipe Details" in the top center.
- Cooking Information:** "Dinner" (highlighted in pink), "Medium • 25 min • 4 servings".
- Ingredients Section:** "Ingredients" with a list:
 - 400g Spaghetti pasta
 - 200g Pancetta or bacon
 - 3 large eggs
 - 100g Parmesan cheese
 - 2 garlic cloves
- Buttons:** "Start Cooking" (red background) and "Edit Recipe" (gray background).
- Bottom Navigation:** Icons for "Recipes" (magnifying glass), "Add" (plus sign), "Favorites" (heart), and "Profile" (person).

9:41 100%

My Favorites

Search favorites...

Fluffy Pancakes

Breakfast | Easy

15 min | 2 servings

Spaghetti Carbonara

Step 2 of 6

Heat a large pan over medium heat. Add the chopped pancetta and cook until crispy and golden, about 4-5 minutes.

04:32

Pause Stop

