Spare Spoon App: Technical Documentation

Team Members

Gowtham Padala (201900149): GitHub
Abhishek Gujjar(202057931): GitHub
Harsh Sharma (201961844): GitHub
Faiyez Noor (201928975): GitHub

Overview

Spare Spoon is an innovative application designed to transform home cooking. It offers recipe recommendations based on available ingredients, aiming to reduce food waste, save money, and introduce diverse culinary experiences. The app caters to a wide audience, from college students to families and seasoned cooks.

App Description

Spare Spoon focuses on utilizing available ingredients to offer diverse recipe recommendations. It's an ideal solution for minimizing food waste and exploring new culinary ideas.

Target Audience

Spare Spoon appeals to college students, families, and cooking enthusiasts looking for creative and sustainable cooking options.

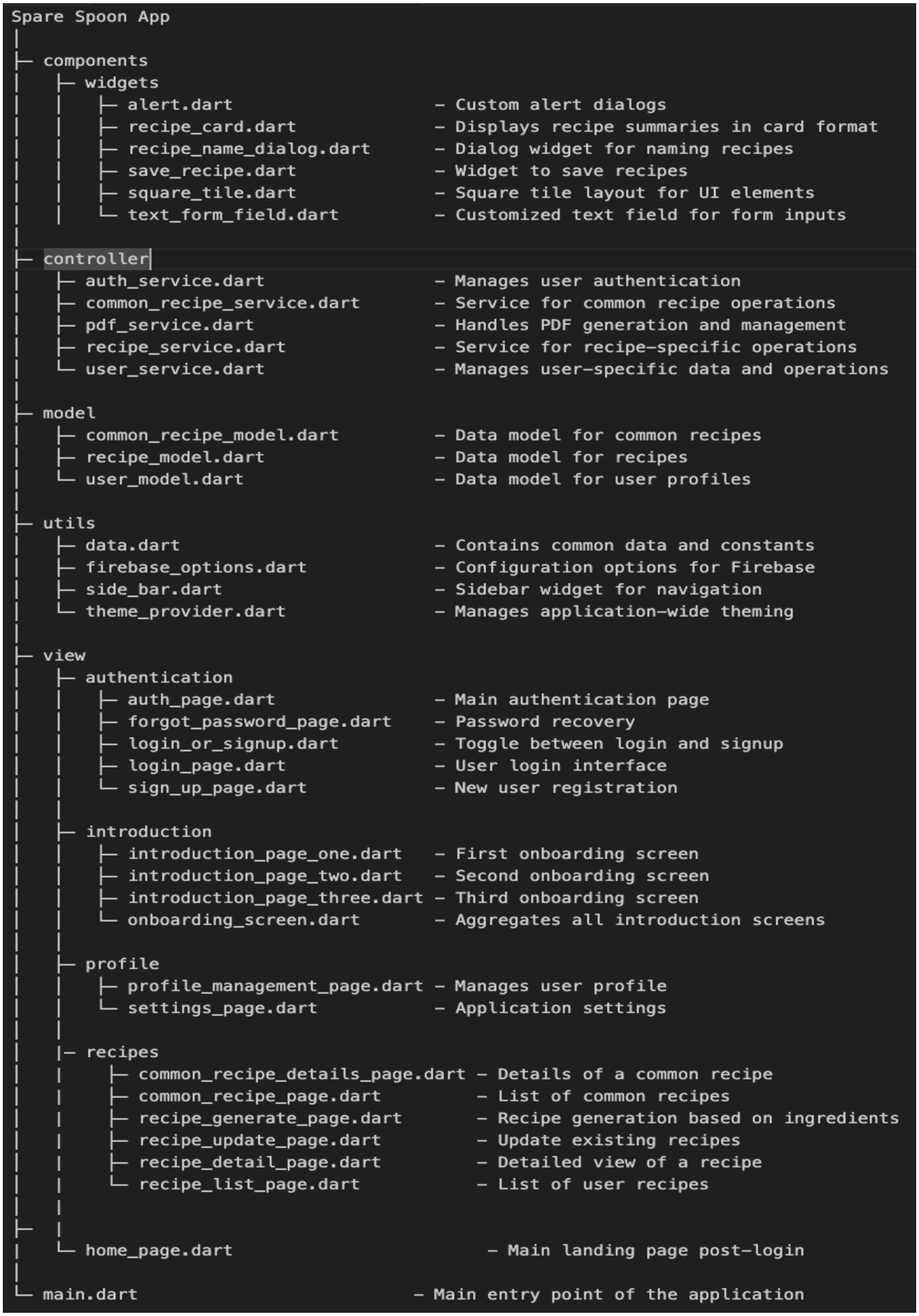
Key Features

- Authentication and Data Security: Secured user data with authentication.
- OPEN AI Integration: For recipe suggestions and regeneration.
- Favorite Recipes: Mark and filter favorite recipes.
- User Profile Management: Personal information management in the profile section for dietary preferences.
- Recipe Customization: Editing existing recipes and adding images.
- Download recipes as PDFs.
- Recipe text filtering.
- Auto-generated shopping lists for recipes.
- Chat support using tawt.to
- Popular recipes recommended by famous chefs.

Technical Stack Used

- UI Development: Flutter, Figma
- User Management: FirebaseAuth
- Database: Firestore
- Image Storage: Firebase Cloud Storage
- APIs: OpenAI, Tawk.to

Technical Documentation: File Structure



```
Spare Spoon App
│
├─ components
│   ├─ widgets
│   │   ├─ alert.dart                   – Custom alert dialogs
│   │   ├─ recipe_card.dart             – Displays recipe summaries in card format
│   │   ├─ recipe_name_dialog.dart      – Dialog widget for naming recipes
│   │   ├─ save_recipe.dart             – Widget to save recipes
│   │   ├─ square_tile.dart             – Square tile layout for UI elements
│   │   └─ text_form_field.dart         – Customized text field for form inputs
│
├─ controller
│   ├─ auth_service.dart                – Manages user authentication
│   ├─ common_recipe_service.dart       – Service for common recipe operations
│   ├─ pdf_service.dart                 – Handles PDF generation and management
│   ├─ recipe_service.dart              – Service for recipe–specific operations
│   └─ user_service.dart                – Manages user–specific data and operations
│
├─ model
│   ├─ common_recipe_model.dart         – Data model for common recipes
│   ├─ recipe_model.dart                – Data model for recipes
│   └─ user_model.dart                  – Data model for user profiles
│
├─ utils
│   ├─ data.dart                        – Contains common data and constants
│   ├─ firebase_options.dart            – Configuration options for Firebase
│   ├─ side_bar.dart                    – Sidebar widget for navigation
│   └─ theme_provider.dart              – Manages application–wide theming
│
├─ view
│   ├─ authentication
│   │   ├─ auth_page.dart               – Main authentication page
│   │   ├─ forgot_password_page.dart    – Password recovery
│   │   ├─ login_or_signup.dart         – Toggle between login and signup
│   │   ├─ login_page.dart              – User login interface
│   │   └─ sign_up_page.dart            – New user registration
│   │
│   ├─ introduction
│   │   ├─ introduction_page_one.dart   – First onboarding screen
│   │   ├─ introduction_page_two.dart   – Second onboarding screen
│   │   ├─ introduction_page_three.dart – Third onboarding screen
│   │   └─ onboarding_screen.dart       – Aggregates all introduction screens
│   │
│   ├─ profile
│   │   ├─ profile_management_page.dart – Manages user profile
│   │   └─ settings_page.dart           – Application settings
│   │
│   |– recipes
│   │    ├─ common_recipe_details_page.dart – Details of a common recipe
│   │    ├─ common_recipe_page.dart         – List of common recipes
│   │    ├─ recipe_generate_page.dart       – Recipe generation based on ingredients
│   │    ├─ recipe_update_page.dart         – Update existing recipes
│   │    ├─ recipe_detail_page.dart         – Detailed view of a recipe
│   │    └─ recipe_list_page.dart           – List of user recipes
│   │
├─  │
│   └─ home_page.dart                        – Main landing page post–login
│
└─ main.dart                          – Main entry point of the application
```

1. File Structure and Responsibilities

a. Components

- Alert (`components/alert.dart`)
  - Purpose: Handling modal dialogue operations.
  - Functionality: Error, warning, success modals; recipe operations; team and user account management.
  - Interactions: RecipeService, AuthService, UI elements.
  - Contributors: Abhishek Gujjar, Gowtham Padala, Harsh Sharma.
- RecipeCard (`components/recipe_card.dart`)
  - Purpose: Displaying recipe information.
  - Functionality: Details and image display; user interaction handling.
  - Interactions: Alert class, RecipeModel.
  - Contributors: Abhishek Gujjar and Harsh Sharma.
- RecipDialog (`components/recipe_name_dialog.dart`)
  - Purpose: Managing recipe-related dialogs.
  - Functionality: Obtaining recipe names and favourites status.
  - Interactions: UI elements for user input.
  - Contributors: Harsh Sharma.
- SavedRecipeItem (`components/saved_recipe.dart`)
  - Purpose: Representing saved recipes.
  - Functionality: Editing or deleting saved recipes.
  - Interactions: RecipeModel.
  - Contributors: Harsh Sharma.
- SquareTile (`components/square_tile.dart`)
  - Purpose: Displaying square image tiles.
  - Functionality: Image display and event handling.
  - Interactions: Image paths, callback functions.
  - Contributors: Harsh Sharma.
- CustomTextField (`components/text_form_field.dart`)
  - Purpose: Custom text field for input.
  - Functionality: Styled text field, password input.
  - Interactions: TextEditingController.
  - Contributors: Harsh Sharma.

b. Controllers

- AuthService (`controller/auth_service.dart`)

    - Purpose: Authentication operations.
    - Key Functions: Google Sign-In, email/password authentication, user registration, user management.
    - Firebase Interaction: FirebaseAuth.
    - Contributors: Gowtham Padala and Harsh Sharma.
- CommonRecipeService (`controller/common_recipe_service.dart`)
    - Purpose: Managing common recipes.
    - Key Functions: Fetch and update recipes.
    - Firebase Interaction: Firestore.
    - Contributors: Abhishek Gujjar and Harsh Sharma.
- PdfService (`controller/pdf_service.dart`)
    - Purpose: PDF management.
    - Key Functions: Generate and save PDFs for recipes.
    - Dependencies: PDF package, HTTP, File and Path Providers.
    - Contributors: Abhishek Gujjar, Gowtham Padala, Harsh Sharma.
- RecipeService (`controller/recipe_service.dart`)
    - Purpose: CRUD for recipes.
    - Key Functions: Add, delete, and update recipes; fetch user-specific recipes.
    - Firebase Interaction: Firestore.
    - Contributors: Harsh Sharma.
- UserService (`controller/user_service.dart`)
    - Purpose: User data management in Firestore.
    - Key Functions: Create, retrieve, update, and delete user data.
    - Firebase Interaction: Firestore.
    - Contributors: Abhishek Gujjar and Harsh Sharma.

c. Model Classes

- CommonRecipeModel (`model/common_recipe_model.dart`)
    - Purpose: Representing common recipes.
    - Attributes: ID, name, description, category, creation date, rating, vegetarian status, image URL.
    - Dependencies: Firestore.
    - Contributors: Abhishek Gujjar and Harsh Sharma.
- RecipeModel (`model/recipe_model.dart`)
    - Purpose: Representing user-specific recipes.
    - Attributes: ID, user ID, basic details, favourite status, timestamps, images.
    - Dependencies: Firestore.
    - Contributors: Harsh Sharma, Faiyez Noor.

- UserModel (`model/User_model.dart`)
    - Purpose: Representing user data.
    - Attributes: ID, basic info, dietary preferences, first-time login flag.
    - Dependencies: Firestore.
    - Contributors: Abhishek Gujjar and Harsh Sharma.

d. Utility Classes

- Data (`utils/data.dart`)
    - Purpose: Static data storage.
    - Attributes: Settings data.
    - Contributors: Harsh Sharma.
- FirebaseOptions (`utils/firebase_options.dart`)
    - Purpose: Firebase configuration.
    - Class: DefaultFirebaseOptions.
    - Contributors: Abhishek Gujjar, Gowtham Padala, Harsh Sharma.
- Sidebar (`utils/side_bar.dart`)
    - Purpose: Sidebar widget for navigation.
    - Attributes: User, selectedIndex, update function.
    - Contributors: Abhishek Gujjar, Gowtham Padala, Harsh Sharma.
- ThemeProvider (`utils/theme_provider.dart`)
    - Purpose: Theme settings management.
    - Class: ThemeProvider.
    - Attributes: Dark theme flag.
    - Contributors: Harsh Sharma.

2. Firebase Integration

- Details: Used for authentication and database operations across modules.
- Key Pages:
    - AuthPage (`view/authentication/auth_page.dart`): Manages user authentication.
    - ForgotPasswordPage (`view/authentication/forgot_password_page.dart`): Password reset functionality.
    - LoginOrSignUpPage (`view/authentication/login_or_signup.dart`): Gateway for login/signup.
    - LoginPage (`view/authentication/login_page.dart`): Handles user login.

    - SignUpPage (`view/authentication/sign_up_page.dart`): New user registration.

3. Customer Support and Introduction Modules

- CustomerSupport (`view/chat/customer_support.dart`): Chat interface for customer support.
- IntroPage1-3 (`view/introduction/intro_page_[1-3].dart`): Introduction sequence pages.
- OnboardingScreen (`view/introduction/onboarding_screen.dart`): Manages onboarding experience.

4. Profile and Recipe Management

- ProfileManagementPage (`view/profile/profile_management_page.dart`): User profile management.
- SettingsPage (`view/profile/settings_page.dart`): User settings and account management.
- Recipe Pages:
    - CommonRecipesDetailsPage (`view/recipe/common_recipe_details_page.dart`): Details of common recipes.
    - CommonRecipesPage (`view/recipe/common_recipe_page.dart`): Listing and filtering common recipes.
    - GenerateRecipe (`view/recipe/recipe_generate_page.dart`): Recipe generation.
    - UpdateRecipePage (`view/recipe/recipe_update_page.dart`): Updating existing recipes.
    - RecipesDetailsPage (`view/recipe/recipe_detail_page.dart`): Detailed recipe information.
    - RecipesPage (`view/recipe/recipe_list_page.dart`): Displaying user's recipes and favourites.

5. Main Application Structure

- HomePage (`home.dart`): Main navigation hub of the app.
- SpareSpoonApp (`main.dart`): Root of the app, initializing Firebase and setting up UI.