Class Project Phase IV

CSE 360 - Fall 2022

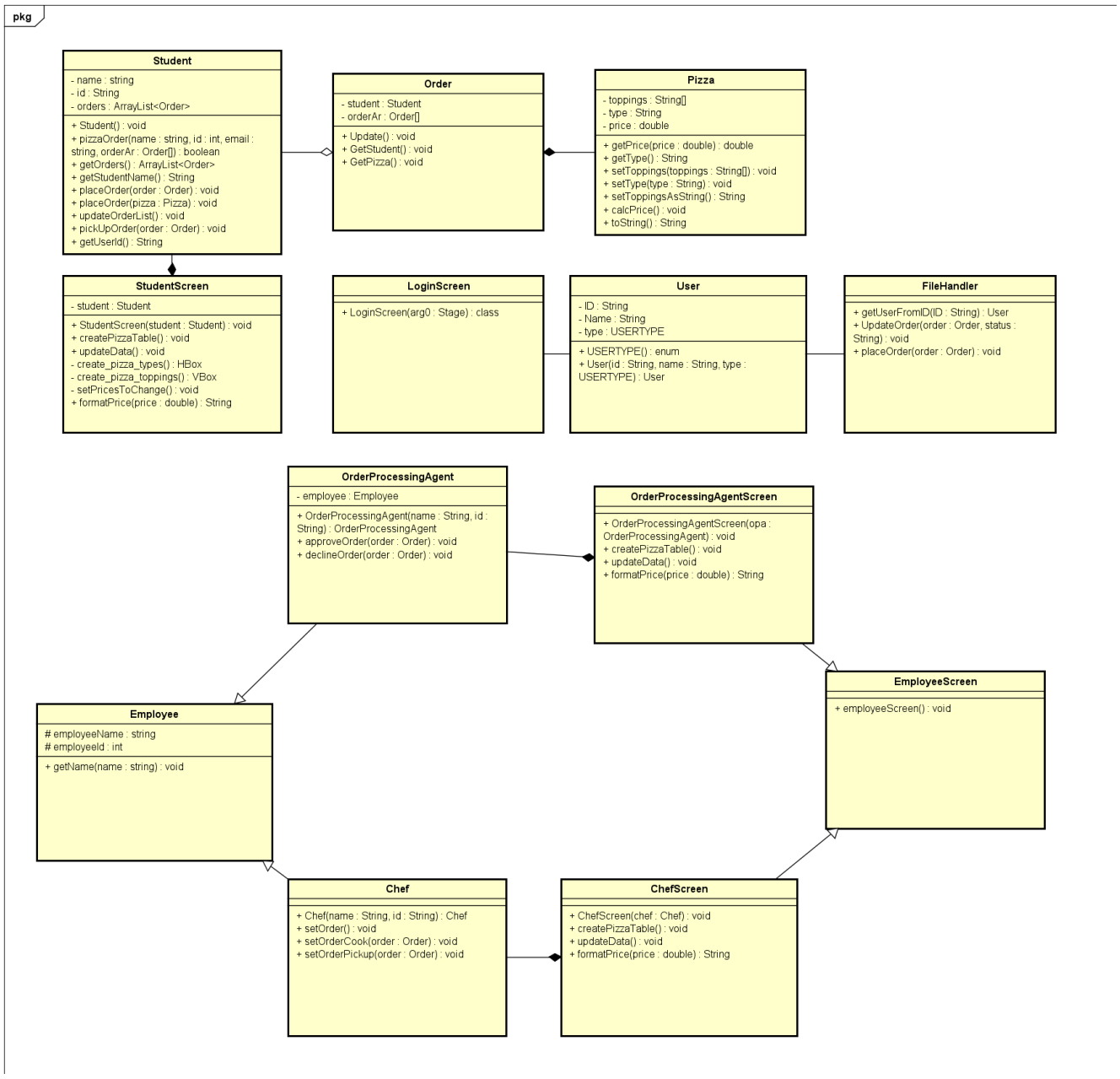
Team Number: 36

Team Members: Cameron Sharp, Adem Cufurovic, Salvador Gomez, Joshua Inman, Atharva Dhoke

Technical Presentation:

https://drive.google.com/file/d/1c_ai4ImNJIbxKkT_GV6Z0utlp2wpZdJ1/view?usp=share_link

Final Class Diagram:



Functional Testing:

The following contains a copy of the functional test plan from Phase II with an additional column of what the actual output was for each outcome. It also contains a statement on whether the test was successful and any changes that need to be made. For tests that contain redundant features with a previous test, only the new features are evaluated.

Use Case: Select Pizza Type. Test Result: SUCCESS

Use Case Number	Scenarios/Steps	Expected Output	Actual Output
1	The student selects pizza type “cheese” and no additional toppings.	The price displayed will be $10.00.	The price displayed is $10.00.
2	The student selects pizza type “pepperoni” with no additional toppings.	The price displayed will be $12.00.	The price displayed is $12.00.
3	The student selects pizza type “veggie” with no additional toppings.	The price displayed will be $15.00.	The price displayed is $15.00.

Use Case: Select Additional Toppings. Test Result: SUCCESS

Use Case Number	Scenarios/Steps	Expected Output	Actual Output
1	The student does not select any additional toppings.	The price displayed will not change from the base output (see Select Pizza Type Use Case).	The price does not change from the base output.
2	The student selects any one additional topping.	The price displayed will be $1.50 above the base output.	The price increases from the base output by $1.50.
3	The student selects any two additional toppings.	The price displayed will be $3.00 above the base output.	The price increases from the base output by $3.00.
4	The student selects any three additional toppings.	The price displayed will be $4.50 above the base output.	The price increases from the base output by $4.50.
5	The student selects all four additional toppings.	The price displayed will be $6.00 above the base output.	The price increases from the base output by $6.00.

Use Case: Order Pizza. Test Result: SUCCESS.

Use Case Number	Scenarios/Steps	Expected Output	Actual Output
1	The student, after selecting a pizza type and additional toppings, presses the “Place Order” button. (Since the user has already logged in with a valid ASU ID, and the app does not allow one to order an invalid pizza, there is only one test case scenario.)	The pizza will be accepted. The pizza description, price and status will be shown in the “My Orders” pane on the user’s screen. The pizza description should match the pizza type and toppings that were ordered. The pizza price should match what was displayed prior to ordering. The pizza status should read “Accepted.” The ordering pane should still contain the information for that pizza so that the user could quickly order an identical pizza if desired.	The pizza is accepted, and the correct description, price, and status are shown in the “Ordered Pizza” table on the student screen. The pizza status reads “Approved”, a small difference from “Accepted”. The ordering pane remains unchanged.

Use Case: Accept Order. Test Result: Minor Changes Needed

Use Case Number	Scenarios/Steps	Expected Output	Actual Results
1	The order processing agent presses the “Accept” button on a pizza in the queue.	The pizza should be removed from the order processing agent’s queue and should be added to the chef’s queue. The pizza order status, in both the view of the chef and the student, should then read “Ready to Cook”.	The pizza’s status is changed to “Approved”, rather than “Ready to Cook”, and is not removed from the Pizza Ordering Agent’s screen. However, it is sent to the chef’s queue.
2	The order processing agent presses the “Decline” button of a pizza in the queue.	The pizza should be removed from the order processing agent’s queue, and the status of the order on the student’s view should read “Declined”	The pizza’s status is changed to “Declined”. A message for the student (simulated email) is sent to the

		until the end of the day. An email should be sent to the student notifying the student that the order was declined and providing a justification.	student. However, the pizza is not removed from the pizza ordering agent's screen.

Changes to make: We will fix the statuses of the pizzas for improved clarity and will remove pizza from the pizza ordering agent's screen once they are no longer needed.

Use Case: Change Order Status: SUCCESS

Use Case Number	Scenarios/Steps	Expected Output	Actual Output
1	The pizza status is currently in the "Accepted" state, and the order processing agent pressed the "Accept" button.	The new pizza status should be "Ready to Cook".	The new pizza status is "Approved". See notes on the previous test case.
2	The pizza status is currently in the "Accepted" state, and the order processing agent pressed the "Decline" button.	The new pizza status should be "Declined."	The new pizza status is "Declined".
3	The pizza status is currently in the "Ready to Cook" state, and the chef pressed the "Cooking" button.	The new pizza status should be "Cooking".	The new pizza status is "Cooking"
4	The pizza status is currently in the "Cooking" state, and the chef pressed the "Completed" button.	The new pizza status should be "Ready".	The new pizza status is "Completed", matching the buttons in the system.
5	The pizza status is currently "Ready to Cook" and the chef pressed the "Completed" button.	The pizza status should not change. An error message should display that informs the chef that he must first cook the pizza.	The pizza status changes to "Completed", giving the chef flexibility if he forgot to press the "Ready to Cook"

			button.
6	The pizza status is currently “Cooking” and the chef pressed the “Cooking” button.	The pizza status should not change.	The pizza status does not change.

Use Case: Prepare Pizza. Test Result: SUCCESS

Use Case Number	Scenarios/Steps	Expected Output	Actual Output
1	The chef presses the “Cooking” button on a pizza in his queue.	The pizza status displayed in the chef’s queue and the student’s ordered pizzas list should read “Cooking”.	The pizza status changes to cooking for both the student and the chef.

Use Case: Set Pickup for Students. Test Result: SUCCESS

Use Case Number	Scenarios/Steps	Expected Output	Actual Output
1	The chef presses the “Ready to Pick Up” button on a pizza in his queue with status “Cooking.”	The pizza should be removed from the chef’s queue. An email should be sent to the student informing the student that the pizza is available for pick up. Further, the pizza’s status should display “Ready” in the student’s list of ordered pizzas.	The pizza status is set to “completed” and a statement is printed for the student to mock an email. The pizza is not removed from the chef’s queue (see comments on “Accept Order”).

Use Case: Pickup Pizza. Test Result: Not Implemented

Use Case Number	Scenarios/Steps	Expected Output	Actual Output
1	The student picks up the pizza.	The pizza should be removed from the student’s list of ordered pizzas.	No pickup feature was implemented. It will be released in a future version.

Future Changes: This use case, which went beyond requirements, will be implemented in a future version release.

Use Case: Login. Test Result: SUCCESS

Use Case Number	Scenarios/Steps	Expected Output	Actual Output
1	The student enters a valid ASU ID.	The student should be taken to the screen displaying an ordering tool and all active orders belonging to that student.	The student is taken to their personal page.
2	The order processing agent enters a valid work ID.	The order processing agent should be taken to the screen displaying the list of “Approved” but not yet “Ready to Cook” orders.	The order processing agent is taken to their personal page.
3	The chef enters a valid work ID.	The chef should be taken to the screen displaying the list of “Ready to Cook” and “Cooking” orders.	The cook is taken to their personal page.
4	An actor enters an invalid ID.	An error message should display, and the screen should remain on the logout screen.	An error message is displayed, and the user remains at the login screen.

Use Case: Logout. Test Result: SUCCESS

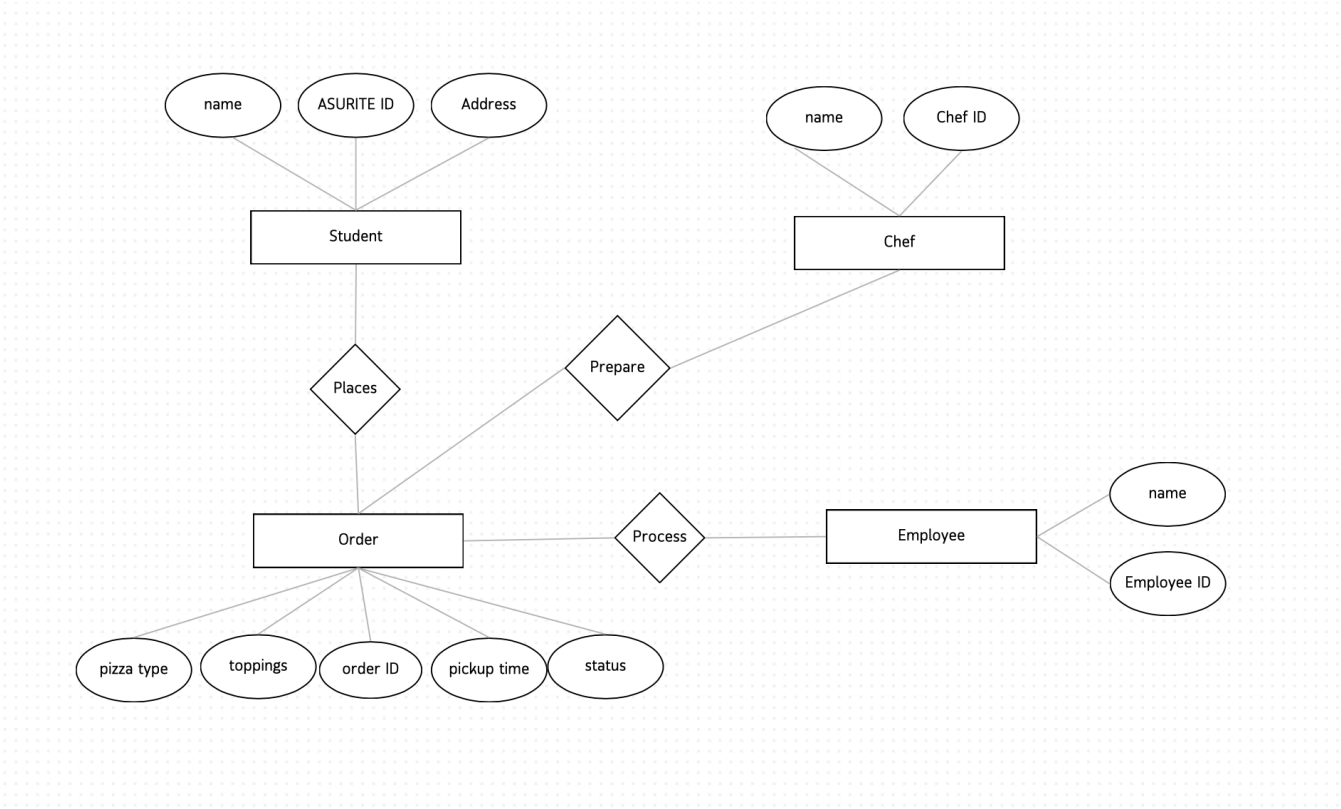
Use Case Number	Scenarios/Steps	Expected Output	Actual Output
1	An actor presses the “logout” button.	The user should be logged out and taken back to the login screen.	The user is logged out and taken to the login screen.

Data Design

Identified Data Entities:

- Student
- Order
- Employee
- Chef

ER Diagram:



Conclusions

During the course of our project, we learned a lot about project management and software engineering. For starters, understanding project deadlines and building a successful project schedule is vital for ensuring that the workload is adequately spread out across the duration of the project. This was a lesson that we learned throughout the project and improved on as the semester progressed.

We also learned the importance of documenting your code thoroughly so that the team members can add functionality to any portion of the code and can help debug code. This helps streamline the project by reducing the amount of time spent trying to understand one another's code.

In the project, we wrote over 1,200 lines of code and spent around 20 hours working on the code, including debugging, initial testing, and GitHub commits. This reduces to around 60 lines of code per minute. This rate was not particularly impressive, which was due to unfamiliarity with JavaFx Tables and editing files, as well as some of the project management issues highlighted above. In general, we found and corrected seven defects per phase.

We tried multiple strategies to work together as a team and complete the project requirements as quickly and efficiently as possible. We worked well in helping eachother out to understand the concepts of the project and how the code would work together. It worked well dividing tasks up between individuals to have completed. It was difficult merging everyone's code together and communicating ideas over text. It was very important to set up meetings in order to get the most efficient work done. We would encourage future groups to have a regular time to meet and consider meeting in-person on occasion so that all of the ideas are effectively communicated.

We would recommend that future CSE 360 students begin writing the code as early as they can in the project in order to ensure that it is completed early and give them additional time to make improvements to the design to go above and beyond the requirements. Such a project would certainly be quite rewarding to create and would be excellent to show to future employers!