



# THE AMERICAN UNIVERSITY IN CAIRO الجامعة الأمريكية بالقاهرة

## Software Engineering

Second Sprint

### Group 3

Submitted To: Dr. Sherif Aly

#### Submitted By:

Abdalla Elragal - [abdalla.elragal@aucegypt.edu](mailto:abdalla.elragal@aucegypt.edu) - 900171502

Ali Salem - [alisalem@aucegypt.edu](mailto:alisalem@aucegypt.edu) - 900183226

Ramy Badras - [ramybadras@aucegypt.edu](mailto:ramybadras@aucegypt.edu) - 900194248

Salma Laban - [salmalaban@aucegypt.edu](mailto:salmalaban@aucegypt.edu) - 900191369

Sherif Wessa - [sherifwessa@aucegypt.edu](mailto:sherifwessa@aucegypt.edu) - 900203171

Tayam badawy - [tayambadawy@aucegypt.edu](mailto:tayambadawy@aucegypt.edu) - 900183911

## **Table of Content**

<b>Planning of our Second Sprint</b>	<b>2</b>
<b>Sprint Backlog</b>	<b>3</b>
<b>User Stories</b>	<b>6</b>
<b>Development of Sprint</b>	<b>9</b>
Prototype	9
Demo	9
<b>Testing of Sprint</b>	<b>10</b>
Test cases	10
<b>Review of the status of the sprint</b>	<b>17</b>
<b>A Sprint Retrospective</b>	<b>18</b>
What went well	18
What could be improved	18
<b>A refined object oriented design for the features we developed</b>	<b>19</b>
Class Diagram	19
Sequence Diagram	20

## **Planning of our Second Sprint**

- Duration of the second sprint: two weeks.
- We started off by conducting a group meeting to discuss what we did in the first sprint, what went well, and what needs improvement.
- In this sprint, we decided to work on the remaining functionalities that were not included in the previous sprint.
- First of all, we decided to work collectively on the prototype. We started from scratch because we thought that the prototype of the first sprint was not efficient enough and needed a lot of changes, and also we chose a different prototyping software that allows us to share and work on it simultaneously.
- We divided the set of features on us and set a deadline for their delivery.
- Second, we conducted a meeting with the TA to get his feedback on our previous deliverables to make sure that we are on the right track, and to know what we did wrong and work on it.
- We agreed that we needed to work on the following:
  - Review the backlog for the next sprint.
  - Add test cases for the new functionalities and improve the old ones.
  - Add user stories for the new functionalities.
  - Modify the UML diagrams (class diagram and sequence diagram).
- Finally, we concluded our planning stage by defining our goal for this sprint and for the final one, dividing the tasks on the team members, setting deadlines for each task, and adding them to Trello so that we can have a clear vision and work towards achieving the sprint goals on time.

## **Sprint Backlog**

### **1. Customize item**

After selecting an item the user has the ability to customize what they ordered ( if applicable) for example a sandwich without olives.

### **2. Selecting location**

After the user has confirmed his order, they should choose whether to pick up the order from the food vendor or the designated pick-up stations at each building.

### **3. Payment Confirmation**

After the amount has been paid a message appears on the screen: payment of: X amount has been successfully made.

### **4. Order confirmation**

An email will be sent to the user's AUC email containing the invoice of the order made which includes the order number, the amount paid and the items that have been ordered.

### **5. Cancel order**

The user can cancel an order if the order has not yet been made by the food vendor.

### **6. Track the order**

The user has the option to track the order and know whether the order is still in the food vendor or is on its way.

### **7. Track the order**

The user has the option to track the order and know whether the order is still in the food vendor or is on its way.

### **8. Review Order**

The user has the option to review the order by giving it a rating on a scale from 1 to 5.

## **9. Support.**

If the user faces any problems the user can chat with an employee to help him through the application.

## **10. Orders history**

The user can view the orders history up to 10 orders. The order history contains which food vendor the user chose, the items, and the amount paid.

## **11. Edit menu**

The vendors would be able to add new items or remove old ones as well as update the items according to inventory (stock).

## User Stories

As a user of the app	I want to be able to select the location	So that I can choose whether to pick up the order from the food vendor or the designated pick up locations	<ul style="list-style-type: none"> <li>- Ensure that the user can choose his preferred pickup method</li> <li>- Given that the user chose to pick up the order from the designated pick up locations, ensure that the user can select the location (building) that he wants from the university map</li> </ul>	<b>5 points</b>
As a user of the app	I want to be able to customize my selected items	So that it matches my food preferences	<ul style="list-style-type: none"> <li>- Given that the customer has a customizable item selected, the option to customize that item should be available</li> <li>- Ensure that the price of the ingredients that can be added is available for the user</li> <li>- Ensure that the price of any added item is reflected in the total price</li> </ul>	<b>5 points</b>
As a user of the app	I want to be able to cancel an ongoing order	I no longer need my order to complete	<ul style="list-style-type: none"> <li>- Given that the order has been placed and the order is still incomplete from the vendor's end, the user should be able to successfully cancel their order.</li> <li>- Given that the order is canceled successfully, a refund will be issued to the respective user.</li> <li>- Given that the order is canceled successfully, the user should receive a confirmation message.</li> </ul>	<b>3 points</b>

As a user of the app	I want to track my order	So that I can know if the order is prepared by the vendor or if it's on its way.	<ul style="list-style-type: none"> <li>- Given that the user has an ongoing order, the option to track the status of the order should be available.</li> <li>- Given that the user has an ongoing order, the user can view the status of the order (accepted, preparing, prepared, ready to pickup)</li> <li>- Given that the user has an ongoing order, and the order is on its way, the user can view and contact the courier</li> </ul>	<b>3 points</b>
As a user of the app	I want to be able to review a completed order	So that i can give it a rating depending on how well the service was	<ul style="list-style-type: none"> <li>- Given that the user has successfully picked up their order, the user can add a star and textual rating</li> </ul>	<b>2 points</b>
As a user of the app	I want a type of order confirmation after placing an order.	To have an invoice of how much i paid and to have a receipt in case there's a problem with the order	<ul style="list-style-type: none"> <li>- Given that the user successfully placed an order, they should receive an email for their AUC account. The email contains the order ID, amount paid and the details of the order.</li> <li>- Given that the user successfully placed an order, a confirmation message should appear in the application</li> </ul>	<b>3 points</b>
As a vendor registered on the app	I want to receive pickup confirmation after a customer has picked up their order	To ensure that the customer successfully picked up their order	<ul style="list-style-type: none"> <li>- Given that the customer successfully picked up their order from the designated location, an email will be sent to both the vendor and the customer, confirming the success of the process.</li> </ul>	<b>3 points</b>
As a user of the app	I want to be able to contact either	In case there is a bug in the app or	<ul style="list-style-type: none"> <li>- An option to contact an employee or food vendor is</li> </ul>	<b>3 points</b>

	the admin of the app or the food vendor	an error in my order, I want to be able to receive support in this situation.	<p>available for the user at all times.</p> <ul style="list-style-type: none"> <li>- Ensure that the user is automatically redirected to his phone application if he selects the “Support” feature in the application</li> </ul>	
As a vendor registered on the app	I want to be able to edit my menu	To keep the menu up to date in terms of prices, offers and items	<ul style="list-style-type: none"> <li>- The ability for food vendors to edit specific parts of their menu should be available at all times.</li> <li>- Given that the vendor modified (added, updated, deleted) an item in the menu, it should be reflected immediately in the application</li> </ul>	<b>5 points</b>
<b>Total Points for this Sprint</b>				<b>32 Points</b>



## Development of Sprint

### Prototype:

<https://www.figma.com/file/5DPIZTZWzw04qiHXLhUtqR/SWE---Project?node-id=12%3A425&t=gJ0fFH4imYwzKmod-1>

### Demo:

<https://drive.google.com/drive/folders/1y9q4rYvla49ErlAFSkrwhfOt7vI-8F1O?usp=sharing>

## Testing of Sprint

### Test cases

1	<p><b>If the user chose to sign up, and entered the correct AUC email, the program sends a unique confirmation code of six numbers to their AUC email. Then, the user will be asked to check their email and enter the code they received to ensure a valid and protected registration process.</b></p> <ul style="list-style-type: none"><li>● Make sure the AUC email lies within AUC's domain.</li><li>● Check validity of username after entering correct credentials at sign up page.</li><li>● Send a unique confirmation code to the user (6 digits).</li><li>● The user should enter the confirmation code and complete signing up.</li></ul>
2	<p><b>If the user did not receive a confirmation code within 30 seconds, he may select the “resend code” option so that the system resends the code and sends it.</b></p> <ul style="list-style-type: none"><li>● Press the “resend code” button and await response.</li><li>● Resend code if the user clicked the “Resend Code” button after 30 seconds.</li></ul>

3	<p><b>Given that the user confirmed their AUC email, they should be asked to enter a password consisting of 8 characters with 1 upper case character and 1 special character. Then, the program creates a new account with the user's entered data.</b></p> <ul style="list-style-type: none"> <li>• Enter a password less than 8 characters.</li> <li>• Enter an 8 character password with no upper case letters.</li> <li>• Enter an 8 character password with 1 upper case and no special characters.</li> <li>• Enter a password with all specifications above.</li> <li>• Given a password that satisfies all the requirements the password creation process should be complete.</li> </ul>
4	<p><b>Given that the user wants to sign up, if the user did enter a valid AUC email and a password containing 8 characters with 1 upper case character and 1 special character, display an error message "Please enter a password that contains 8 characters of which 1 is an uppercase and 1 special character"</b></p> <ul style="list-style-type: none"> <li>• Display an error message "Invalid Password"</li> <li>• Ask the user to re-enter a new password that contains 8 characters of which 1 is uppercase and 1 is a special character by displaying this message "Please enter a password that contains 8 characters of which 1 is an uppercase and 1 special character"</li> <li>• Validate new password</li> </ul>
5	<p><b>If the user chose to log in, and given that the user entered the correct AUC email and password, display the list of food vendors (restaurants).</b></p>

	<ul style="list-style-type: none"> <li>• Enter invalid email at the login page.</li> <li>• Enter valid email but an invalid password at the login page.</li> <li>• Enter valid credentials at the login page.</li> </ul>
6	<p><b>Given that the user entered the wrong AUC email and/ or password, display an error message and ask the user to re-enter the email and password</b></p> <ul style="list-style-type: none"> <li>• In case of invalid login credentials, ask the user to re-enter the credentials and redirect them back to the login screen.</li> </ul>
7	<p><b>Given that the user is logged in, the list of restaurants should be scrollable</b></p> <ul style="list-style-type: none"> <li>• If not working, the app should close and reload again</li> </ul>
8	<p><b>Given that the user selected a food vendor (restaurant) to order from, a menu should be displayed.</b></p> <ul style="list-style-type: none"> <li>• Redirect the user to the chosen vendor's menu.</li> </ul>
9	<p><b>If the user selects an item from the menu, the item and the quantity should be reflected in the cart.</b></p> <ul style="list-style-type: none"> <li>• Add item to cart and check results.</li> <li>• Remove item and add another item.</li> <li>• Edit an item in the cart.</li> <li>• Change the number of an item in cart to an invalid amount e.g. negative or a number too large.</li> <li>• Given an invalid item amount the app automatically sets it to the previous value.</li> </ul>

10	<p><b>Given that the user selected an item, a list of the ingredients that can be added or removed from the item is displayed and the user can select any ingredient.</b></p> <p><b>The added or removed ingredients should also be displayed in the cart</b></p> <ul style="list-style-type: none"> <li>• Edit an item in the cart and check the final price.</li> <li>• Undo edits and check the final price.</li> </ul>
11	<p><b>Given that the cart has only one item, if the user chooses to remove this item, the cart should be empty</b></p> <ul style="list-style-type: none"> <li>• After removing the last item from the cart, a different screen is displayed where it shows the cart is empty.</li> </ul>
12	<p><b>Given that the cart has multiple items, if the user chooses to remove an item, the item should be removed from the cart, and it should show the rest of the items</b></p> <ul style="list-style-type: none"> <li>• After an item is removed from the cart, the other items are moved upwards to be displayed one after the other (given there were items below the removed one).</li> </ul>
13	<p><b>Given that the user chose the “select pickup location” option, AUC map with different pickup locations should be displayed and the user is asked to select a certain location.</b></p> <ul style="list-style-type: none"> <li>• Choose a location for pickup.</li> <li>• Choose a different location and check results.</li> </ul>

13	<p><b>Given that the user chose a specific location, if they chose the “proceed to payment” option, the user should be able to choose their preferable payment method, either through credit/debit card, or through AUC pay.</b></p> <ul style="list-style-type: none"> <li>• Display payment methods after the user finalized the choosing pickup locations process.</li> </ul>
14	<p><b>If the user chose to pay with credit/ debit card. If the user entered the correct information, send a confirmation email and display a payment confirmation message.</b></p> <ul style="list-style-type: none"> <li>• Enter invalid card credentials.</li> <li>• Enter valid card credentials and await response.</li> </ul>
15	<p><b>Given that the user chose to pay with credit/ debit card. If the user entered the wrong payment information (name on card, card number, expiry date, security code), the app will display an error message and ask the user to re-enter the information.</b></p> <ul style="list-style-type: none"> <li>• The user is asked to re-enter the payment information.</li> <li>• Redirecting them back to the payment screen.</li> </ul>
16	<p><b>Given that the user chose to pay with AUC pay, redirect the user to the AUC SSO to enter their credentials and validate it.</b></p> <ul style="list-style-type: none"> <li>• Enter invalid credentials.</li> <li>• Enter valid credentials and check the result.</li> </ul>

17	<p><b>Given that the user chose the “Cancel order” option, they should be redirected to the home page, and the cart should be emptied.</b></p> <ul style="list-style-type: none"> <li>• After confirming an order, press the “cancel order” button and check results.</li> <li>• Press the “cancel order” button after an order is ready for pickup and check response.</li> </ul>
18	<p><b>Once payment is confirmed, a message appears to the user with the confirmed order and payment details along with the estimated time of arrival.</b></p>
19	<p><b>Given that the user chose the “Track order” option, the program displays a window with four states: Accepted, Preparing, Prepared, Ready to pickup.</b></p> <ul style="list-style-type: none"> <li>• Press the “track order” button after confirming an order.</li> </ul>
20	<p><b>Given that the user selects the “View order history” option, the user should be able to view the details of their order history.</b></p> <ul style="list-style-type: none"> <li>• Confirm an order and check order history while it’s ongoing.</li> <li>• Check history after canceling an order.</li> <li>• Check history after completing an order.</li> <li>• Display completed orders with the status “Completed”</li> <li>• Display ongoing orders with their current status.</li> </ul>
21	<p><b>Once the user confirms receiving their order, a window pops up asking the user to give a rating and review the order.</b></p> <ul style="list-style-type: none"> <li>• Display the option to rate the order after it has been completed</li> </ul>





## **Review of the status of the sprint**

For the second sprint, we were able to achieve 100% of our sprint goals. We worked closely together on the list of tasks and functionalities that we decided on in the beginning of this sprint.

First off, we developed a set of user stories and test cases for the newly added functionalities and modified the old ones.

Then, we focused on building and improving our prototype, and adding new features to it while ensuring a user-friendly interface that is easy to use and understand.

Lastly, we were able to modify the class diagram by adding new methods and functions to better accommodate the new features, and edited the class relationships. Also, we improved the sequence diagrams to better illustrate the interactions between objects as well as the overall flow of the program.

- **Team velocity:**

**Salma Laban:** 24 points per sprint

**Sherif Wessa:** 22 points per sprint

**Abdalla Elragal:** 22 points per sprint

**Ramy Badras:** 21 points per sprint

**Tayam Badawy:** 5 points per sprint

**Ali Salem:** 5 points per sprint

# **A Sprint Retrospective**

## **What went well**

- We were able to develop a functional (UI) prototype that showcases almost all the features of the app.
- A different prototyping software was used, which later proved to be more suitable for our requirements.
- We were able to include features that we were not able to include in the initial sprint due to time constraints.
- Diagrams from the previous sprint facilitated the development of the diagrams in the current sprint.

## **What could be improved**

- Consistency with one prototyping software can improve our efficiency, instead of having to switch to a different software after gaining some proficiency in the former.
- Dividing the workload of the sprint more equally amongst the team members.
- Improve and polish any issues with the prototype or the current features of the app.

# A refined object oriented design for the features we developed

## Class Diagram



## Sequence Diagram



