

# **Project**

## **Part 2**

### **CMPT 363**

**Team 29**

**Chitransh Motwani**

**301435651**

**cma115@sfu.ca**

**Manpreet Singh Sarna**

**301416796**

**msarna@sfu.ca**

## **Part 2a- Context of Use & Requirements Establishment**

### **Context Identification:**

1. **When:** The Dining feature of the SFU Snap app will be used when SFU students need to plan their meals or dining activity while attending classes or are studying at any of the campuses.
2. **Where:** Since the dining feature is designed specifically for on campus dining, it will be used by students and staff when they are on campus or when they are at home planning their day at the campus.
3. **Who:** The dining feature of the app has specifically been designed for students, staff, and faculty members at Simon Fraser University.
4. **What:** This feature has been developed to help students navigate through various dining choices on the different campuses to help them decide regarding their eating plan.
5. **How:** Users can open and use this feature via the SFU Snap app; hence the user interface of the app has been designed based on mobile usage.

### **User Identification:**

The dining feature on the SFU Snap app is designed for students that are taking courses at any of the campuses of the Simon Fraser University, it provides them choices in accordance with their campus and helps them choose their dining option.

**Persona 1:** Anasthasia is a 3rd year SFU SIAT student who has a four-hour long break between her morning lecture and evening lab. She often gets hungry during this break period and needs to be looking for food as she does not get time to prepare lunch in the morning. She tries to find an efficient app that would help her find restaurants within and around the campus.

**Persona 2:** Gabriella is a Computer Science professor and teaches at SFU Surrey Campus. She teaches a class every morning from 8:30 to 11:30. Before her class, she uses various apps on the phone to look for coffee and a quick snack in the morning.

**Persona 3:** Kaz is a Security officer at SFU Burnaby, and he usually starts working at 6AM in the morning and works until 3PM. He uses his phone to look for affordable options to snack during his long workday.

**Functional Requirements:**

1. Users can look at the rating for each of the listed restaurants from the list of the restaurants itself without clicking on the restaurant or going further into the user interface.
2. When the user clicks on a specific restaurant, it lists the number of reviews for different ratings and show the overall rating of the restaurant. They can scroll down and read reviews left by other users.
3. When Users click on a specific restaurant, they can drop a review for the restaurant and are able to rate the restaurant 1-5 stars in accordance with their experience.

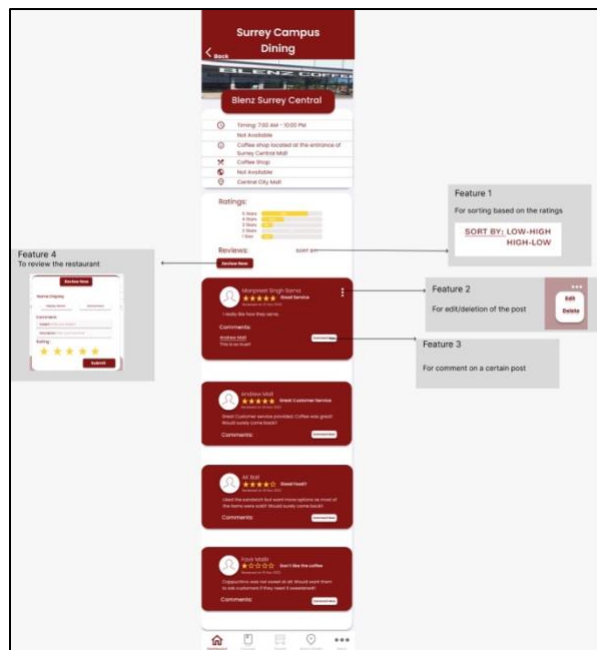
**Non-functional Requirements:**

1. After posting a review, the user can see their review on the top and is able to edit the review, change their rating according to their latest experience.
2. Users can sort restaurants using the ratings and the number of reviews for the restaurants which can account for their popularity.
3. Users can comment and react on other Users' reviews. This will show up as a thread below the original review.

## Part 2b- Medium Fidelity Prototype

For the generation of MFPs, we have combined multiple different LFPs made during the project cycle. We took the layout for the restaurant listings from one of the LFPs as the listings needed to be clear and interactive to promote user interaction. For the drop a review section, we developed various LFPs after learning from other apps on the internet and combining our requirements to the prototypes, we were able to generate an MFP which had our ideal design. We also designed a prototype for the list of reviews where the user was able to look at all the reviews, and comment on them. All our functional requirements were kept in mind when combining the various prototypes to generate the final MFPs, which resulted in our MFPs being highly user-friendly.

### H-MFP:



For our first functional requirement, both our prototypes have a dining screen where after selecting the campus, the user can look at the rating of any restaurant in the list along with its name and a photograph. This gives our design a good match between the system and the real world, as the ratings update and users can see the photograph of the restaurant without going further into the design, making user interaction very easy in less steps.

For our second functional requirement, when a restaurant is selected from the list of restaurants, the user can look at a list which indicates the number of user ratings for each number of stars just after the restaurant information. Following it, the user can scroll down to the reviews and read the reviews and look at the ratings given by each user. The highly informative nature of the restaurant page results in the user being able to make informed decision regarding their dining option.

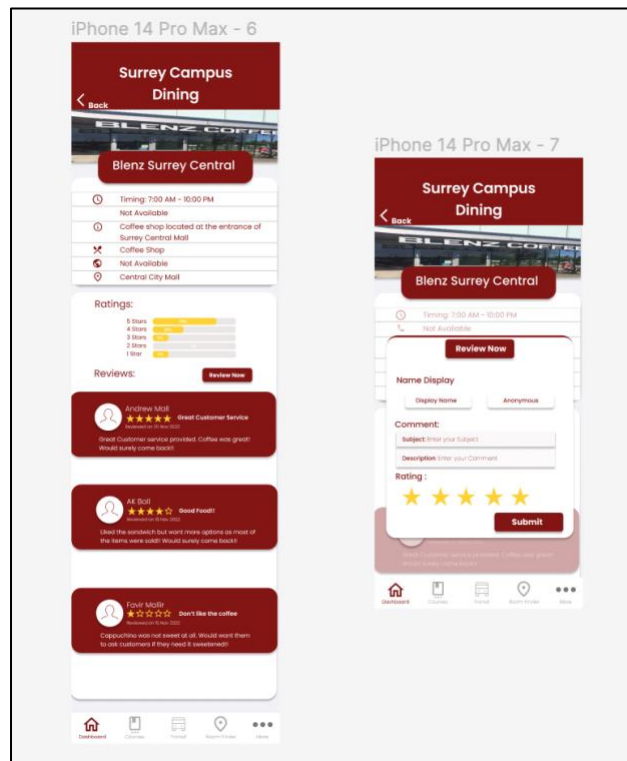
For the third functional requirement, the user can drop a review along with a rating for any selected restaurant, they can do this from the restaurant page or even the list of restaurants. This makes it very easy for users to drop a review as most users would not drop a review if it were a complicated process.

For the non-functional requirements, we have listed the user's review at the top whenever the user opens the restaurant page, there is an option to edit or delete the review for the user in case they want to update it or remove it.

For our second non-functional requirement, we have the sorting function under the reviews section of each of the restaurants, this will give user the choice to look for the reviews that they want to read and make a better decision about their food choice as well.

For the third non-functional requirement, all reviews have an option for the user to comment, which can result in a more informed choice for users as it will begin a thread regarding that review and give users more information.

## V-MFP:



### For the meaningful features:

The MFP focuses on the review feature in the dining option of the SFU Snap App. It allows the user to rate and review restaurants. The user can delete or edit their reviews at any point in time giving them full control over their posted review.

The users can look at the restaurants rating from the list of restaurants itself making it very easy to decide the restaurant based on the rating rather than opening each of the restaurant separately.

The review feature will also maintain validity of data and error handling, as it won't let user post multiple reviews for the same restaurant, and they need to fill the review with the title and at least 50 words to successfully post the review along with giving the restaurant a rating.

### **For the interaction steps:**

When the review now button is clicked the user can select whether they want to display their name or be anonymous and the selected option will change colour indication successful selection.

Clicking the submit button will show the review being posted to the user hence providing successful user feedback regarding their actions.

User can select the sort by options to sort the reviews immediately and the selection will be highlighted.

### **Learnings from designing MFPs:**

#### **Considerations:**

These MFPs were designed keeping in mind the importance and the high functionality of the review feature in in the dining section of the Snap App.

We would've added more features such as narrowing down the reviews to selected for a better user experience unless the user wants to go through all the reviews but since the vertical MFP was focusing on the review feature for each restaurant, our focus was to give the user a whole experience where they are able to make an informative decision.

#### **Couldn't Do:**

We could not provide further functionality to the sort and filter function as we focused on the user experience for dropping a review.

The reactions feature could not be added to the comments as that would further deviate our main feature of reviewing and confuse the user, also it is difficult to be able to accommodate different reactions in a prototype.

**Strengths/ Weaknesses:**

**Strength:** We were able to successfully implement the review feature in the dining function with multiple abilities for the user to edit and delete their reviews.

**Strength:** We were able to successfully provide the user looking for dining choices an option to sort their reviews, read reviews and the comments on the reviews and look at various ratings.

**Weakness:** The design is largely dependent on the user flow, and various steps are connected to each other making it very complicated to make sure the user is going the right way.



## **Part 2c- Analytical Evaluation and Reflection**

### **Cognitive Walkthrough:**

#### **Context and Scenario:**

During their education period at Simon Fraser University, university students tend to be looking for various food options around the different campuses. They use the SFU Snap app's dining feature to look for food options and decide regarding their dining choices.

In a regular academic semester, Jane, a 22-year-old, and year-four computing science major, tends to spend most of his weekdays on campus, going from one class to another. On most days she gets on campus at 8AM and leaves late in the evening. Jane wants to use the dining feature of the SFU Snap app to navigate through various dining choices when she is on campus and make a choice regarding her meals.

#### **Potential/ Target Users:**

Students, staff, and faculty of the Simon Fraser University who tend to spend extensive hours on the various campuses while working or studying and frequently look for dining choices while being on campus.

#### **Representative tasks for V-MFP & Cognitive Walkthrough**

The team invited **Sahil** to evaluate the vertical MFP and complete the cognitive walkthrough.

**Representative Task 1:** In this representative task, Jane wants to rate and drop a review for his favourite cafe at the SFU Surrey Campus, Blenz Coffee. He wants other users to see his name with his review.

Action Sequence	Does the user know what to do given the action?	Can the user find the right interface component to perform this action?	Can the user associate the feedback from the interface to the correct action they perform?	Does the user understand the feedback so that they know where they are in the task after performing the correct action?
<b>Action 1:</b> Login using your details	Yes, the user knows the action as it is common.	Yes, the user finds the login button to be in-line with other applications.	Yes, the user can move forward in the task.	Yes, the user finds themselves accustomed with the application interface.
<b>Action 2:</b> Go to the dining option of the app	Yes, the user knows the action as it is common.	Yes, the dining icon is easy for the user to distinguish.	Yes, clicking the icon takes them to the dining platform.	No, the user was a bit confused about the next step.
<b>Action 3:</b> Select the campus you are looking for	Yes, the user knows the action as it is common.	Yes, the arrow makes it easy to establish the selection required.	Yes, the user finds the selection feedback easy.	Yes, the user can move forward with the task after the feedback.
<b>Action 4:</b> Select the restaurant you are reviewing	Yes, the user knows the action as it is common.	Yes, the user can easily make a selection.	Yes, the user is satisfied with the feedback.	Yes, the user finds themselves moving forward in the task.
<b>Action 5:</b> Click on the review now button for the restaurant	Yes, the user knows the action as it is common.	Yes, the button makes it obvious for the user.	Yes, the user is satisfied with the feedback from the dialog.	Yes, the dialog leads them further in the task.
<b>Action 6:</b> Select the display name option	Yes, the user knows the action as it is common.	Yes, the two options are identical for the user.	Yes, the selection feedback is satisfying to the user.	Yes, the user finds themselves accustomed to the flow of the application.
<b>Action 7:</b> Type your comment and submit	Yes, the user knows the action as it is common.	Yes, the user can find the input boxes and submit easily.	Yes, the user is satisfied with the feedback.	No, the user is not notified about the success of the review.

**Representative Task 2:** In this representative task, Jane is looking to get coffee on the Surrey campus and wants to read reviews before making a choice.

Action Sequence	Does the user know what to do given the action?	Can the user find the right interface component to perform this action?	Can the user associate the feedback from the interface to the correct action they perform?	Does the user understand the feedback so that they know where they are in the task after performing the correct action?
<b>Action 1:</b> Login using your details	Yes, the user knows the action as it is common.	Yes, the user finds the login button to be in-line with other applications.	Yes, the user can move forward in the task.	Yes, the user finds themselves accustomed with the application interface.
<b>Action 2:</b> Go to the dining option of the app	Yes, the user knows the action as it is common.	Yes, the dining icon is easy for the user to distinguish.	Yes, clicking the icon takes them to the dining platform.	No, the user was a bit confused about the next step.
<b>Action 3:</b> Select the campus you are looking for	Yes, the user knows the action as it is common.	Yes, the arrow makes it easy to establish the selection required.	Yes, the user finds the selection feedback easy.	Yes, the user can move forward with the task after the feedback.
<b>Action 4:</b> Select the restaurant you want to read the reviews for	No, the user was a bit confused about selecting the restaurant.	No, the user is confused where to click.	Yes, the user finds the interface responsive and is satisfied.	Yes, the user understands the feedback as it is in line with other applications.
<b>Action 5:</b> Scroll down to read various reviews and look at different ratings.	Yes, the user knows the action as it is common.	Yes, the user finds it is easy to look at various reviews.	Yes, the user finds the feedback easy and is satisfied.	Yes, the user is accustomed to the list and finds it familiar.

**Summarising results:**

This MFP has several strengths, the MFP is able to direct the user towards the right direction in the flow of the tasks, most of the time. The icons and the buttons are obvious to the user and there is a match between system and the real world. The prototype is also able to support various actions required for the app and the overall structure and layout along with the colour scheme is highly user friendly.

But there is also a scope for improvement in various sections of the MFP, the app's feedback is a little less obvious for the user at various points and the user is unable to deduce the course of action at some points in a task. Some buttons and input boxes are not clear enough to the user. A better visibility of system status can be introduced.

Overall, the MFP ticks a lot of boxes for a prototype which is highly indicative of the final product and provides the user with a great functionality for a range of tasks. But the prototype can be further improved by making more obvious interactions which would align the prototype more tightly with recognition rather than recall. The design is although very minimal and aesthetic, we can further work on the review feature to make it easier for user to navigate. Visibility of the system status can be further worked upon by introducing more dialogs and confirming user actions.

## Reflection

### **Design for the Application Interface:**

To be able to design the dining feature of the SFU Snap app which can facilitate rating and reviews parameter from the user, we took the following steps:

1. The first step is to identify and understand the context and user for the application being designed. This could be done by performing surveys and interviews with randomly selected SFU Students, so the designers can learn about the users' thoughts about the app, their expectations from the app and their usage requirements and conditions.

At the same, the designers should look for similar products, like Yelp, and conduct heuristic evaluation to identify the products' usability problems and good usability features. This would make the developers better familiar with the context of the user.

2. The second step is to gather requirements from SFU students. Features of software and products need to be designed based on the target users' demands and requirements. Designers should find the important aspects of the dining feature in the SFU Snap app, as it will only be able to attract more users, if the users' demands are met through these aspects. Also, the requirement gathering will improve the product's popularity and attract more users.

After the previous steps, the designers will now have raw data of the students' views on the app's dining feature. The designers will now need to analyse the requirements put forward by the students and develop on the functional and non-functional requirements for the feature.

Furthermore, the designers should be able to end up with a clear and concise specifications document listing the context and user identification of the product and the functional and non-functional requirements.

3. The third step is to design the prototype adhering to the listed requirements. We do not need to design an MFP or HFP at this point, as it will consume a lot of time. The designers can begin by designing various LFPs with different UI frameworks. It is significant in the product's development for the designers to put forward the requirements in various LFPs, comparing them and selecting their respective advantages, integrating them into new LFPs and repeating this process until a satisfactory LFP is designed which can be developed into an

MFP. Designers can also involve some students in this design process as our goal is for the design to be user-friendly.

4. After developing interactive MFPs, the designers can make a short presentation introducing the prototypes to collect suggestions.

Furthermore, the designers can perform product evaluation for the MFP. With the help of an expert, they can predict user behaviour. The expert shall be informed about the context and scenario of the product and can conduct a heuristic evaluation to evaluate the prototype for general and usability issues. The designers can also complete one or more cognitive walkthroughs. This will result in detailed problems which can be further worked upon.

5. The final step will be to keep UCD in mind, at all steps of the design process. After each evaluation, the designers shall return to the first step of the process until the final design has been achieved.

## **Methods:**

1. Heuristic Evaluation:

Heuristic evaluation can help designers check the usability of the design during the process itself, as it is the standard evaluation method used by designers all around the world, it makes it easier to find improvements to the design. When we conducted heuristic evaluation of other online review and rating apps, we were able to find various good usability examples and usability problems. Hence, during the design process of our project, we were able to avoid the problems and adapt the good usability ideas in our design to improve the quality of our design.

2. Requirements Gathering & Specification:

Gathering requirements and identifying specifications are an important step when creating and designing usability. Only when the designers have accurate context and user identification along with functional and non-functional requirements, they can continue designing products which are user-friendly. The collection of the information began with finding the when/where/who/what/how for the review feature of the app and then figuring out some personas of the user group for the product. We then found the functional and non-functional requirements for the app's feature, through which we were able to outline the requirements of the users.

3. Prototyping:

Creating prototypes is an important and highly essential part of design as they can represent the core functional requirements of the product. Prototypes also help designers to check and further evaluate the practicality of the design and look for the details of the design and propose solutions to potential issues, as we move forward in the design cycle by making modifications to the prototypes.

4. Cognitive Walkthrough:

Analysing cognition will help the designers improve the design and further develop features for the final product. But first, the designers need to specify the context, scenario, target users and then design representative tasks with several actions. The designers can then invite someone with knowledge in User Interaction to perform the cognitive walkthrough where the designers follow each step and take note of the feedback and comments and it will give them a user perspective. This step helped us find drawbacks of the current design in detail and make further improvements like rearrangement of events and interface components. This pushed the design further close to the final product.

5. As the design needs to be user friendly, familiarity with the UCD process is vital. Identification of target users, clarification of the users' requirements, solving the users' needs, inviting users to evaluate, and repeating this process while improving until the final product is developed. This makes the designers focus on the user requirements and experience, which lets the product be user-driven instead of making users adapt to the design.

## **Tools:**

1. Balsamiq:

Balsamiq can be majorly used to design the low fidelity prototypes. It provides us with a toolbar which has different design components making the design easier and faster. Especially for learners, it is convenient to draw, add marks and comments for the design process. Even though team projects are not supported in Balsamiq, it simplifies the design process for LFPs

2. Figma:

Figma is majorly used for designing medium and high-fidelity prototypes. It has various functions and is easy to use. The component features are easy to use and provide various interactive methods for design development. Team projects are supported in Figma, and design teams can communicate and invite users for evaluation.