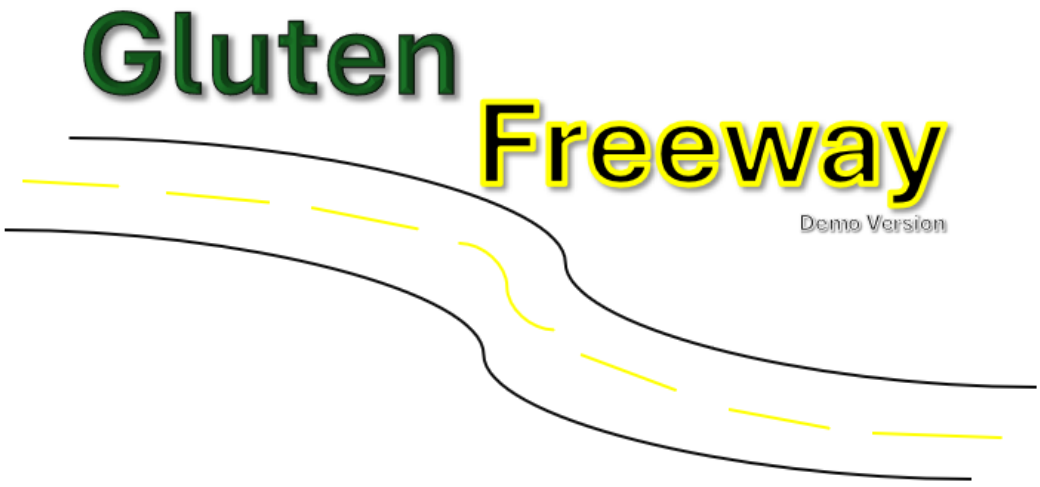
GRAND CANYON UNIVERSITY



STEVEN CUSEY

CST-451 CAPSTONE PROJECT
IMPLEMENTATION PLAN
GRAND CANYON UNIVERSITY

INSTRUCTOR:
PROFESSOR AMR ELCHOUEMI

REVISION: 6.0

DATE: 03/16/2025

ABSTRACT

For this project, our team is going to be developing an android application that allows the user to find local gluten-free restaurants . It will be similar to other navigation apps that the user can find on their phone's app store. Our team will need to implement features that ensure a smooth user experience. It will be simple to use, in order to ensure that everyone with a gluten allergy is able to use it. It will need to have two main features: the ability to navigate to a restaurant, and the ability to review a restaurant.

The first feature is a navigation tool. It will allow the user to find a restaurant, see what gluten free food it has available, then go there. It will be seamless, with an emphasis on being as accessible as possible. This will ensure that as many people as possible can use it, providing an enjoyable user experience.

The second feature it will have is the ability to review restaurants on a 5 star scale. This will ensure that a user can state whether or not a restaurant truly has good gluten free food available, and allow them to share this information with other people with gluten allergies. This ensures that people don't end up finding gluten-free food that tastes bad. To see all the features that are in scope for this project, see the Excel doc submitted with it.

History and Signoff Sheet

Change Record

Date	Author	Revision Notes
		Initial draft for review/discussion

Overall Instructor Feedback/Comments

Overall Instructor Feedback/Comments

Integrated Instructor Feedback into Project Documentation

☒ Yes ☐ No

TABLE OF CONTENTS

DESIGN OVERVIEW

DETAILED HIGH-LEVEL SOLUTION DESIGN

DETAILED TECHNICAL DESIGN

APPENDIX A – TECHNICAL ISSUE AND RISK LOG

APPENDIX B – REFERENCES

APPENDIX C – EXTERNAL RESOURCES

DESIGN OVERVIEW

Design Introduction

Gluten Freeway will be using a typical Flutter design structure that uses one Widget for each feature. For example, if I want to edit a review, I should have a widget called EditReview. The application allows the user to find and review gluten free restaurants. The app is connected to a MongoDB database called "glutenfreeway" that runs on an AWS server. This database has 3 collections called "users", "restaurants" and "reviews". The users collection is responsible for storing login information. Gluten Freeway is also connected to the Google Maps API, which allows the user to navigate to the restaurants featured in the database. In addition, users will be able to see and review restaurants as well. The previously mentioned MongoDB uses a free AWS connection that costs nothing, making it extremely cost effective.

Deliverable Acceptance Log					
ID	Deliverable Description	Comments	Evaluator (internal or external as applicable)	Status	Date of Decision
1	Data Dictionary	This is a data dictionary of all my MongoDB fields.	Steven Cusey	Complete	11/12/24

For more details on what features the app will need to perform, see the user stories excel sheet that has been submitted with this file.

DETAILED HIGH-LEVEL SOLUTION DESIGN

Introduction

GlutenFreeway will need will be made using Flutter and Dart in VS code, and it will require 3 MongoDB database tables. Flutter Version 3.24.3 is the version currently being planned for development, though it is possible our team will update Flutter if we know for certain we can safely migrate to a newer version. Similarly, we are using Dart 3.5.3, but may update if we can do it safely. There will be 5 pages in the app: login, registration, restaurants, navigation, create review, and edit review. Each user who registers must have a username less than 20 characters long and provide an email and password. There will be a navbar to navigate between all 4 pages, and the restaurant navigation pages will only be accessible if the user is logged in. The app will require 3 MongoDB database collections: users, restaurants and reviews. The app will be use the AWS connection service, providing connection to the MongoDB. The users collection will need fields for a username, email and password, the restaurants collection will need fields for the restaurant name, address and available foods, and number of stars the restaurant has, and a short review summary for the restaurant. The app will be tested locally and will use a cable while testing and use the Flutter Build APK to download the app onto a phone when completed for demonstration. The Flutter Build APK allows the app to run without the computer being connected. While our team would like to get our app uploaded to the Google Play Store for public use, for now we will just give it to our testing team and download it locally to use.

In Scope technologies:

- **Android 13 is the operating system being developed for**
- **MongoDB used to store users and restaurants**
- **Uses Dart Programming Langauage**
- **Uses Flutter for development**
- **AWS Server in MongoDB**
- **Google Maps API**

Out of Scope technologies:

- **Google Play Store**

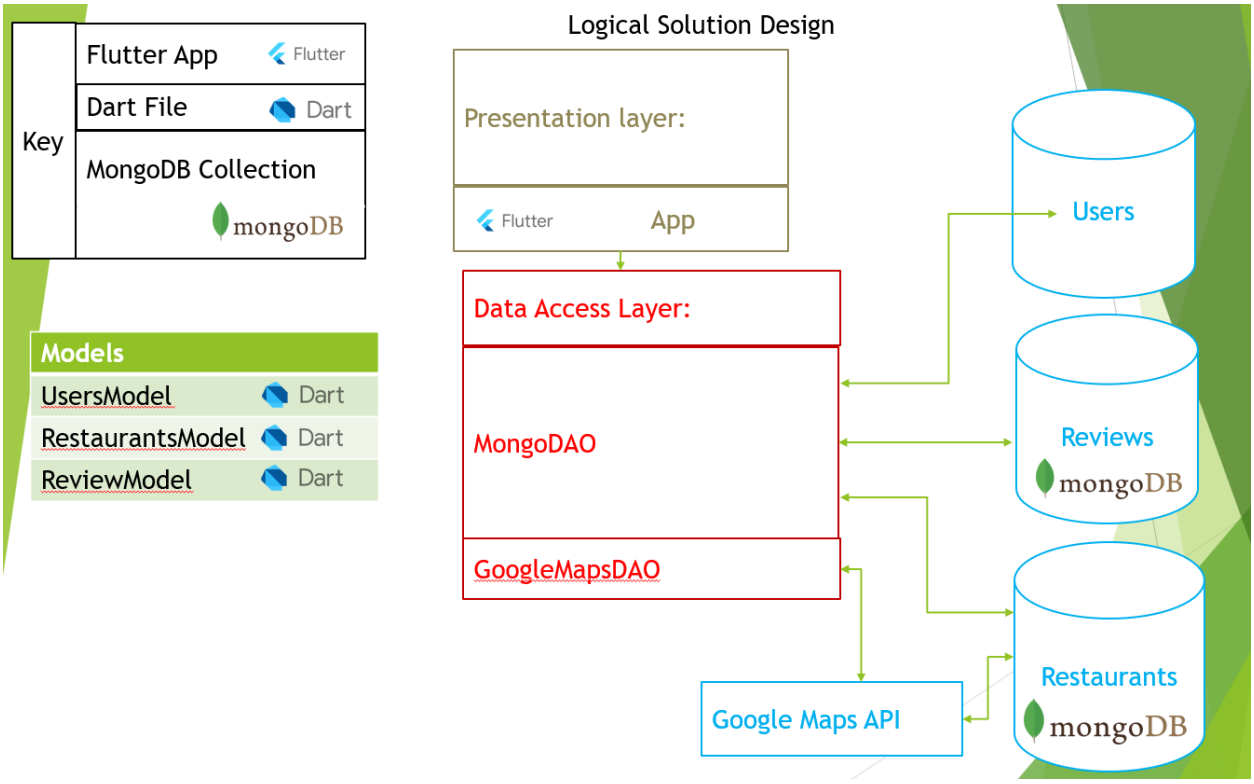
Here are my in-scope technologies in table form.

Hardware and Technology	Justification
Flutter Version 4.24.3	**This is the software development kit that the project will use. It is useful for**

	mobile development for android.
Dart Version 3.5.3	**This is the programming language used for the project. It has built in logging will provide the views and business logic.**
Google Maps	**This is the API the project will be using for navigation. It allows the user to navigate to various locations**
MongoDB Atlas	**This is the database software that will hold the data for our restaurants, users and reviews. It is browser based.**
Android OS Version 13	**This is the operating system we will be developing for.**
AWS	**The server connected to our MongoDB database. While it is Managed in MongoDB atlas, and we never have to access it directed, our team still included it here since it is the server our MongoDB is on.**
Motorola Edge 5G EW	**This is the phone we will be using to test our application.**

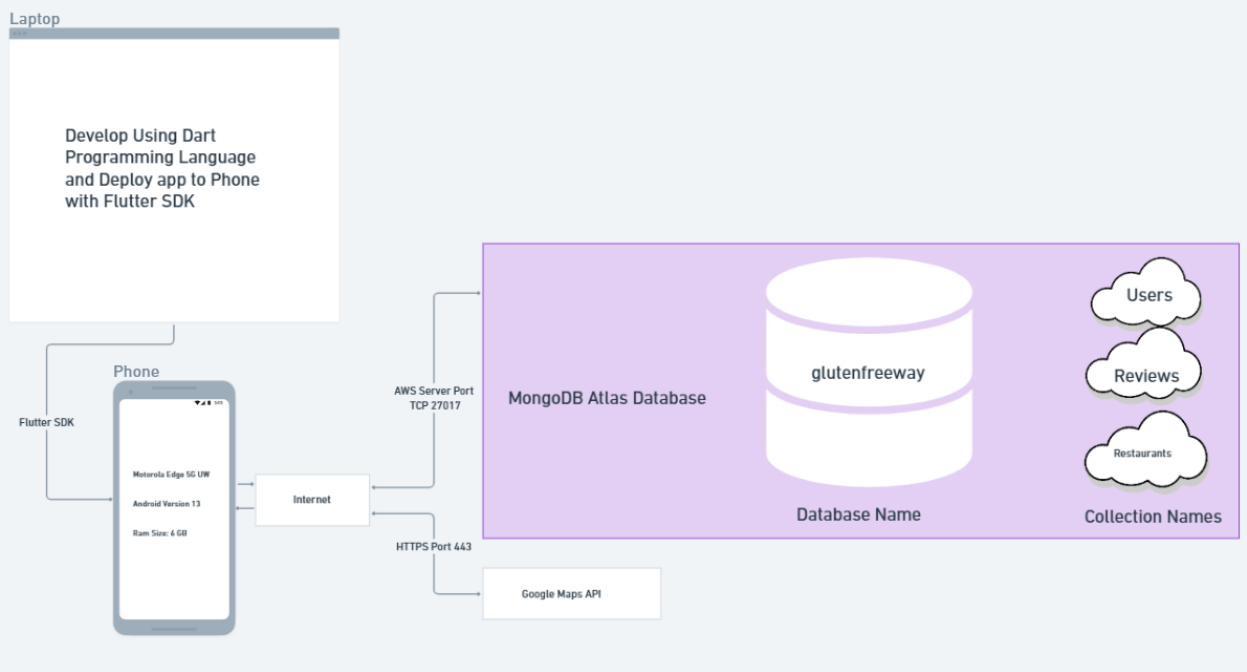
LOGICAL SOLUTION DESIGN:

The app will have 3 primary technologies powering it: MongoDB, Google Maps, and Flutter. Flutter handles more of the front end elements, though all 3 work with the UI to a certain extent, as you coldn't see the list of restaurants if the connections to MongoDB is disconnected and you can't see the route to the restaurant if Google Maps is disconnected. MongoDB uses a connection string powered by AWS to connect to Flutter. (MongoDB supports other connection strings, AWS is just the one our team has chosen for this project) The GoogleMaps API requires a key for connection. Since the Map Logic for Google Maps is connected to the address, city, state and zip field in our restaurants collection, our GoogleMapsDAO will need to connect to our restaurant collection. Additionally, since a restaurant can have more than one review, our team will need to have a review collection separate from the restaurants collection.



PHYSICAL SOLUTION DESIGN:

The app will be developed on a laptop using Dart and Flutter. When the App is being tested, it will use a cable to connect the development laptop to the phone. Once the App is finished, the App will be downloaded to the phone using Flutter SDK. The app requires an internet connection, and is connected to both MongoDB using the free AWS connection available with MongoDB Atlas. The connection connects to a MongoDB database called glutenfreeway with 3 collections: users, reviews and restaurants. In addition to MongoDB, our app uses the Google Maps API for navigation.



GENERAL TECHNICAL APPROACH:

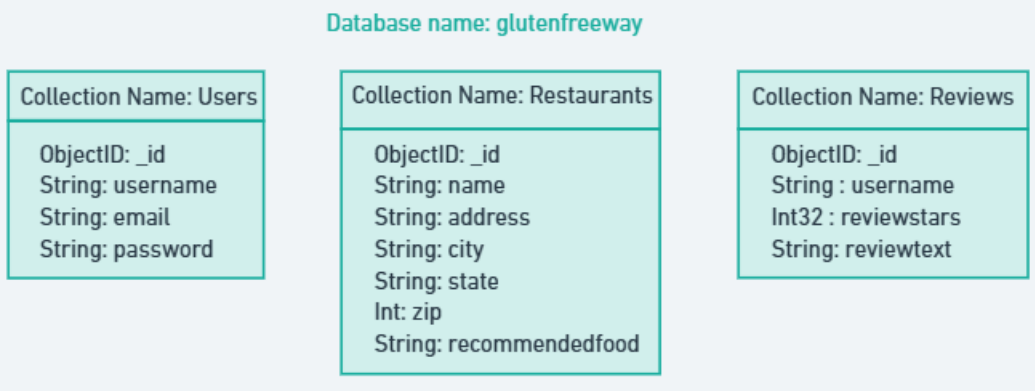
For our general technical approach, we are using a design pattern in flutter that is similar to an MVC. Flutter doesn't usually use MVC though, as navigating between pages is easier, so it is more like an MV since the is no true controller.

KEY TECHNICAL DESIGN DECISIONS:

Hardware and Technology used in design	Justification
Flutter Version 4.24.3	**This was chosen because it is compatible with our team's phone, and is popular for app development.**
Dart Version 3.5.3	**This was chosen because it connects with Flutter nicely.**
Google Maps	**This is the API the project will be using for navigation. It allows the user to navigate to various locations**
MongoDB Atlas	**This is the database software that will hold the data for our restaurants, users and reviews. It is browser based.**
Android OS Version 13	**This is the operating system we will be developing for.**
AWS	**The server connected to our MongoDB database. This was chosen due to it being free and easy to use.**
Motorola Edge 5G EW	**This is was chosen because it is the only phone our team has access to.**

DATABASE ER DIAGRAM AND DATA DICTIONARY:

The Users collection stores the user info for login and registration. If a user registers an account with their username, email and password, they will be added here. The Restaurants collection stores the restaurants. A restaurant cannot be added or modified within the application itself, as to add a restaurant to our database, it would have to approved by our team after a request and added to our database by our database manager. This is to ensure that every restaurant in our app is 100% safe for those who need a gluten free experience. A user can review a restaurant. The review will be on a 5 star scale, and the user can share their experience in their review.

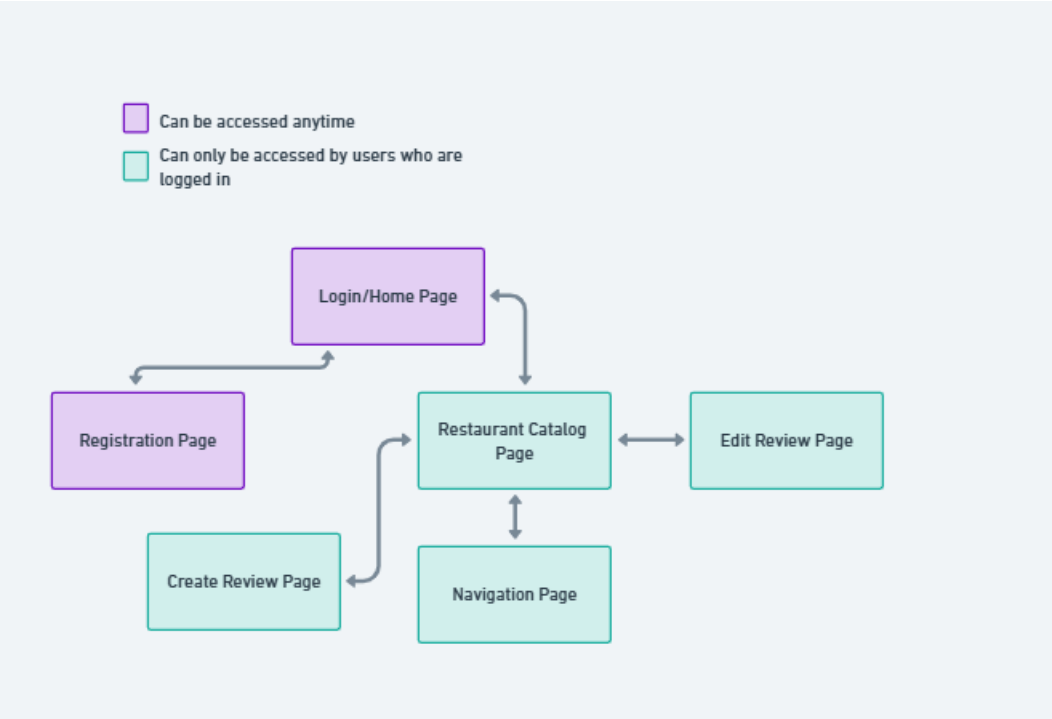


Name	Mongo Collection	Data Type	Field Size	Description	Example Comment	Nulls (Y/N)
_id	all	Id	Automatic	Generates A Unique ID for Each Item	ObjectID(#)	N
username	users	String	6-20	The User's Username	Steven	N
email	users	String	less than 40	The User's Email	scusey@my.gcu.edu	N
password	users	Password	6-20	The User's Password, Must contain an uppercase letter, lowercase letter and number	Testing1	N
restaurantname	restaurants	String	1-30	The restaurant's name	Pizza Ranch	N
address	restaurants	String	1-50	The restaurant's street address	1431 E LaSalle Dr	N
city	restaurants	String	1-25	The city the restaurant is in	Bismarck	N
state	restaurants	String	2	The two letter abbreviation of the state the restaurant is in	ND	N
zip	restaurants	Int32	5	The zipcode the restaurant is in	58503	N
recommendedfood	restaurants	String	1-50	A summary of the gluten free food that can be found at the restaurant	This restaurant has gluten free pizza available. They also have ice cream and potatoes available as well.	N
reviewstars	reviews (only objects in the reviews Mongo collection can be null, as a restaurant that just opened may not have any reviews yet)	Int32	1	This shows how many stars a restaurant has using an int from 1 to 5	★★★★☆	Y
reviewtext	reviews	String	1-200	This allows the user to give their thoughts on a restaurant	The pizza is delicious, and their soft serve Ice cream is splended.	Y

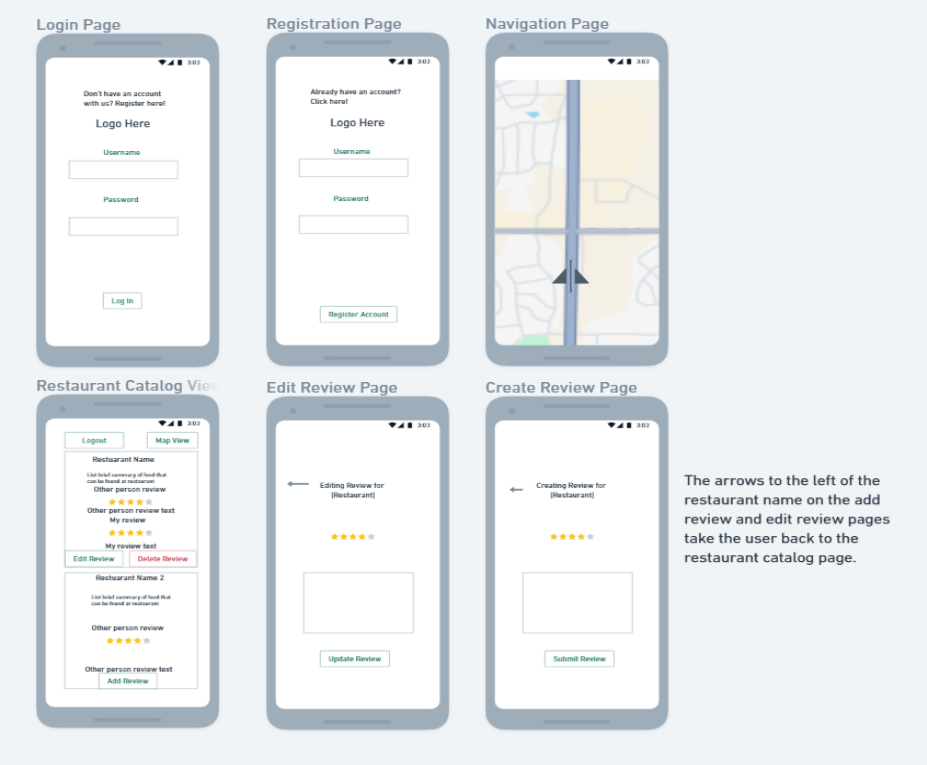
SITEMAP DIAGRAM:

Here is my sitemap. The purple pages indicate pages that can be accessed by anybody, and the mint pages indicate pages that can only be accessed by users who are logged in. You can use the sitemap to see how the pages in the wireframe should interact on the restaurant app.

Software Layout Sitemap:



USER INTERFACE DIAGRAMS:



SERVICE API DESIGN:

Google Maps Uses a key in order to access it. It is a fairly straightforward process. This key can be inserted to a Data Access object in order for a map to appear on the screen.

NFR'S (SECURITY DESIGN, ETC.):

Non-Functional Requirements

GlutenFreeway will need to work on Android Version 13 devices. While our team would like it to work on other Android versions and Apple devices, we will just be focusing on Android version 13 for now. Our team would like to add support to other mobile operating systems eventually, if possible. Passwords will be hashed using BCrypt.

In scope:

Android 13

BCrypt

Out of scope:

Android 8-12, 14-15

iPhone OS

Google PixelOS

TEST CASES AND TRACEABILITY MATRIX

For the test cases and traceability matrix for this program, please see the excel files "CST-452GlutenFreewayTestCases.xls" and "GlutenFreewayTraceabilityMatrix1" included with this program.

APPENDIX A – TECHNICAL ISSUE AND RISK LOG

Issues and Risk Log								
Issue or Risk	**Description**	**Project Impact**	**Action Plan/Resolution**	**Owner**	**Importance**	**Date Entered**	**Date to Review**	**Date Resolved**
I/R	What is the issue or risk?	How will this impact scope, schedule, and cost?	How do you intend to deal with this issue?	Who manages this issue?				
R	None	None	None	Steven Cusey	Low	2/13/2024	2/16/2024	2/16/2025

APPENDIX B – REFERENCES

Flow Chart and UML Diagrams made Using Draw.io

Data Dictionary, ER diagrams and Wireframes made with Whimsical

draw.io - free flowchart maker and diagrams online. (n.d.). https://app.diagrams.net/

Whimsical. (n.d.). Whimsical. https://whimsical.com/login

APPENDIX C – EXTERNAL RESOURCES

GIT URL:	*https://github.com/StevenCusey/Gluten-Freeway*
Hosting URL:	*N/A*