Project Part 1 Documentation

Bhavneet Singh Rajpal
bsr1@sfu.ca, 301435220

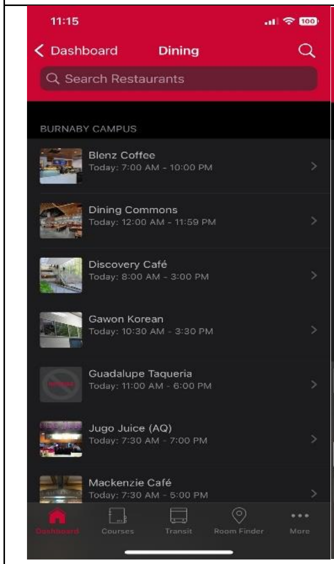
Part 1a

#: 1

Problem/Good: Usability Problem

Name: Ineffective search tool (no use of filter)

Relevant Heuristic: Flexibility & Efficiency of Use



BURNABY CAMPUS

Blenz Coffee
Today: 7:00 AM - 10:00 PM

Dining Commons
Today: 12:00 AM - 11:59 PM

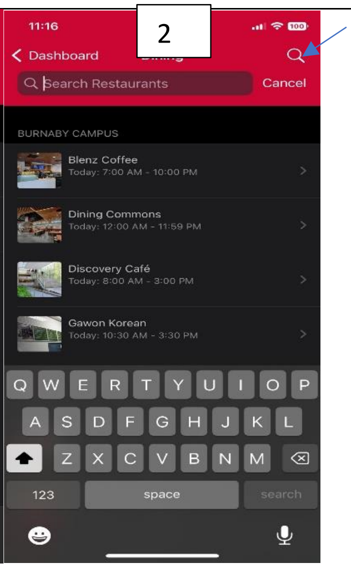
Discovery Café
Today: 8:00 AM - 3:00 PM

Gawon Korean
Today: 10:30 AM - 3:30 PM

Guadalupe Taqueria
Today: 11:00 AM - 6:00 PM

Jugo Juice (AQ)
Today: 7:30 AM - 7:00 PM

Mackenzie Café
Today: 7:30 AM - 5:00 PM



BURNABY CAMPUS

Blenz Coffee
Today: 7:00 AM - 10:00 PM

Dining Commons
Today: 12:00 AM - 11:59 PM

Discovery Café
Today: 8:00 AM - 3:00 PM

Gawon Korean
Today: 10:30 AM - 3:30 PM

Evidence of Issue:
The issue is found in the dining feature as shown in image 1 and 2. Selecting the search utility shows up “Search Restaurants” rather than the app having more filter options for food.

Detailed Explanation:
The SFU snap app provides plenty of restaurants for users to select from but only allows users to search for a restaurant by its name, restricting them from exploring. If a user is new to campus and is craving a particular cuisine, the person might have to go through each restaurant hence increasing the time spent and decreasing the efficiency and effectiveness of the app. A new student doesn’t know the campus well, they might be exhausted going through each restaurant to find their fav cuisine. This leads it to violate the heuristic of “Flexibility & Efficiency of Use”.

Severity or Benefit (Low, Medium, High):
3 (major)

Justification:
This issue is major because the filter utility is needed by the user to reduce the time spent on finding the right restaurant. Other than that the app is running perfectly without interrupting the other features. Having a filter eases a user’s findings and does the task as timely as possible for an app.

Possible Solution and/or trade-offs:
A possible solution is to introduce a filter search which can take user input for location, expensiveness, ratings and food type. Each restaurant could have its menu displayed inside the dining application more like a food delivery app but without the delivery option.

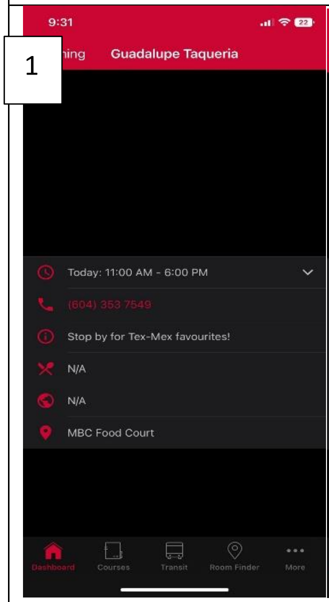
#: 2

Problem/Good: Usability Problem

Name: Tapping on the number for Guadalupe Taqueria doesn’t do anything

Relevant Heuristic: Help users recognize, daignose, and recover from errors

1


9:31

ing Guadalupe Taqueria

Today: 11:00 AM - 6:00 PM

(604) 353 7549

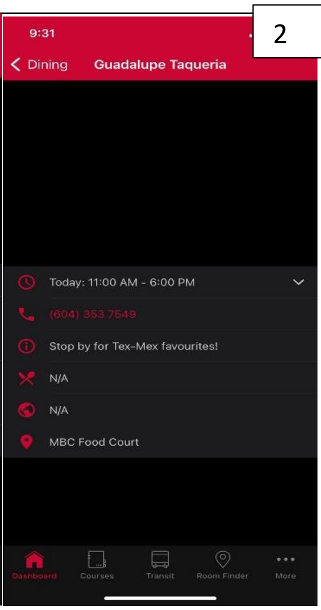
Stop by for Tex-Mex favourites!

N/A

N/A

MBC Food Court

2


9:31

Dining Guadalupe Taqueria

Today: 11:00 AM - 6:00 PM

(604) 353 7549

Stop by for Tex-Mex favourites!

N/A

N/A

MBC Food Court

Evidence of Issue:
Each of these images were taken before and after tapping on the phone number for the restaurant. Nothing shows up on the screen as shown by image 2.

Image 1- Before

Image 2- After

Detailed Explanation:
The SFU dining feature has some flaws with its design, take for example the number for Guadalupe Taqueria. The app doesn’t respond to double-tapping of the number which violates “Help users recognize, diagnose, and recover from errors”. The user should be presented with an error message whenever they are interrupted or discomforted while using the app. These error messages hold a key to smooth interaction between the app and the interface, and not having them pop up at valuable moments makes an app unreliable and difficult to use.

Severity or Benefit (Low, Medium, High):
2 (Minor)

Justification:
The issue is minor because the other restaurants don’t produce any errors and work perfectly fine. This issue doesn’t cause the app’s other interface to be interrupted and could be fixed with a little software update. This could have a critical issue if the user was not able to do anything after double-tapping the number.

Possible Solution and/or trade-offs:
Any error or problem in an interface should always have a report generated so that a user can be assured that it will be fixed and the author could be notified.

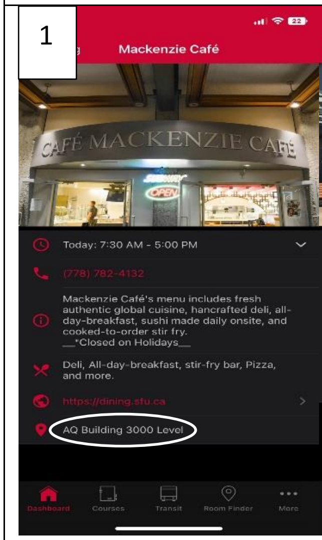
#: 3

Problem/Good: Good Usability

Name: Location of each restaurant given.

Relevant Heuristic: Match between system and the real world

1



Mackenzie Café

Today: 7:30 AM - 5:00 PM

(778) 782-4132

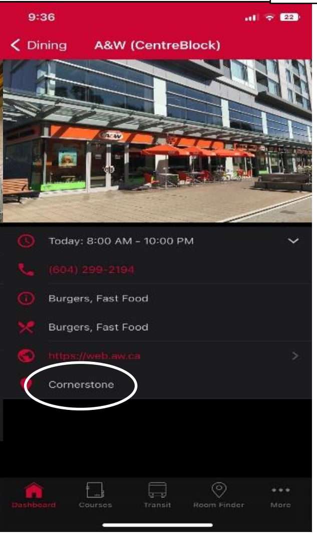
Mackenzie Café's menu includes fresh authentic global cuisine, hancrafted deli, all-day-breakfast, sushi made daily onsite, and cooked-to-order stir fry.
__*Closed on Holidays__

Deli, All-day-breakfast, stir-fry bar, Pizza, and more.

https://dining.sfu.ca

AQ Building 3000 Level

2



9:36

Dining

A&W (CentreBlock)

Today: 8:00 AM - 10:00 PM

(604) 299-2194

Burgers, Fast Food

Burgers, Fast Food

https://web.aw.ca

Cornerstone

Evidence of Issue:
Each of the circled locations are SFU terms.

Detailed Explanation:
Image 1 shows how each of the restaurants has its location provided in the app to help the users find it. But what stands out is the location provided are terms familiar to Sfu students (the target audience). Using terms like "MBC Food Court" or "AQ Building 3000 Level" rather than Google locations makes it easier for the user to look it up on the room finder interface's map on the same app. This simple yet brilliant natural way of representing a location matches the thoughts of a user and connects with them comfortably.

Severity or Benefit (Low, Medium, High):
4 (Critical)

Justification:
The app effectively uses Sfu's jargon to tell the location of a restaurant place an important role because firstly it's efficient and secondly provides information quickly without confusing the user.

Possible Solution and/or trade-offs:
A possible upgrade is that users who are new to campus might not be able to interact quickly so a possible solution is to add affordance to the location part so that we can link it to google maps. The transit interface is an example of this.

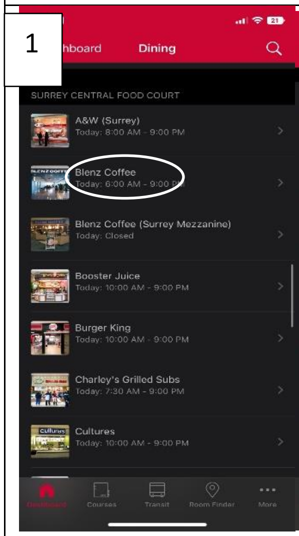
#: 4

Problem/Good: Good Usability

Name: Timings shown below a restaurant.

Relevant Heuristic: Visibility of system status

1



SURREY CENTRAL FOOD COURT

A&W (Surrey)
Today: 8:00 AM - 9:00 PM

Blenz Coffee
Today: 6:00 AM - 9:00 PM

Blenz Coffee (Surrey Mezzanine)
Today: Closed

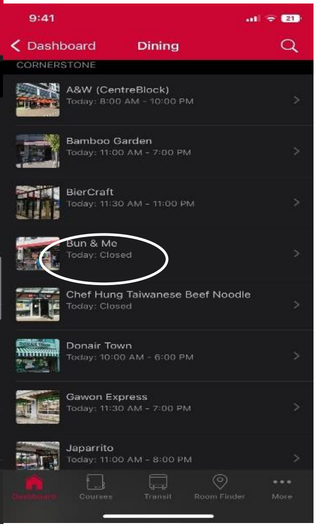
Booster Juice
Today: 10:00 AM - 9:00 PM

Burger King
Today: 10:00 AM - 9:00 PM

Charley's Grilled Subs
Today: 7:30 AM - 9:00 PM

Cultures
Today: 10:00 AM - 9:00 PM

2



CORNERSTONE

A&W (CentreBlock)
Today: 8:00 AM - 10:00 PM

Bamboo Garden
Today: 11:00 AM - 7:00 PM

BierCraft
Today: 11:30 AM - 11:00 PM

Bun & Me
Today: Closed

Chef Hung Taiwanese Beef Noodle
Today: Closed

Donair Town
Today: 10:00 AM - 6:00 PM

Gawon Express
Today: 11:30 AM - 7:00 PM

Japarrito
Today: 11:00 AM - 8:00 PM

Evidence of Issue:
Each of the circled timings show how the system keep the user up to date.

Detailed Explanation:
The SFU snap app helps keep a clear vision of what restaurants are currently open and their timings. Each restaurant has its timing listed below their names so that the user is up to date with the restaurant's current timings. As soon as the user opens the dining feature the users are presented with and made aware of today's timings which helps them to decide what restaurants to look through and manage time efficiently. A user who is craving food might not have to go through each restaurant to view its availability but instead could just view it before. The user doesn't have to remember what days the restaurant might be open instead the person could just check the app which makes less load on their memory.

Severity or Benefit (Low, Medium, High):
3 (Major)

Justification:
The app makes sure that the user is presented with timings as soon as the restaurant's name is clicked so this feature falls into the minor category. But the fact can't be ignored that presenting it under the restaurant's name helps make decisions faster for users and increases the reliability of the app thus changing the category to major.

Possible Solution and/or trade-offs:
Having important timings is already an advantage for the user but one upgrade would be to highlight them: 1) open can be green and 2) closed has expected red.

Summary

Part 1a

Part 1a of the assignment was conducted on the SFU Snap app's dining feature. The SFU snap at the time of the evaluation was on version 2.14 and was conducted by a student (user). Each of the usability errors or good usability was found so that it could ease a student's interaction with the system. Screenshots were taken of the dining feature as evidence of these usabilities. All the evaluations were done at home. I never thought I could find more errors in a professional app but designing a user-centred interface is difficult. The finding from the Heuristic evaluations indicates that the dining feature has two errors which are categorized into minor and major. The first error is no filter option available for a user so that the app could be time efficient, which directly violates the "Flexibility & Efficiency of Use". This error is from Assignment 1. Second, no error message is displayed when there is no feedback on double tapping the phone number for Guadalupe Taqueria. This error is similar to my app crashing evaluation from assignment 1 but it is a minor issue. But the fact can't be ignored that a user might become uncomfortable if no response is provided by the app. There are 2 good findings in the app which contribute to its good usability. The terms used for locations help students as they might be familiar with the term rather than the actual location. Today's timing is shown below each restaurant name to cut students' time exploring each availability of the restaurant.

Context Identification: Two SFU students would be carrying out tasks related to the dining app and each user will perform a different task so has to make good use of the SFU snap app. Each of the tasks is being carried out at the SFU Burnaby campus near SWH 10041 to keep a starting location and it is being done in the afternoon. Each of the SFU students' tasks is separate and the pre-requisite for them is that the SFU Dining feature is already open. One of them must find its favourite cuisine through the search option and can take any cuisine and finds its restaurant's location and both should be done promptly.

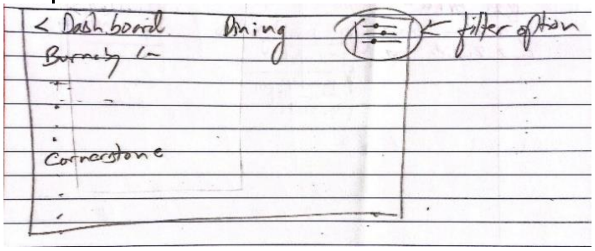
User Identification: As stated in the context in context identification we have two SFU students performing a task. Each of these students has their own personality and characteristic which might affect their app usage.

- **Person 1**: Michael Scott is a 22-year-old SFU 3rd year student who is studying computer science. He loves coding but written assignments bite him. He loves eating at different places and tends to explore different cuisine available at a restaurant. Cooking is never an option for Michael. He loves to play video games with his friends and loves partying on weekends. Task:
  - **Overall Task**: Search for a Fast-Food Restaurant
  - **User**: Michael Scott
    - **Sub-task 1**: Tried searching for it by typing "Fast Food" in the search option.
  - **Consequences**: No results displayed. Michael becomes frustrated and just closes the app to look for alternatives to not waste his time.

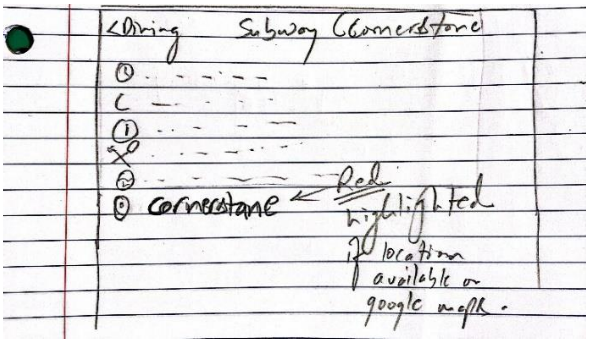
- **Person 2**: Carl Johnson is a 19-year-old SFU 1st year student. He is currently deciding on switching his major from computing to business as he is currently not doing so well in his courses. He is new to Canada and lives on campus, so he doesn't know the area well. He becomes nervous and confused when he must multitask. He tends to use the SFU snap app a lot for his daily routine. He can't cook properly as he is not that experienced. Task:
    - **Overall Task**: Find the location of a restaurant and set up a navigational route.
    - **User**: Carl Johnson
        - **Sub-Task 1**: Search for his cuisine by going through each restaurant.
        - **Sub-Task 2**: Found a perfect fit for his type.
        - **Sub-Task 3**: Find the location.
        - **Sub-Task 4**: Is not familiar with the SFU jargon so must look up into room finder interface.
        - **Sub-Task 5**: If not found on room finder he must go to Google map service and type the name of the restaurant. (Some of the locations like Central City Food Court are not available on room finder)
    - **Consequences**: Constant switching between room finder, google maps and dining features make Carl uncomfortable and hesitant to use the app again. So, Carl decides to order food from other delivery apps wasting his resources on delivery.

Functional Requirements:

1. SFU Dining feature should give a filter option for the users so that each user can explore and find their favourite cuisine.



2. Location of each restaurant should be highlighted to add affordance to it so that it can be linked to the map making interaction quicker and smoother. For example, double-clicking over the Burnaby location in the filter search would just show all the available restaurants in that area.



Non-Functional Requirements (Related to Functional Requirements)

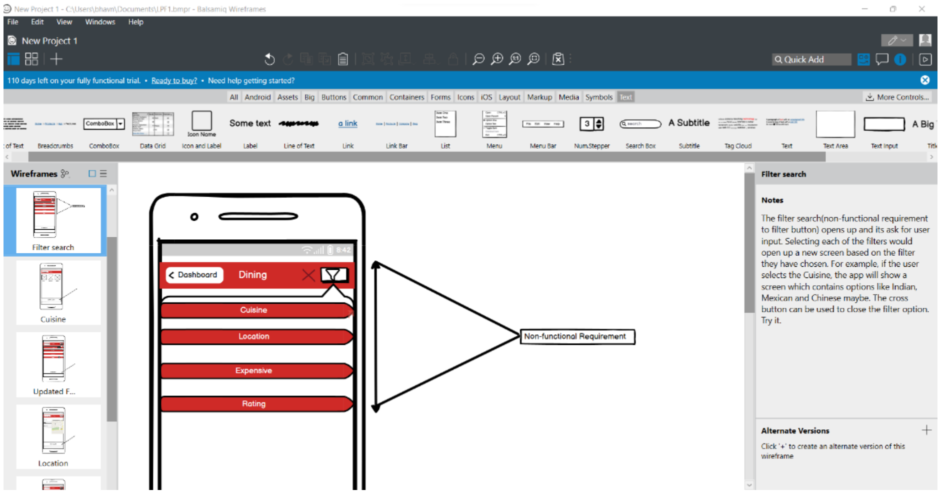
1. **Related to Point 1 of Functional Requirements**
    a. The filter search should take input from the users for location, expensiveness, rating, and the most important type of cuisine. These filter options will make a user's search more effective and on point rather than letting them waste around searching for the right restaurant. Michael would benefit from this feature the most as he loves trying different cuisines, at different locations.
2. **Related to Point 2 of Functional Requirements**
    a. The location can be highlighted so that when clicked the location would be added to Google Maps creating a navigational route. Like the transit interface of the app. Once the location is clicked a new page opens showing the exact location. Double-clicking over the Burnaby location in the filter search would show all the available restaurants in that area. This would help Carl a lot because of him not know the area so well. But the map in the filter search can't be interacted with.

Next Step

After collecting the functional and non-functional requirements the next step is to design a prototype. Tasks performed by users should always be recorded so that they can be further analyzed to break down user requirements. Once the developer has their requirements, they can start developing prototypes. Prototypes help Each design a developer makes needs to be evaluated by stakeholders or users themselves to know whether the design or prototype does the work it is supposed to be doing and if the user is satisfied with the outcome. Each prototype developed should be evaluated depending on the environment it is going to be used in, what type of user is going to use the design and whether is there a level of control that is needed. If the outcome of the evaluation is not satisfactory, then a developer must start again by collecting more requirements that are stronger and better than the previous one. When implementing a new design or interface into a system, a prototype must be developed to ensure it can be evaluated by different groups and meets their requirements. A developer's designed interface should be reliable, valid, bias and should have a scope in it. The next step describes above is the iterative meaning if user requirements are not met, the developer must start over again. Even while making improvements to the existing interface, the developer must follow the UCD iterative process: to not get ahead of themselves.

Part 1b: Low-Fidelity Prototypes

LFP1



Filter search

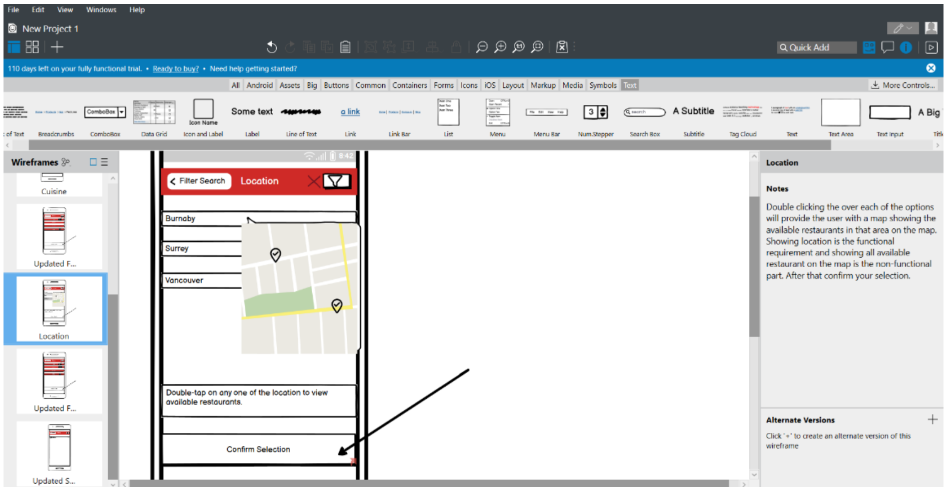
Notes

The filter search(non-functional requirement to filter button) opens up and its ask for user input. Selecting each of the filters would open up a new screen based on the filter they have chosen. For example, if the user selects the Cuisine, the app will show a screen which contains options like Indian, Mexican and Chinese maybe. The cross button can be used to close the filter option. Try it.

Alternate Versions

Click '+' to create an alternate version of this wireframe

Image 1



Location

Notes

Double clicking the over each of the options will provide the user with a map showing the available restaurants in that area on the map. Showing location is the functional requirement and showing all available restaurant on the map is the non-functional part. After that confirm your selection.

Alternate Versions

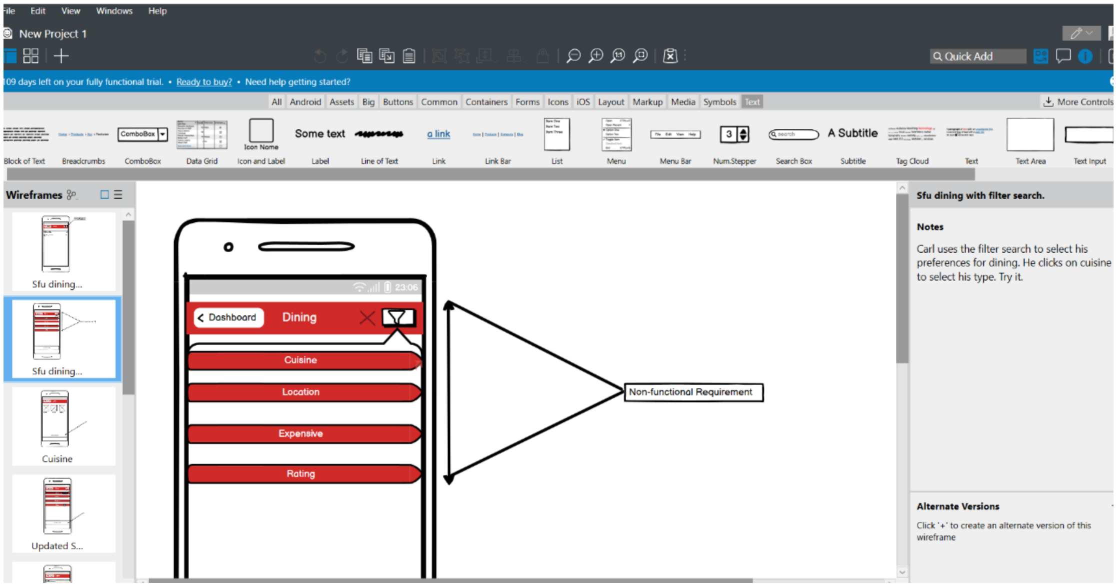
Click '+' to create an alternate version of this wireframe

Image 2

Image 1 shows the filter option implemented into the interface with its non-functional requirements of having filter preferences. The 2nd image shows the implementation of 2nd functional requirement of showing the location to the user. As the user double-clicks any one of the locations, a small map google map or transit map appears on the screen with all the available restaurants for that location. This is my 2nd non-functional requirement. This is Michael's Point of view as he just needs to find the restaurant with his cuisine. To complete the functional requirement I added location, cuisine, expensiveness, and rating to it, which are the non-functional requirement. Locations as you can see from image 2 we could add a transit map to complete the functional requirement of location.

- **Functional requirement**: Filter, **Non-functional requirement**: Having only 4 filter searches.
- **Functional requirement**: Having location displayed, **Non-functional requirement**: Display of all restaurants in that area.

LFP2

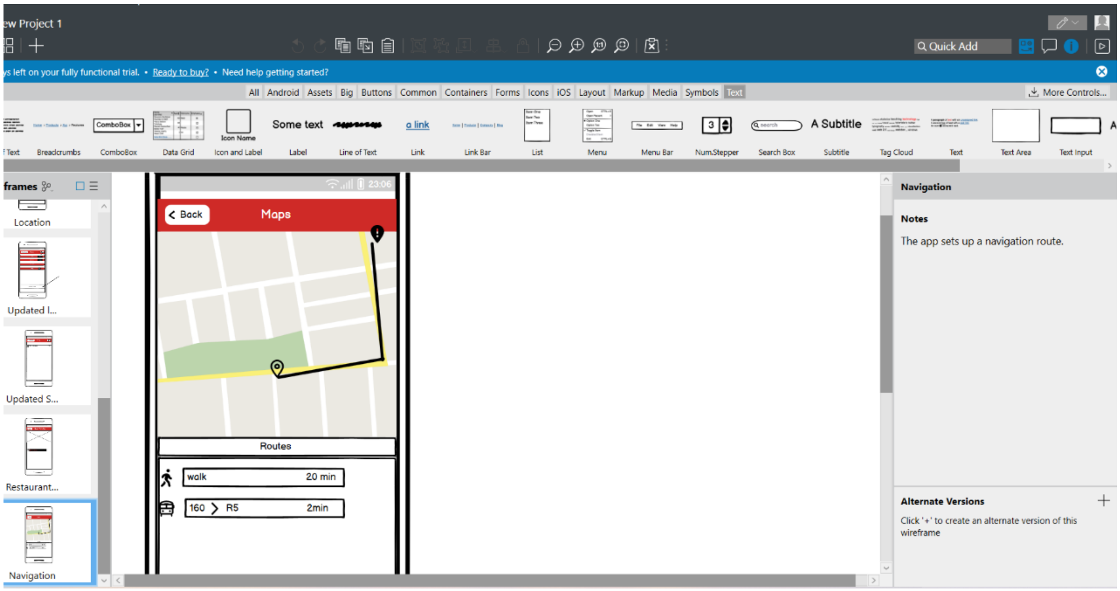


Sfu dining with filter search.

Notes

Carl uses the filter search to select his preferences for dining. He clicks on cuisine to select his type. Try it.

Image 1


ew Project 1
Quick Add
s left on your fully functional trial. • Ready to buy? • Need help getting started?
All Android Assets Big Buttons Common Containers Forms Icons iOS Layout Markup Media Symbols Text
More Controls...
frames
Location
Updated I...
Updated S...
Restaurant...
Navigation
Back
Maps
Routes
walk
20 min
160
R5
2min
Alternate Versions
Click '+' to create an alternate version of this wireframe

Navigation

Notes

The app sets up a navigation route.

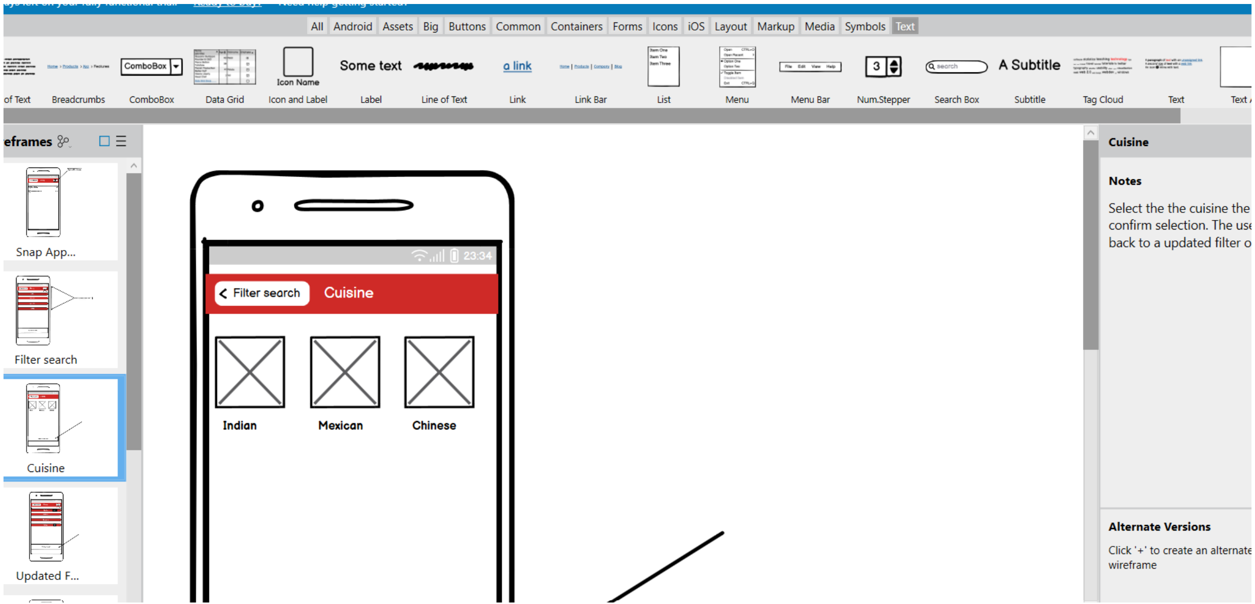
Image 2

Sketches created in LFP2 are based on Carl's point of view. The functional requirement of having a filter is implemented the same for the LFP but here Carl uses the 2nd functional requirement to help him set up a navigation route.

- **Functional requirement**: Filter, **Non-functional requirement**: Having only 4 filter searches.
- **Functional requirement**: Having location displayed, **Non-functional requirement**: setting up a navigation path between the user and the location.

Appendix

LFP1



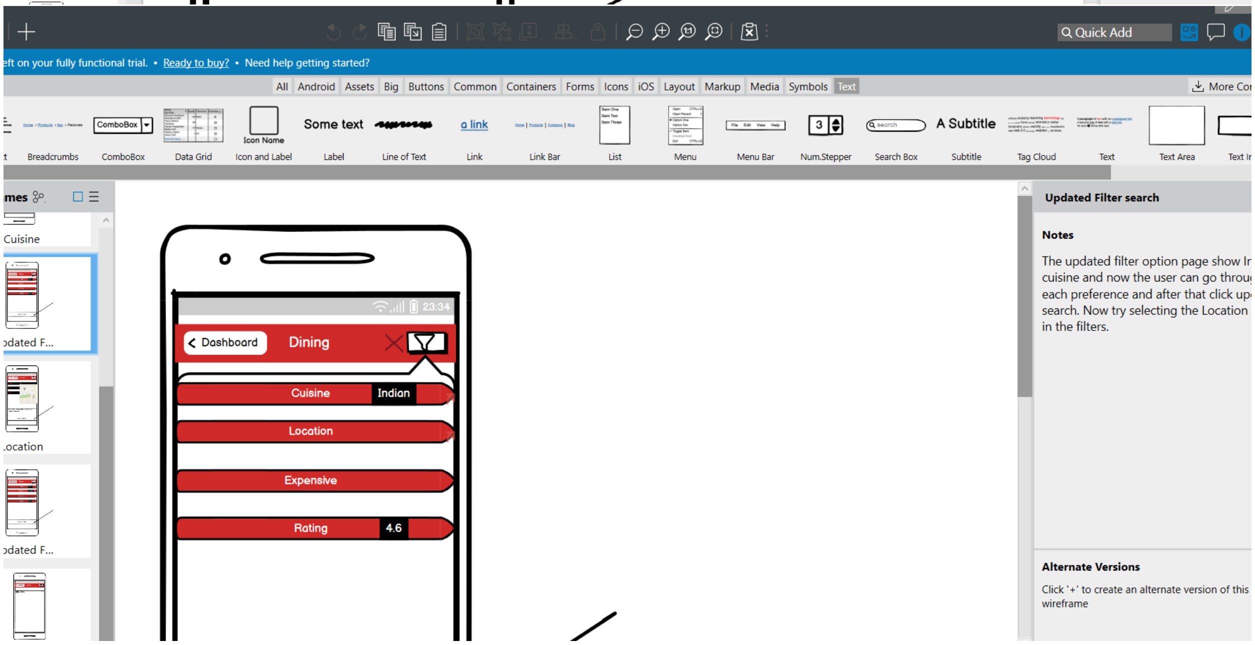
Cuisine

Notes

Select the the cuisine the confirm selection. The use back to a updated filter o

Alternate Versions

Click '+' to create an alternate wireframe



eft on your fully functional trial. • Ready to buy? • Need help getting started?

Updated Filter search

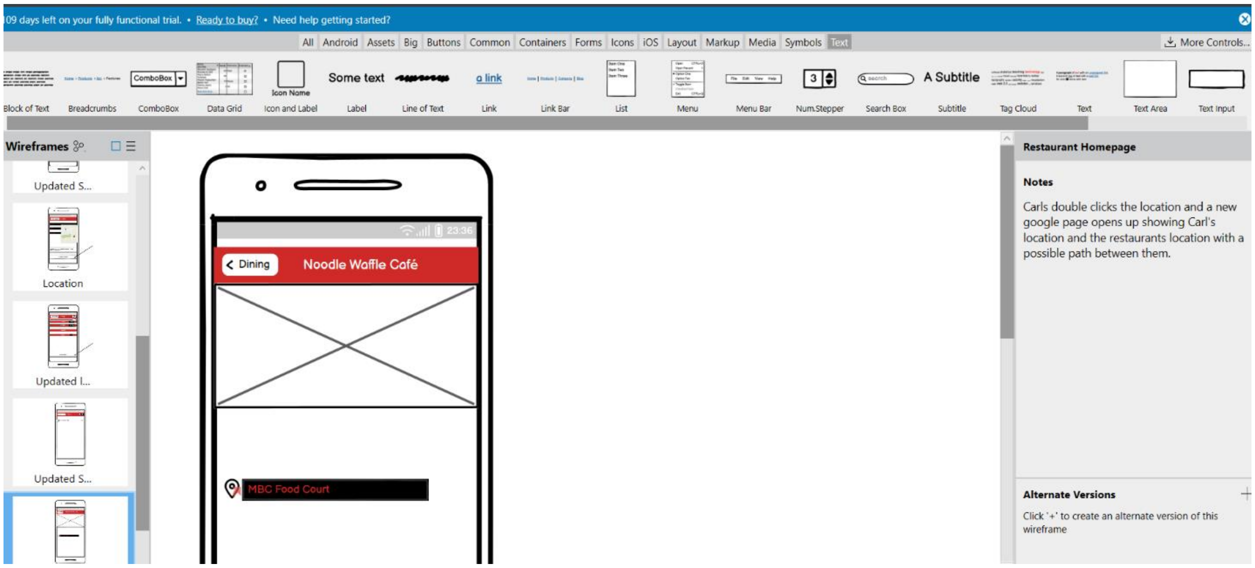
Notes

The updated filter option page show Ir cuisine and now the user can go throu each preference and after that click up search. Now try selecting the Location in the filters.

Alternate Versions

Click '+' to create an alternate version of this wireframe

LFP2



Restaurant Homepage

Notes

Carls double clicks the location and a new google page opens up showing Carl's location and the restaurants location with a possible path between them.

Alternate Versions

Click '+' to create an alternate version of this wireframe