Restaurant Finder

Filimon Gebrekidan

Ethan Kolkmeier

James Dulfinger

Carl Trautwein

Final report

Fall 2019

1. Introduction

Restaurant Finder (RF) is an app that helps people decide on what to eat. RF allows a user to create an account and view restaurants in their area in a view that is less overwhelming than similar services such as Google Maps or Yelp. The app is capable of expanding it's search radius to fit user needs and reports the price, rating, address, and other details about each restaurant. This document goes into detail about the functions and creation of RF, as well as shows off several features.

2. Requirements status

Functional requirements status table:

	Description	**Status**	**Justification or comments**
3.2.1	**Log-in**	**Ongoing**	**Log-in is currently local instead of over firebase.**
3.2.2	**Main Page**	**Updated and Complete**	**Random complete, list changed to scrolling button.**
New	**Restaurant Display**	**Complete**	**Page for the restaurant which shows extra information**
3.2.3	**List**	**Updated and ongoing**	**List of restaurants updated in main page (removed from list). Map and favorites ongoing. Filter removed.**
3.2.4	**Survey**	**TBD**	**Not implemented yet.**

User interface requirements status table:

ID	Description	Status	Justification or comments
	Log-in Screen	Complete	Allows user to enter a username and password and create an account
	Main Screen	Complete	Shows restaurants in your area
	Restaurant Screen	Complete	Shows details/location of chosen restaurant

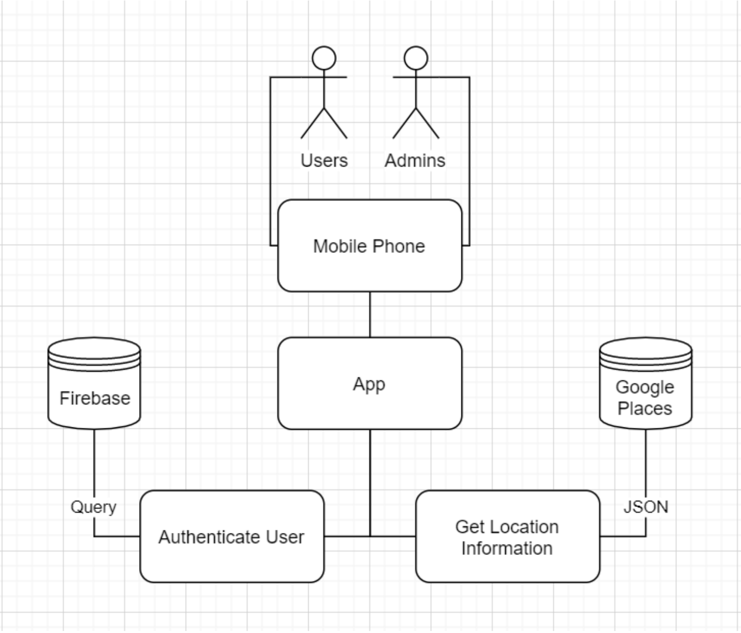
Non-functional requirements status table:

3.3.1	Performance	Ongoing	Performance can be slow based on the internet speed
3.3.2	Ease of Use	Ongoing	Should be fairly easy to use for people who are used to mobile apps.
3.3.3	Quality	Ongoing	User Interface needs some improvement

3. Implementation

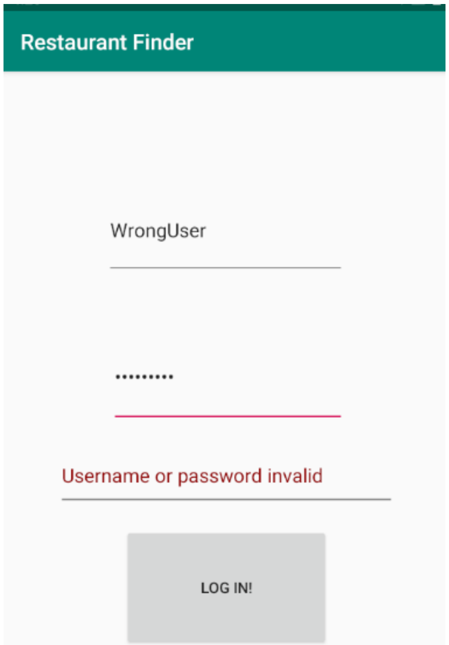
3.1 Implementation details

We used Android Studio as our IDE for this app. Android studio comes with a built-in emulator for the Google Pixel which we used for all of our testing. For Log-in we use google Firebase, a free cloud database program. Lastly we used the Google Places API to get information about device location and restaurants around the device



.

3.2 Snapshots of application

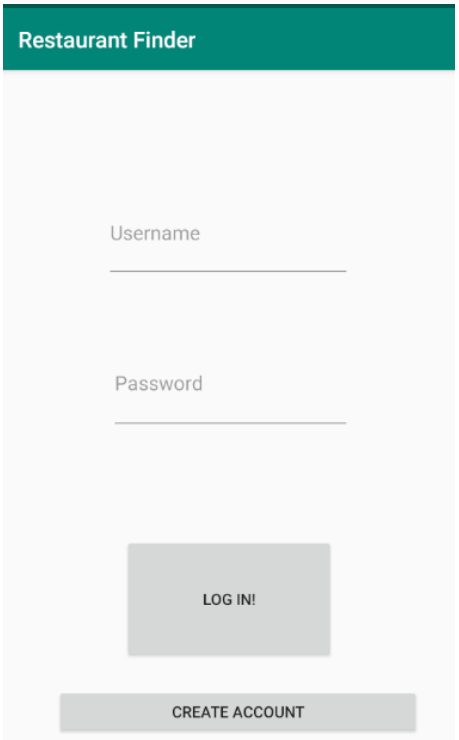


WrongUser

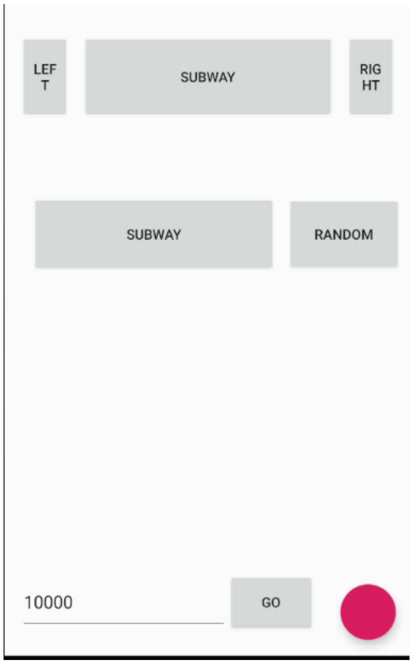
•••••••••

Username or password invalid

Shows log in screen.

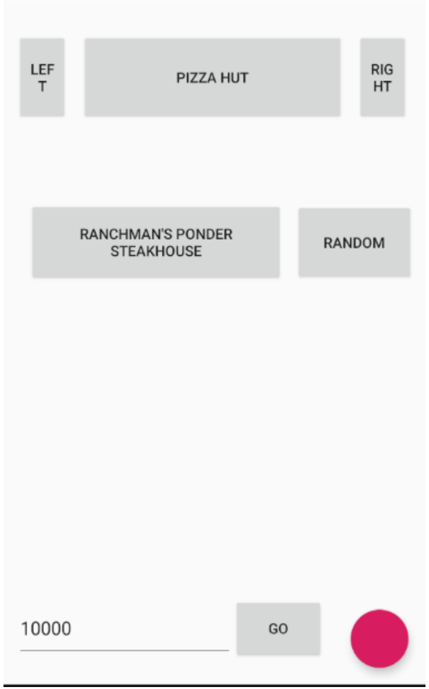


User failed to enter the correct credentials.



10000

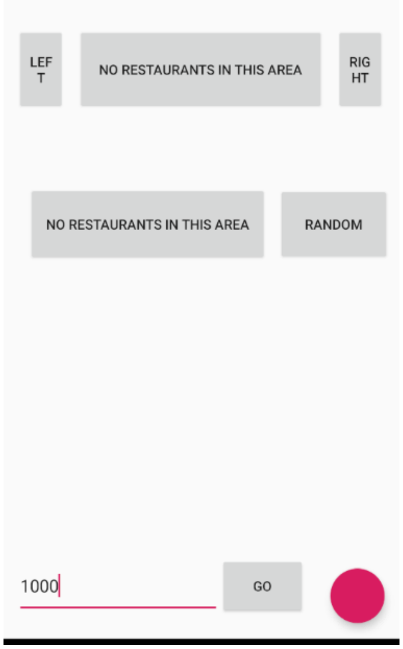
User entered correct credentials and has been taken to the main page. Here two restaurant buttons are displayed. LEFT and RIGHT will move the selected restaurant of the first button to another restaurant in a deterministic way. Random will choose a random restaurant from the list and display it in the second button. The text at the bottom of the page it how far to search for restaurants with a default of 10000. Last the purple button takes you to the map page.



10000

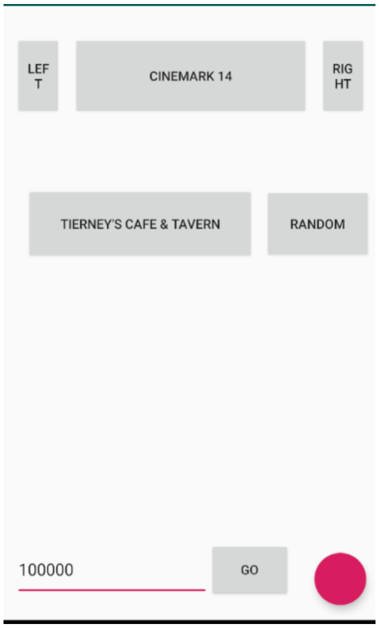
After hitting the RIGHT button SUBWAY turned into PIZZA HUT. If we now hit LEFT we will return to SUBWAY or we could go RIGHT for another restaurant from the list.

The RANDOM button was also hit, selecting a random restaurant from the list. If we hit it again the odds are we get a totally different restaurant.



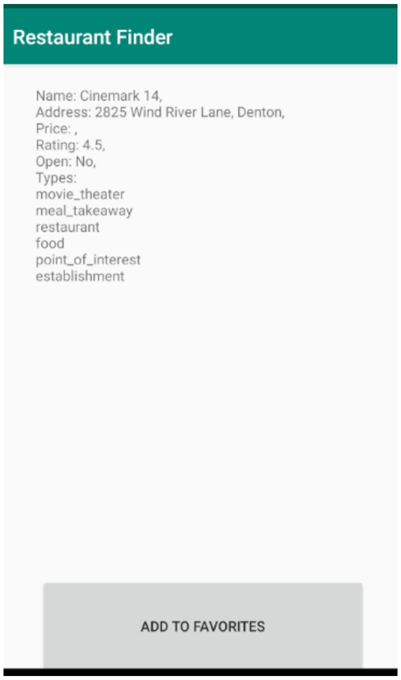
1000

We change the number in the box from 10000 to 1000 and now there are no restaurants in the area. Hitting the RIGHT, LEFT, or RANDOM buttons gives the same “NO RESTAURANTS IN THIS AREA” result.



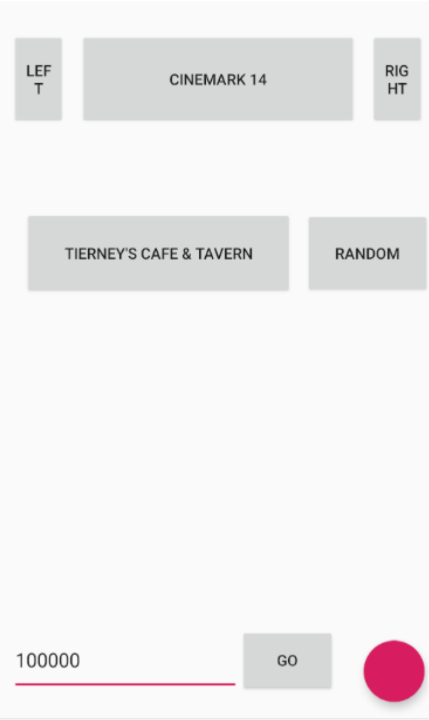
100000

We now changed the radius from 1000 to 100000 (10X higher than the default). We can see that some new restaurants have populated our list. Now CINEMARK 14 is the first restaurant in the list whereas SUBWAY has been moved further down.



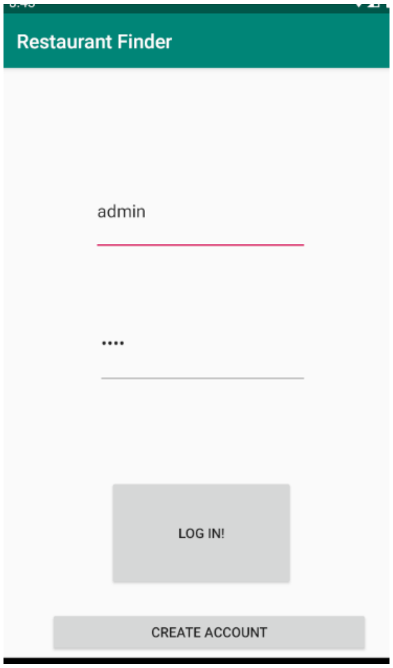
Name: Cinemark 14,
Address: 2825 Wind River Lane, Denton,
Price: ,
Rating: 4.5,
Open: No,
Types:
movie_theater
meal_takeaway
restaurant
food
point_of_interest
establishment

Now we have pressed Cinemark 14 which has taken us to a page with info about the restaurant including its address, rating, and if it is open. We can see from the ‘types’ field that this restaurant is also a theater. Hitting the back arrow brings us back to the previous page without having to requery google.



100000

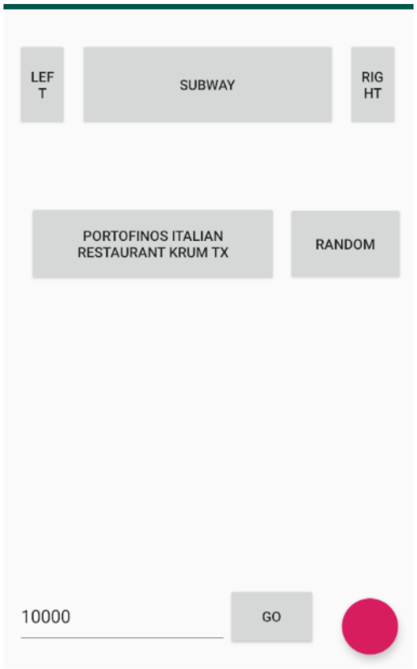
Upon hitting the back button all information is the same including the random. The query was not made again so we are using the same list of restaurants before.



admin

••••

Using the back button again we see that text fields are already filled in from before.



10000

When we use those credentials to log on you can see that the restaurants have been reset to their default values.

4. Testing

Write at least 25 test cases involving different functionalities in your system. (Refer to rest case template provided for individual assignment 4)

Related components/ functions	Pre-conditions	Test input	Expected Outcome	Actual Outcome	Pass /Fail
Log-in	Not logged-in	“Fake” as user	Some error showing no entry	Error showing incorrect user/pass	Pass
Log-in	Not logged-in	Admin user/pass	Directed to Main Page	Taken to main page	Pass
Log-in	No account	Hit “Create Account”	Taken to create account screen	Nothing	Fail
Restaurant Selection	MainPage, restaurants available	Hit “RIGHT”	Go to new restaurant	“PIZZA HUT”	Pass
Restaurant Selection	MainPage, restaurants available	Hit “LEFT”	Return to previous restaurant	“SUBWAY”	Pass
Random	MainPage, restaurants available	Hit “RANDOM”	Random restaurant appears	“BRUNCH CAFF”	Pass
Restaurant Info	MainPage, restaurants	Click on a restaurant	Go to restaurant	Successfully sent to page	Pass

	available		page		
Restaurant Info	MainPage, no restaurants available	Click on a restaurant	Go to “No Restaurant” page	Crash	Fail
Restaurant Info	Restaurant Info page	Hit back	Return to Main Page	Return to Main Page	Pass
Search Radius	Main Page	10000, hit GO	Change restaurants to match radius	Radius changed	Pass
Search Radius	Main Page	0, hit GO	Show error	Nothing	Pass
Search Radius	Main Page	-1, hit GO	Correct to positive value	Radius changed to 1	Pass
Search Radius	Main Page	ASKJDB, hit GO	Text box takes no input	No value	Pass
Log-in	Main Page	Hit back button	Logs user out	Logs user out	Pass
Restaurant Selection	Main Page	Hit "LEFT" until several times	List restarts at the end	Goes to end of list	Pass
Restaurant Selection	Main Page	Try to overflow by hitting "RIGHT"	List starts at 0	List starts at 0	Pass
Restaurant Selection	Main Page	Hit "RANDOM" 20 times	Shows random restaurant	Shows random restaurant	Pass
Restaurant Selection	Main Page	Change device location drastically before log in	Change to new device location	Changes to new location	Pass
Restaurant Selection	Main Page	Change device location drastically after log in	Change to new device location	Uses old device location	Fail

Log-in	Not logged-in	No input just hit "Log-In"	Some error showing no entry	Error showing incorrect user/pass	Pass
Create Account	No account created	User: "test", Pass: "test"	Create account	Account created	Pass
Create Account	No account created	Try to destroy filesystem with invalid name User: "--"	Error saying no entry	Account created (I didn't think this would work)	Fail
Create Account	Username already taken	User: "test"	Error saying user already exists	Writes over old account	Fail
Create Account	Account is already made	User: "test" Pass: "testing"	Error saying user already exists	Updates password	Fail

5. Risk management

Risk	**Probability**	**Impact**	**Exposure**	**Classification**	**Mitigation Strategy**
Poor UI	High	Low	Medium	Quality	Spend more time on UI, receive help from people or tutorials
Integration Risk	Medium	High	High	Technical	Ask people who were familiar with FireBase and Google’s API
Lack of team Collaboration	Low	High	Medium	Personal	Make mandatory weekly meeting

Final Product does not meet needs	Medium	High	High	Quality	Set weekly goals, check that all teammates are doing their parts.
Tight Deadlines	High	High	High	Technical	Set short deadlines, so we can make sure we are working at all times

6. References

https://ieeeauthorcenter.ieee.org/wp-content/uploads/IEEE-Reference-Guide.pdf