1.1 Introduction

This document presents the architecture and detailed design involving the software developed for the PureDine project. This project performs as a web application that can be used to search up local restaurants with their menus to check for different types of allergies possible on them.

1.1.1 System Objectives

The objective of this application is to provide an interface to provide users, either signed in through Firebase integration or standalone, the ability to search for local restaurants in ways including, but not limited to, specific name, city, or zip code. Once searched, the software will list either specific or local restaurants in a local area range each defined with a separate box containing information about them. These boxes can be clicked on to show basic menus of the restaurants with checkboxes at the bottom to mark certain allergens. Once marked, the website will analyze items on the menu with their ingredients to give recommendations on what is safe to eat and what could possibly be harmful to a consumer.

1.1.2 Hardware, Software, and Human Interfaces

The following Hardware, Software, and Human Interfaces are used in PureDine design and development.

1.1.2.1 Hardware

The hardware used for the PureDine project can be summarized as follows. The current developer of PureDine develops on a Windows 10 interface presented on a Dell laptop. The laptop itself can be linked to both home and LMU servers to allow for development in different

locations. The laptop comes equipped with a keyboard and trackpad to allow for interaction with the machine but can also support plugins like external mice or keyboards if needed. Presentation of the project will be done on external devices such as projectors provided by LMU.

1.1.2.2 Software

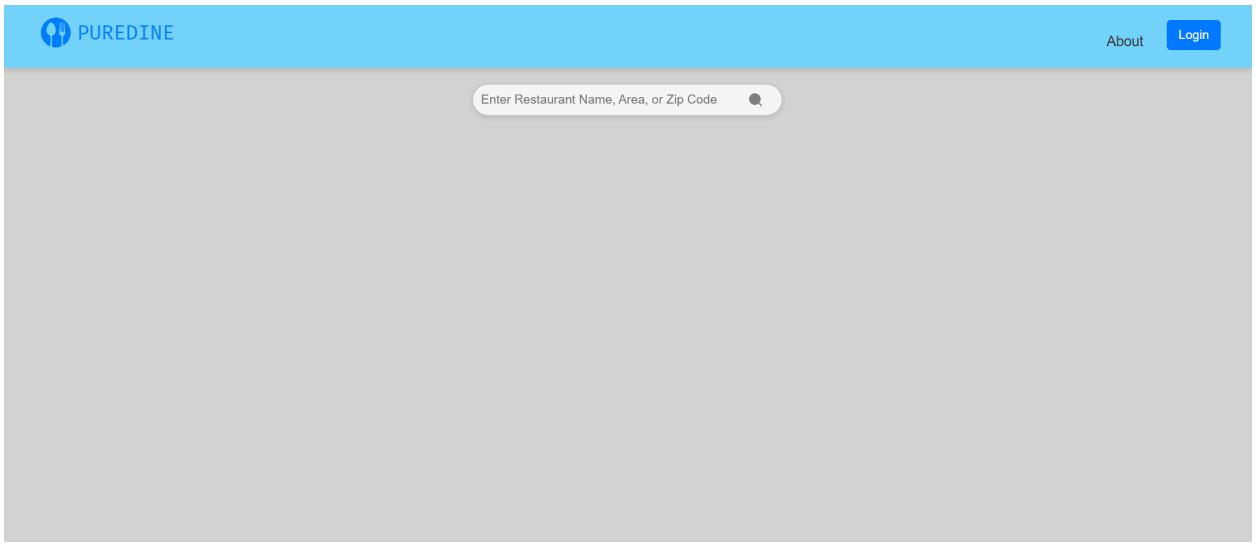
The software used for the PureDine project can be summarized as follows. External browsers such as Safari, Google Chrome, Opera, and Firefox will be used to host the broadcast of the website to users when used. Visual Studio Code will be used to interact with and modify files for development use. React is used for development purposes in association with Node.js to work on frontend and backend processes of the PureDine system and its associated applications. APIs such as the Google Businesses API, Possible MenuPortal API, and Edamam API are used for development of certain pieces of the project including restaurant searching, menu sharing, and allergen information and analysis with menu items. Firebase will be used primarily for hosting of the websites itself, but will also be used for user profiles in tuning of specific allergies involving frequent use of the account.

1.1.2.3 User Interface

The UI for the PureDine project can be summarized as follows. The UI of PureDine is split into four main pages: the main homepage, the about page, the restaurant listing page, and the menu / allergen page. These pages will be described in the following paragraphs.

1.1.2.3.1 Main Homepage

The main homepage of PureDine will display the following when first seen. When loaded, the website will show a header containing the following objects: The logo of PureDine that can be clicked on to reload the page, the link to the “About Page”, and a login button for personal accounts. In the center of the page, there will be a search bar with the following text within it “Please enter a specific restaurant, city, or zipcode”. The user will be able to type in this search bar to go to the “Restaurant Listing Page” when entered. There may also be images and other graphics or text around the website to improve presentation and overall style.



1.1.2.3.2 About Page

The about page of PureDine will display the following when first seen. When loading, the website will have the header from the first page with the following objects still present: a back button for returning to the main webpage, and the logo of the site in the center of the header. The image of the logo can also be clicked onto return to the main page, similarly to the original

image. The page under the header will contain a description of the website's origin + story and a small instruction guide on how to use the website properly with and without an account. There will also be notices on the bottom informing the user that all information provided may not be 100% accurate.

1.1.2.3.1 Restaurant Listing Page

The restaurant page of PureDine will display the following when first seen. The page itself can only be accessed through use of the search bar from the home page. Once a location is searched up, restaurants in a local area will display in boxes that go down the page. Either the page will display restaurants with infinite scroll or with several pages containing about 20 or so restaurants. These restaurant boxes will list the following: restaurant name, restaurant type, restaurant hours, restaurant address, and an associated image. Each of these boxes can be interacted with to go to the menu / allergen page.

1.1.2.3.1 Menu / Allergen Page

The menu / allergy page of PureDine will display the following when first seen. Once a restaurant box from the restaurant page is selected, the menu of the restaurant will be displayed generated from the MealMe API. The menu will be listed in a strong format aiming for the look of a typical restaurant menu to be achieved. Under this menu, there will be several checkboxes denoting allergies, including but not limited to: Nuts, Soy, Shellfish, and more. These boxes can be checked and, in association with the Edamam API, the menu will highlight specific items in specific colors with a guide to the right of the screen highlighting if something is safe to eat or

dangerous based on selection. This page will also include the message from the about page where it highlights that everything given may not be 100% accurate.

1.2 Architectural Design

The Architectural Design of PureDine will be broken down into multiple different subsystems. These include the UI Control, API Control, and Database Control described as the following:

UI Control:

The UI for PureDine is designed based on the main App.js file that structures the main page and links to other webpages. The main class is based in classes and subclasses to deal with and handle different parts of the design. In App.js in particular, there are classes dealing specifically with the header and the search bar of the site allowing them to function and interact with the main page. Using the 'react router dom' extension, the site can also link to different pages through different files and functions. These files include the about page (About.js), restaurant listing page (Restaurant.js), and the menu / allergy page (Menu.js). Each of these files contains separate classes and functions to host UI for those separate pages including features such as restaurant listings, menu display, and information on the general site. These files mainly interact with each other through the use of UseEffect functions that allow for site operations and modifications to happen in real time.

API Control:

The API's of PureDine are designed to work based on a main file containing the APIs themselves labeled Api.js. This file contains constants labeled "getRestaurants", "getMenus",

and “getAllergies” specifically which are async functions that contain the API keys for each of the APIs used (Google Places, MealMe, Edamam) along with urls to specific parts of the sites to draw information from. These work in parallel with a constant named response that fetches data from the url and throws an error if no information is collected. Finally, each function has a constant labeled json that collects and returns the information from the site if found. At the bottom of this file, there is an export that exports all of the main functions to allow them to be utilized in other files. Other uses for these functions can include using the search bar on the home page, using the restaurant listing information on the restaurant page, and using the menu selection and allergy tailoring of the menu / allergy page. These API keys are soon to be hidden with environmental variables for safety purposes.

Database Control:

The database of PureDine is designed based on the main firebase_config.js file that allows for connection to the firebase database and allows specifically for the initialization, hosting, and authentication of the project. These constraints are set up as objects which are used throughout the files. Hosting can be done in the back end near the end of the project and can be updated regularly through the terminal. Authentication will use functions such as SignIn, SignOut, and useAuthentication to allow for interaction with objects such as the login and logout buttons.

1.2.1 Major Components

Major subsystems present in PureDine that correlate to the functional requirements of the software present in the Requirements Specification document can be described in the following:

- From 1.3.1.1-1.3.1.4, The UI subsystem is used to render the main webpage and interact with the React systems done specifically by the App.js file.
- At 1.3.1.5, the Database Control system will deal with Firebase in its own firebase config file that will handle specific hosting and authentication.
- From 1.3.1.6-1.3.1.15, the UI subsystem will be used for header design, and website design specifically of the main homepage and the about page, allowing for reloading and access of the two pages thanks to the 'react-router-dom' extension allowing page switching between the App.js and About.js files.
- From 1.3.1.16-1.3.1.24, the API control subsystem will be used to pull information on restaurant data, nutrition information, and menu information from the three APIs in use (Google Places, MealMe, Edamam) and display them specifically in combination with the UI subsystem in the Restaurant.js and Menu.js files.
- From 1.3.1.25-1.3.1.26, the UI and Database control systems will allow for user login through authentication and specifically the SignIn, SignOut, and useAuthentication functions will be used to host and display changes to the webpage based on the Firebase status.
- From 1.3.2.1-1.3.2.6, the API control subsystem will be utilized to display certain information depending on the search bar and restaurant selected involving location data given and specific restaurants selected (Google places, MealMe). The website will also highlight different allergy recommendations and guidelines based upon what is given by the Edamam API.

- From 1.3.2.7-1.3.2.15, The Database Control systems will allow for hosting due to the pushing of files to Firebase through the use of the terminal and login due to the authentication features built into Firebase extensions.

1.2.2 Major Software Interactions

Major software interactions throughout the PureDine application can be described by the following:

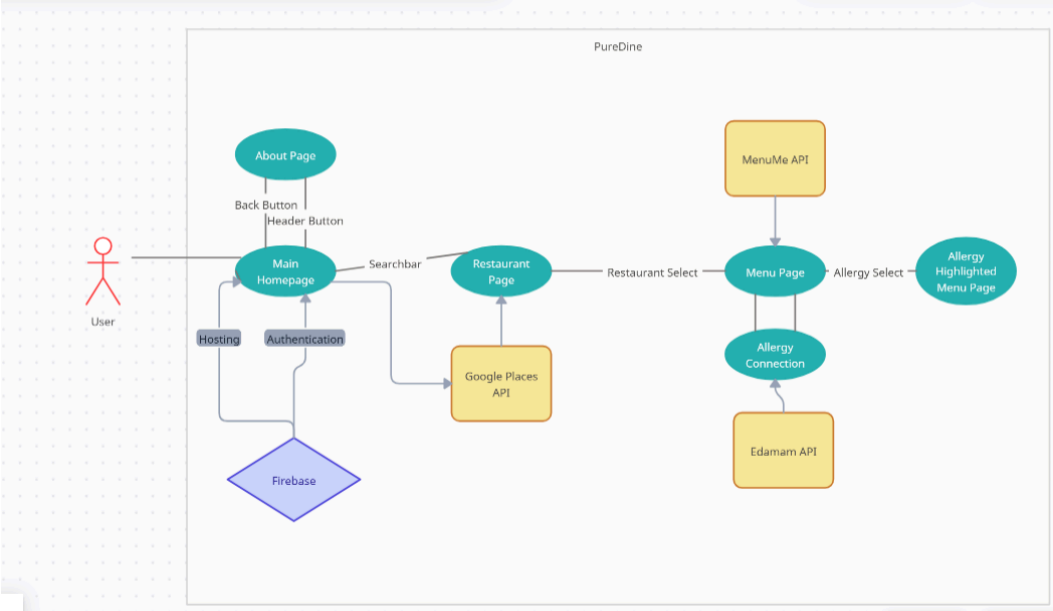
- The main header of the main page can be interacted with to not only reload the main page, but also link to the about page through the use of a link with the text “About the site”. The header will also contain the login button which will allow for a user to sign in or sign out of the site to adjust allergy preferences.
- The search bar of the main page, when given information based on a specific restaurant, city, or zipcode, will interact with the Google Places API to draw information on restaurants in a user's local area and list them specifically on the restaurants page, which the user will be linked to once selections are entered.
- Each restaurant on the display page can be clicked on to give access to the menu / allergy information given by the MealMe and Edamam APIs.
- The restaurants page will list up to 10 restaurants and will either have selections to another page for more listings or will allow for infinite scroll (still undecided).
- The user will have the option to interact with checkboxes on the menu page to select certain allergies or dietary restrictions that may affect them. This will lead to the menu

highlighting specific items with different categories of safety for consumption. These choices can also be made through the use of an account login.

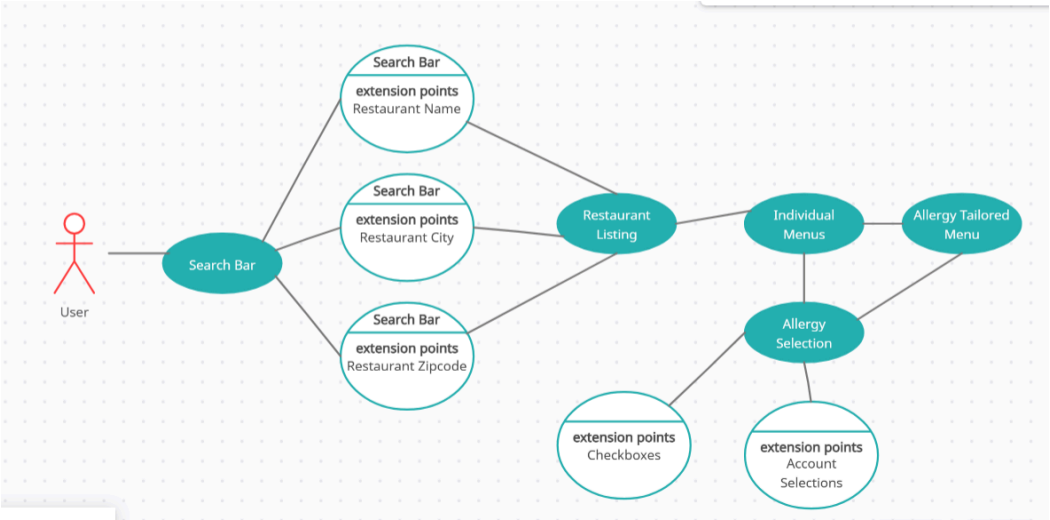
1.2.3 Architectural Design Diagrams

Here are some diagrams for PureDine applications at an architectural level:

Overall System



Restaurant / Menu Selection



Firebase Interactions