**Final Deliverable**

**Introduction:**

Team: CoffeeBreadSnek

Members: Jakob Hoveland, Morgan Du Bois, Anna Rowena Waldron

Project: Inventory and Recipe Finder for Grocery Store

**Business Problem:**

The problem that we are trying to solve is a recipe finder and inventory finder for a grocery store. Many stores have an app or a function on their website that allows customers to find where various groceries are located within the store. Few, however, also have the functionality of finding recipes for grocery items they already have or want to use. Many customers are not able to determine if a specific store has a necessary ingredient for a recipe for sale, or if that product is in stock. This leads to frustrated customers not being able to find what they want, or having to travel to multiple locations and wasting time. This may cause the business to lose valuable customers.

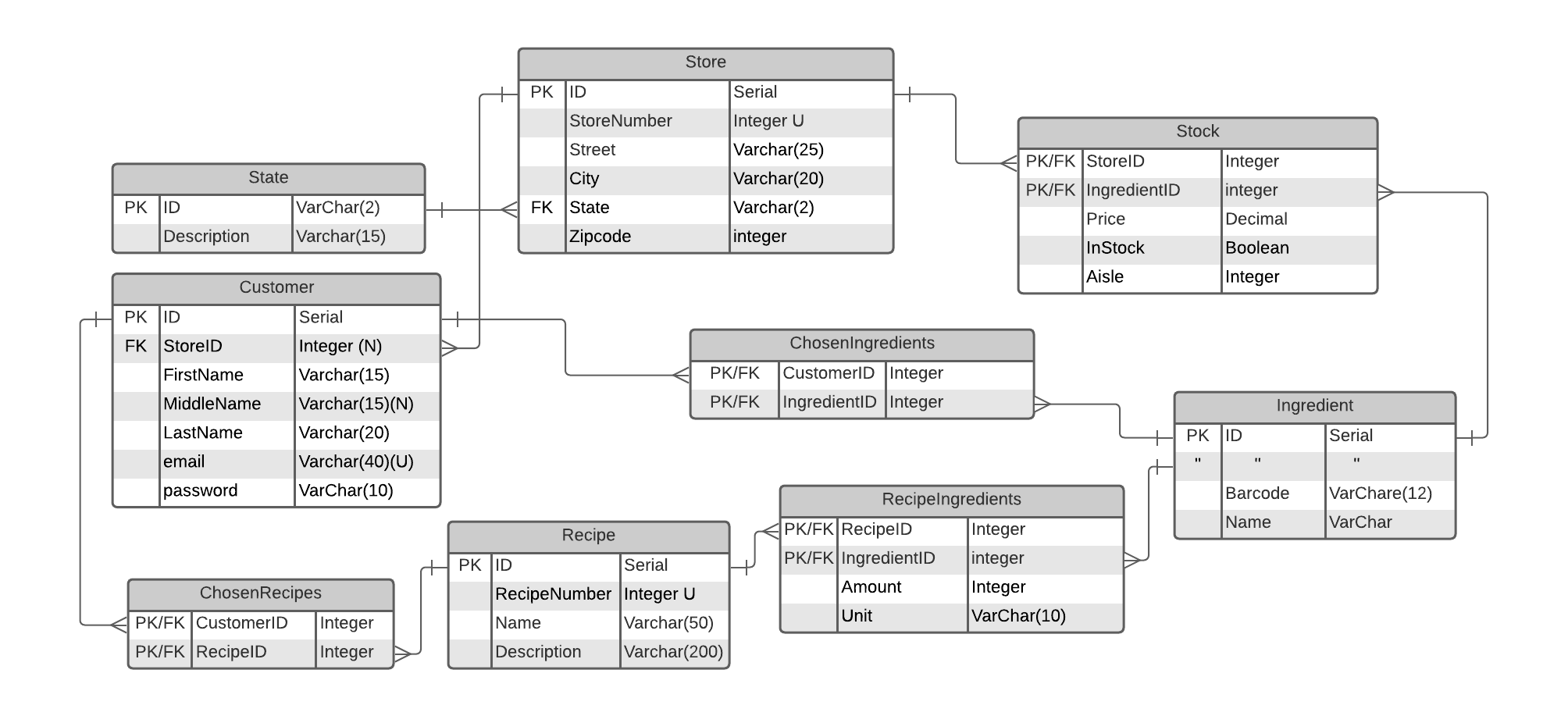
**High Level Interface:**

This app will be a tool for users and will store the users information after the app is closed; an account will be made for the users but no actual purchases will be made through the app. The app will provide a list of store locations within the given zip code given by the user, and if no store is within that zip code, it will return a list of all stores. It will be able to return a list of recipes that contain the food items the customer has chosen which are currently saved in the customer’s digital shopping cart. The customer may also select recipes instead of food items. Once a store has been selected, a list of items in the recipe selected is returned to the user, and information about those stock items (price, stock, location, etc.).

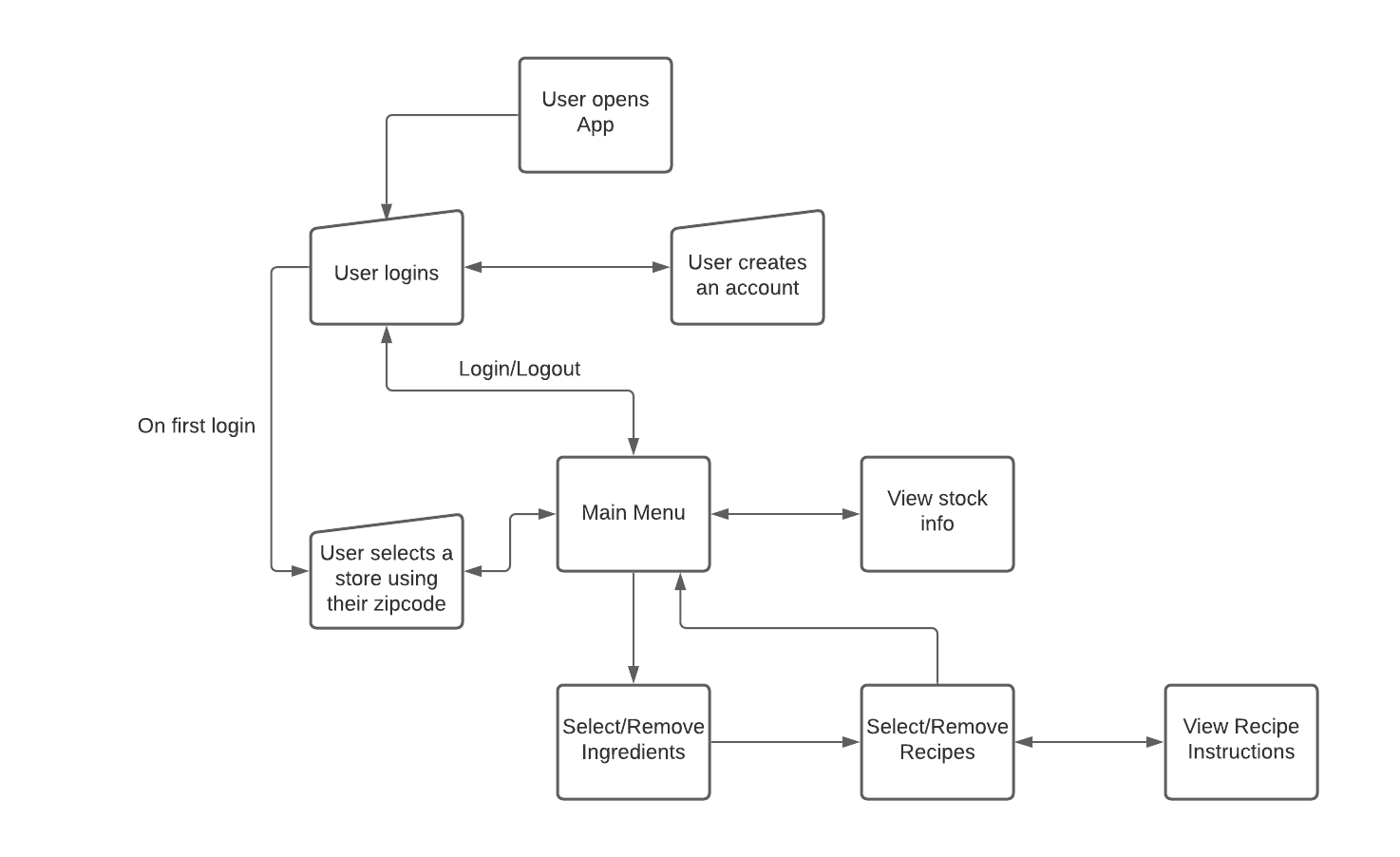
**User Interactions:**

* First, the customer will login or create an account with their name, email and password.
* Then the customer Inputs a zip code and will be shown a list of store addresses within that zip code to choose from. They can remove or add stores.
* the customer inputs the ingredients they want to use in a recipe. Ingredients are first typed and a list is shown with potential options of that ingredient from a list of ingredients (e.g. onion would have a drop down bar of red onion, yellow onion, walla walla onion, ect.). A customer can also instead choose to put the name of a recipe in the search bar and a similar list will be shown with similar or like recipes, in that way selecting recipes instead of individual ingredients.
  + The user will have a search bar that they fill in. As they do, their entry will be searched for in the relevant table, and as they modify their entry the new string will be re-searched to update the list of relevant items.
* The customer is provided a list of recipes using the ingredients already provided by the customer, or the customer is provided a list of recipes based on their recipe selection from earlier. If the customer does not choose any ingredients then they are able to select recipes from the entire list of recipes. They are shown the recipes they have selected, and the recipes they have selected that were filtered by the ingredients they have chosen.
* The customer can view information on the recipe by inputting the recipes number, showing them all the information about the recipe and ingredients it uses.
* Then they return to main menu where they can press the stock button to review information about the ingredients in their recipes, such as price and store aisle location.
* The customer can change their store selection to see if another store has the products that they are searching for by inputting a different zip code, and then selecting a different store
* The customer can logout and have their recipes and stores saved.

**Schema Design**



**User Interface Design**

****

**Refer to flow chart**

1. The user opens the application and is prompted to login or create an account. If the user chooses to login with a previously created account, they type in their email and password and login. Otherwise the user can create an account with their first name, middle name as optional, last name, email, and password. After creating their account they are prompted to login.
2. Next The user then inputs a zip code. A list of stores that are within that zip code is returned to the screen for the user to choose from. If there are no stores within the searched zip code, a message indicating that there are no stores in that location will be returned and a list of all stores will be given to the user. The input of zip code is searched against Store.zipcode for any matches, which are then returned.
3. From the returned lists of stores, the user can select a store. The list the user is given is of stores within that zip code along with information on where that store is located. The user selects a store by typing in the store number and adding it to their saved store. Only one store may be saved at a time. When a user selects a store, Customer.StoreID is updated with the Store.ID from the chosen table.
4. If the user has a store already saved they are taken to the main menu with options including select ingredients, manage recipes, view stock information, and quit the program.
5. The user is then prompted to select ingredients (or do so by pressing the ingredients button) by typing the names of the ingredients they have or want to be used in a recipe (e.g. onion would have a drop down bar of red onion, yellow onion, walla walla onion, ect.). If the user wants to view all ingredients in the database, they may search using no input to view all ingredients. The user can add or remove multiple ingredients but only one ingredient is searched for at a time. As every ingredient is added, the ChosenIngredients table is updated with a new entry. These ingredients are used to find corresponding recipes.
6. The user can then search for recipes. This is accomplished by finding all entries in the Recipe Table that have ingredients that match the ChosenIngredients table. The user can add or remove recipes here to their selected recipes from the shown recipes. The selected recipes are added to ChosenRecipes.
7. the user can also use the recipe search bar where the user can search for a specific recipe by name, or continue to add recipes after they have added recipes based upon their ingredient search. They may also view all recipes in the database by searching with no input to view all recipes. When the user selects a recipe, the ChosenRecipe table is updated with a new entry linking it to the chosen Recipe table entry.
8. The user can view information on a recipe by inputing the recipe number into the get info search bar. Once a recipe is selected, the ingredient list for that recipe is displayed, along with the selected store's information on those ingredients. Based on the value stored in Customer.StoreID, a table is returned containing the information from the Ingredient table, joined onto the stock table with ingredientID, where Stock.StoreID is equal to Customer.StoreID.

**Development Tools and Environment**

* *Database Software*
  + Postgress SQL
* *UI Tools*
  + Python
* *Database Hosting*
  + Postgress
* *Other Supporting Tools*
  + Tkinter
  + Psycopg2
  + Pycharm/Visual Studio
  + GitHub
  + Trello