



GROCAFAST

Smart Grocery Shopping Application



GROCAFAST

LIST YOUR GROCERIES .. WITH NO WORRIES



GROUP MEMBERS

Elaf Aloufi

Manar Altaiary

Layan Fakhurji

Raneem Alshareef

SUPERVISIOR

Dr. Abeer Hakeem



OUTLINE

Problem

Solution

Features

System
Design

Project
Limitations

Implementation

Results

Testing

Future
Work

Conclusion



PROBLEM

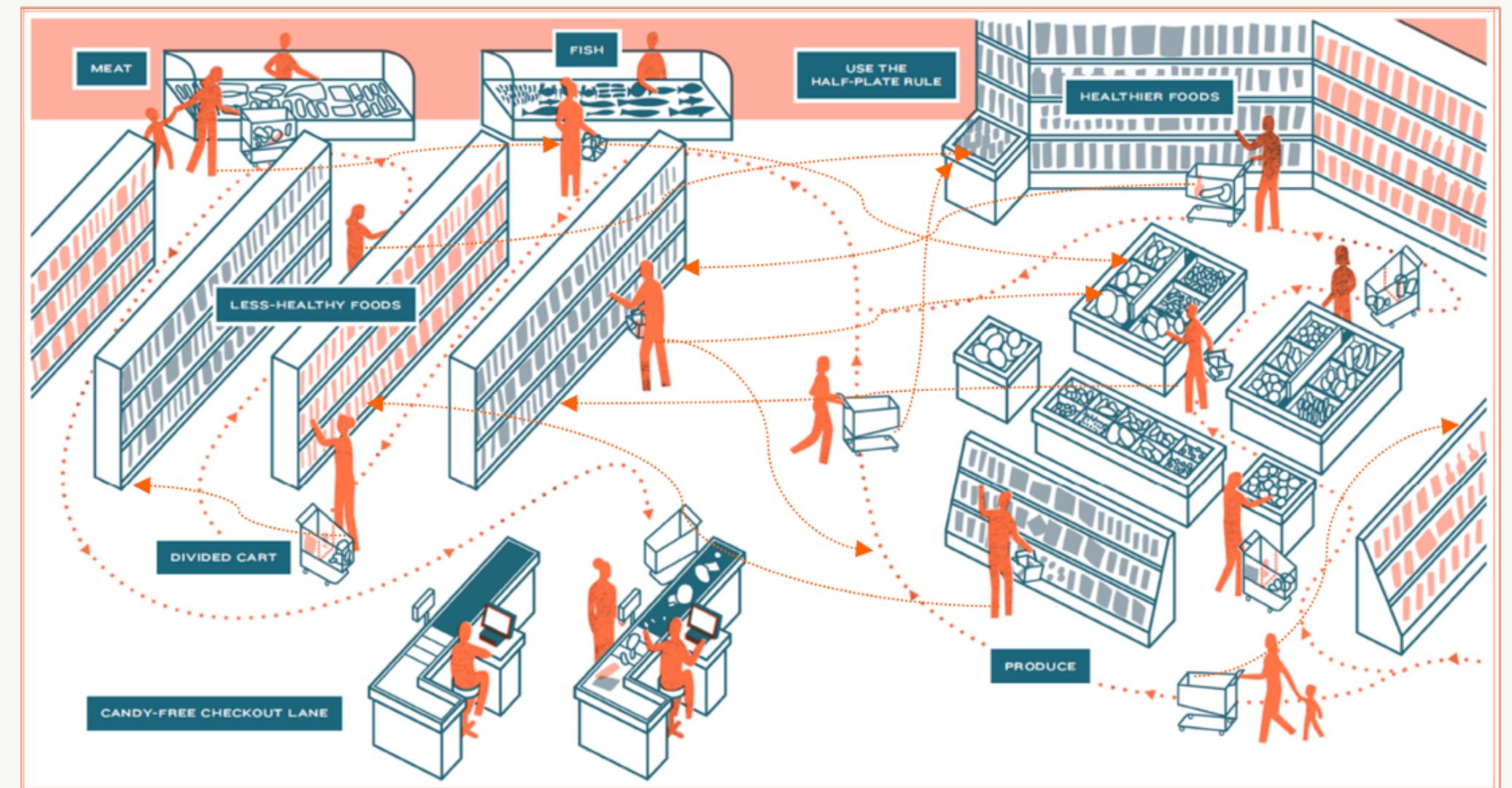


GROCAFAS†

LIST YOUR GROCERIES .. WITH NO WORRIES

PROBLEM

- Grocery Stores a place where you can easily get lost.
- Cross the same aisles over and over again.
- Time-consuming.



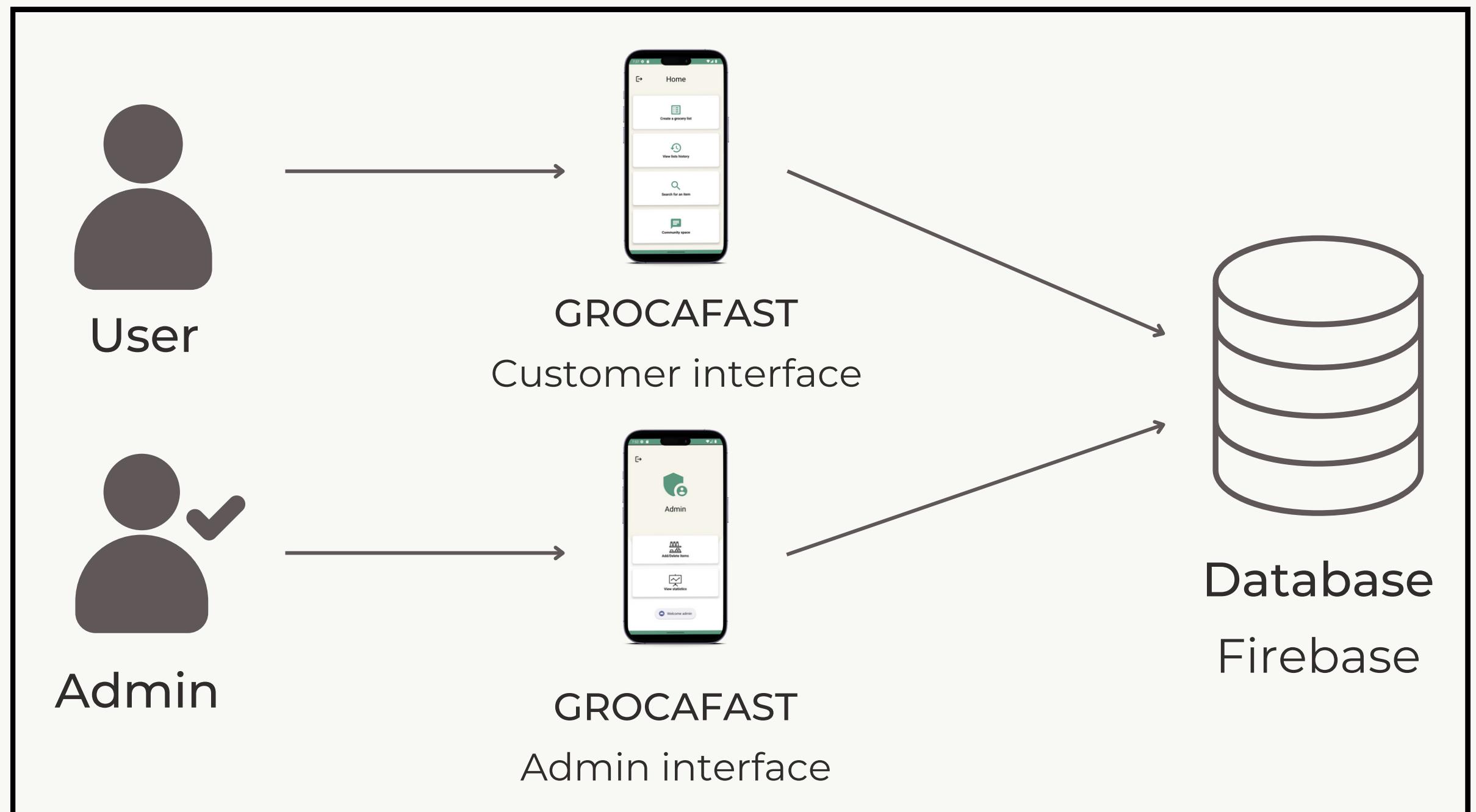
SOLUTION



GROCAFAS†

LIST YOUR GROCERIES .. WITH NO WORRIES

SOLUTION



FEATURES



GROCAFAS~~T~~

LIST YOUR GROCERIES .. WITH NO WORRIES

FEATURES



Create a grocery list



Search for an item and view its location



Modify a previous list (if any)



Receive personalized shopping suggestions



Obtain the shortest path



Create posts in the online space between users of the same supermarket



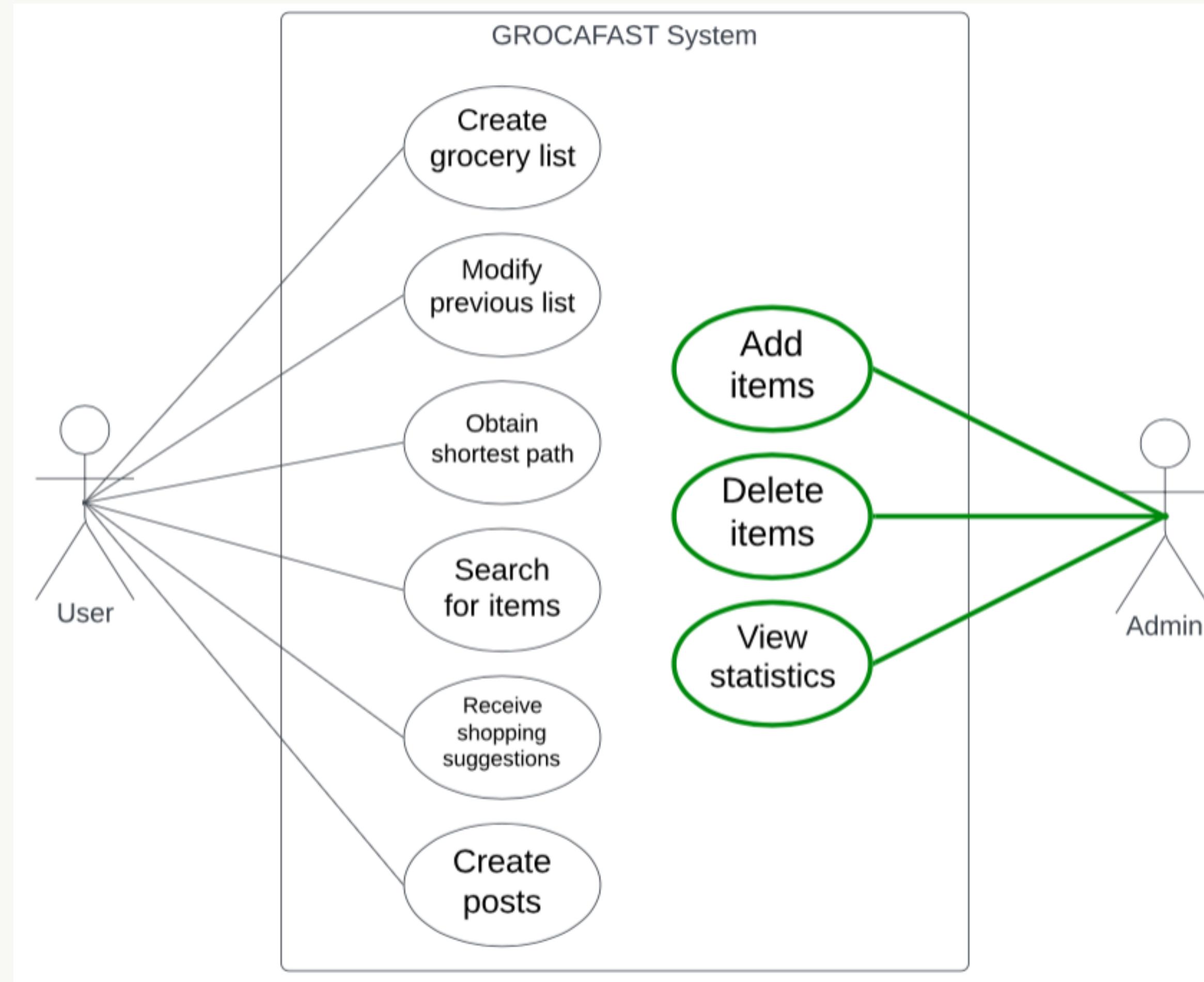
SYSTEM DESIGN



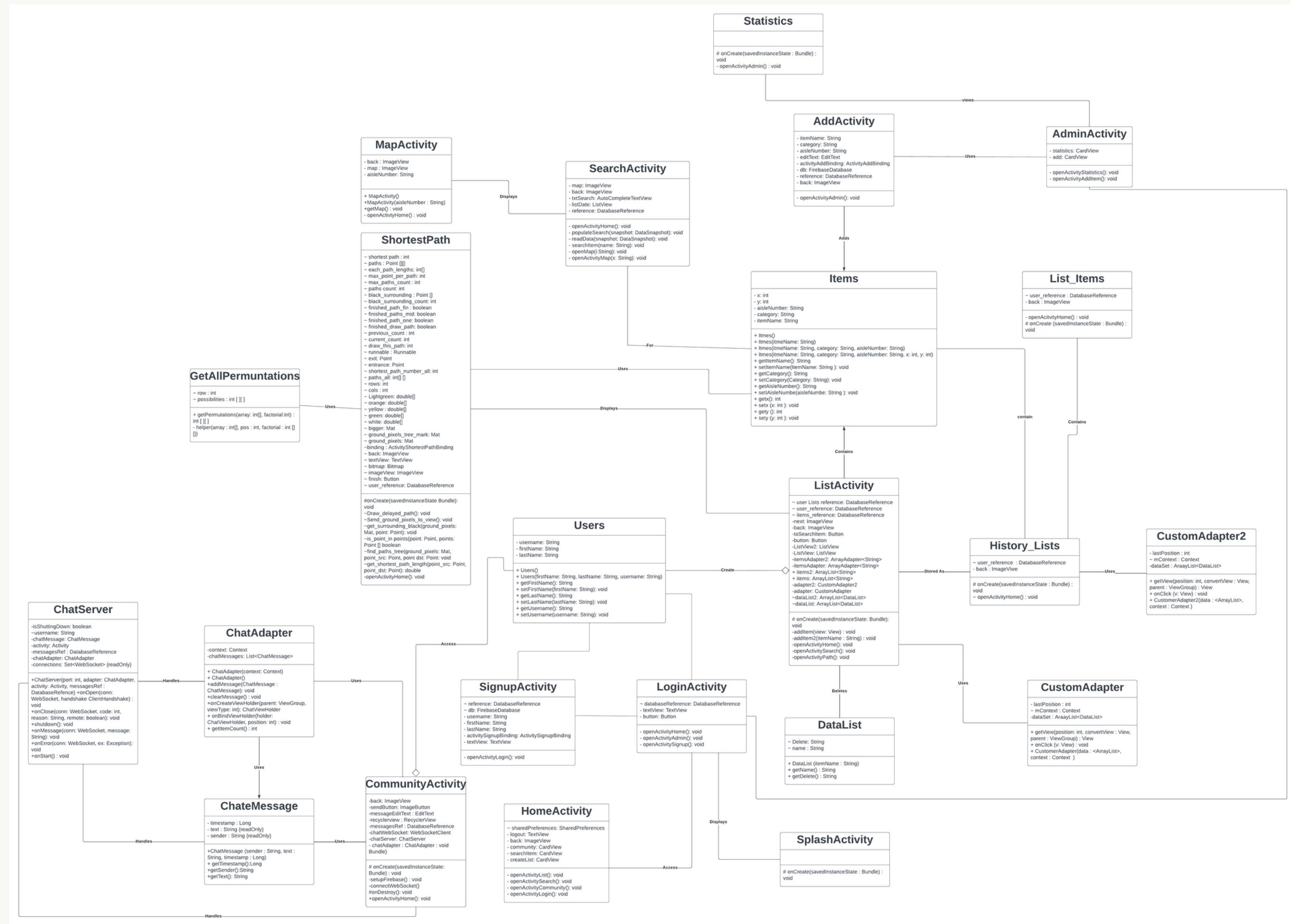
GROCAFAS†

LIST YOUR GROCERIES .. WITH NO WORRIES

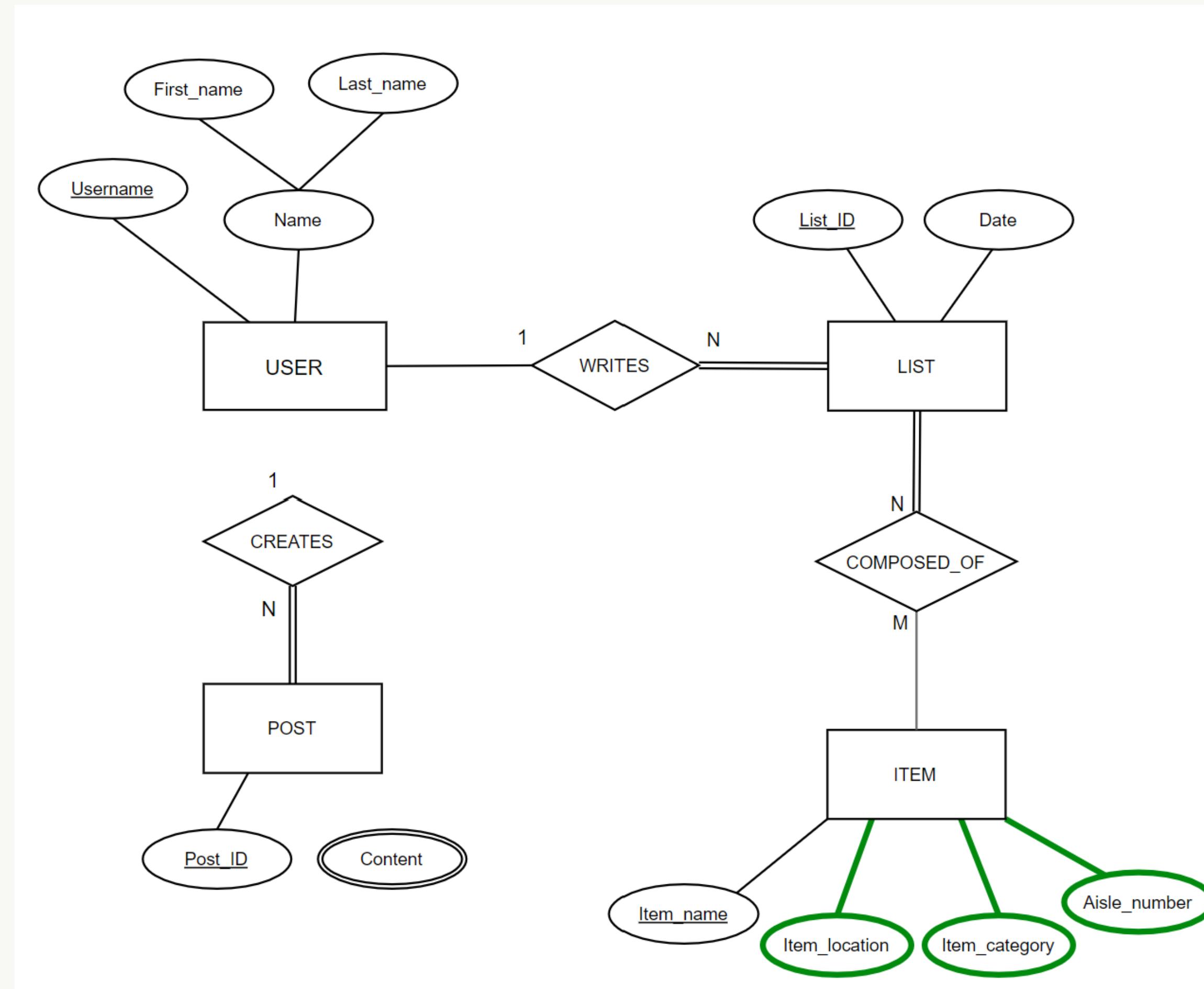
SYSTEM DESIGN: USE CASE DIAGRAM



SYSTEM DESIGN: CLASS DIAGRAM



SYSTEM DESIGN: ER DIAGRAM



PROJECT LIMITATION



GROCAFAS†

LIST YOUR GROCERIES .. WITH NO WORRIES

PROJECT LIMITATION

- Integration with Indoor Atlas



IMPLEMENTATION

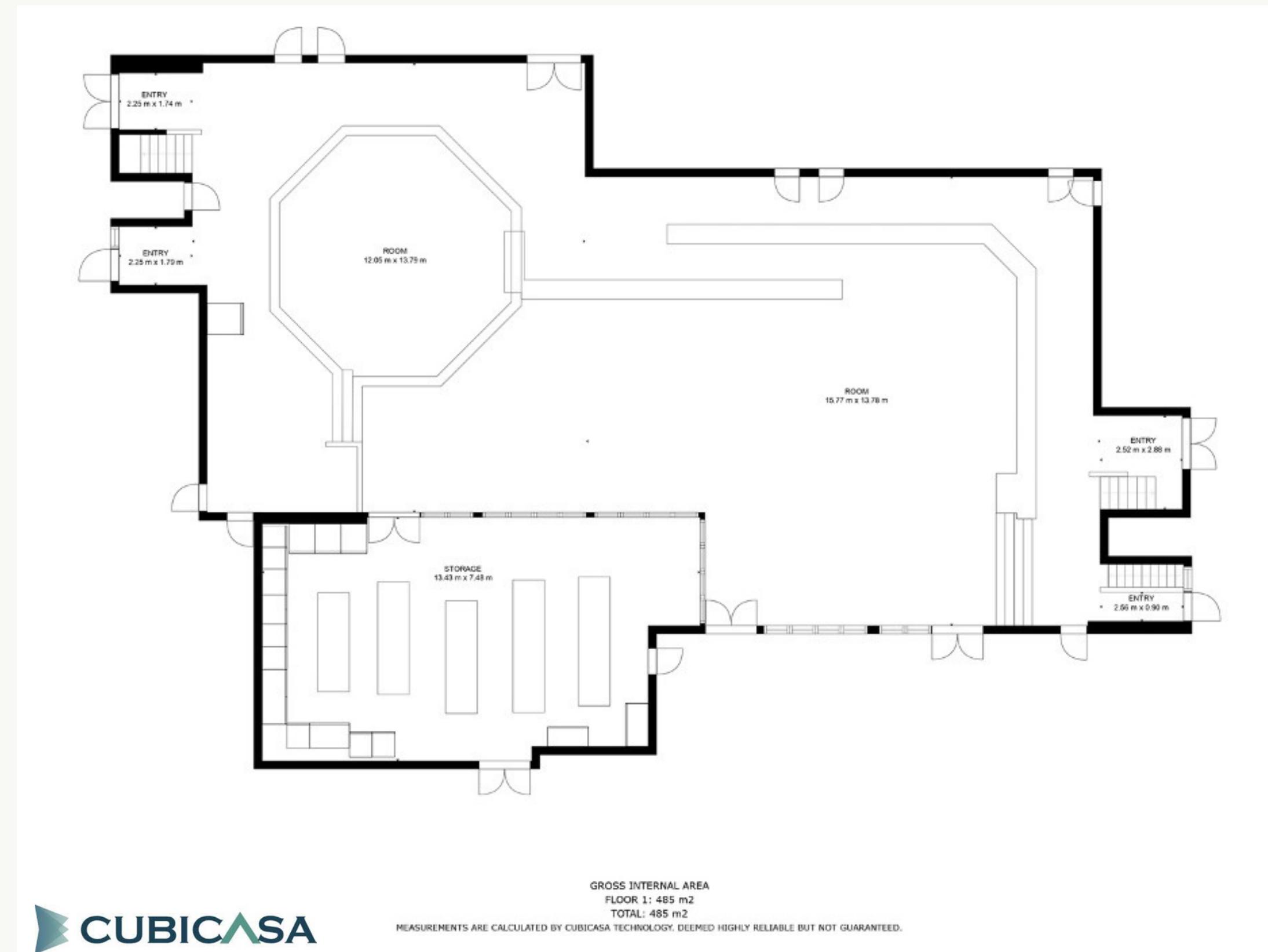


GROCAFAS†

LIST YOUR GROCERIES .. WITH NO WORRIES

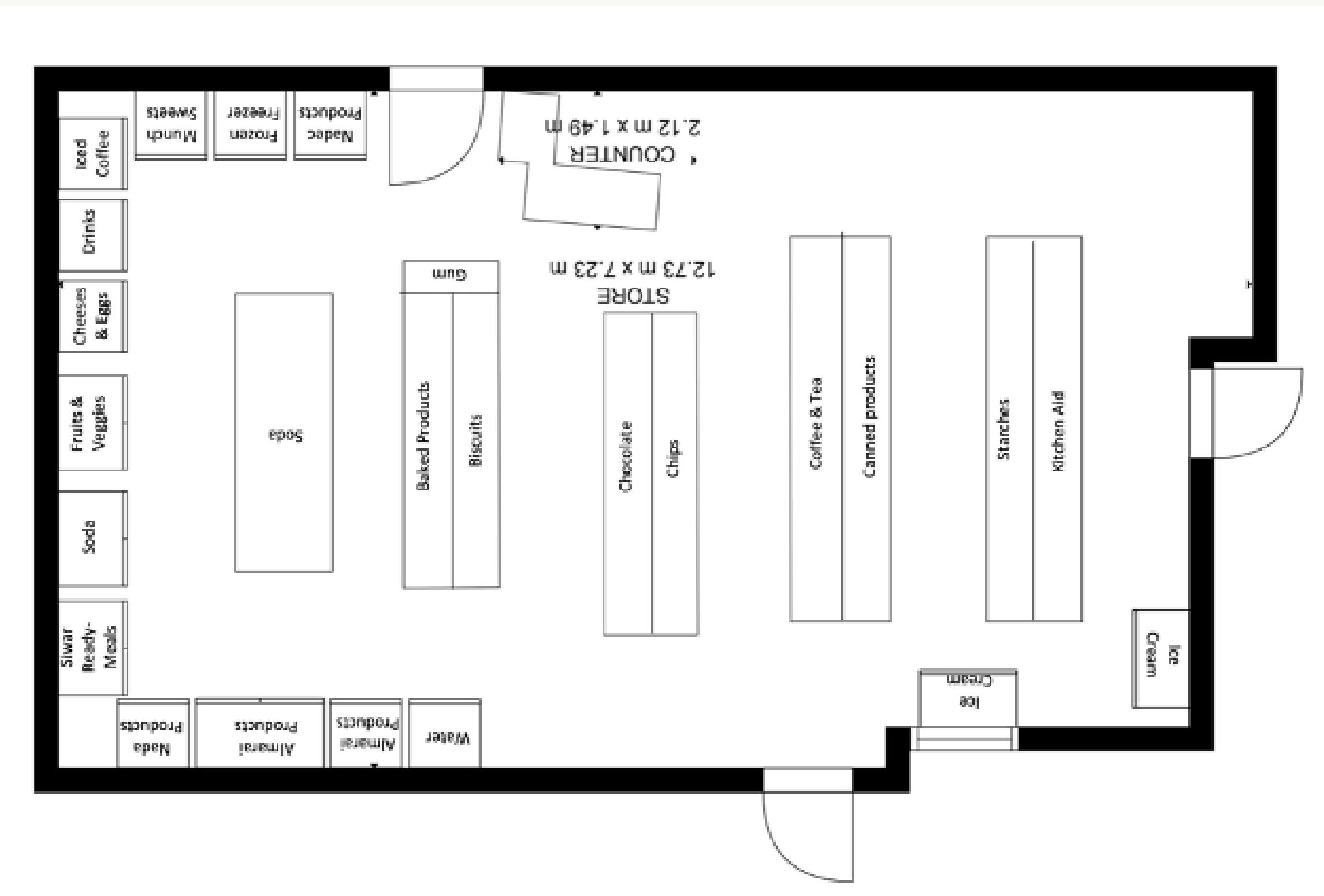
IMPLEMENTATION

CREATING FLOOR PLAN



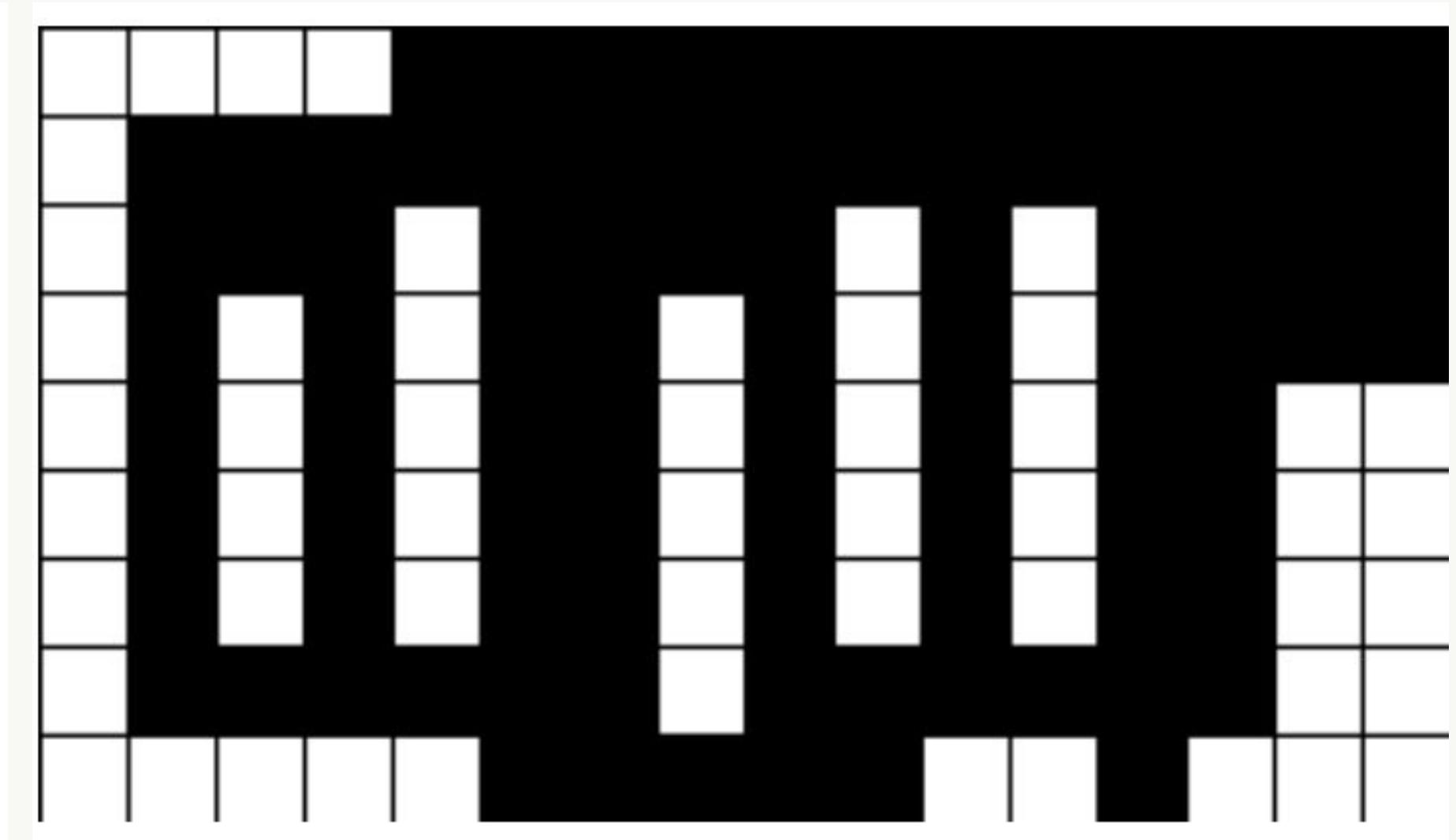
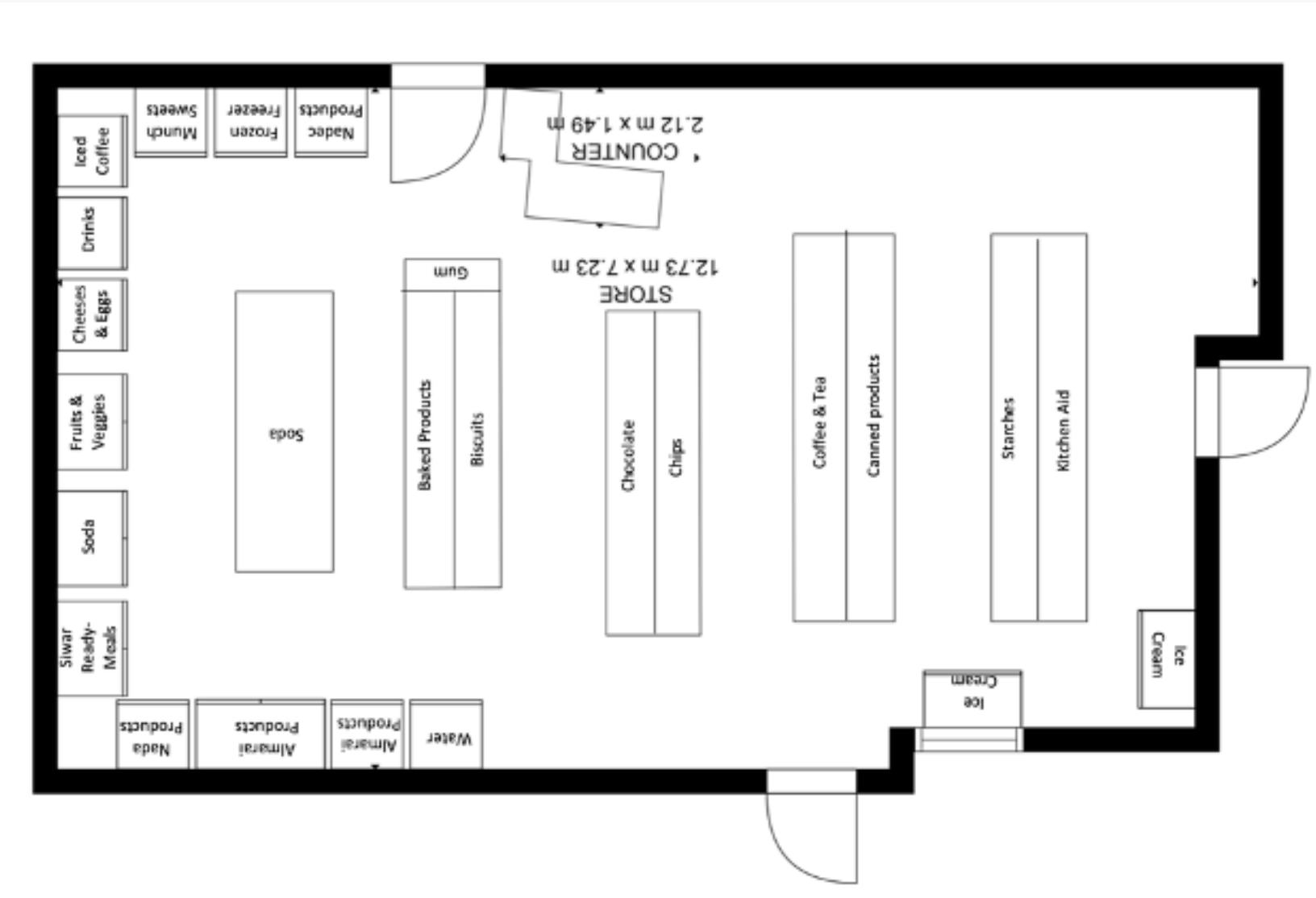
IMPLEMENTATION

ADDING LABELS



IMPLEMENTATION

MAP THE FLOOR PLAN TO A MATRIX



IMPLEMENTATION

FIND THE SHORTEST PATH

DIJKSTRA ALGORITHM

- ✓ Matches GROCAFAST objective.
- ✓ Simple to understand and implement.
- ✓ Reduce space and memory effort .



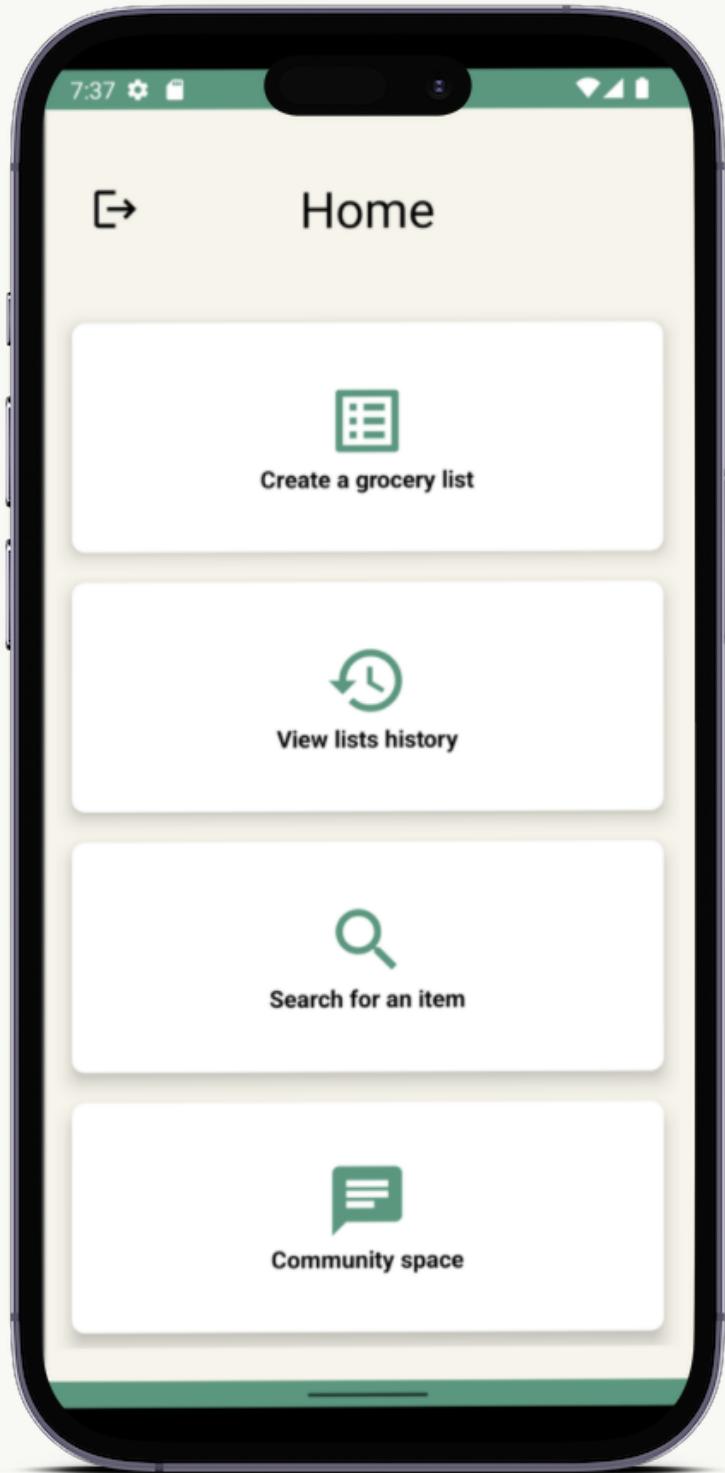
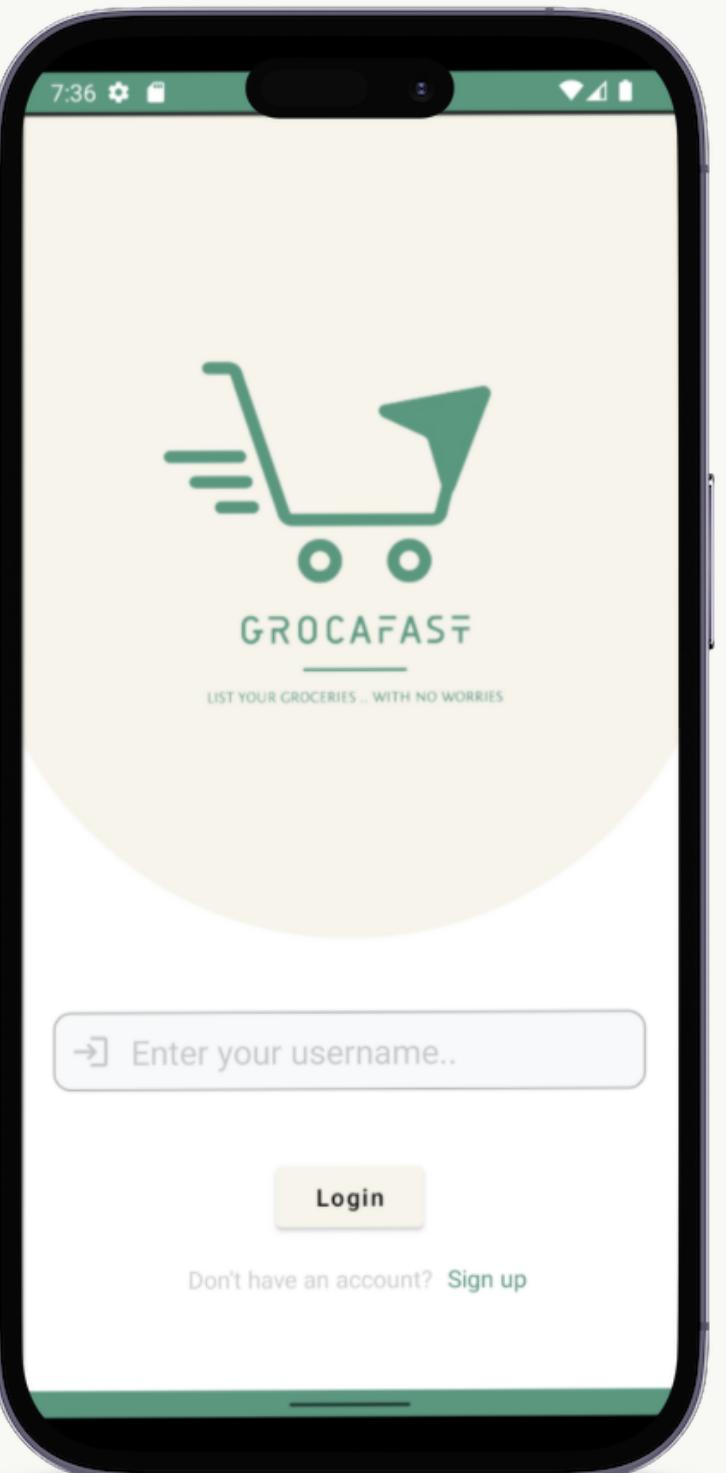
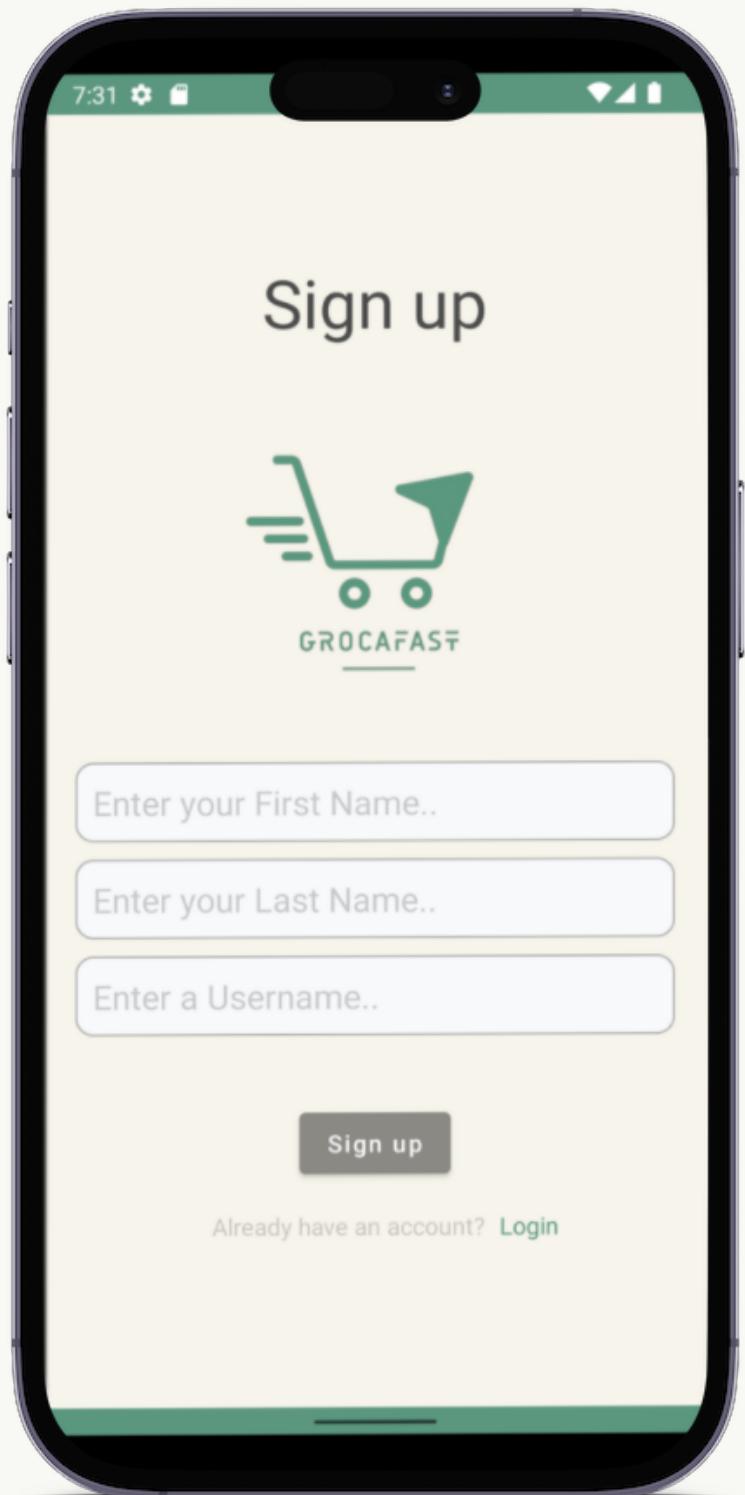
RESULTS



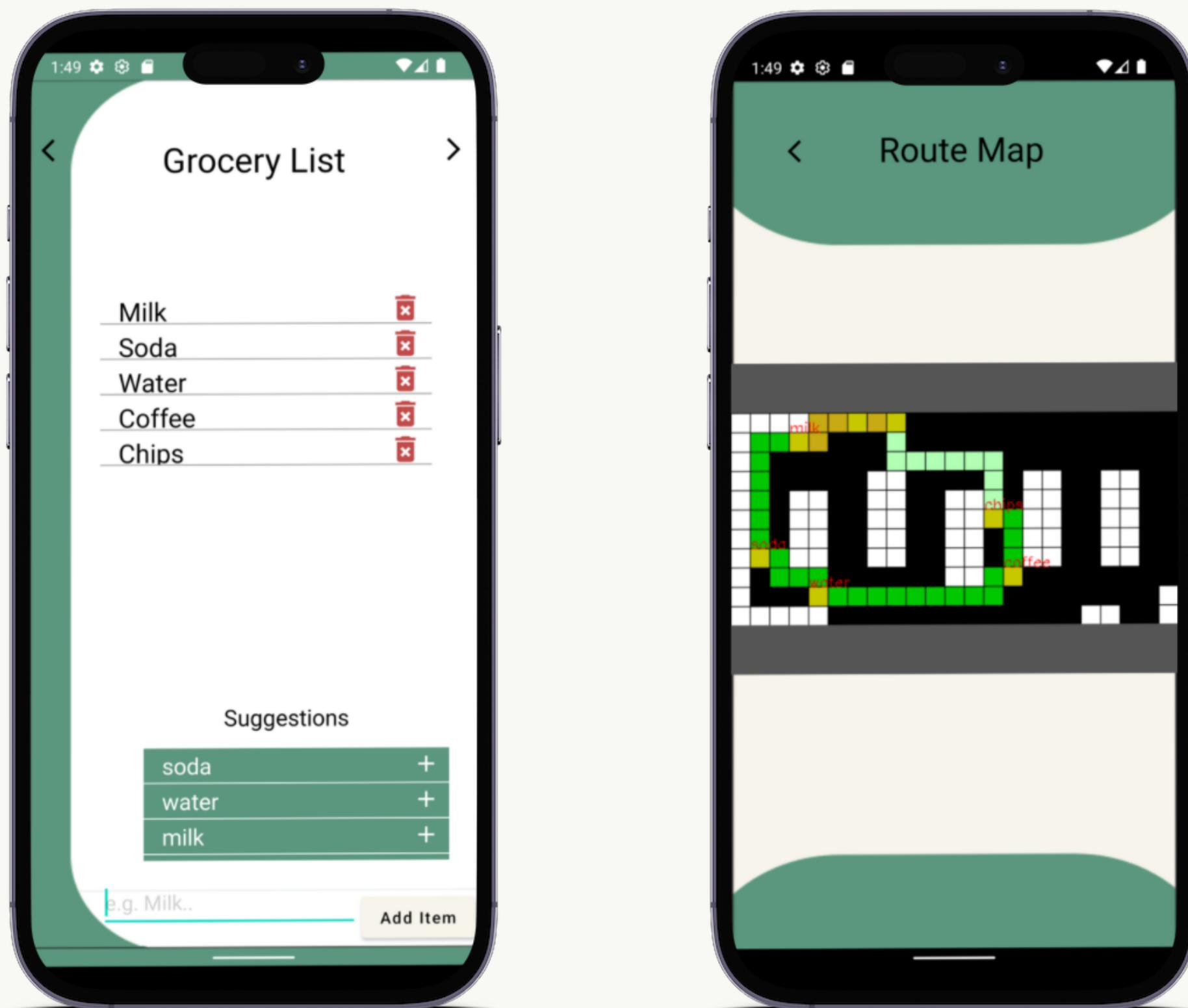
GROCAFST

LIST YOUR GROCERIES .. WITH NO WORRIES

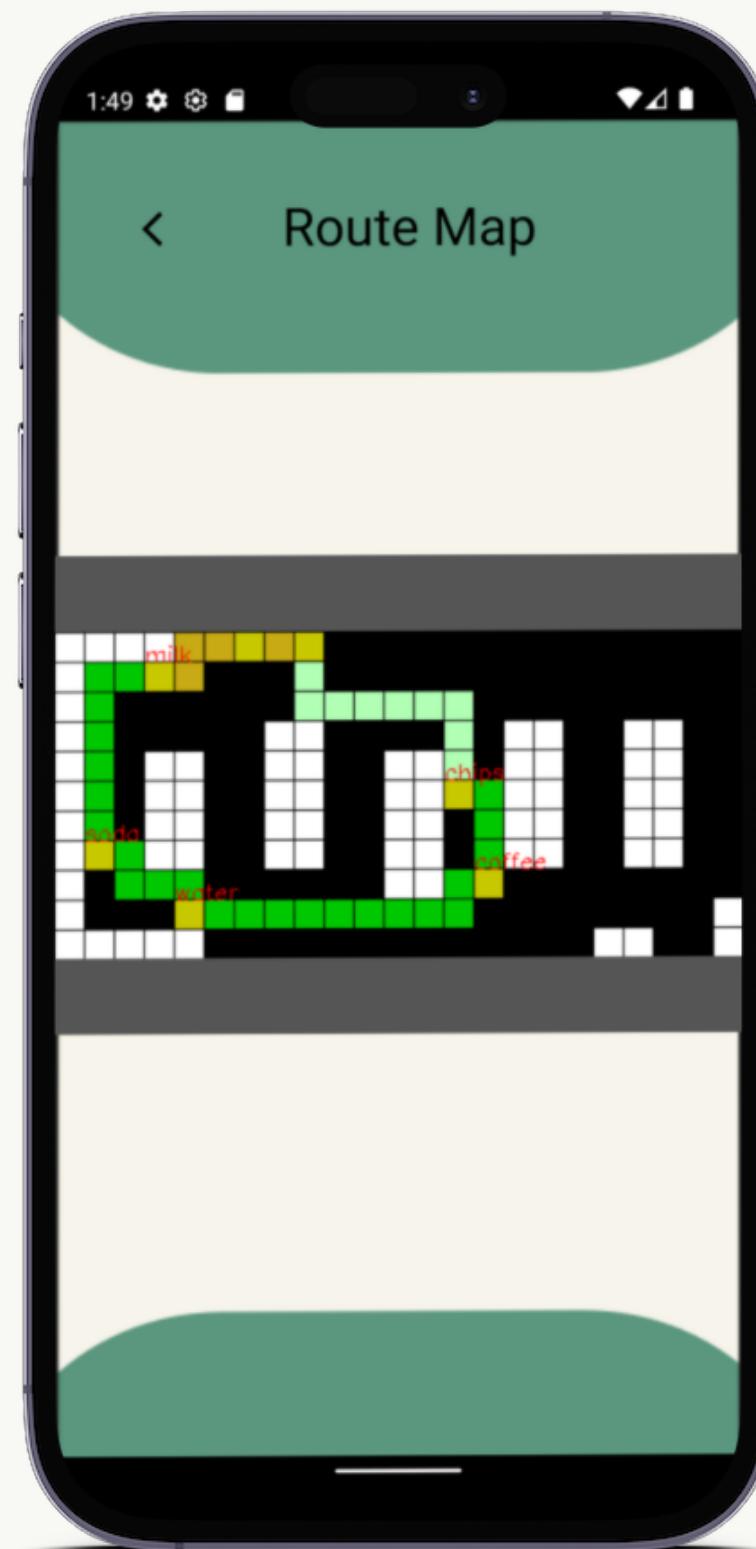
RESULTS



RESULTS

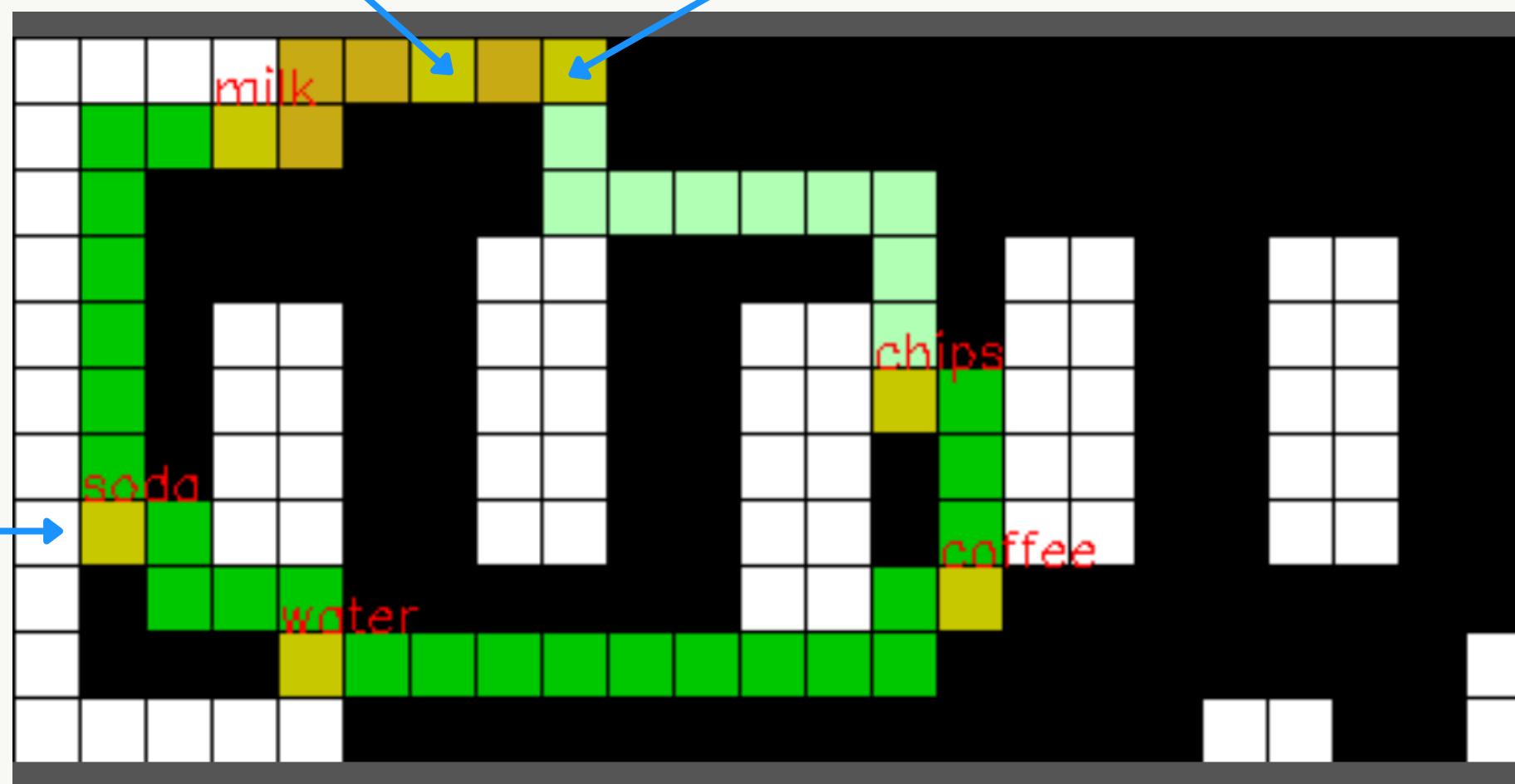


RESULTS



Item
(Soda)

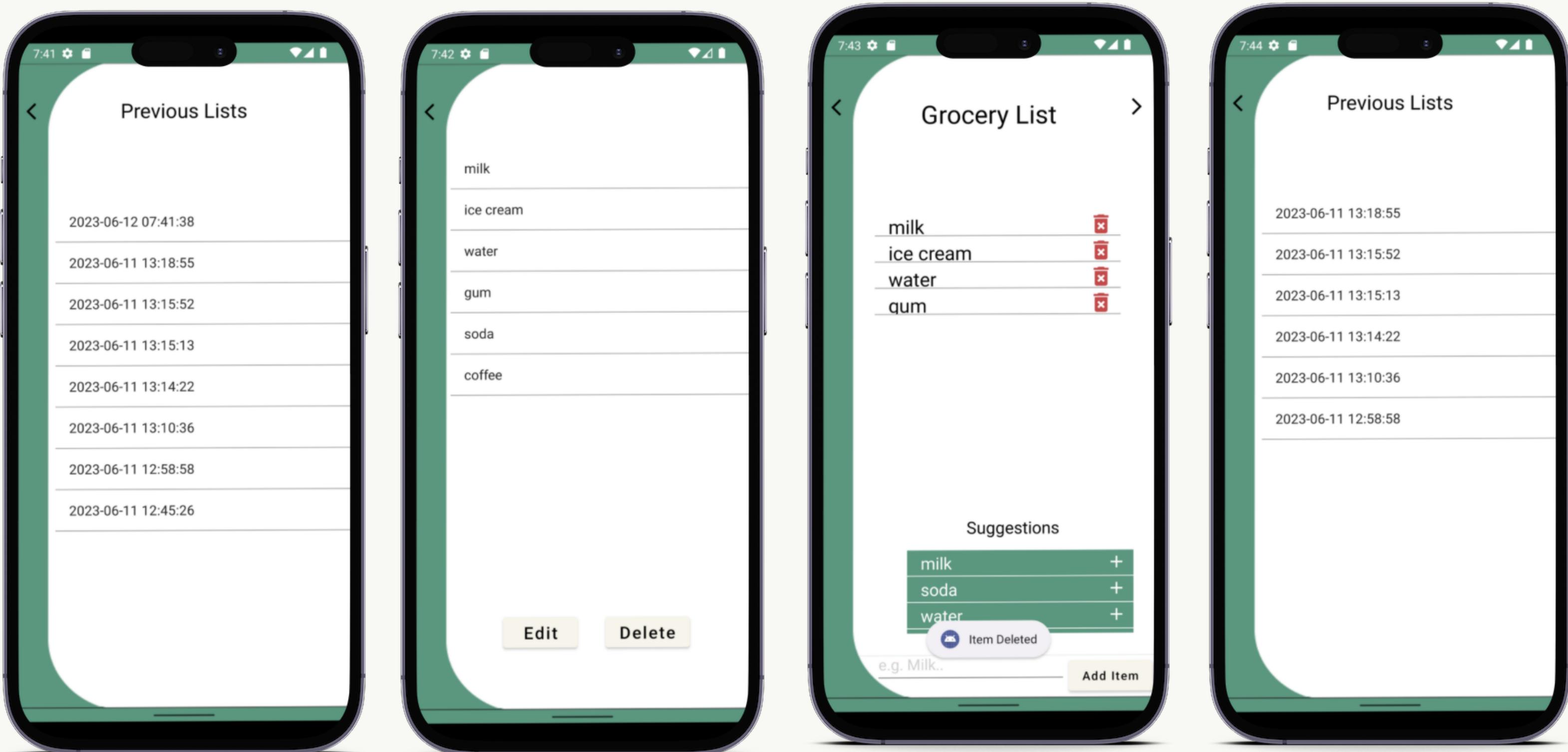
Entrance



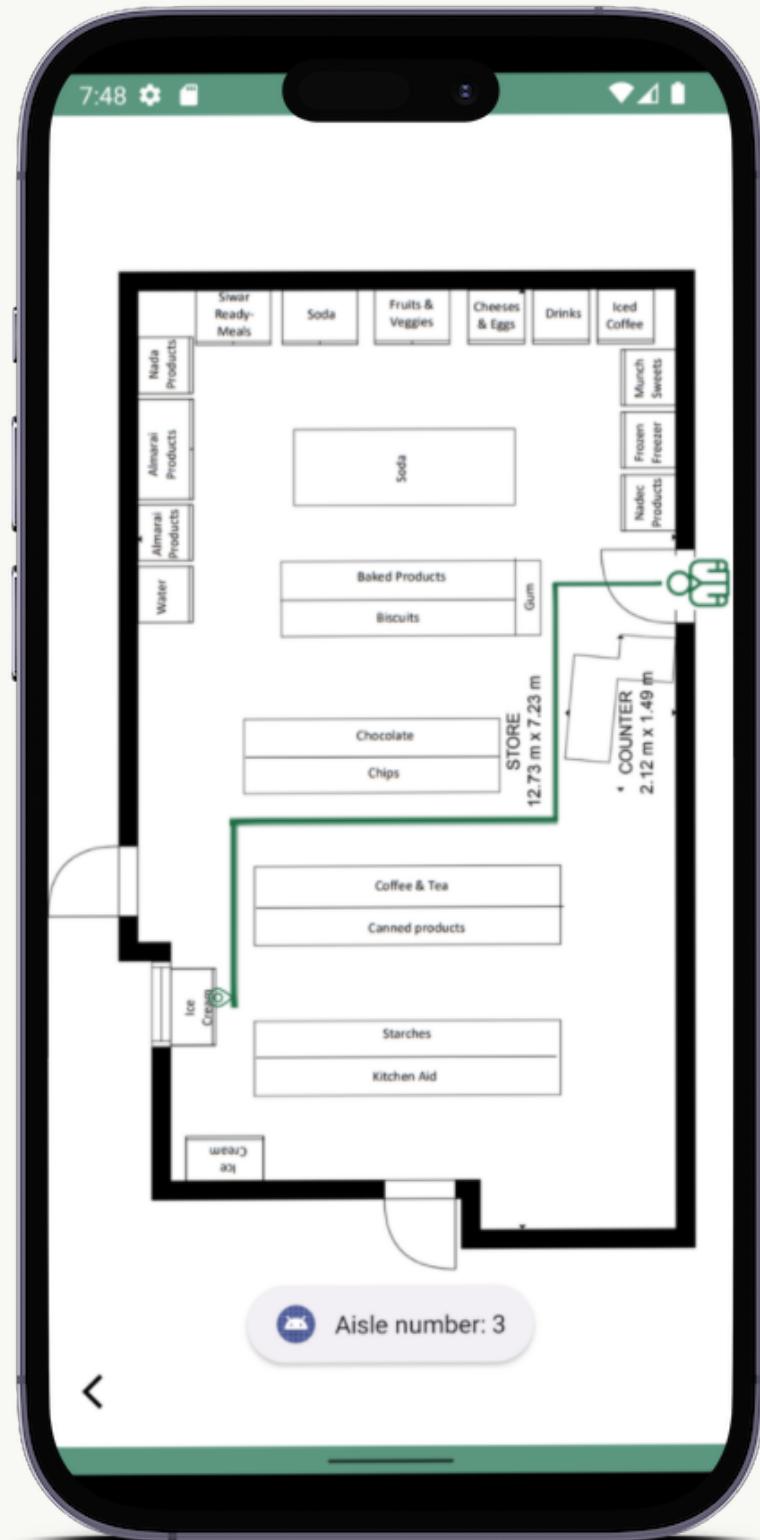
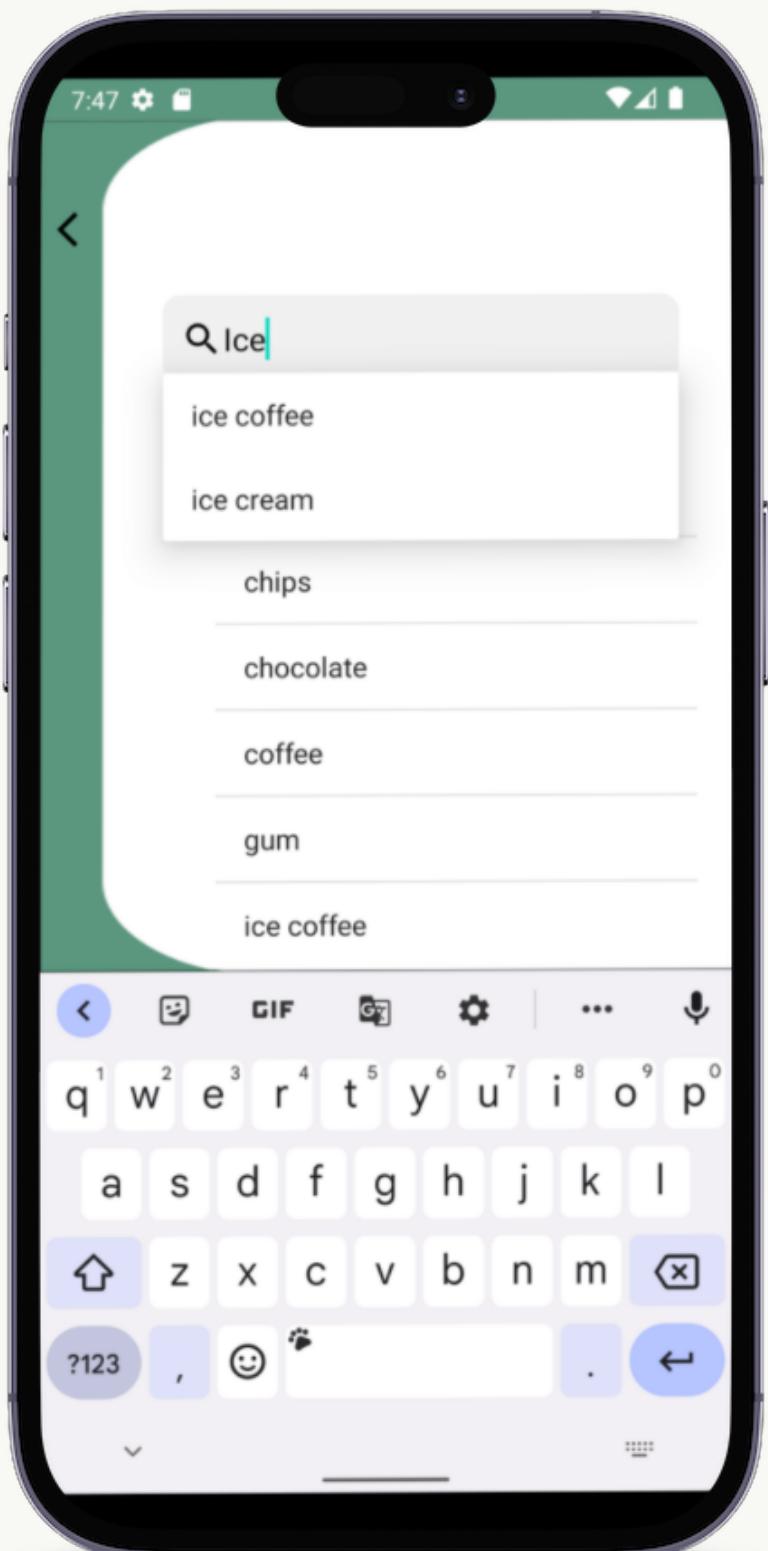
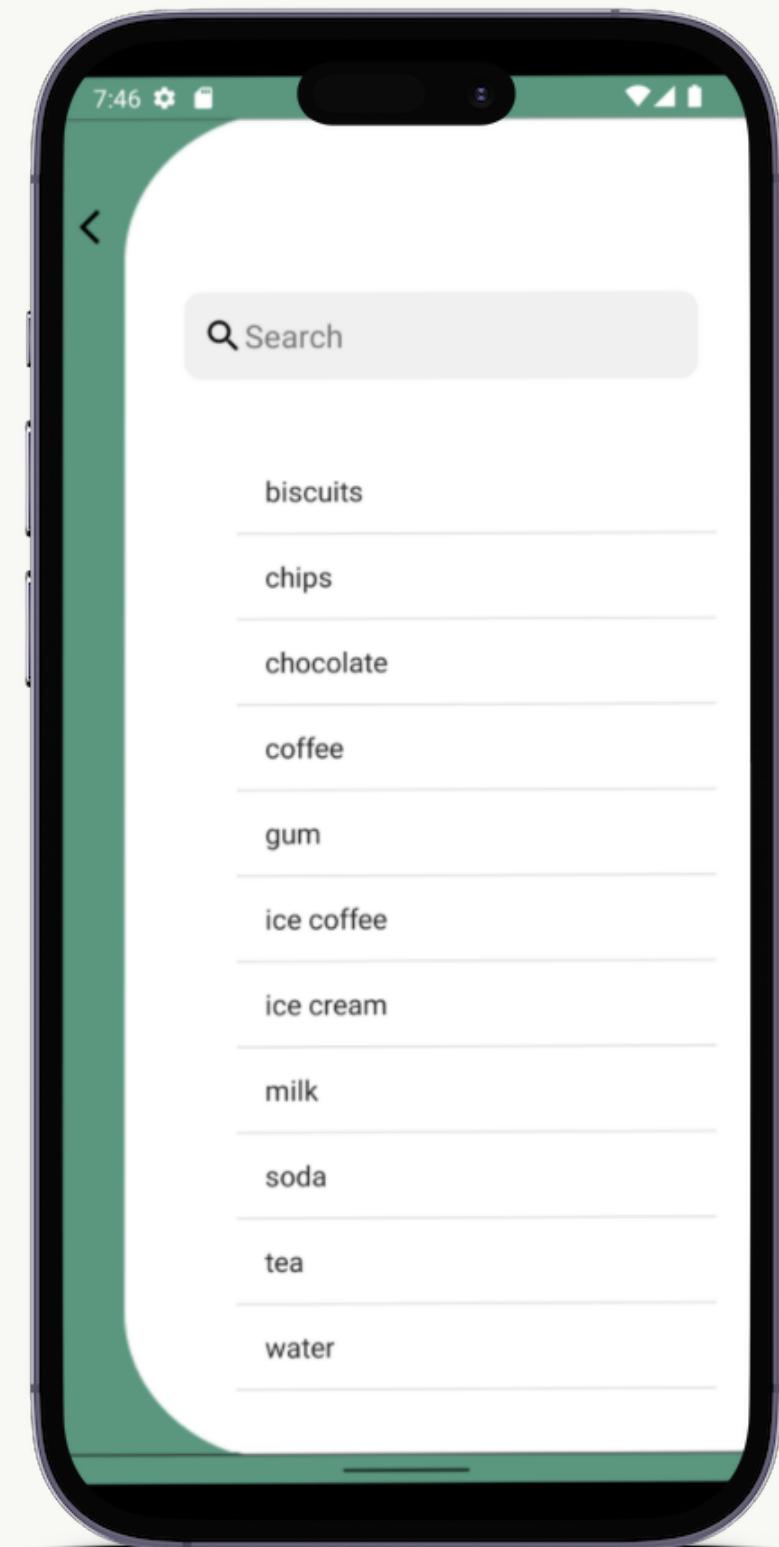
Checkout



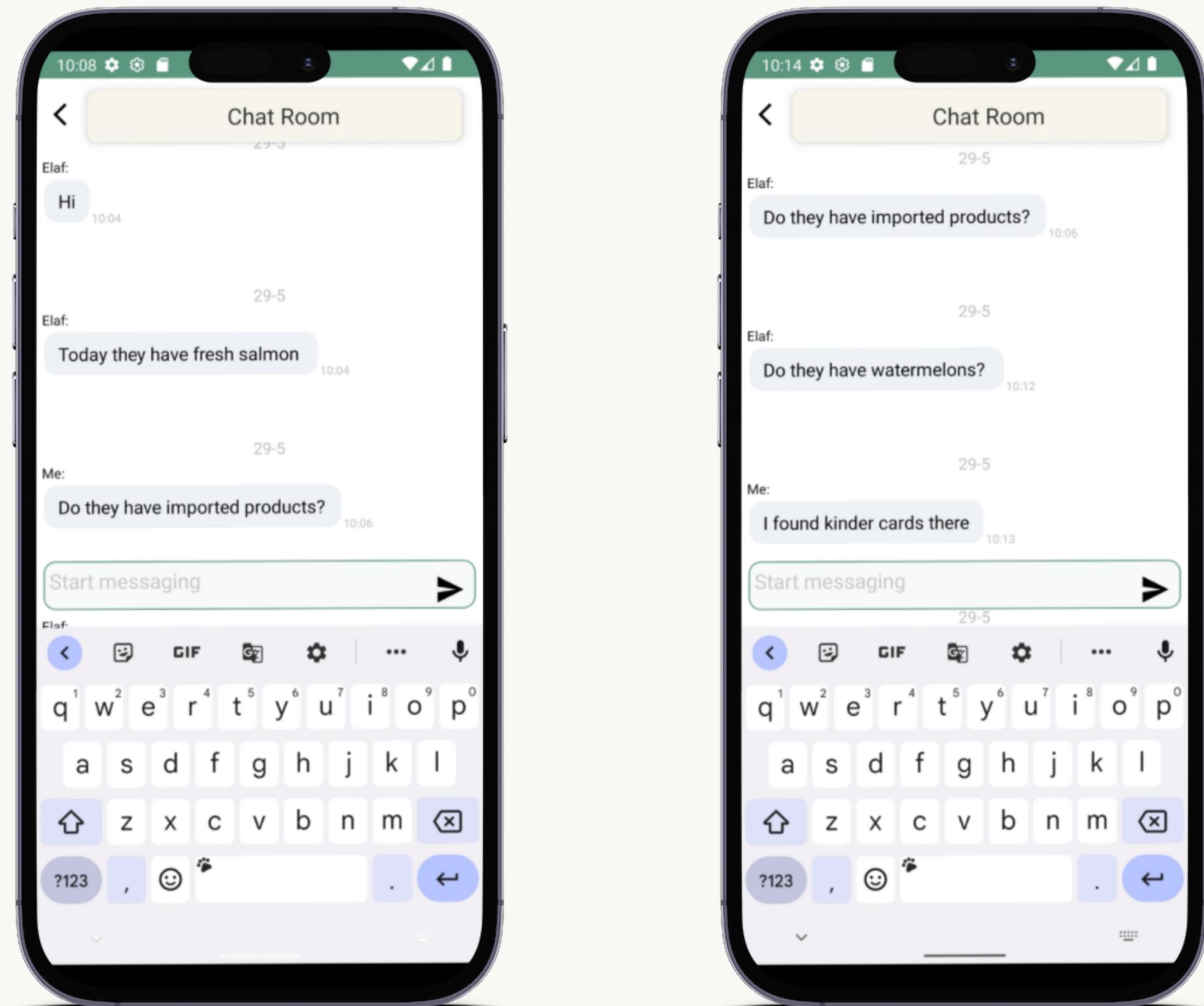
RESULTS



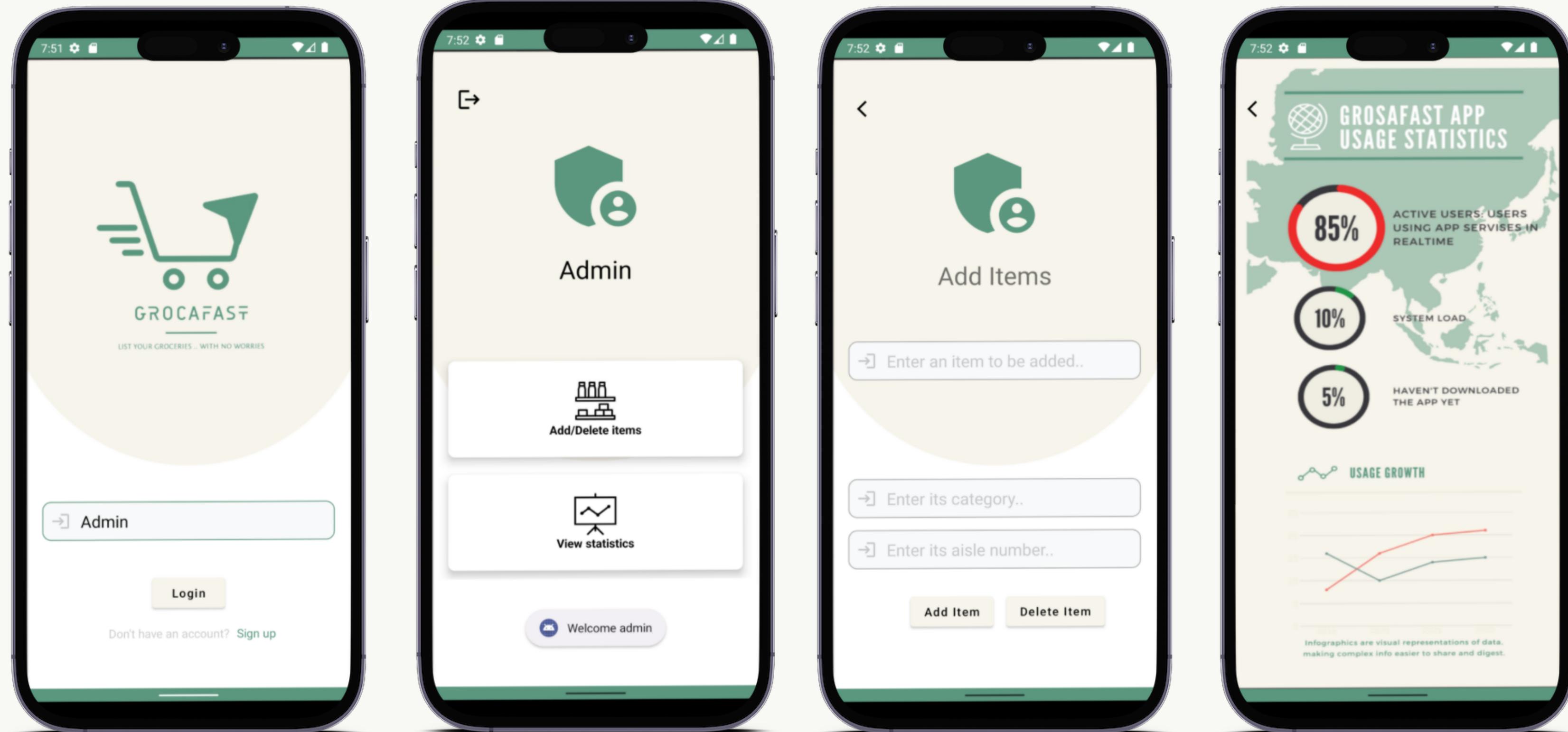
RESULTS



RESULTS



RESULTS



TESTING



GROCAFAS†

LIST YOUR GROCERIES .. WITH NO WORRIES

TESTING

FUNCTIONAL TESTING

UNIT TESTING
INTEGRATION TESTING
SYSTEM TESTING

NON-FUNCTIONAL TESTING

COMPATIBILITY TESTING
USABILITY TESTING
PERFORMANCE TESTING



TESTING

UNIT TESTING

Test case 1: GetTOPThreeItems()

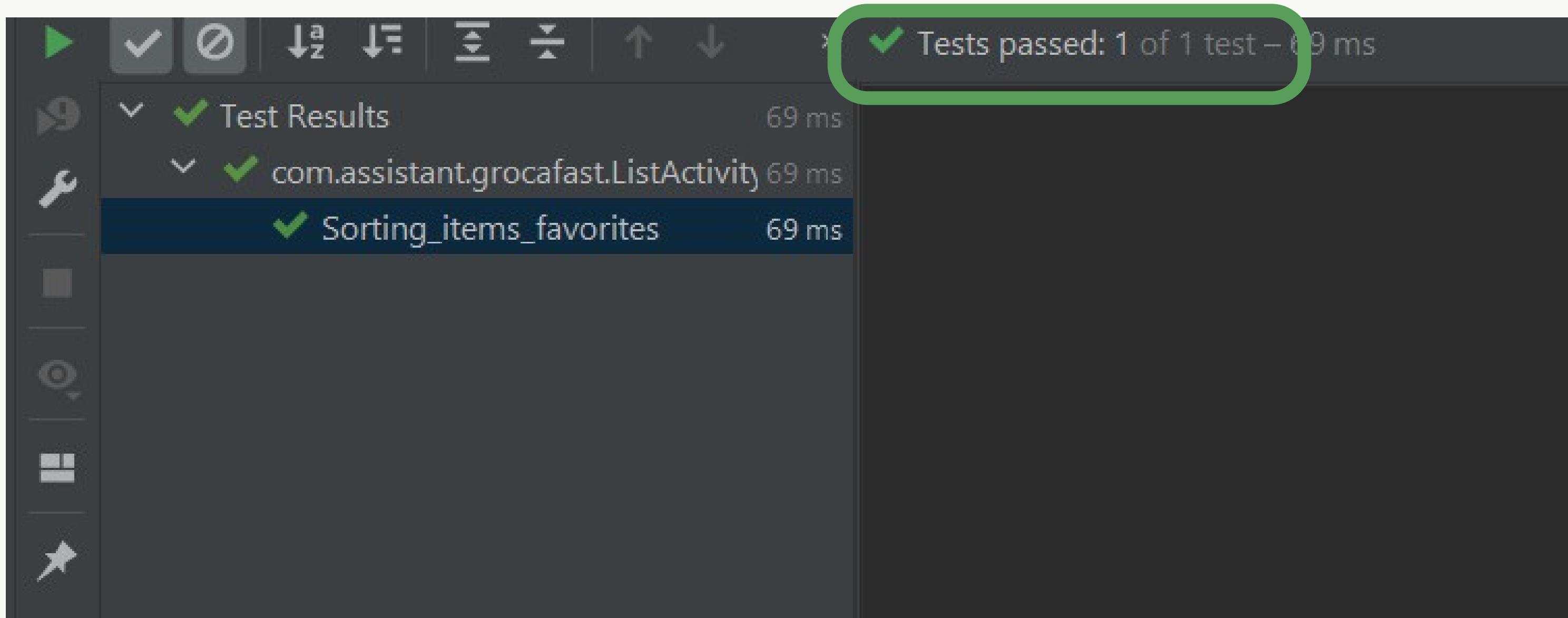
Functionality	Input	Expected result	Result
Check whether the method will return the user's top three most purchased items.	Two input arrays: one with integers for item repetitions and another with strings for the items. (tops[0],items_names[3]) (tops[1],items_names[4]) (tops[2],items_names[2])	The first item will be the most written item: (tops[0],items_names[3])	Pass



TESTING

UNIT TESTING

Test case 1: Result



TESTING

UNIT TESTING

Test case 2 : getPermutations()

Functionality	Input	Expected result	Result
Check whether the method will return permutations of all paths for a given item.	Two inputs. The first array of integers to be tested and the second is the factorial according to a number of items.	The output from the getPermutations() method matches the expected value of both possible permutations of the input array.	Pass



TESTING

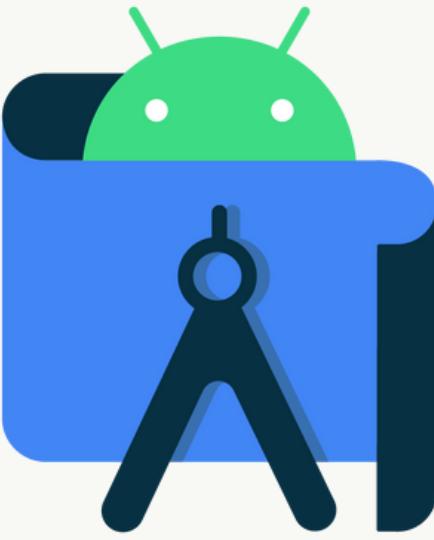
UNIT TESTING

Test case 2 : Result



TESTING

INTEGRATION TESTING



Firebase



TESTING

INTEGRATION TESTING

Adding Dependencies

```
implementation 'com.google.firebaseio:firebase-analytics'  
implementation 'com.google.firebaseio:firebase-auth'  
implementation 'com.google.firebaseio:firebase-database'  
implementation 'com.google.firebaseio:firebase-storage'  
implementation 'com.google.firebaseio:firebase-crashlytics'  
implementation 'com.google.firebaseio:firebase-firebase'
```

```
1 usage  
DatabaseReference databaseReference = FirebaseDatabase.getInstance().getReference( path: "Users");
```



TESTING

INTEGRATION TESTING

Database updated successfully

GROCAFAST ▾

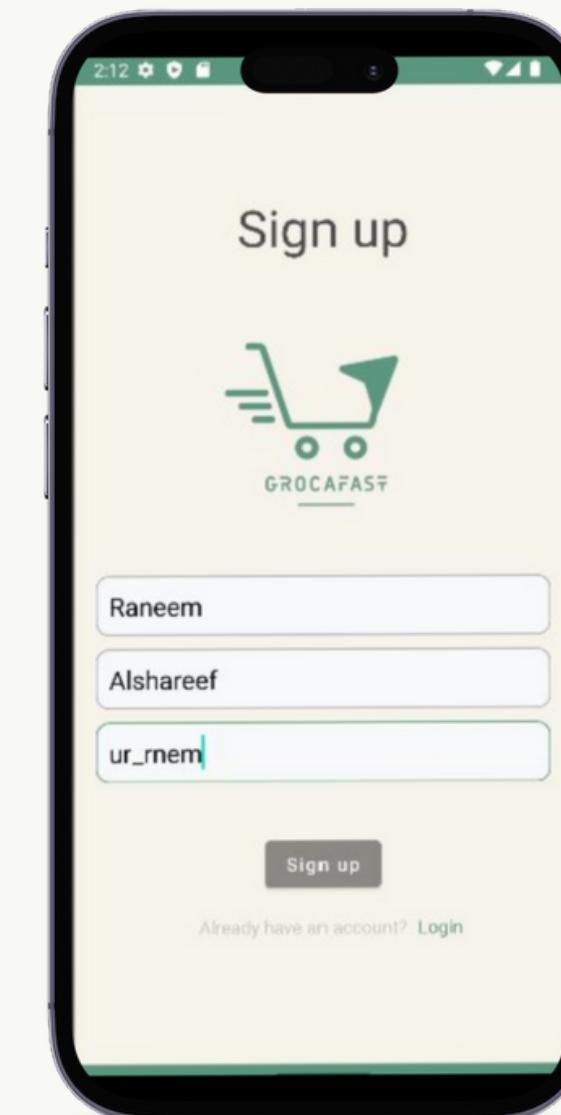
Realtime Database

Data Rules Backups Usage Extensions NEW

https://application-a6598-default-rtdb.firebaseio.com

https://application-a6598-default-rtdb.firebaseio.com/

- Community
- Community_Chat
- Items
- Users
 - Abeer1092
 - Elaf
 - Lolo
 - Loofa
 - Manar12
 - Roro



Before

User sign up

GROCAFAST ▾

Realtime Database

Data Rules Backups Usage Extensions

https://application-a6598-default-rtdb.firebaseio.com

Users

- Abeer1092
- Elaf
- Lolo
- Loofa
- Manar12
- Roro

ur_rnem

- firstName: "Raneem"
- lastName: "Alshareef"
- username: "ur_rnem"

After



TESTING

INTEGRATION TESTING

Test each function with its related functions

Test Case Scenario	Link between pages
Create a grocery list: Create a new account → login → create a grocery list.	Verified
View and modify a previous list: Login → View lists history → View and modify a previous list.	Verified
Obtain the shortest path: Login → Create a grocery list → Obtain the shortest path.	Verified
Search for an item: Login → Search for an item → View item location.	Verified
Receive personalized shopping suggestions: Login → (Have list history) → Create a grocery list → Receive personalized shopping suggestions.	Verified
Create posts: Login → Community space → Create posts.	Verified



TESTING

SYSTEM TESTING

1. User:

Test case scenarios	Result
View splash screen – signup valid username - login - view home page	Pass
View splash screen – login incorrect username - login correctly - view home page	Pass
View splash screen – login correct username - login correctly - view home page	Pass
View splash screen – signup invalid username - signup correctly - login correctly - view home page	Pass
View splash screen – login - view home page – create grocery list service – add valid items – get route map	Pass
View splash screen – login - view home page – create grocery list service – add invalid items	Pass
View splash screen – login - view home page – view lists history service – select previous list – edit list – get route map	Pass
View splash screen – login - view home page – view lists history service – select previous list – delete list	Pass
View splash screen – login - view home page – search service - search for an item – route map	Pass
View splash screen – login - view home page – search service - search for an undefined item – Search for valid item – get route map	Pass
View splash screen – login - view home page – community space service – write comment	Pass



TESTING

SYSTEM TESTING

2. Admin:

Test case scenarios	Result
View splash screen – login incorrect username - login correctly - view home page	Pass
View splash screen – login correct username - login correctly - view home page	Pass
View splash screen – login - view home page – add items service – fill item fields correctly – add item	Pass
View splash screen – login - view home page – add items service – fill item fields invalid input – fill item fields correctly - add item	Pass
View splash screen – login - view home page – delete items service – fill item fields correctly – remove item	Pass
View splash screen – login - view home page – delete items service – fill item fields invalid input – fill item fields correctly - remove item	Pass
View splash screen – login - view home page – view statistics service	Pass



TESTING

COMPATIBILITY TESTING

Device	Android version	Memory space	Memory available	Pass/Fail
Google pixel 2 XL	11	4 GB	612 MB	Pass
Galaxy A10	11	2 GB	314 MB	Pass
ZTE Blade 10 Smart	9	4 GB	1 GB	Pass



TESTING

USABILITY TESTING



TESTING

USABILITY TESTING

Measuring Criteria

Task5: Obtain the shortest path shortest path to collect all items on the grocery list

* **Learnability:** How easy is it for users to accomplish basic tasks the first time they encounter the design

4 3 2 1

Strongly Disagree Strongly Agree

* **Efficiency:** Once users have learned the design, how quickly can they perform tasks

4 3 2 1

Strongly Disagree Strongly Agree

* **Memorability:** When users return to the design after a period of not using it, how easily can they reestablish proficiency

4 3 2 1

Strongly Disagree Strongly Agree

* **?Errors:** Can users easily recover from the errors

4 3 2 1

Strongly Disagree Strongly Agree

* **?Satisfaction:** How pleasant is it to use the design

4 3 2 1

Strongly Disagree Strongly Agree

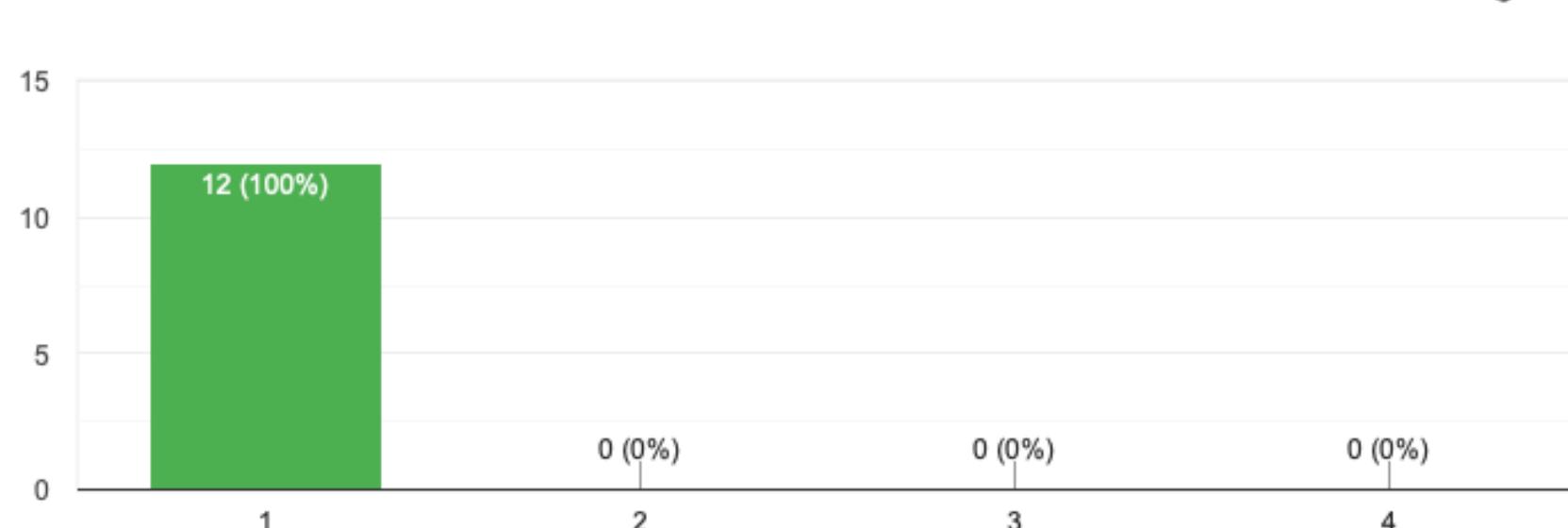
TESTING

USABILITY TESTING

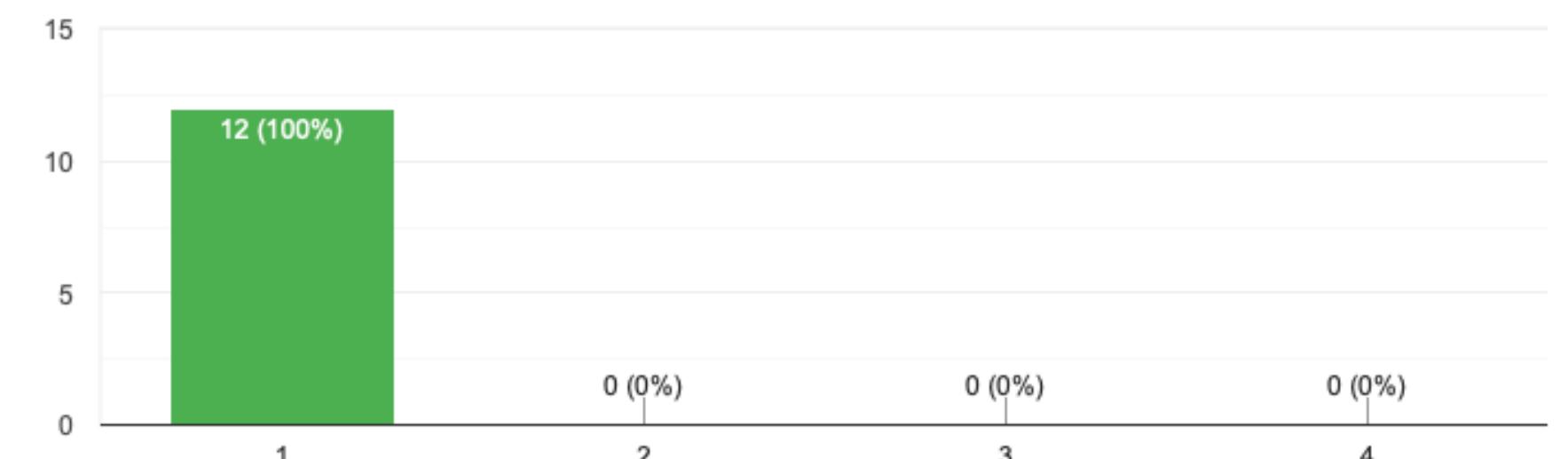
Analyze the acquired information

Task5: Obtain the shortest path shortest path to collect all items on the grocery list

نسخ □ Learnability: How easy is it for users to accomplish basic tasks the first time they encounter the design



نسخ □ Efficiency: Once users have learned the design, how quickly can they perform tasks



TESTING

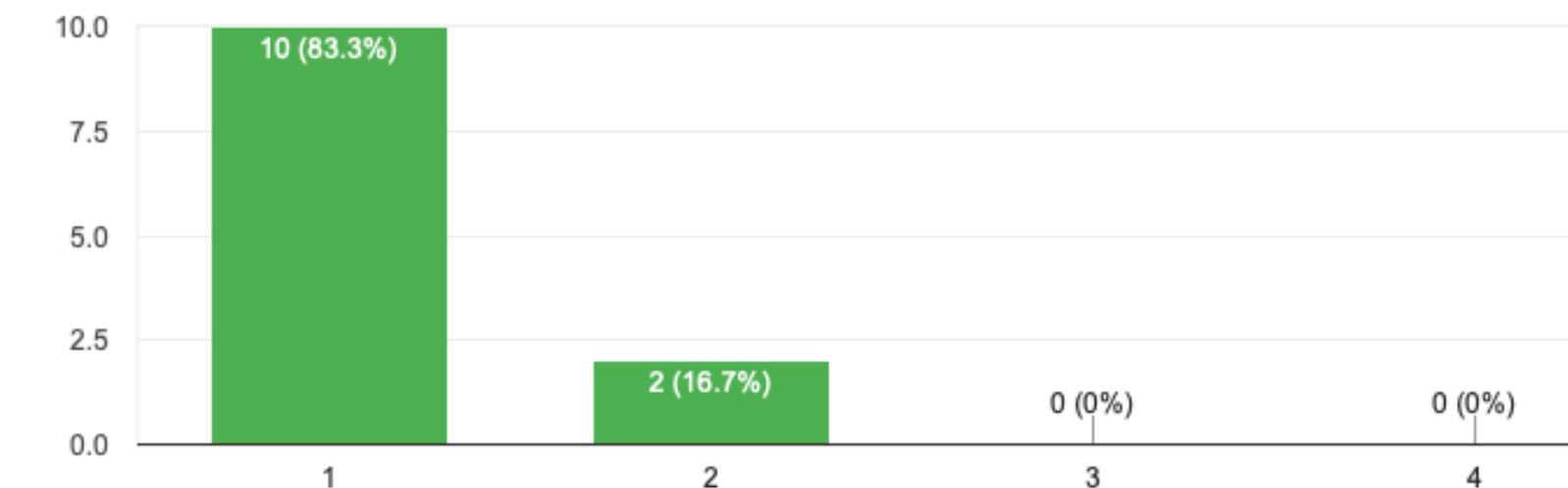
USABILITY TESTING

Analyze the acquired information

نسخ

Memorability: When users return to the design after a period of not using it, how easily can they reestablish proficiency

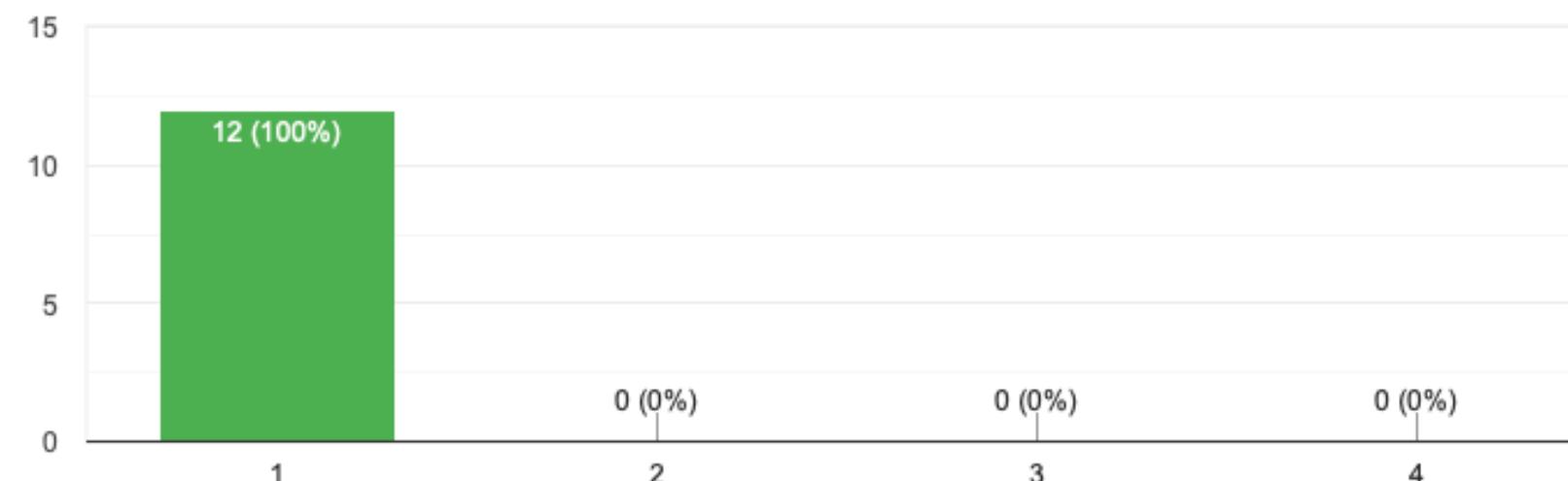
ردا 12



نسخ

?Errors: Can users easily recover from the errors

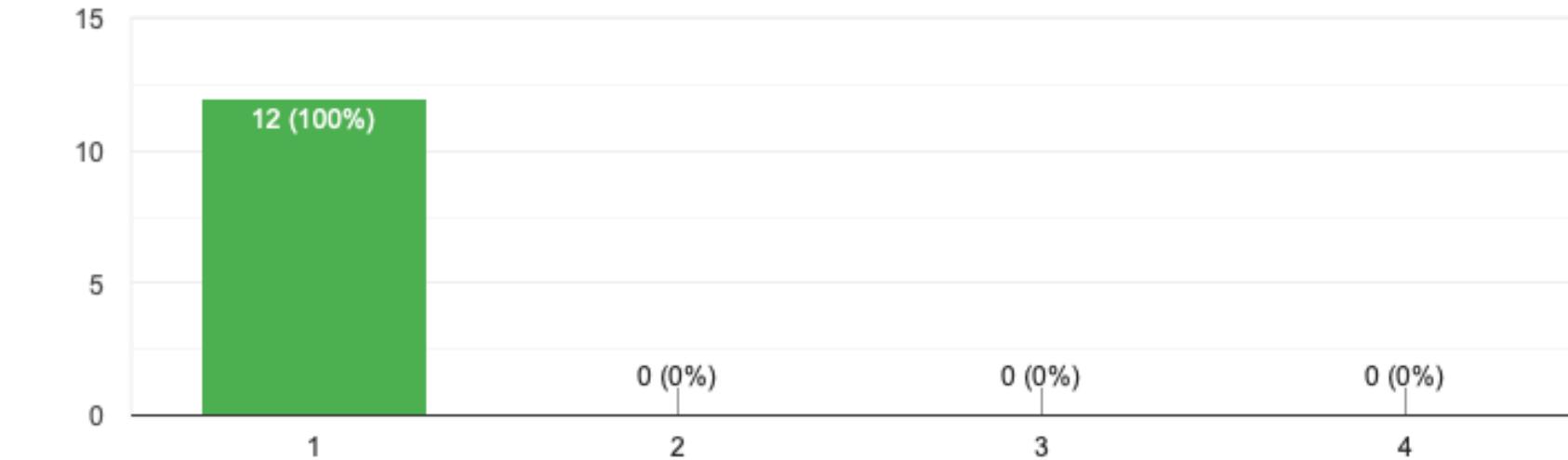
ردا 12



نسخ

?Satisfaction: How pleasant is it to use the design

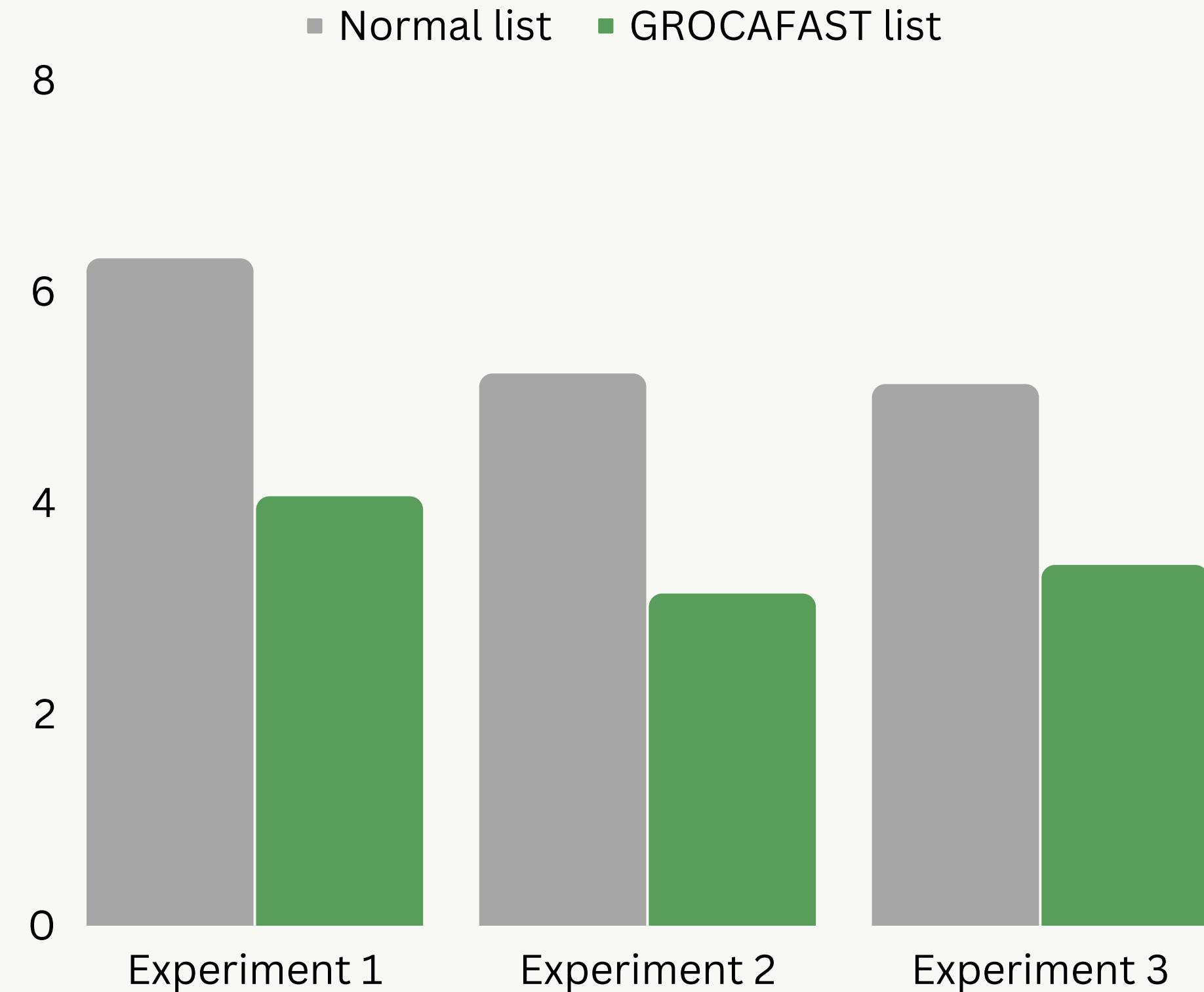
ردا 12



TESTING

PERFORMANCE TESTING

- Experiment 1: 7 items
- Experiment 2: 5 items
- Experiment 3: 5 items



FUTURE WORK



GROCAFAS†

LIST YOUR GROCERIES .. WITH NO WORRIES

FUTURE WORK

- ✓ Integrate with Indoor Atlas.
- ✓ Add more functionality for Admin use.
- ✓ Include more grocery stores.



CONCLUSION



GROCAFAS†

LIST YOUR GROCERIES .. WITH NO WORRIES



GROCAFST

LIST YOUR GROCERIES .. WITH NO WORRIES

Thank You

تم بحمد الله و توفيقه

APPENDEX

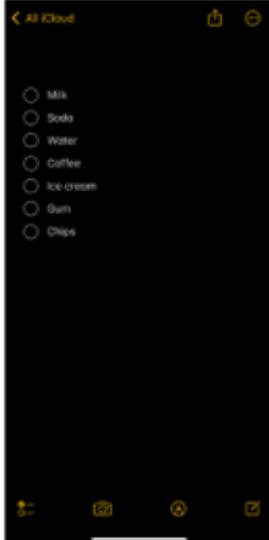
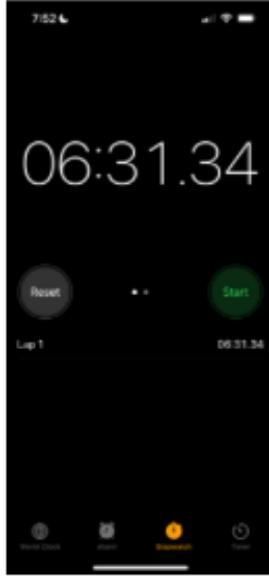
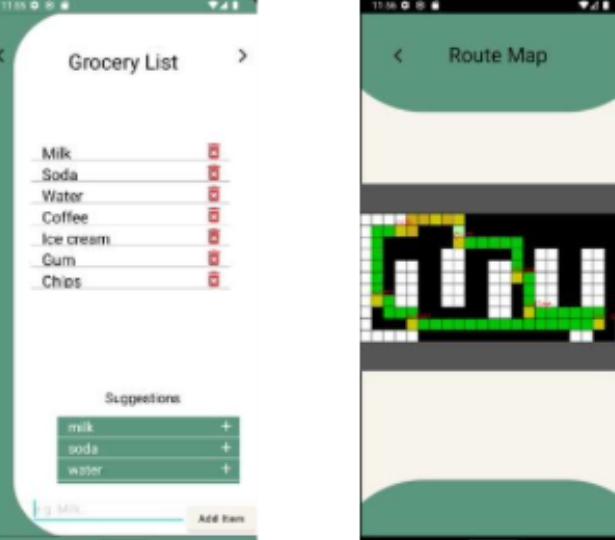
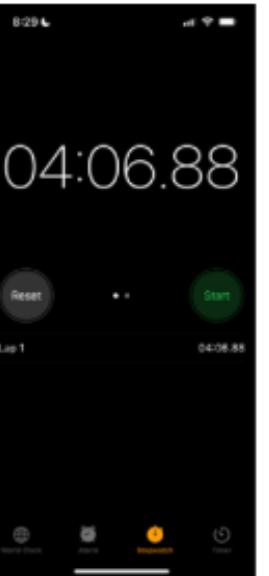
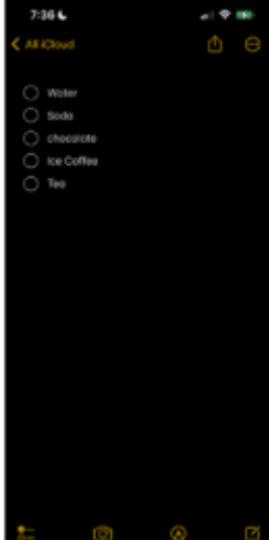
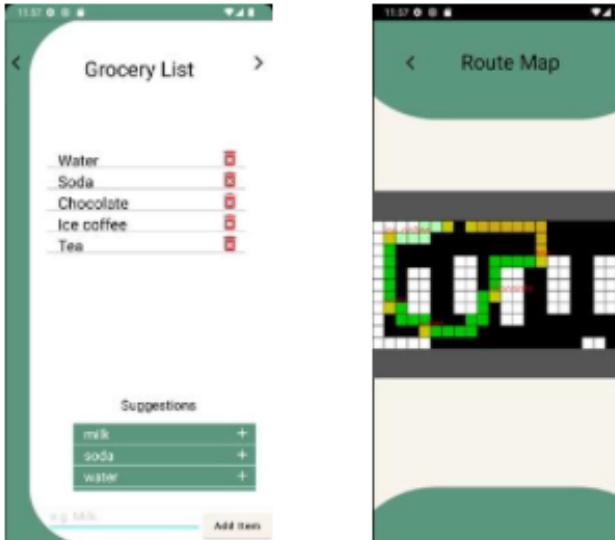
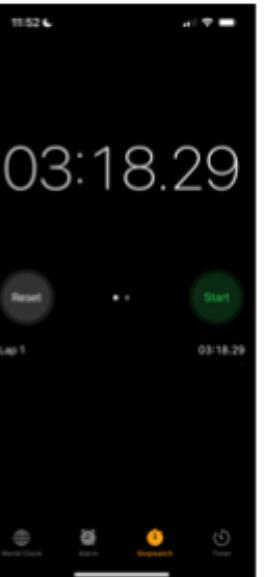
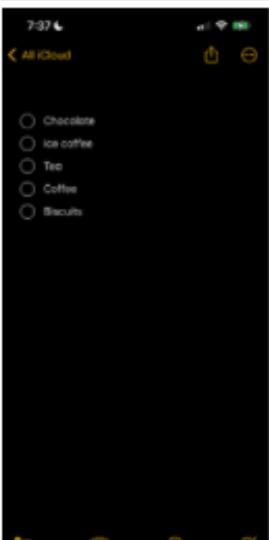
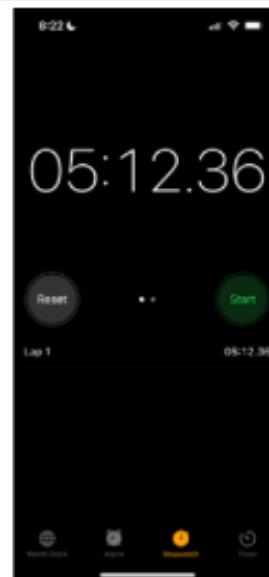
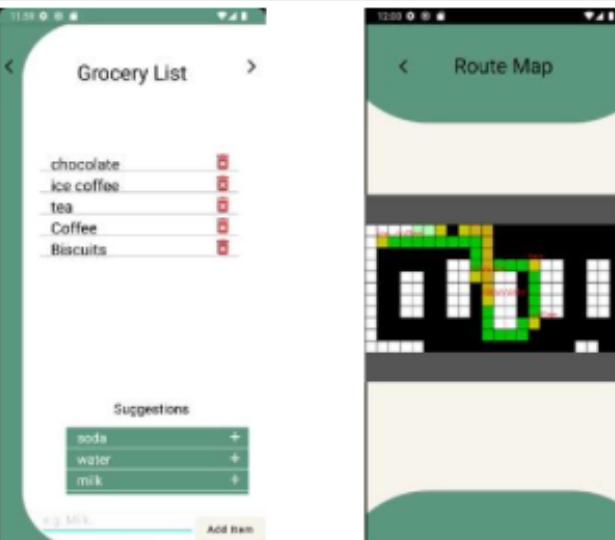
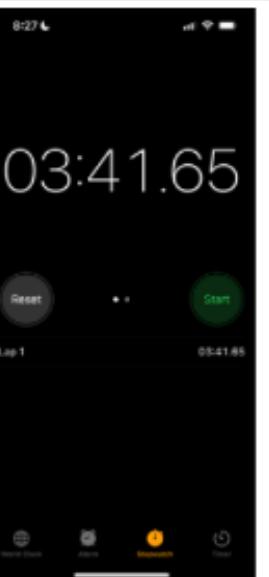
Unit Testing Codes

```
public class ListActivityTest {  
    @Test  
    public void Sorting_itemsFavorites() {  
        int[] items_repeataion = new int[5];  
        items_repeataion[0]=0;  
        items_repeataion[1]=0;  
        items_repeataion[2]=3;  
        items_repeataion[3]=5;  
        items_repeataion[4]=4;  
        String[] items_names = new String[5];  
        items_names[0] = "item0";  
        items_names[1] = "item1";  
        items_names[2] = "item2";  
        items_names[3] = "item3";  
        items_names[4] = "item4";  
        String[] tops = ListActivity.GetTOPThreeItems(items_repeataion, items_names);  
        assertEquals(tops[0], items_names[3]);  
        assertEquals(tops[1], items_names[4]);  
        assertEquals(tops[2], items_names[2]);  
    }  
}
```

```
@Test  
public void testGetPer() {  
    int[] inputArray = {1,2};  
    int factorial = 2; // 2! = 4, since there are 2 elements in the input array  
  
    int[][] expectedOutput = {  
        {1,2},  
        {2,1}  
    };  
    int[][] actualOutput = GetAllPermutations.getPermutations(inputArray, factorial);  
    assertEquals(expectedOutput, actualOutput);  
}
```



Performance Testing

Experiment #	Items #	Normal List (iPhone)	Time Elapsed	GROCAFAST List	Time Elapsed	Difference in Time
1	7 items					2 min and 25 seconds approximately
2	5 items					2 min and 4 seconds approximately
3	5 items					2 min and 11 seconds approximately