

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

Introductions -----	2
Contributions -----	3
UML diagram -----	5
Screenshots -----	6
Lessons Learned -----	32

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

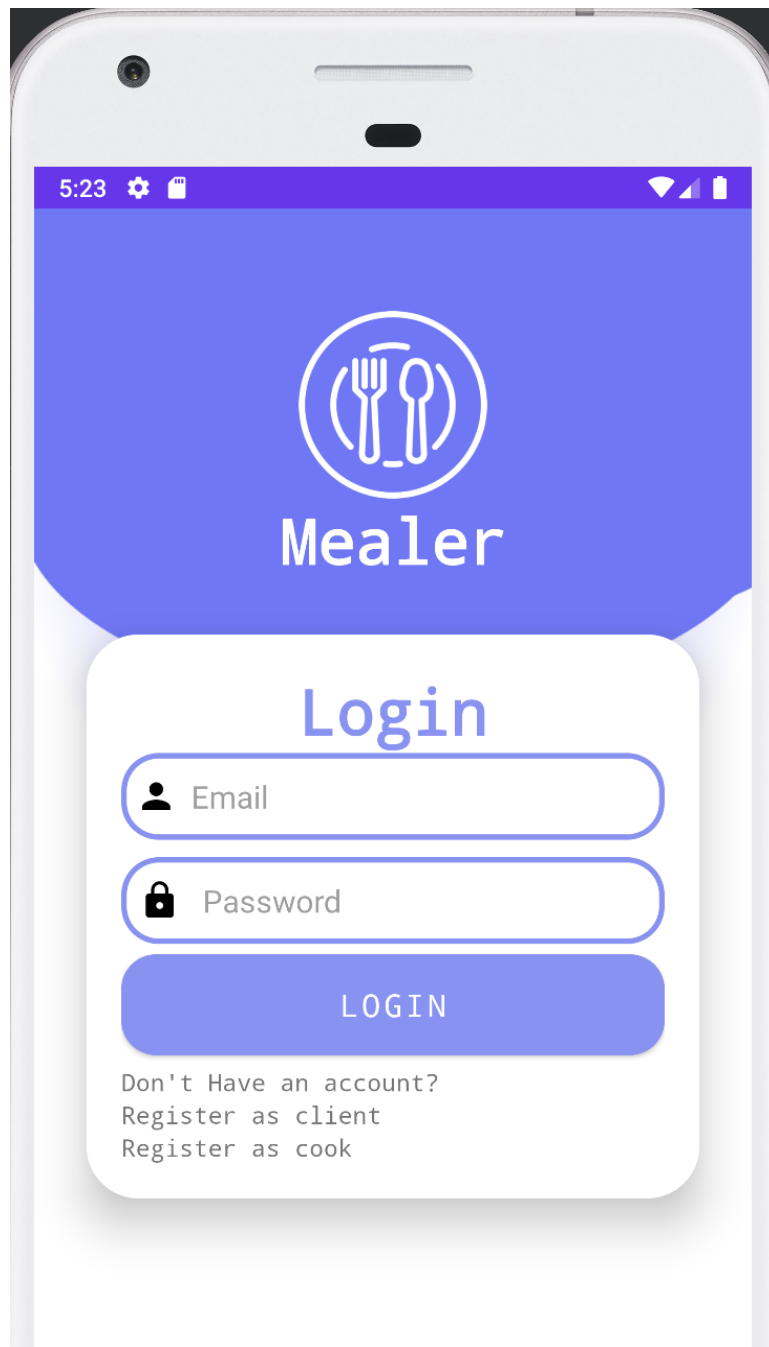
All in all, the application should allow a functional interaction-style system between cook and client, where cooks can offer meals, and clients can find those meals and make requests on them.

	Jad Mghabghab	Yasmeen Aburamadan	Joey Lee	Kevin Naveen
Deliverable 1	<p>Implemented classes below:</p> <ul style="list-style-type: none"> • 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 <p>Implemented firebase for project.</p>	<p>Designed xml files for classes listed below:</p> <ul style="list-style-type: none"> • AdministratorProfile Activity • ClientProfileActivity • CookProfileActivity • RegisterCook • RegisterClient <p>Added different custom drawables.</p>	<p>Implemented Activity client Profile class</p> <p>UML design for D1</p>	<p>Implemented classes below:</p> <ul style="list-style-type: none"> • Cook class • CookProfileActivity class • MainActivity class fixes • RegisterCook class • Activity_cook_profile class
Deliverable 2	<p>Edited classes below:</p> <ul style="list-style-type: none"> • Client class • Cook class • Administrator class • User class • CookProfileActivity class • MainActivity class <p>Edited Complaint class</p>	<p>Edited design for xml files for classes listed below:</p> <ul style="list-style-type: none"> • ComplaintView • RegisterClient • RegisterCook • CookProfileActivity • AdministratorProfile Activity class 	<p>Edited AdminProfile class</p> <p>UML design for D2</p>	<p>Implemented classes below:</p> <ul style="list-style-type: none"> • Complaint class • ComplaintView class <p>Implemented test cases:</p> <ul style="list-style-type: none"> • CookTest class
Deliverable 3	<p>Implemented and edited classes below:</p> <p>CookProfileActivity class</p> <p>CookProfileBanned class</p> <p>ComplaintView class</p> <p>ClientProfileActivity class</p> <p>CookProfileBanned class</p>	<p>Edited design for xml files listed below:</p> <ul style="list-style-type: none"> • Activity_meal_offered • Activity_cook_profile • Activity_meal_not_offered • Activity_meal_offered 	<p>Implemented and edited classes below:</p> <ul style="list-style-type: none"> • Meal class • CookProfileActivity class <p>Edited design for xml files classes listed below:</p> <ul style="list-style-type: none"> • CreateMeal • MealNotOffered 	<p>Implemented and edited classes below:</p> <ul style="list-style-type: none"> • CookProfileActivity class • Meal class <p>Implemented different test cases for the CookProfileActivity class</p>

		d UML design for D3	<ul style="list-style-type: none"> MealOffered 	
Deliverable 4	<p>Implemented and edited classes below:</p> <ul style="list-style-type: none"> Request class ClientRequests class CookMealsActivity class SearchView class RequestRecyclerAdapter class SearchRecyclerAdapter class MainActivity class <p>Fixed xml ids and debugged</p>	<p>Implemented and edited classes below:</p> <ul style="list-style-type: none"> CookActivity class CookBanned class CookMealsActivity class CookProfile class CookRequests class Meal class SearchView class <p>Added drawables for edittext fields. Final UML</p>	<p>Implemented and edited classes below:</p> <ul style="list-style-type: none"> RequestRecyclerAdapter SearchRecyclerAdapter <p>Edited design for xml files classes listed below:</p> <ul style="list-style-type: none"> Meal_in_recycler_view Request_in_recycler_view Activity_search_view Activity_meal_info Activity_clients_requests_history Activity_clients_requests activity_client 	<p>Implemented and designed/edited the following:</p> <ul style="list-style-type: none"> ClientRequestHistory Old_requests.xml Client_Requests Requests ClientActivity

[illegible]

Screenshots:



Main page for Mealer app

Client:

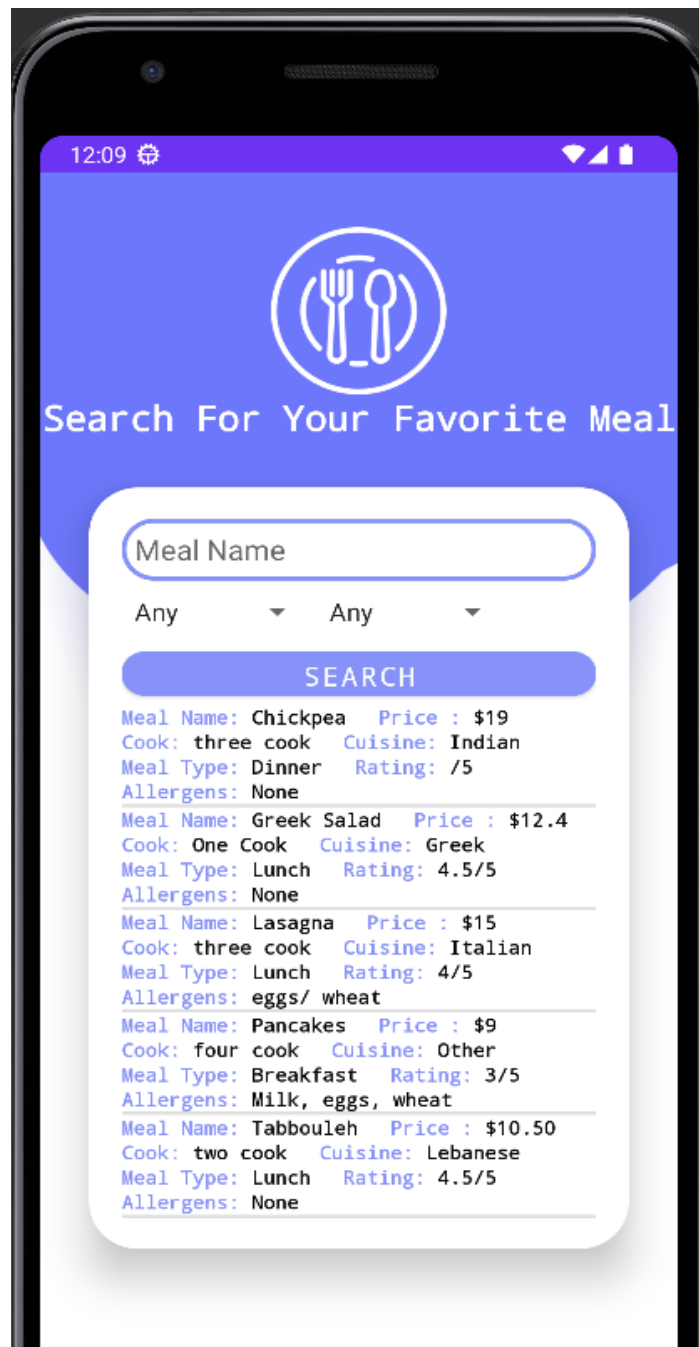
The image shows a mobile application interface for a 'Client Account'. The screen has a blue header with the title 'Client Account'. Below the header is a white rounded rectangle containing several input fields. The fields are labeled as follows: 'Kevin' (name), 'Cor' (company), 'kevcor@gmail.com' (email), '.....' (password), 'kevocr adress' (address), and '12345678901234562' (credit card). A red exclamation mark icon is visible next to the credit card field. Below the credit card field, a black error message box with white text reads 'Credit card must be 16 digits'. At the bottom of the screen, a standard Android keyboard is visible, showing the letters 'q', 'w', 'e', 'r', 't', 'y', 'u', 'i', 'o', 'p' on the top row, and 'a', 's', 'd', 'f', 'g', 'h', 'j', 'k', 'l' on the second row. The keyboard also includes a spacebar, a backspace key, and a search key.

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:



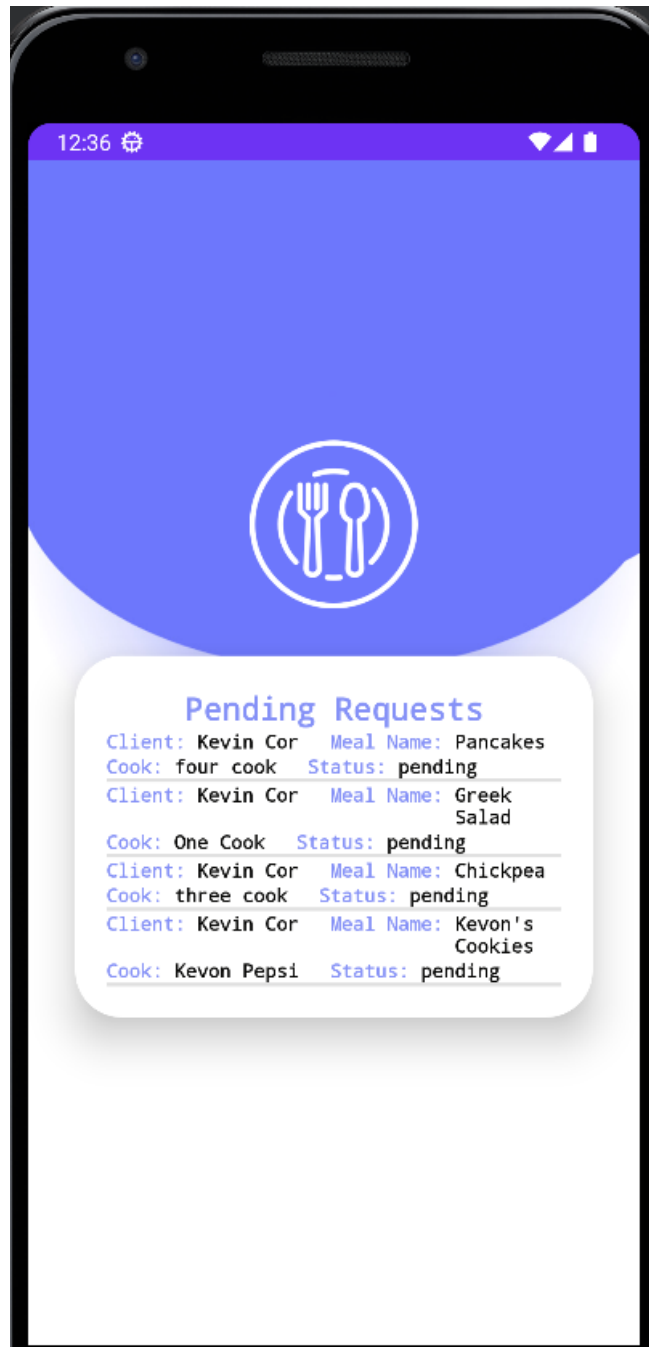
Search Page for Client



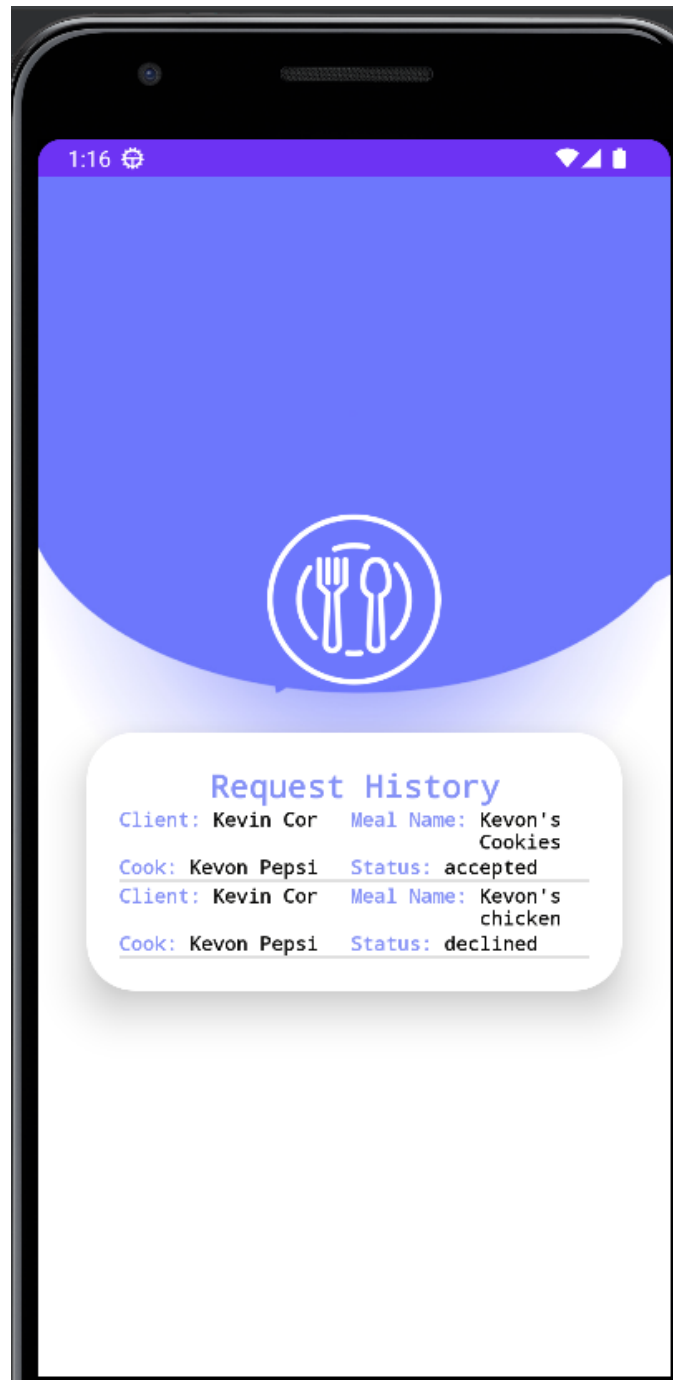
Clicking on a Meal opens the recycler view on the search page to request a meal



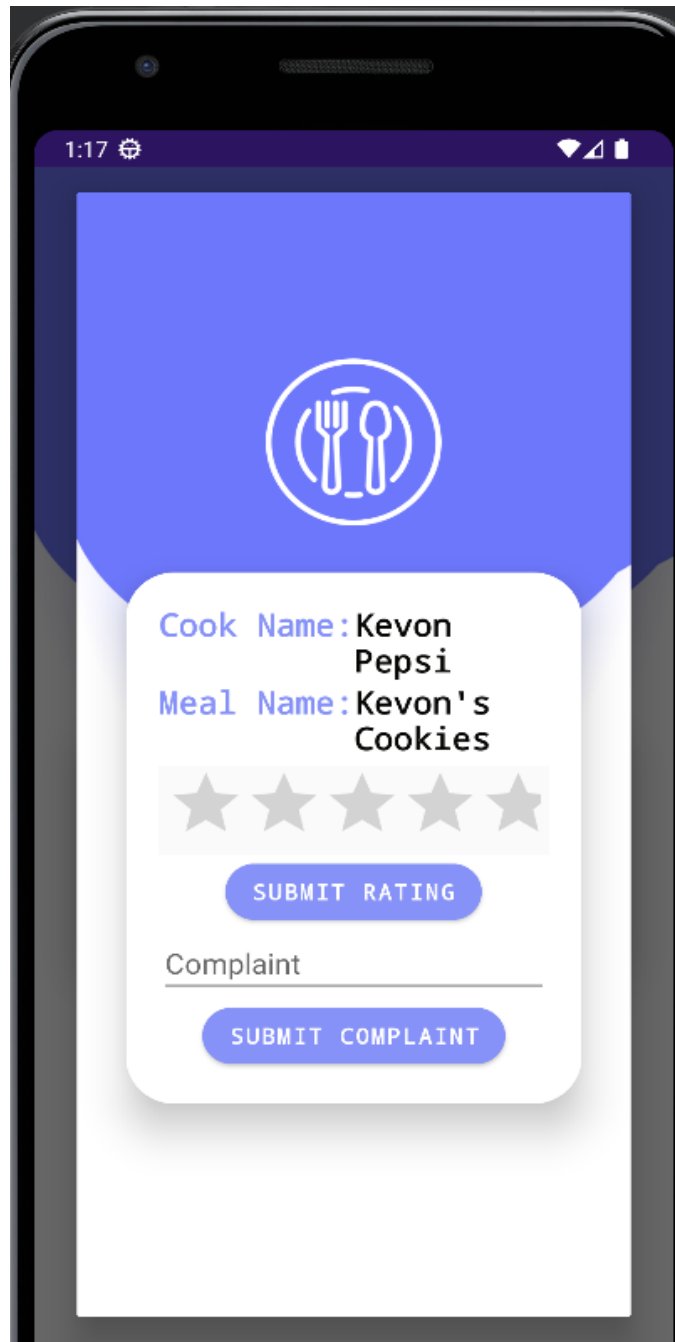
Requesting another meal



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

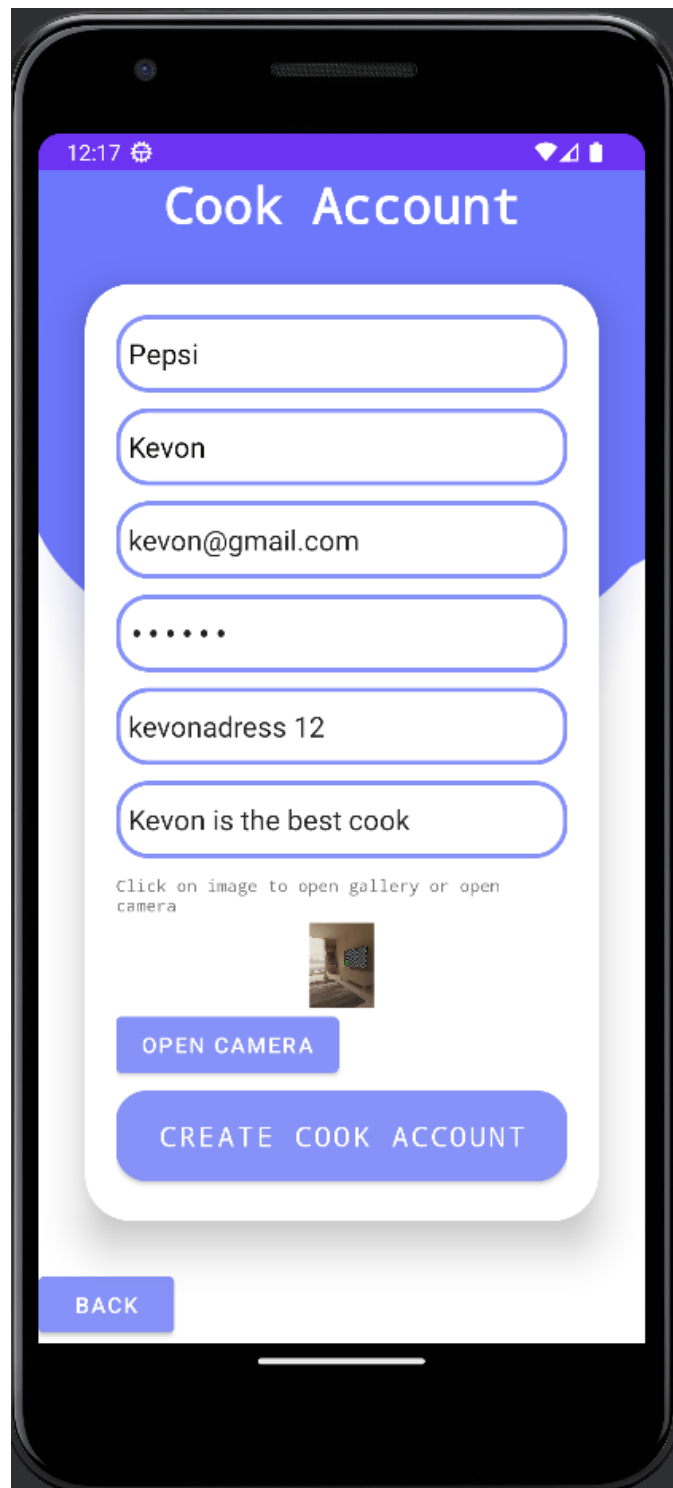


Client complaint and rating activity where they can give a cook rating of 5 stars and submit complaints to the administrator



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

Cook



The image shows a smartphone screen with a registration form titled "Cook Account". The form is set against a blue gradient background. It contains several input fields with the following text: "Pepsi", "Kevon", "kevon@gmail.com", a masked password ".....", "kevonadress 12", and "Kevon is the best cook". Below the text fields is a small image placeholder with the text "Click on image to open gallery or open camera". Underneath the image is a blue button labeled "OPEN CAMERA". At the bottom of the form is a large blue button labeled "CREATE COOK ACCOUNT". A "BACK" button is located at the bottom left of the screen.

12:17

Cook Account

Pepsi

Kevon

kevon@gmail.com

.....

kevonadress 12

Kevon is the best cook

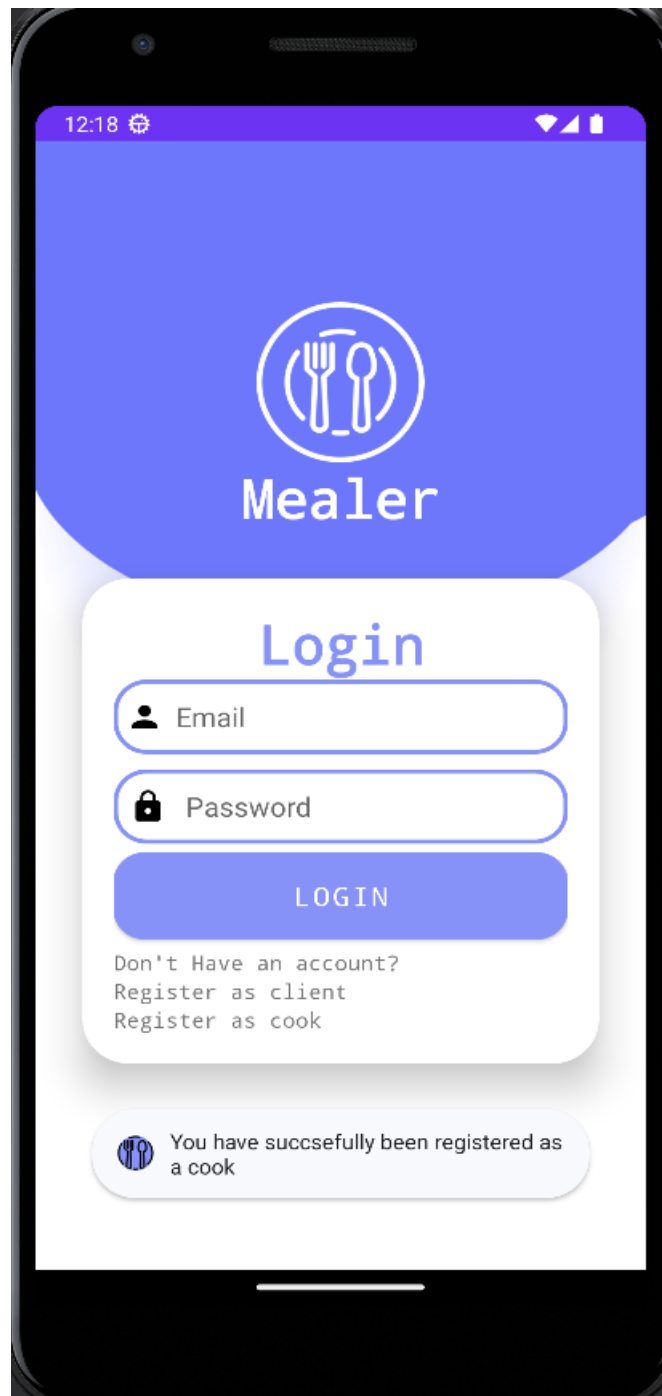
Click on image to open gallery or open camera

OPEN CAMERA

CREATE COOK ACCOUNT

BACK

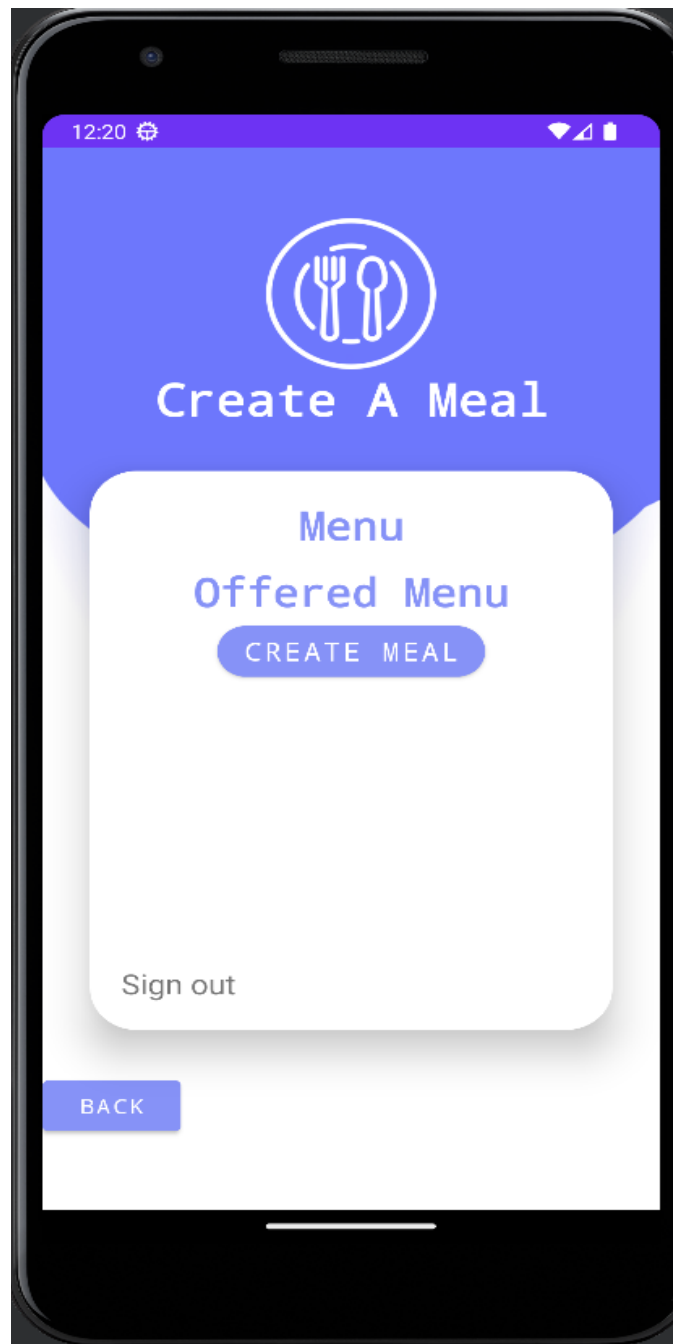
Cook Registration by inputting information and submitting a picture



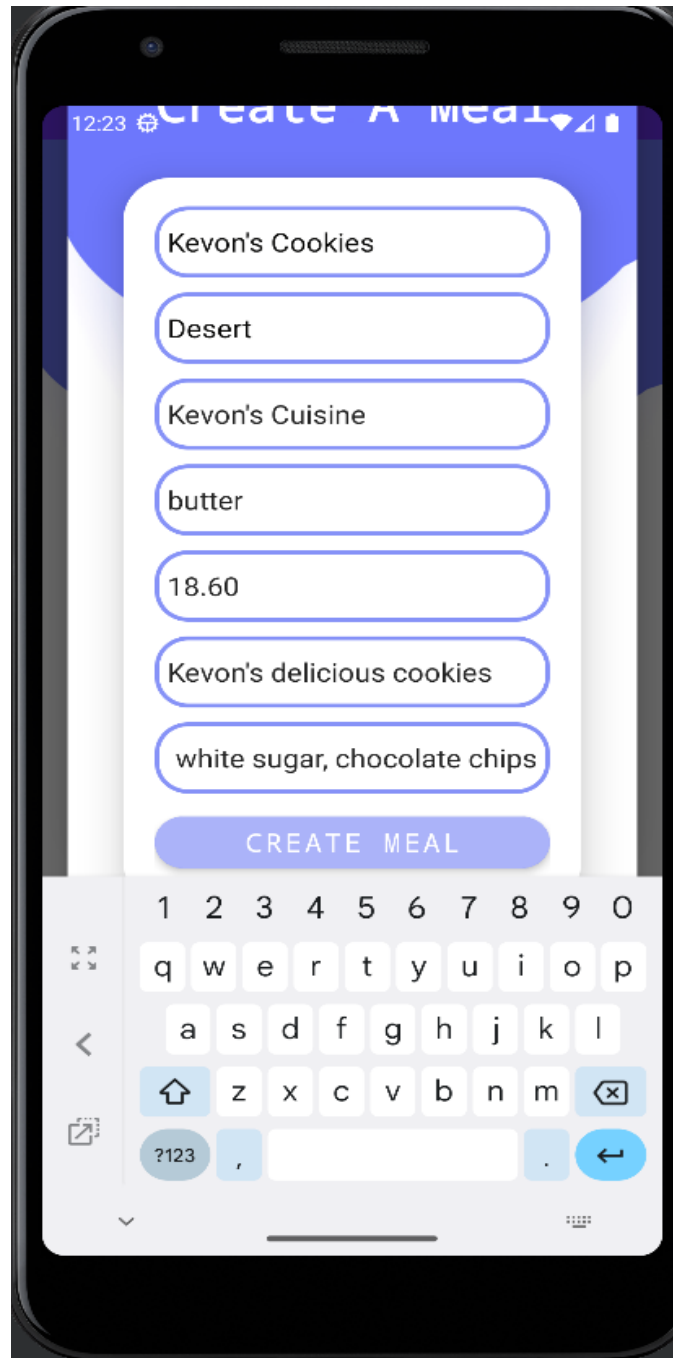
Cook successful pop-up after a successful registration



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



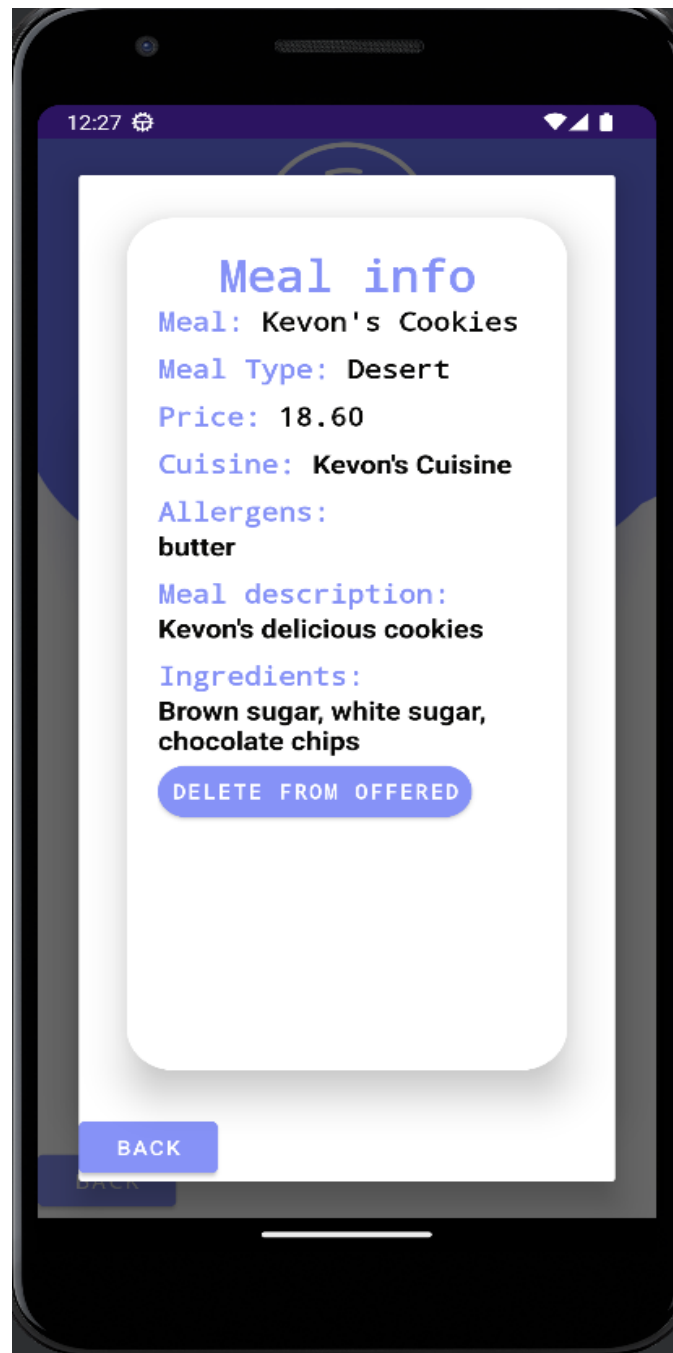
Altered meal view with a newly created meal that is on the menu but not the offered menu



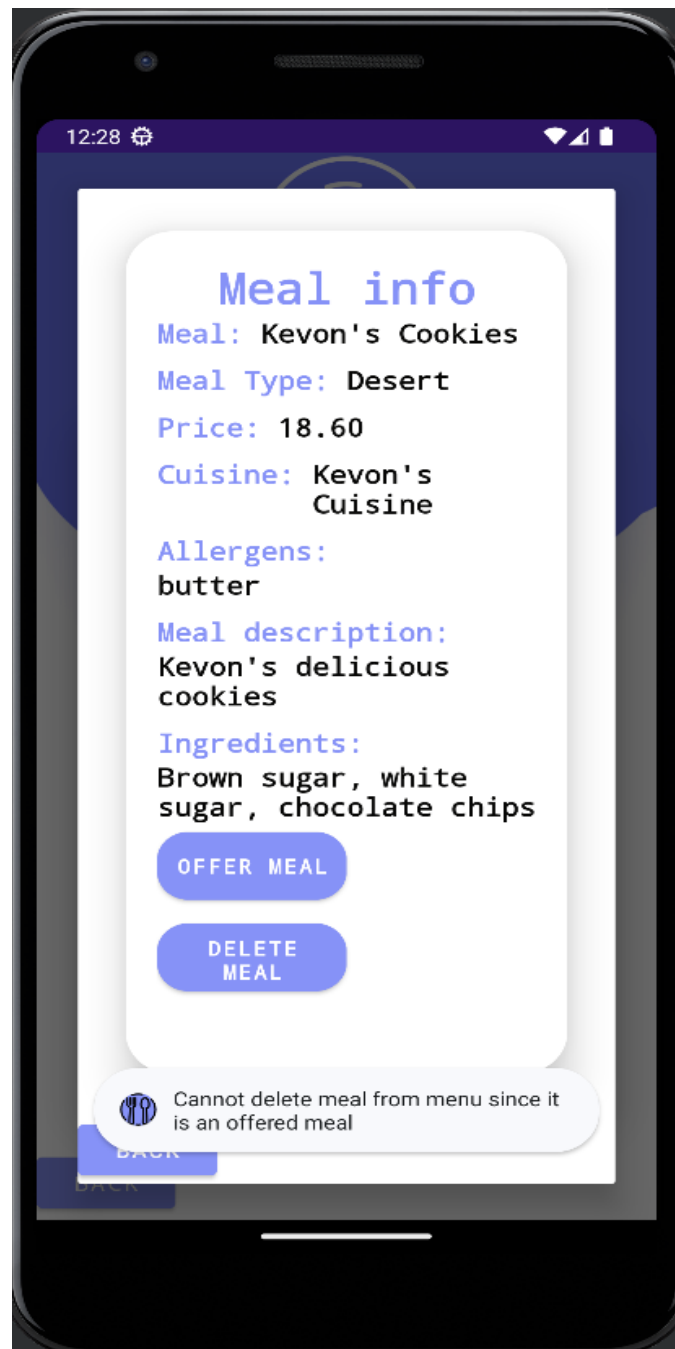
Clicking on the meal to open options to either put the meal on the offered menu



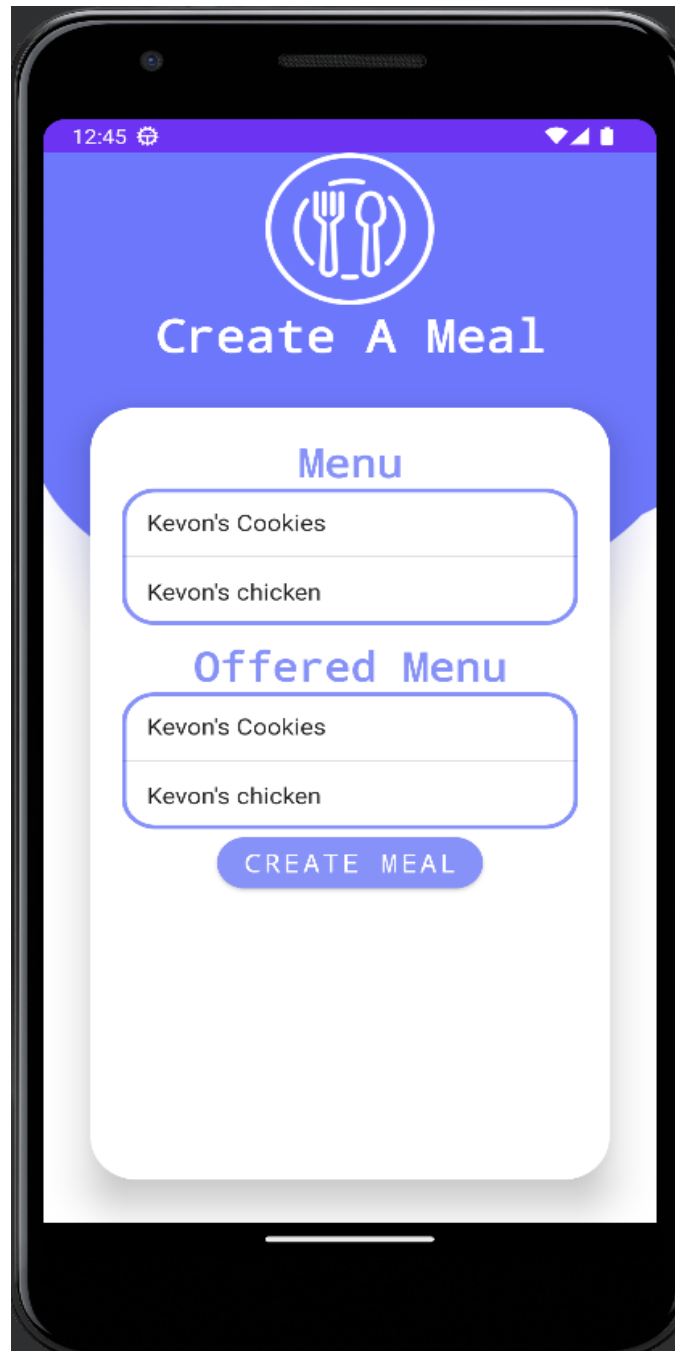
Successfully adding meals creates a popup



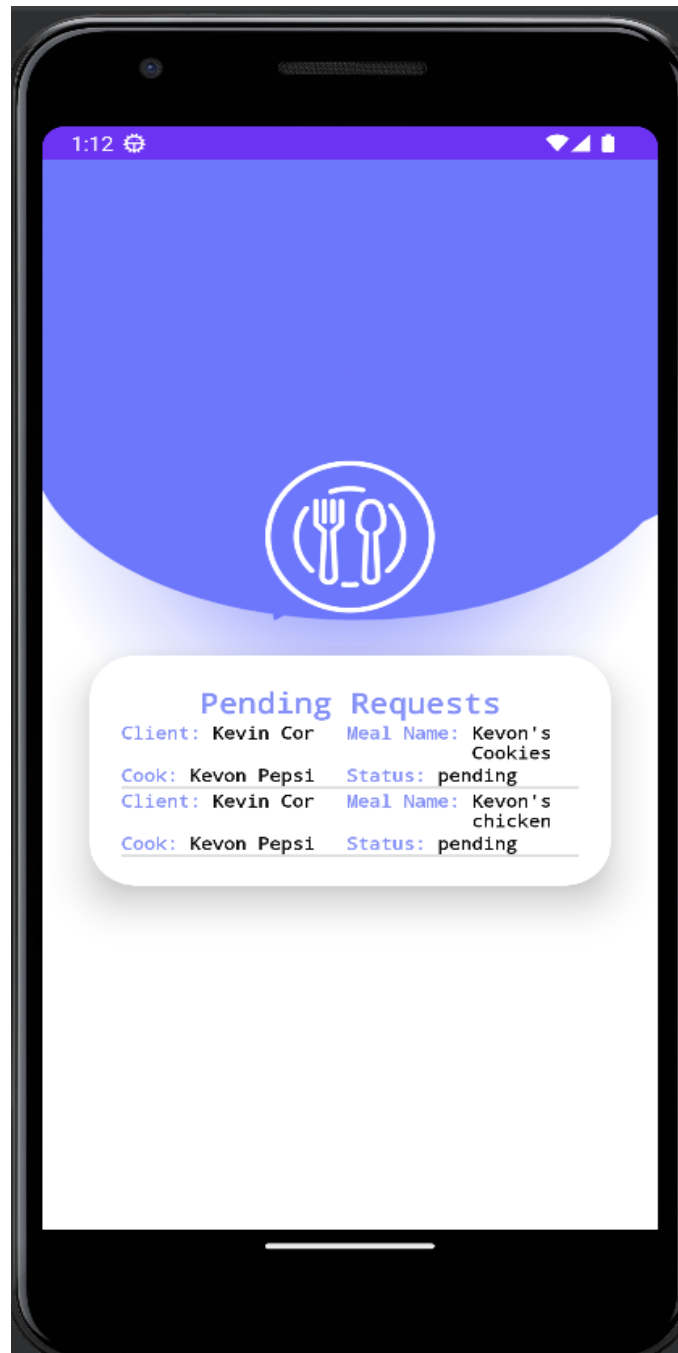
Clicking on an offered meal allows you to delete the meal from the offered menu



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

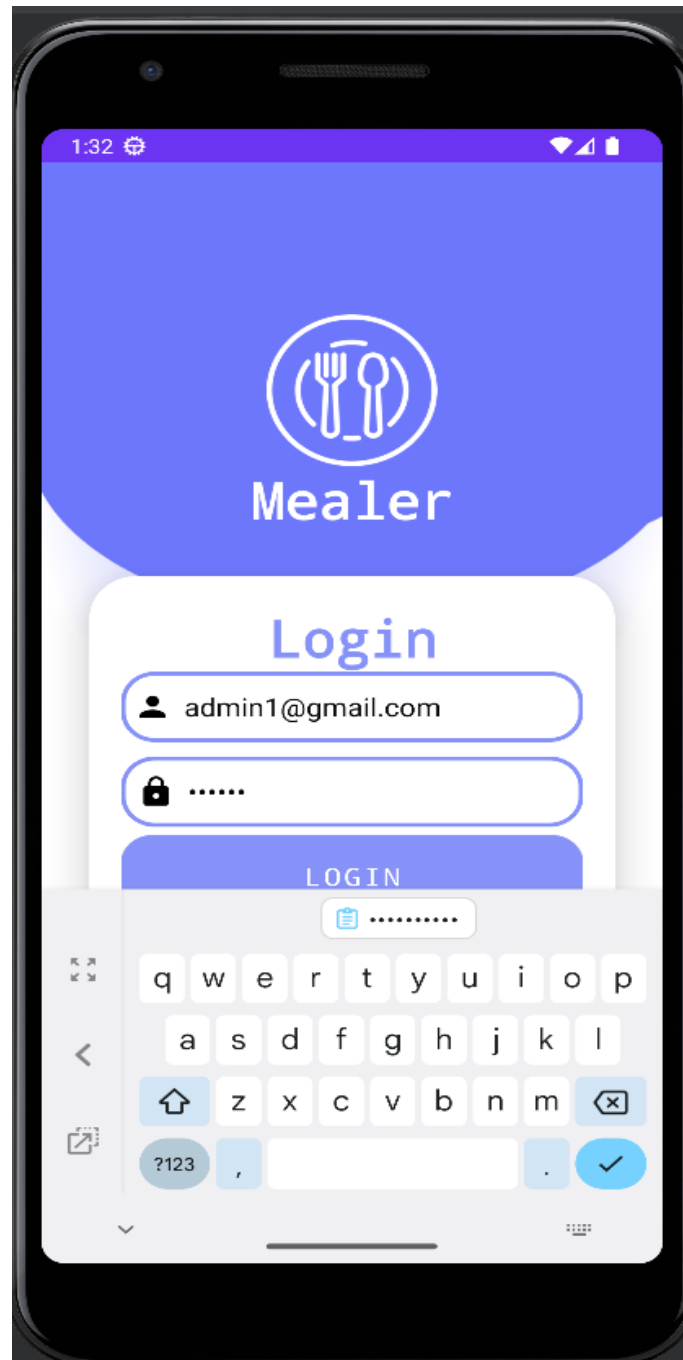


A cook clicking on a meal request will open up the options to either accept or decline the meal request

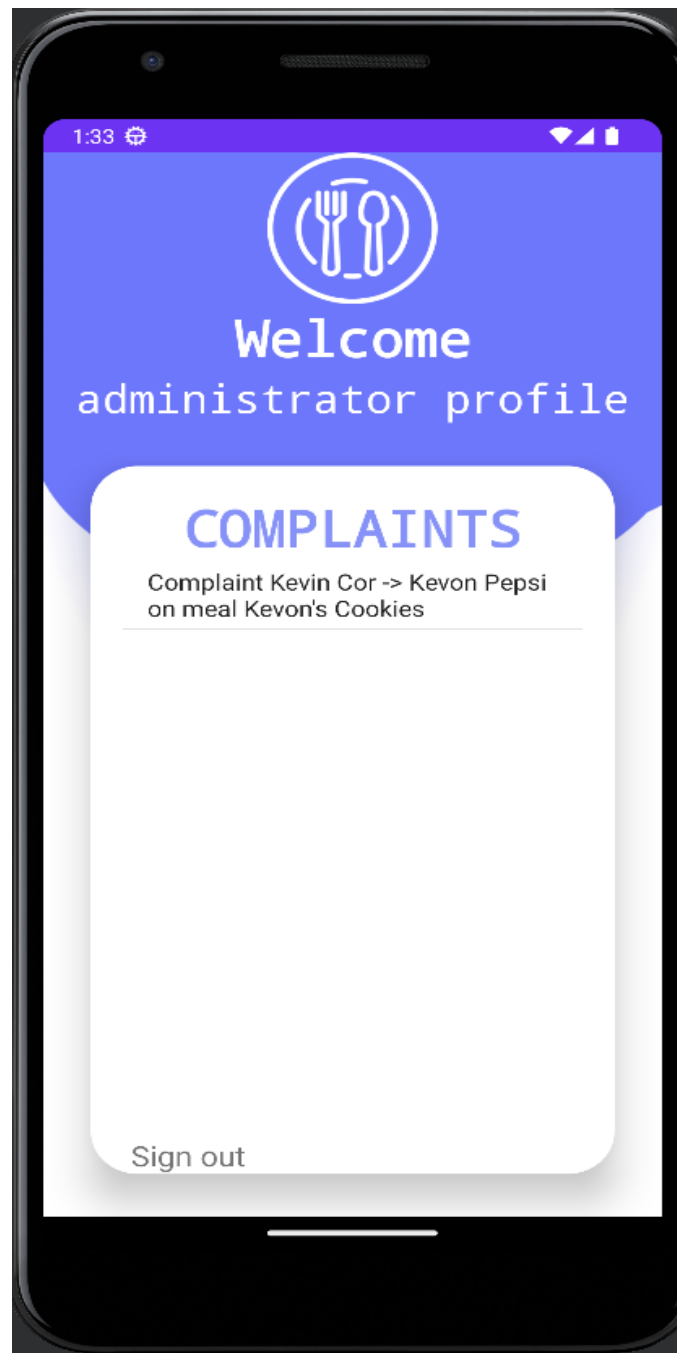


Ban cook alert upon logging in where the cook is unable to do anything but sign out

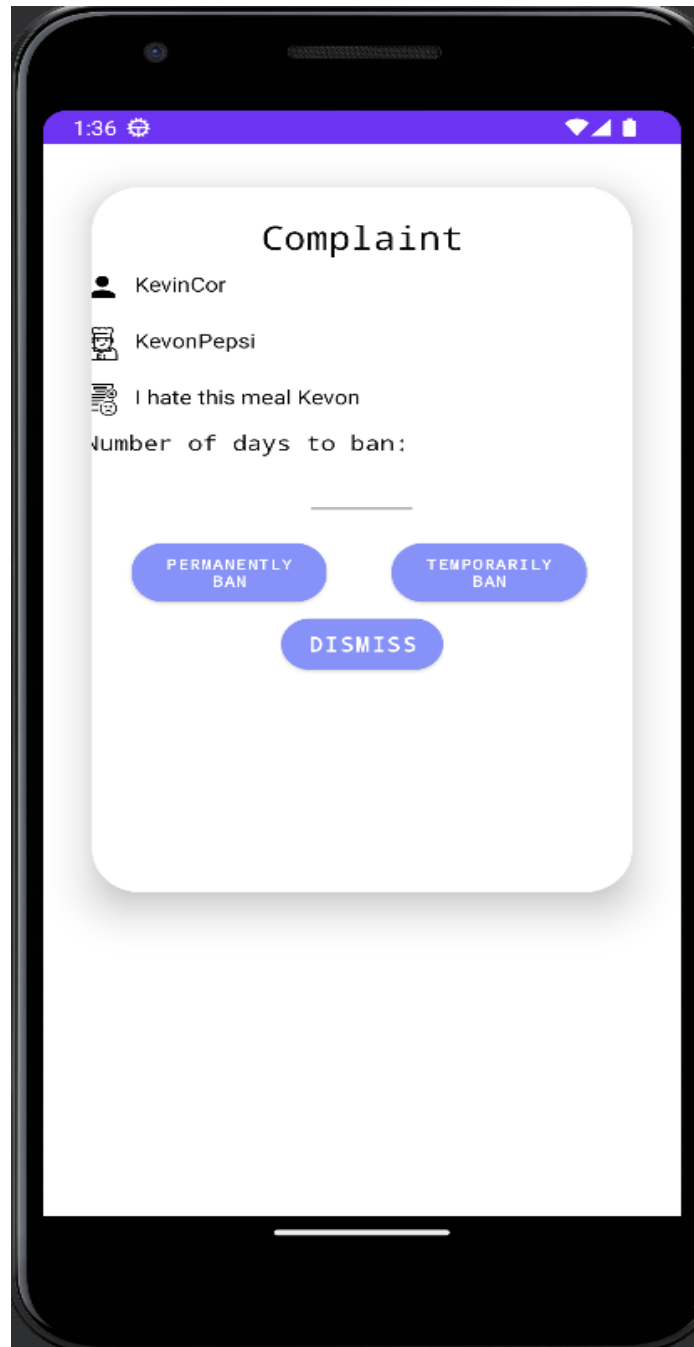
Administrator



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.