SEG2105[B]: Deliverable 4 Report

SEG 2105[B] - Software Engineering

Team:

Student Number	Name
300220683	Mghabghab, Jad
300279906	Aburamadan, Yasmeen
300235454	Naveen, Kevin
300238813	Lee, Joey

Table of Contents

Introduction:

Project Introduction

The SEG Mealer app is an application we developed through means of the IDE called Android Studios. Throughout the span of this course, we have been able to learn and integrate the many features of android studios to devise this application to its fullest. Features like the creation and management of intents, onClick listeners, and firebase authentication and integration were all learned. These, of course, are merely a trifling of the many aspects of Android Studios that were learned.

App Features

The app consists of a multitude of design features in of itself. First and foremost, there exists a client and cook registration and login. Moreover, administrators may also login for wider access to the app, such as suspending cooks for an indefinite amount of time or permanently by receiving complaints. Complaints, by themselves, are submitted by the client.

Clients:

- Can search for meals and submit requests for them as well as view those requests
- View the rating of the cook based on meals
- Can submit complaints about cooks to the administrator

Cook:

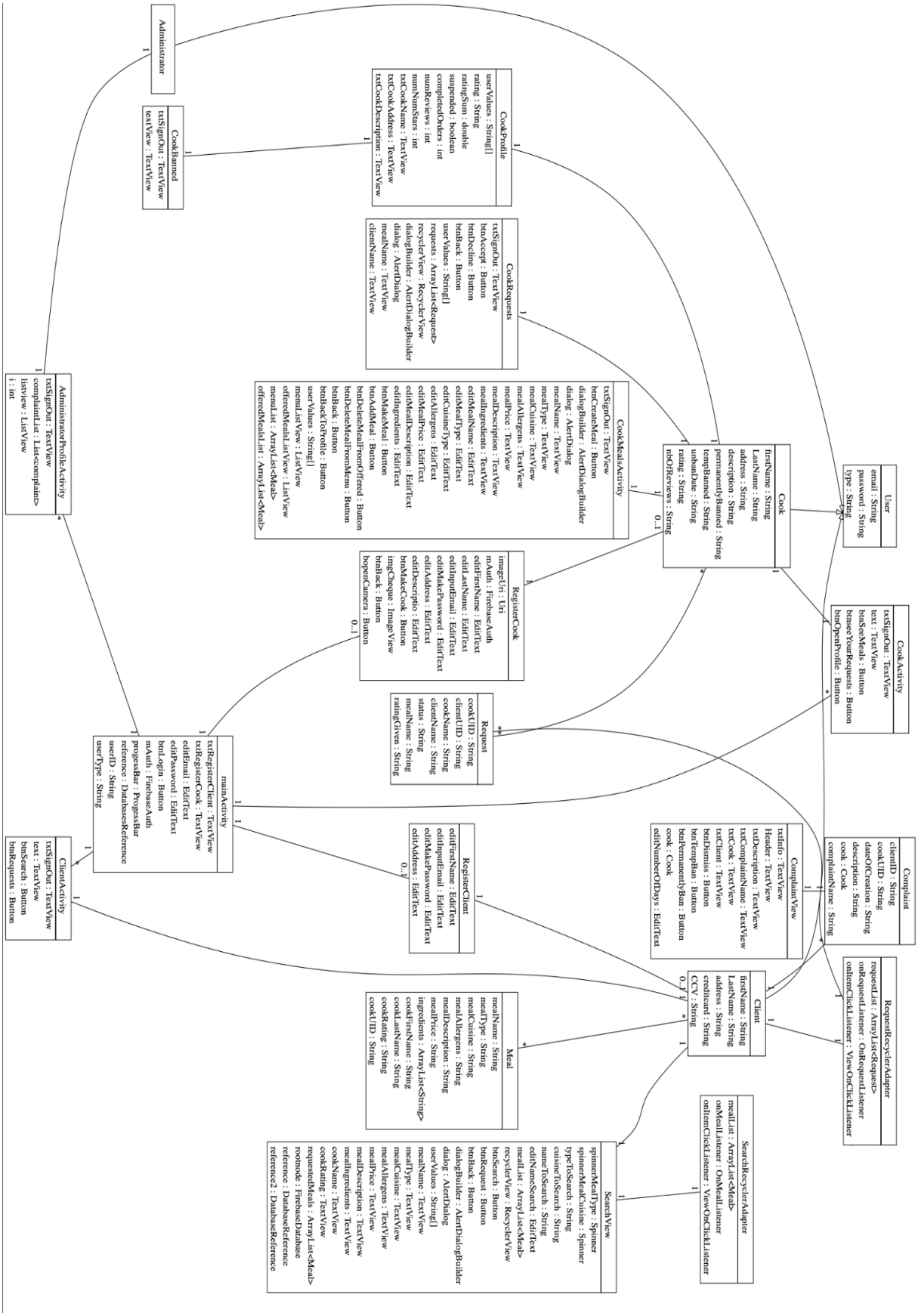
- Can create meals and add those meals to the menu
- Is able to add a meal to an offered meals list
- Cook can delete a meal
- Can view client meal requests and can decline them or accept them

All in all, the application should allow a functional interaction-style system between cook and client, where cooks can offer meals, and clients can find those meals and make requests on them.

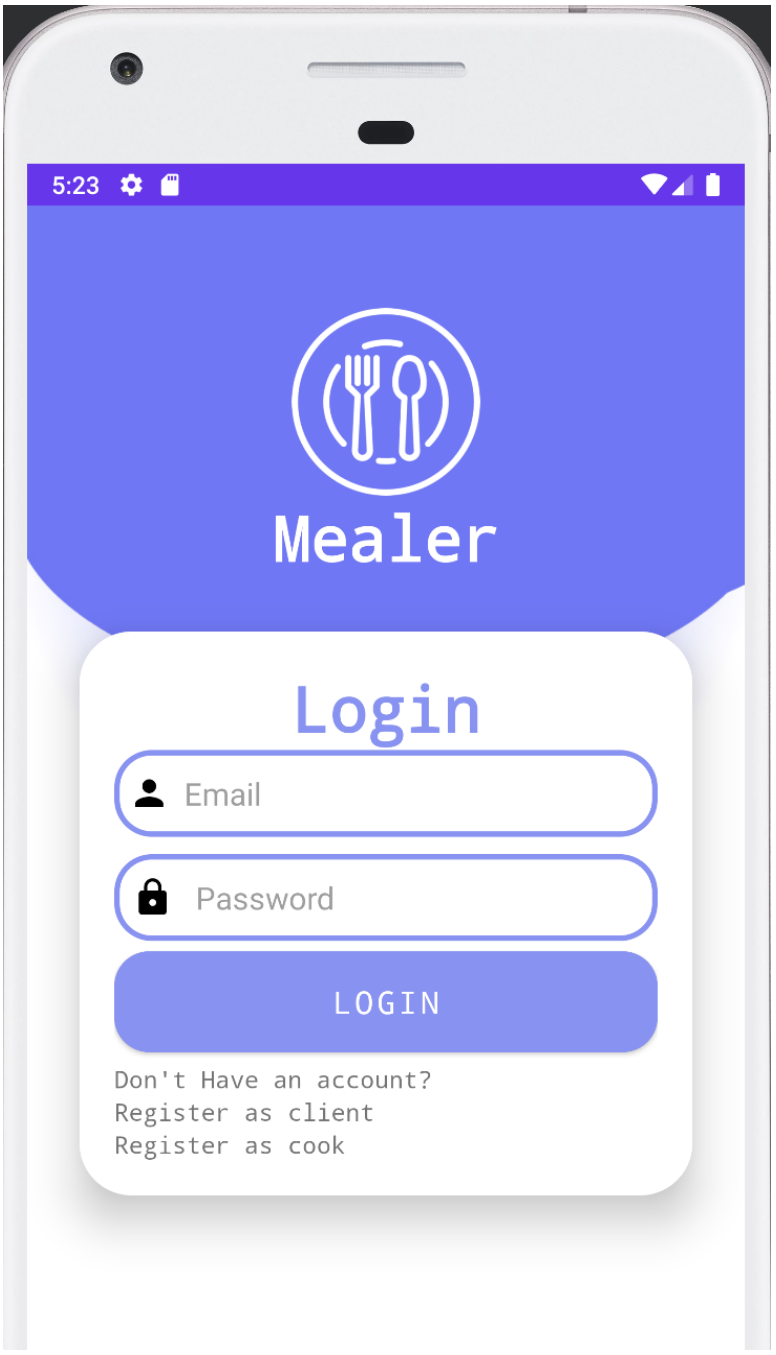
	Jad Mghabghab	Yasmeen Aburamadan	Joey Lee	Kevin Naveen
Deliverable 1	Implemented classes below: ● Client class ● Administrator class ● AdministratorProfile Activity class ● Client class ● CooktProfileActivity class ● ClientProfileActivity Class ● Cook class ● User class ● MainActivity class ● RegisterClient class ● RegisterCook class ● User class Implemented firebase for project.	Designed xml files for classes listed below: ● AdministratorProfile Activity ● ClientProfileActivity ● CookProfileActivity ● RegisterCook ● RegisterClient Added different custom drawables.	Implemented Activity client Profile class UML design for D1	Implemented classes below: ● Cook class ● CookProfileActivity class ● MainActivity class fixes ● RegisterCook class ● Activity_cook_profile class
Deliverable 2	Edited classes below: ● Client class ● Cook class ● Administrator class ● User class ● CookProfileActivity class ● MainActivity class Edited Complaint class	Edited design for xml files for classes listed below: ● ComplaintView ● RegisterClient ● RegisterCook ● CookProfileActivity ● AdministratorProfile Activity class	Edited AdminProfile class UML design for D2	Implemented classes below: ● Complaint class ● ComplaintView class Implemented test cases: ● CookTest class
Deliverable 3	Implemented and edited classes below: CookProfileActivity class CookProfileBanned class ComplaintView class ClientProfileActivity class CookProfileBanned class	Edited design for xml files listed below: ● Activity_meal_offered ● Activity_cook_profile ● Activity_meal_not_offered ● Activity_meal_offere	Implemented and edited classes below: ● Meal class ● CookProfileActivity class Edited design for xml files classes listed below: ● CreateMeal ● MealNotOffered	Implemented and edited classes below: ● CookProfileActivity class ● Meal class Implemented different test cases for the CookProfileActivity class

		d UML design for D3	● MealOffered	
Deliverable 4		Implemented and edited classes below: ● CookActivity class ● CookBanned class ● CookMealsActivity class ● CookProfile class ● CookRequests class ● Meal class ● SearchView class Added drawables for edittext fields.	Implemented and edited classes below: ● RequestRecyclerAdapter ● SearchRecyclerAdapter Edited design for xml files classes listed below: ● Meal_in_recycler_view ● Request_in_recycler_view ● Activity_search_view ● Activity_meal_info ● Activity_clients_requests_history ● Activity_clients_requests ● activity_client	Implemented and designed/edited the following: - ClientRequestHistory - Old_requests.xml - Client_Requests - Requests - ClientActivity

UML

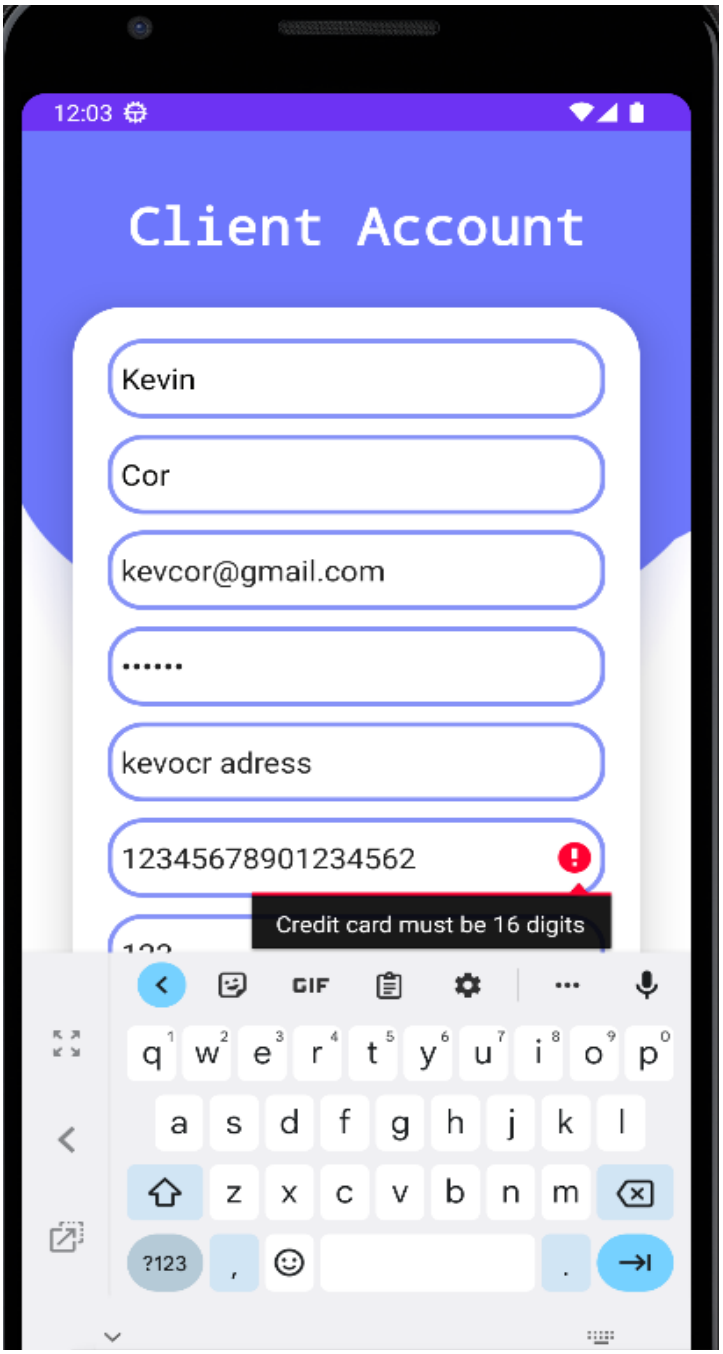

User
email : String
password : String
type : String
CookActivity
txtSignOut : TextView
text : TextView
btnSeeMeals : Button
btnseeYourRequests : Button
btnOpenProfile : Button
Complaint
clientID : String
cookUID : String
dateOfCreation : String
description : String
cook : Cook
complaintName : String
RequestRecyclerAdapter
requestList : ArrayList<Request>
onRequestListener : OnRequestListener
onItemClickListener : ViewOnClickListener
Cook
firstName : String
LastName : String
address : String
description : String
permanentlyBanned : String
tempBanned : String
unbanDate : String
rating : String
nbOfReviews : String
ComplaintView
txtInfo : TextView
Header : TextView
txtDescription : TextView
txtComplaintName : TextView
txtCook : TextView
txtClient : TextView
btnDismiss : Button
btnTempBan : Button
btnPermanentlyBan : Button
cook : Cook
editNumberOfDays : EditText
Client
firstName : String
LastName : String
address : String
creditcard : String
CCV : String
SearchRecyclerAdapter
mealList : ArrayList<Meal>
onMealListener : OnMealListener
onItemClickListener : ViewOnClickListener
SearchView
spinnerMealType : Spinner
spinnerMealCuisine : Spinner
typeToSearch : String
cuisineToSearch : String
nameToSearch : String
editNameSearch : EditText
mealList : ArrayList<Meal>
recyclerView : RecyclerView
btnSearch : Button
btnRequest : Button
btnBack : Button
dialogBuilder : AlertDialogBuilder
dialog : AlertDialog
userValues : String[]
mealName : TextView
mealType : TextView
mealCuisine : TextView
mealAllergens : TextView
mealPrice : TextView
mealIngredients : TextView
cookName : TextView
cookRating : TextView
requestedMeals : ArrayList<Meal>
rootnode : FirebaseDatabase
reference : DatabaseReference
reference2 : DatabaseReference
Meal
mealName : String
mealType : String
mealCuisine : String
mealAllergens : String
mealDescription : String
mealPrice : String
ingredients : ArrayList<String>
cookFirstName : String
cookLastName : String
cookRating : String
cookUID : String
Administrator
CookProfile
userValues : String[]
rating : String
ratingSum : double
suspended : boolean
completedOrders : int
numReviews : int
numNumStars : int
txtCookName : TextView
txtCookAddress : TextView
txtCookDescription : TextView
CookBanned
txtSignOut : TextView
textView : TextView
CookRequests
txtSignOut : TextView
btnAccept : Button
btnDecline : Button
btnBack : Button
userValues : String[]
requests : ArrayList<Request>
recyclerView : RecyclerView
dialogBuilder : AlertDialogBuilder
dialog : AlertDialog
mealName : TextView
clientName : TextView
CookMealsActivity
txtSignOut : TextView
btnCreateMeal : Button
dialogBuilder : AlertDialogBuilder
dialog : AlertDialog
mealName : TextView
mealType : TextView
mealCuisine : TextView
mealAllergens : TextView
mealPrice : TextView
mealDescription : TextView
mealIngredients : TextView
editMealName : EditText
editMealType : EditText
editCuisineType : EditText
editAllergens : EditText
editMealPrice : EditText
editMealDescription : EditText
editIngredients : EditText
btnMakeMeal : Button
btnAddMeal : Button
btnDeleteMealFromOffered : Button
btnDeleteMealFromMenu : Button
btnBack : Button
btnBackToProfile : Button
userValues : String[]
menuListView : ListView
offeredMealsListView : ListView
menuList : ArrayList<Meal>
offeredMealsList : ArrayList<Meal>
RegisterCook
imageUri : Uri
mAuth : FirebaseAuth
editFirstName : EditText
editLastName : EditText
editInputEmail : EditText
editMakePassword : EditText
editAddress : EditText
editDescriptio : EditText
btnMakeCook : Button
imgCheque : ImageView
btnBack : Button
bopenCamera : Button
Request
cookUID : String
clientUID : String
cookName : String
clientName : String
status : String
mealName : String
ratingGiven : String
AdministratorProfileActivity
txtSignOut : TextView
complaintList : List<complain>
listview : ListView
i : int
mainActivity
txtRegisterClient : TextView
txtRegisterCook : TextView
editEmail : EditText
editPassword : EditText
btnLogin : Button
mAuth : FirebaseAuth
progressBar : ProgressBar
reference : DatabaseReference
userID : String
userType : String
RegisterClient
editFirstName : EditText
editInputEmail : EditText
editMakePassword : EditText
editAddress : EditText
ClientActivity
txtSignOut : TextView
text : TextView
btnSearch : Button
btnRequests : Button

Screenshots:

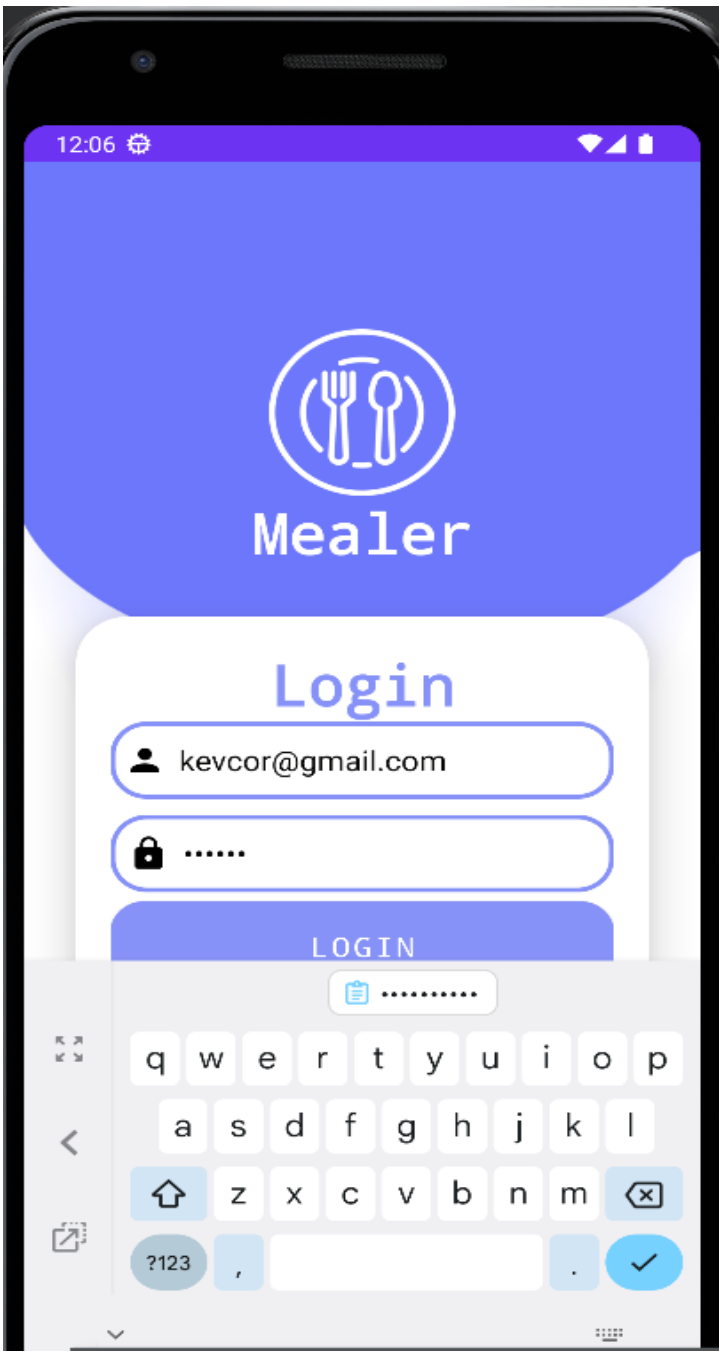


Main page for Mealer app

Client:

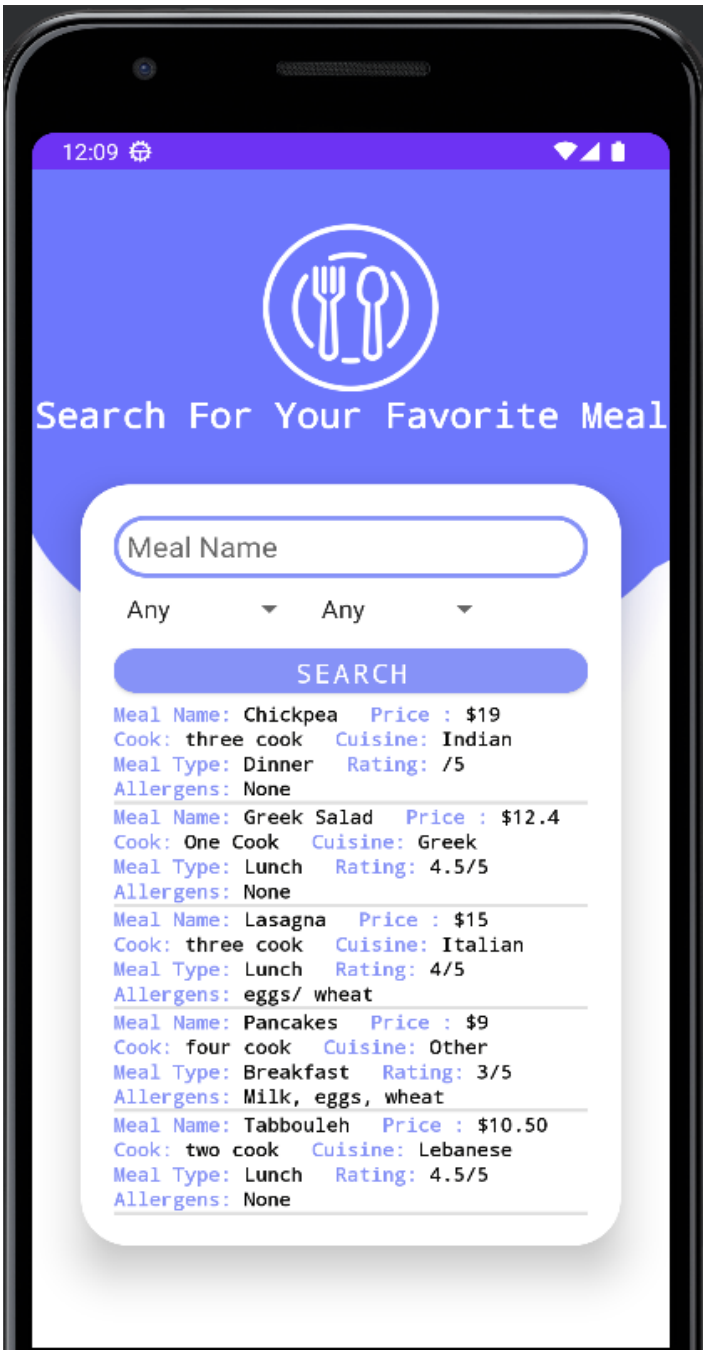


Attempting to sign up for a cook and inputting over 16 digits into the credit card results in an error pop-up.



After successful sign-up, login to the cook account now that it is stored in the database.

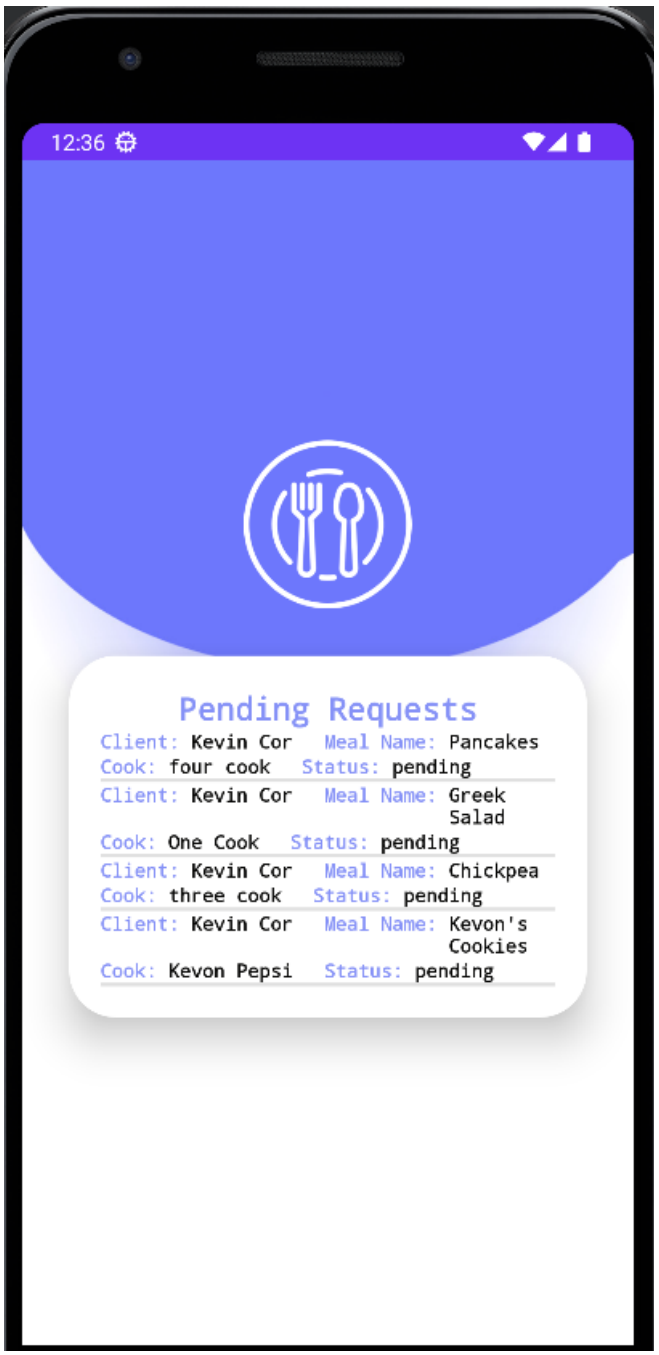
Home page of the client:



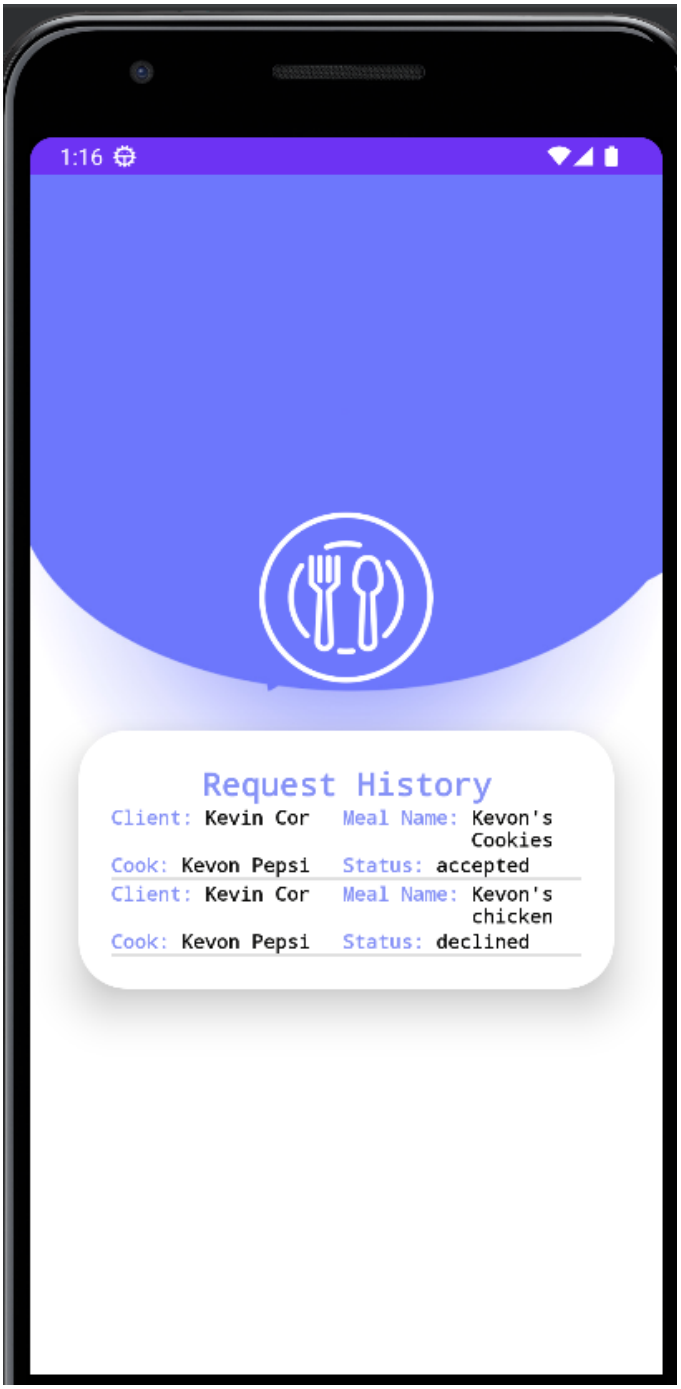
Search Page for Client

Clicking on a Meal opens the recycler view on the search page to request a meal

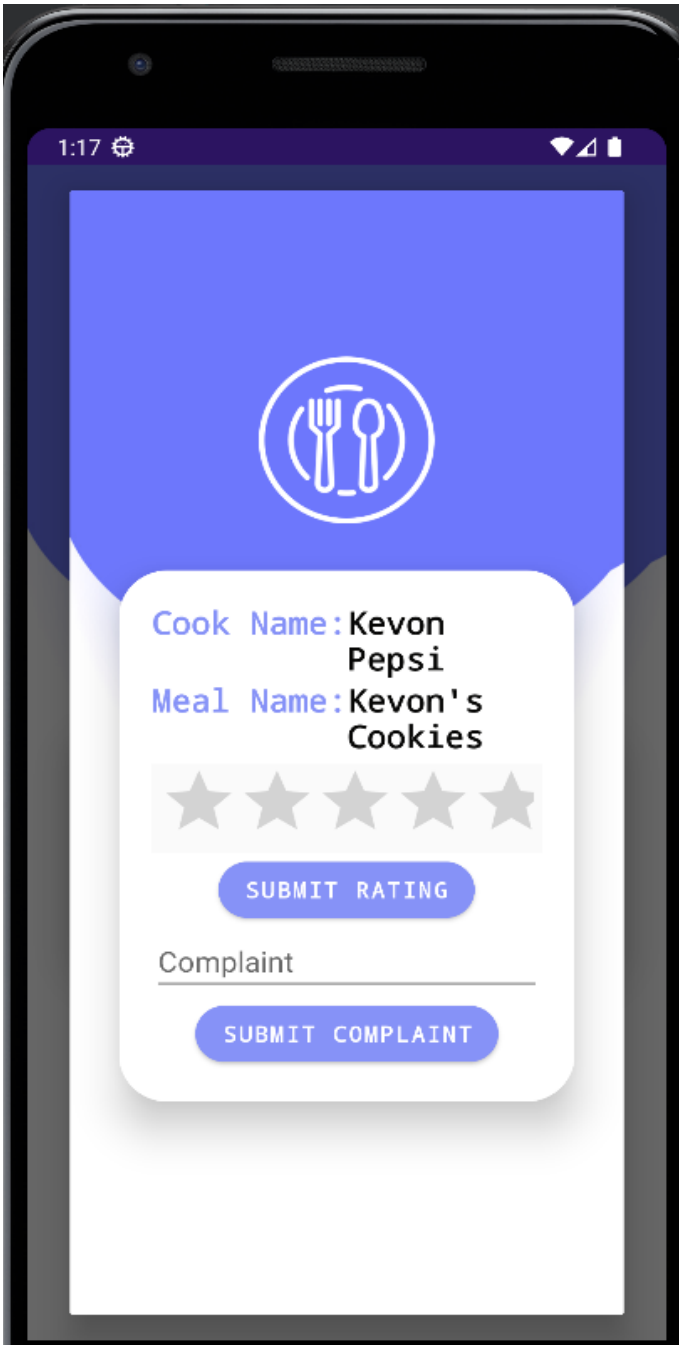
Requesting another meal



A view of the pending requests for meals from the client view



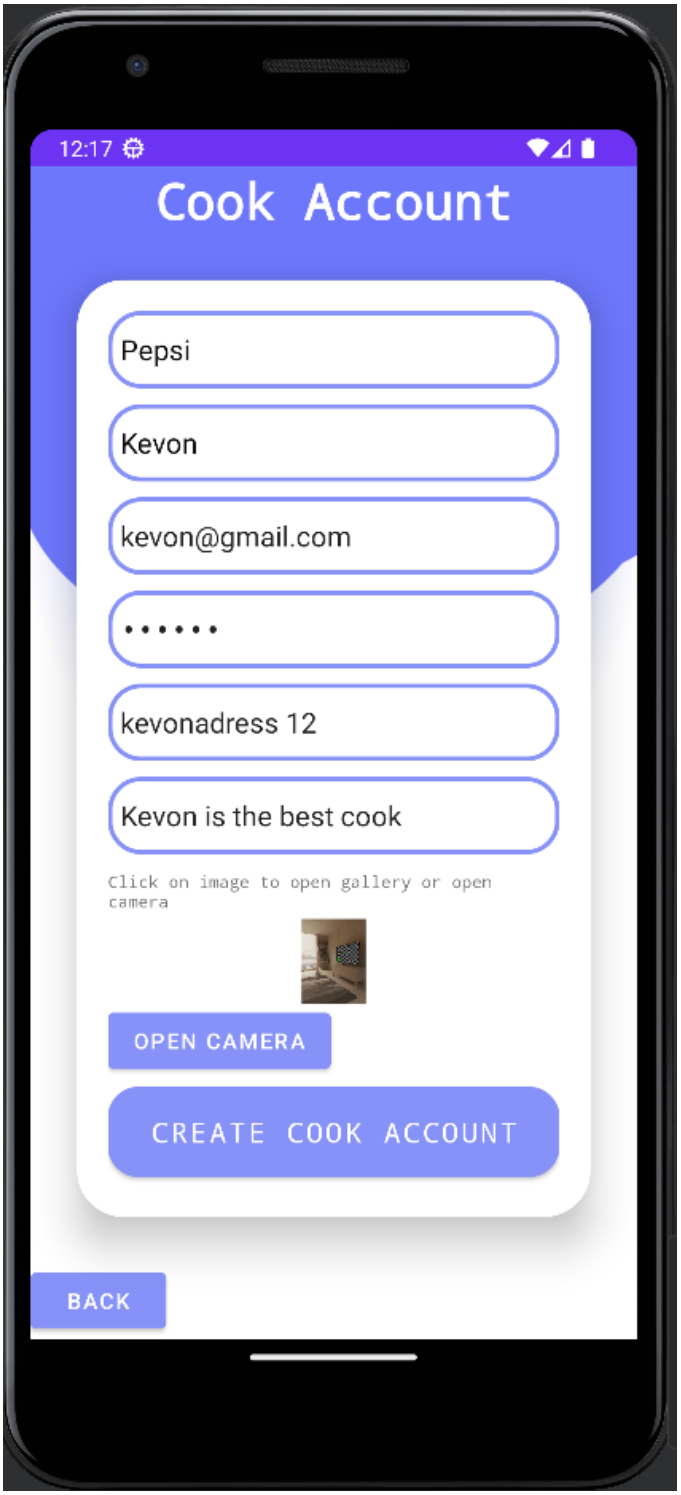
Request history allowing clients to view the status of their requests– in this instance, one meal request is accepted and the other is denied



Client complaint and rating activity where they can give a cook rating of 5 stars and submit complaints to the administrator

Submitting a complaint against the cook “Kevon Pepsi” based off meal “Kevin's Cookies”

Cook

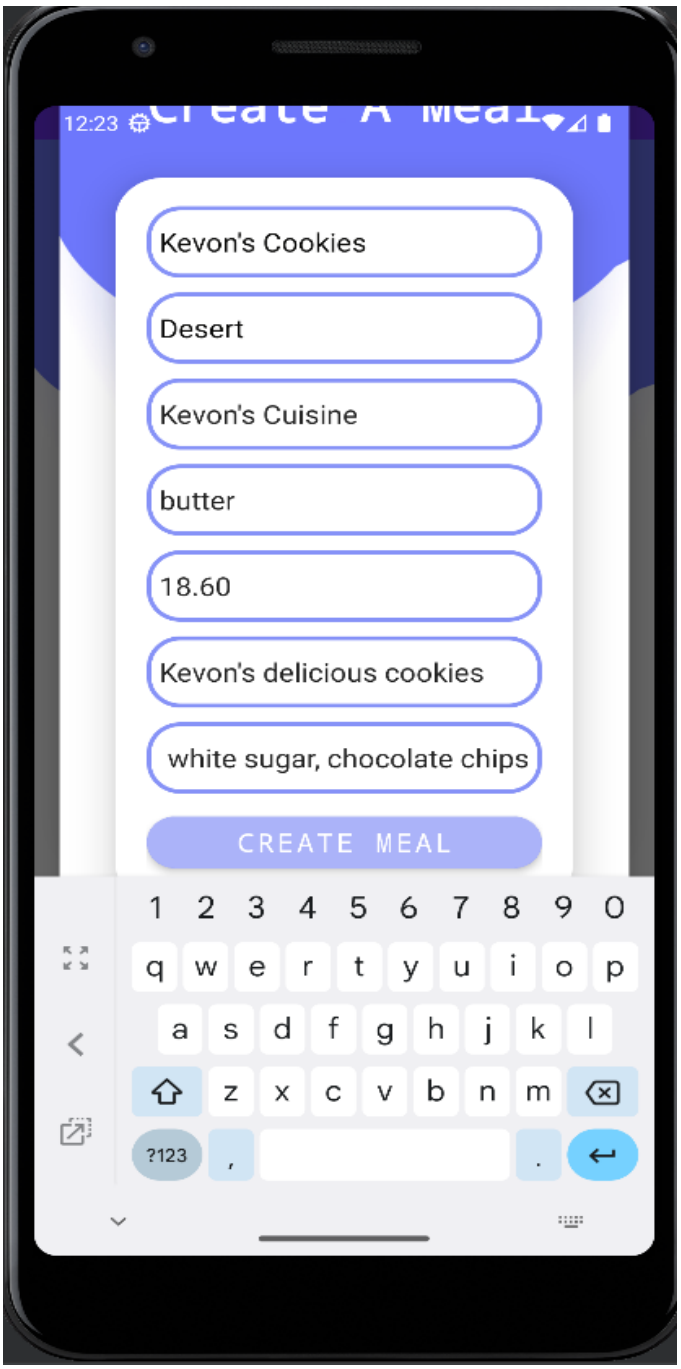


Cook Registration by inputting information and submitting a picture

Cook successful pop-up after a successful registration

Logging into the recently registered cook account

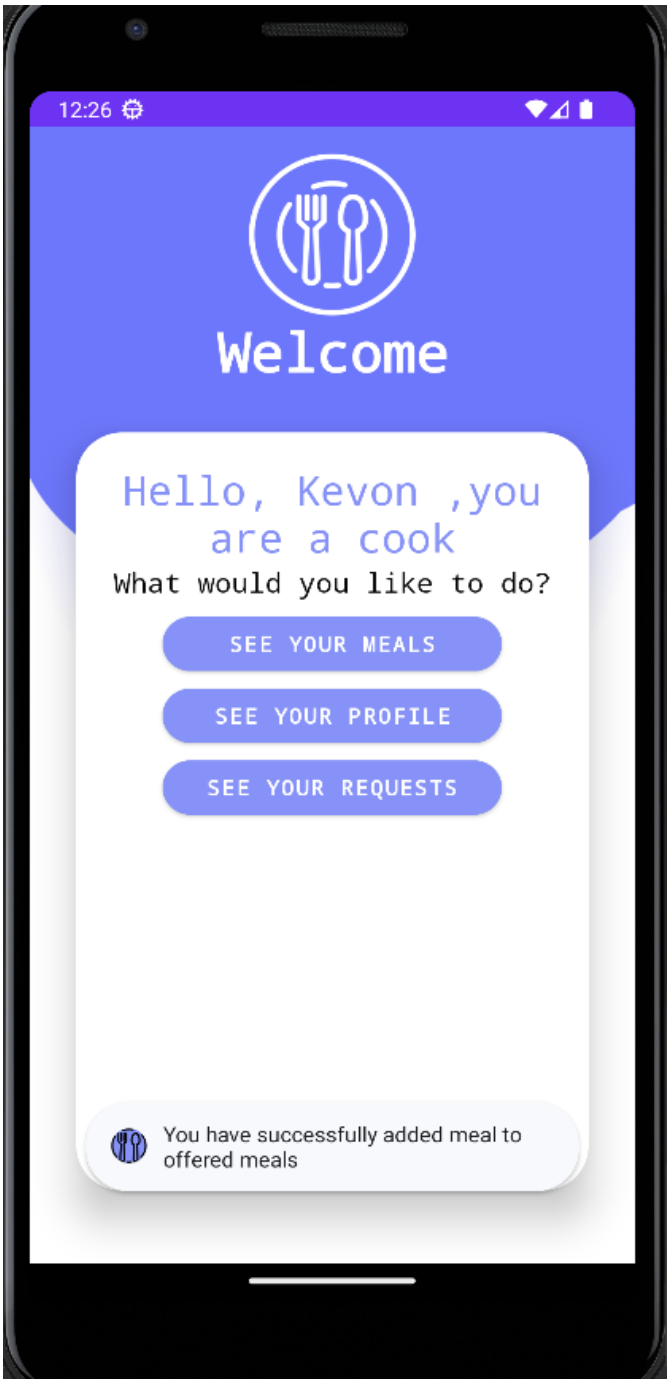
The view when you click “see your meals” from cook main activity, where the cook can see the menu and the offered menu (which are still empty because meals haven’t been created yet), and create meals.



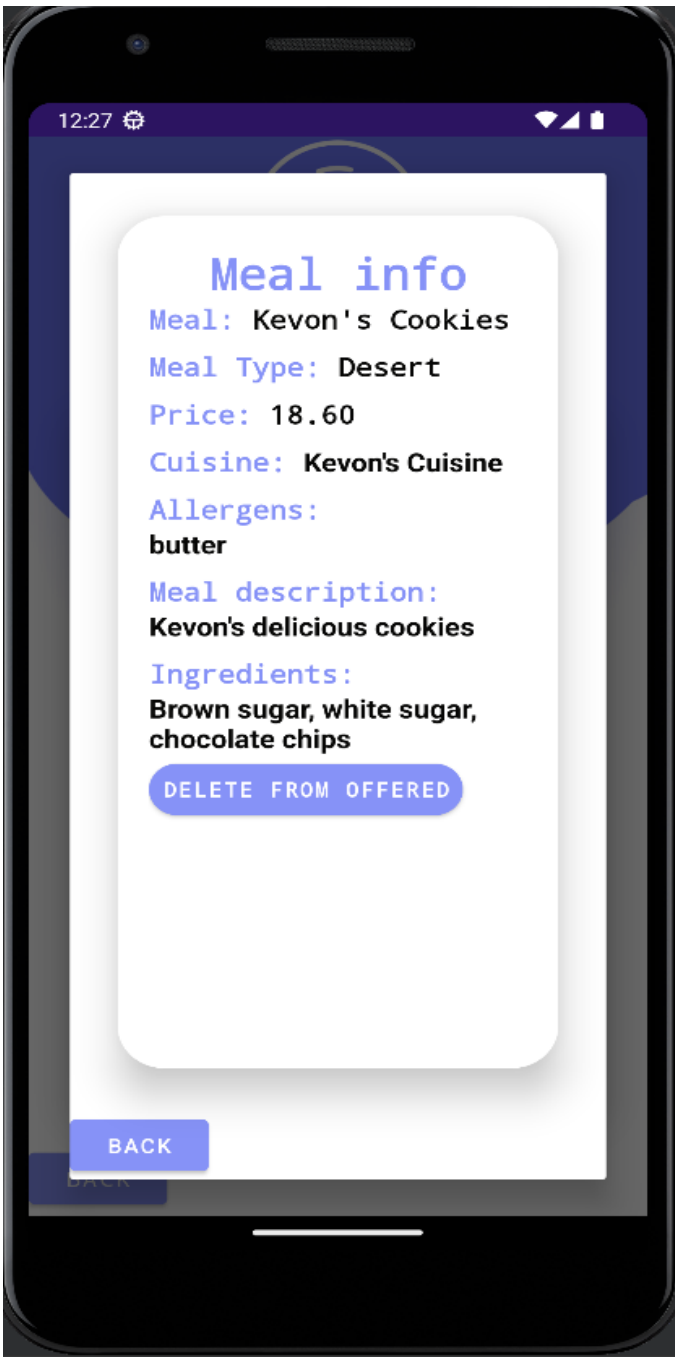
Process of creating a meal by cook by inputting all of the information

Altered meal view with a newly created meal that is on the menu but not the offered menu

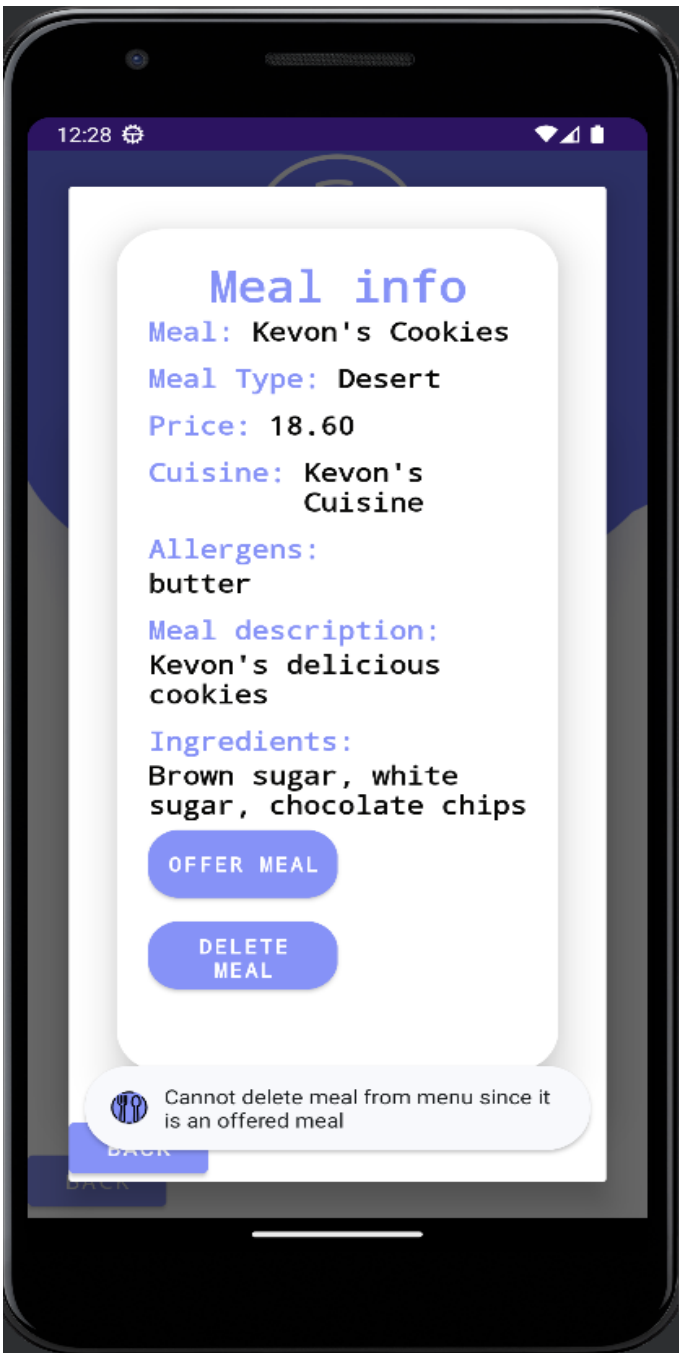
Clicking on the meal to open options to either put the meal on the offered menu



Successfully adding meals creates a popup

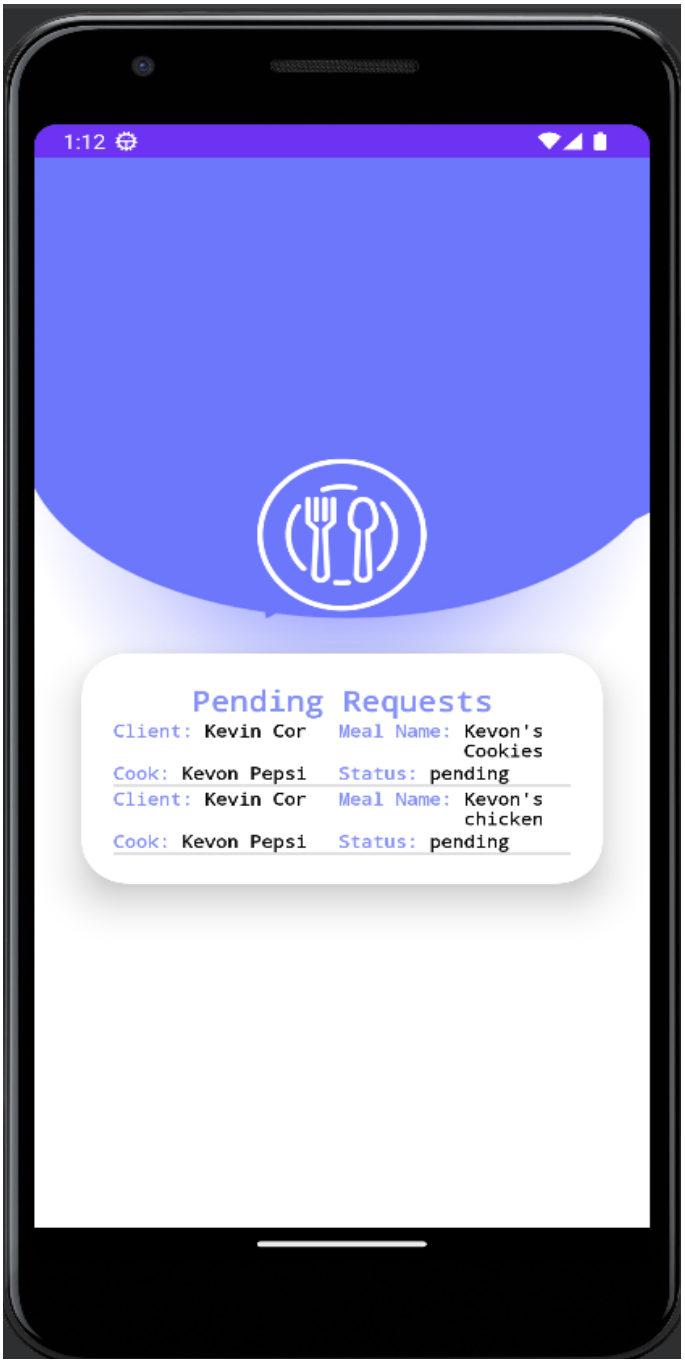


Clicking on an offered meal allows you to delete the meal from the offered menu

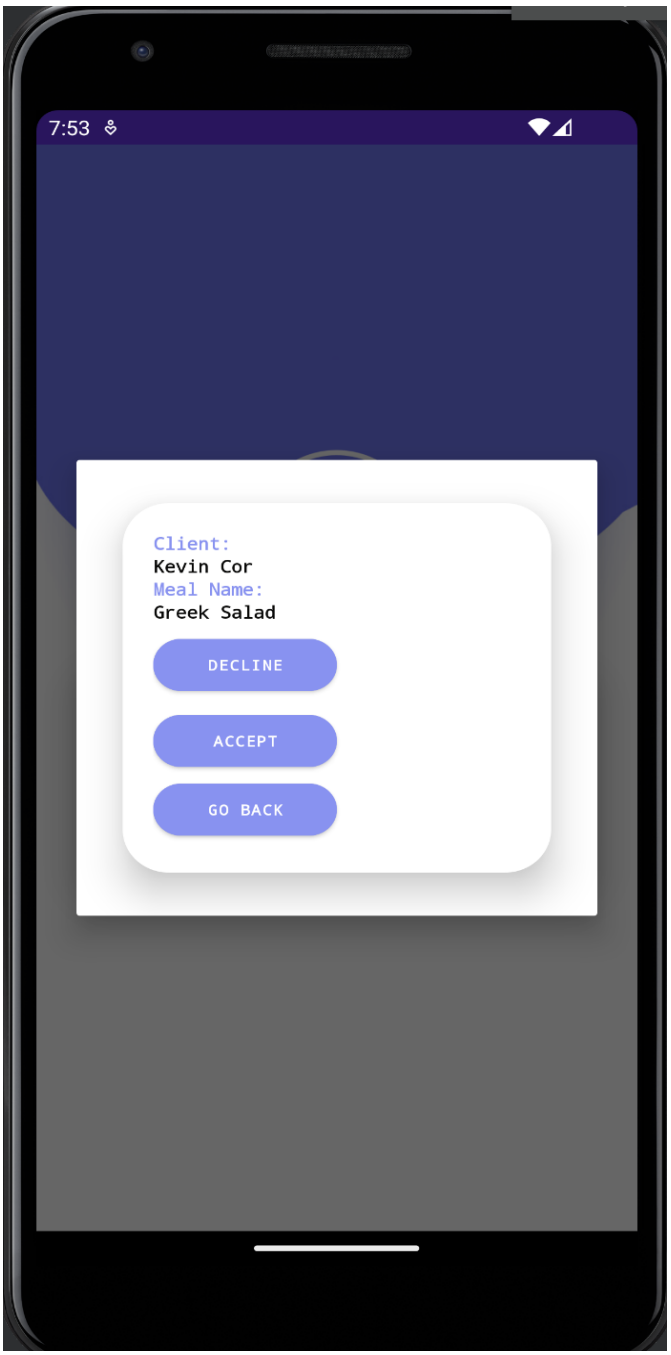


Deleting a meal from the menu when it is already on the offered meal list will result in a message dictating that the action cannot be performed

View displaying the menu and offered menu list



All meal requests for the specified cook sent from clients

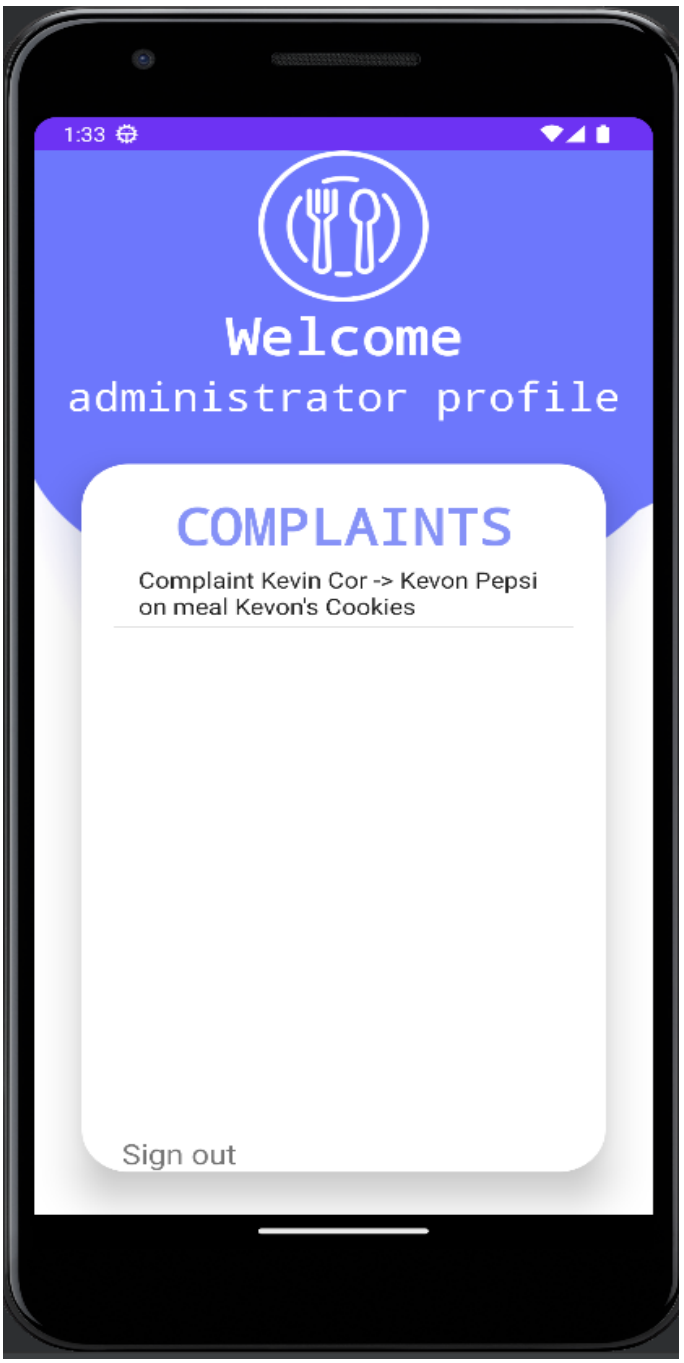


A cook clicking on a meal request will open up the options to either accept or decline the meal request

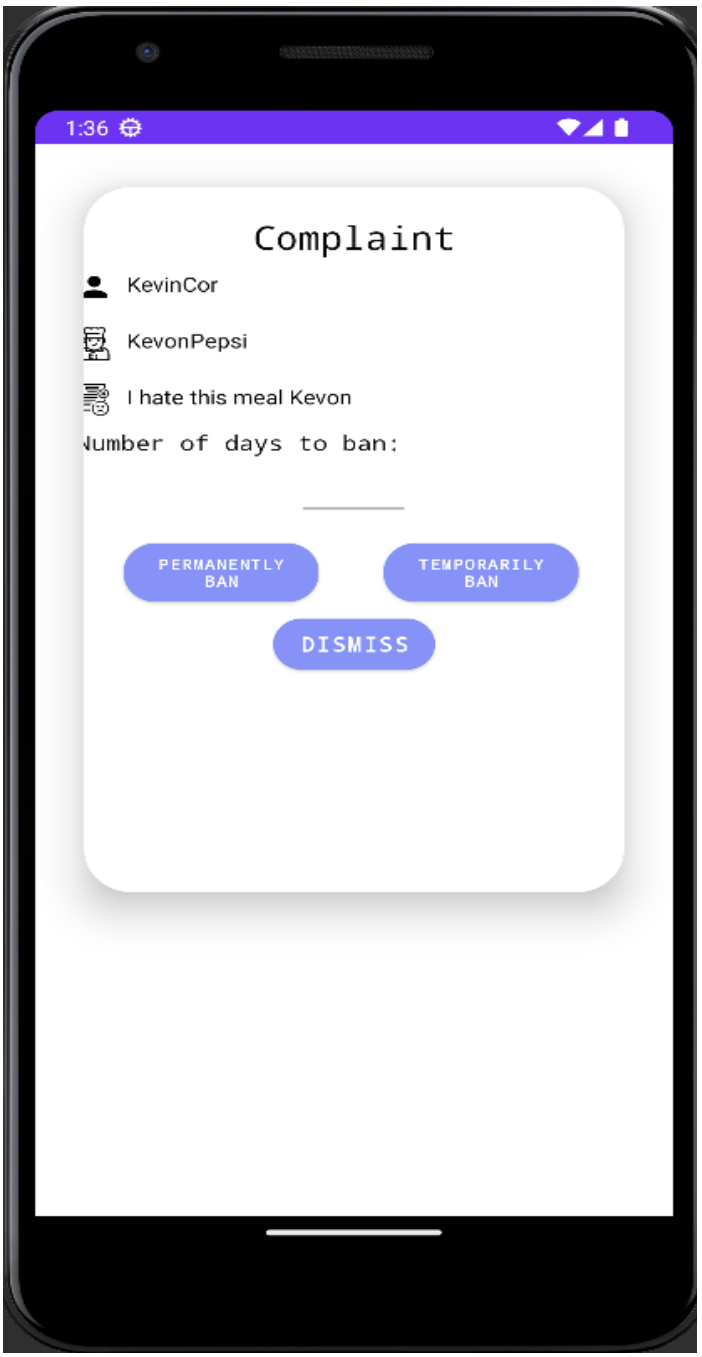
Ban cook alert upon logging in where the cook is unable to do anything but sign out

Administrator

Login to administrator account



View **complaints**



Options when the administrator faces a complaint, they can dismiss it, temporarily ban, or permanently ban the cook

Lessons Learned

Throughout the course of this project, a multitude of skills were acquired. First and foremost, each of us learned how to fully create a functioning application on android studio. Furthermore, we also learned what a firebase is, including how to create an instance and authenticate the firebase. Features like getting the reference to the database, and traversing the nodes by accessing the child elements.

Another critical aspect that was developed was how to use github and how it is useful when working on group projects. Learning to commit and pull has taught us the value of source control and its utility encompassing projects. Additionally, we honed and refined coding skills by devising xml files as well as java files to construct our application.

Most importantly, this project has taught us patience and the importance of debugging. For it is debugging where a vast majority of time has been dedicated towards. A program constructed from scratch is inevitably going to encounter issues predicated on minor and major issues within the code– but nothing a keen perception would not fix. The ability to patiently weather the immense time it takes to debug and fix issues is what truly constitutes a developer– an ability acquired from this project.