AI Chef Recipe Generator – Final Report

Project Team:
Oleg Vasiliev
Moied Ahmed
Eiad Alsafadi

Technology Stack: Flutter (Dart), Firebase Firestore, Firebase Authentication, Gemini API

Project Aim and Description

Aim

The AI Chef Recipe Generator aims to help users quickly find recipes they can cook using ingredients they already have, while considering cooking time, cooking method, and dietary restrictions.

Description

AI Chef is a cross-platform mobile and web application built using Flutter and Dart. Users authenticate through Firebase Authentication and manage ingredient data stored in Firebase Firestore. Based on user selections, the app sends API requests to the Gemini AI API to generate personalized recipes with step-by-step cooking instructions.

Functional and Non-Functional Requirements

Functional Requirements

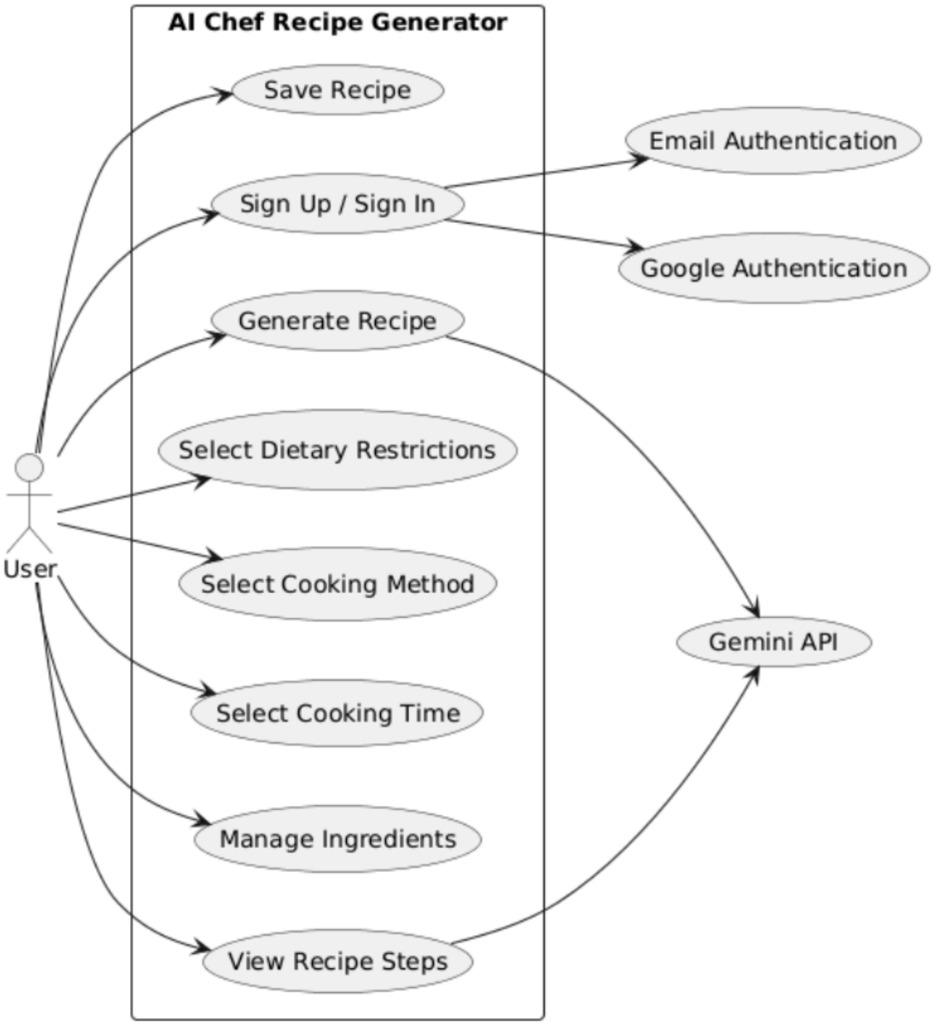
- User sign up and login using Email/Password and Google Authentication
- Ingredient list management
- Selection of cooking time, cooking method, and dietary restrictions
- AI-based recipe generation using Gemini API

- Display and save generated recipes

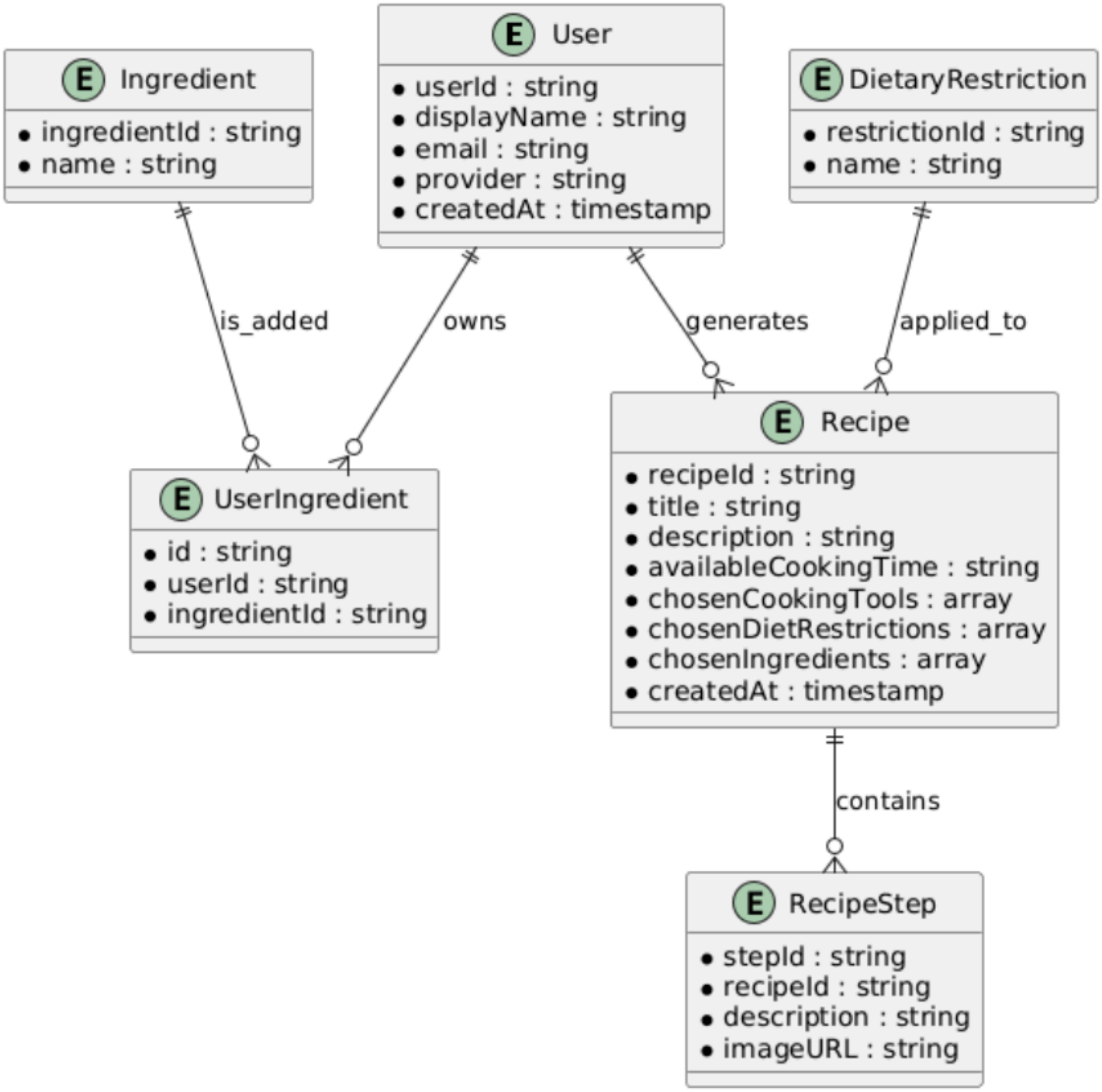
Non-Functional Requirements

- Fast response time for recipe generation
- Secure authentication and data storage
- Simple and intuitive user interface
- Scalable and cross-platform design

USE CASE DIAGRAM



ERD DIAGRAM



User Stories

- As a user, I want to log in using email or Google so my data is saved securely.
- As a user, I want to manage my ingredients for future use.
- As a user, I want recipes that match my available time and cooking method.

- As a user, I want recipes that follow my dietary restrictions.
- As a user, I want clear steps for recipes.

Test Cases

Test Case	Expected Outcome	Status
Email Authentication	User logs in successfully	Passed
Google Authentication	User logs in successfully	Passed
Ingredient Update	Data saved in Firestore	Passed
Recipe Generation	Recipe returned from Gemini API	Passed
Dietary Filtering	Recipe meets restrictions	Passed
App Reload	User data persists	Passed

Individual Roles and Responsibilities

Oleg Vasiliev – Frontend, API Integration & Data Handling

- Developed Flutter UI
- Integrated Gemini API
- Managed API calls and data validation
- Connected Flutter app to Firebase Firestore

Moied Ahmed – Database & Testing

- Designed Firestore database
- Implemented Firebase Authentication

- Assisted in connecting Flutter app to Firebase Firestore
- Performed testing and validation

Eiad Alsafadi – Support Role

- Assisted with application development
- Supported overall system integration

End of Final Report