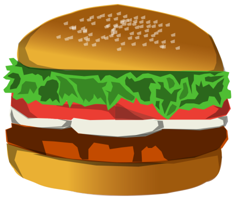


U Eat:

A Dining Concierge App

-Requirements

Group 28
Luke Carter
Ibrahim Mahmoud
Miles McCoy
Mathew McDade
Timothy Tseng

Requirements Definition:

❖Functional Requirements Definitions:

- The customer can create an account that includes a preferred payment method.
- The system can display a list of participating restaurants within a particular area.
- The system can display a list of participating restaurants from a name search.
- The system can display a selected restaurant’s menu.
- The customer can select the number of guests who will be dining.
- The customer can add items from the menu to an order and assign that item to an individual guest or for the entire table.
- The customer can request a table for a selected number of guests at a specific time.
- The system can display the soonest available table for a requested time and number of guests.
- The customer can accept or reject an available table offer.
- The system can process payment for the meal using the customers preferred payment method.
- The system can display a confirmation that an order has been completed.
- The system can display information about a completed order, including table reservation time and number of guests.
- The customer can open an existing order while dining and add additional items from the menu.
- The restaurant can add and remove items from the menu.
- The restaurant can change menu item prices.
- The system can notify the restaurant of a new table request.
- The restaurant can send the user available table times for the requested number of guests.
- The system can display an order that has been placed to the restaurant, including the table reservation information and the food order.

- The system must display confirmation to the restaurant that payment for an order has been accepted.
- The restaurant can begin a customer visit when the customer is seated.
- The restaurant can end a visit when the customer leaves.

❖Non-functional Requirements Definitions:

- Users will be able to log in within 5 seconds.
- The system will display a map of nearby restaurants within 10 seconds.
- The system will display restaurant search results within 5 seconds.
- The system will display the selected restaurant’s menu within 5 seconds.
- The system will allow the customer to select a number of diners between 1 and an upper limit set by the restaurant.
- The system will display a menu item added to an order within 1 second.
- The system will allow the user to select a desired dining time, from the current time to an upper limit set by the selected restaurant on the current day.
- The system will process and confirm payment within 15 seconds.
- The system will confirm customer order and dining time within 15 seconds.
- The system will display an order to the restaurant within 15 seconds of order confirmation.

Use Cases:

❖ **Case One** - Customer places an order and pays.

➢ **Actors**
- Customer
- Restaurant Staff

➢ **Preconditions**
- User downloaded app and created user account
- Restaurant is provider member of UEats
- Restaurant provider uploaded menu and prices correctly
- User has access to network
- Restaurant near user has a table available within a reasonable time period

➢ **Postconditions**
- Customer received food, as ordered
- Customer paid restaurant
- Restaurant received payment

➢ **Flow of Events**
- Customer opens UEats App
- Customer navigates to restaurant, either through search or map view
- Customer reviews wait time to determine if the time is acceptable
- Customer reviews menu items
- Customer selects menu items and place them in their “table”
- Customer reviews their table items
- Customer finalizes order
- Customer pays for order using Apple Pay API
- Restaurant is notified of new UEats order and required time
- Restaurant staff prepare meal and place items on table within required time
- Customer arrives at the time indicated by the app
- Restaurant staff seats customer at prepared table
- Customer eats meal
- Customer confirms receipt of meal
- Customer allocates tip for service
- Customer approves final payment for meal
- Customer leaves restaurant
- Restaurant notified of final payment

❖ **Case Two** - Account Creation

➢ **Actors**

- Customer

➢ **Preconditions**

- Customer has downloaded the UEat app on a smartphone
- Customer has a preexisting email account
- Customer has a preexisting phone number
- Customer has a preexisting online payment method
- Customer has access to network

➢ **Postconditions**

- Customer created a UEat account they have access to
- Customer's email contains a confirmation email from UEat about a new account being created
- UEat created a new entry in its database of accounts

➢ **Flow of Events**

- Customer opens UEats App
- Customer navigates to "Log In"
- Customer navigates to "Create an Account"
- Customer inputs their first name in the "First Name" textbox
- Customer inputs their first name in the "Last Name" textbox
- Customer inputs their email address in the "Email Address" textbox
- Customer inputs their phone number in the "Phone Number" numeric textbox
- Customer inputs a password of their choice in the "Password" textbox, with the minimum requirements of at least one lowercase letter, uppercase letter, number, and special character
- Custom inputs the same password in the "Confirm Password" textbox, which bars the customer from continuing until the passwords are the same
- Customer reads and checks that they have read the UEat's User Agreement and Privacy Policy in the checkbox next to "I have read and agree to UEat's User Agreement and Privacy Policy.", which bars the customer from continuing until the checkbox is ticked
- Customer clicks "Sign Up"

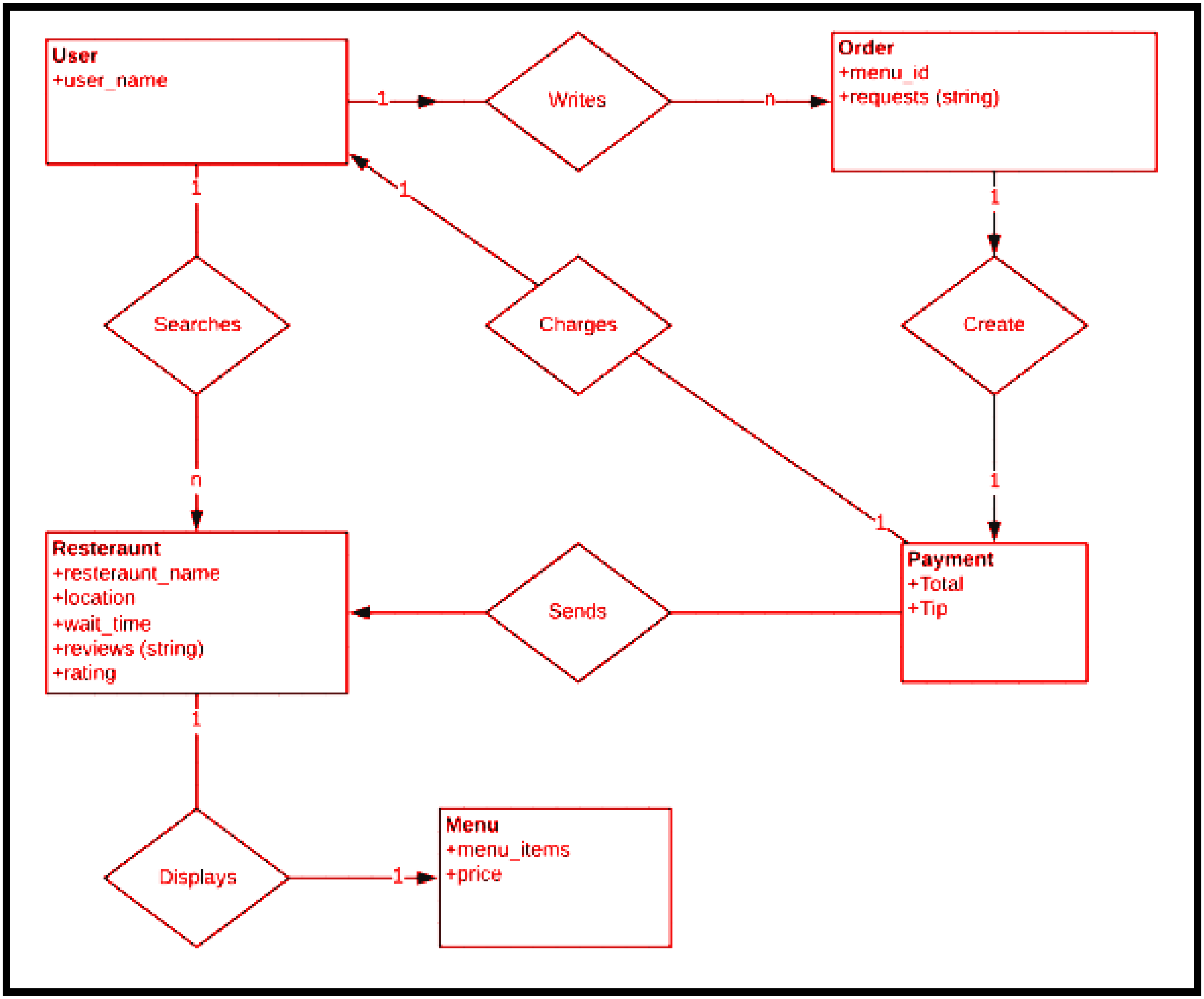
- UEat’s database of accounts gets updated with the new customer’s information
- Upon completion, the customer will receive a confirmation email with the option to undo their action of creating their account

---

❖ **Case Three** – Restaurant adds seasonal items to menu

- **Actors**
  - Restaurant staff.
- **Preconditions**
  - Restaurant staff have UEat app installed
  - Restaurant staff have access to change menu and look of menu.
  - Restaurant have pictures, prices, and description of seasonal items.
- **Post-Conditions**
  - Restaurant staff member removes seasonal items after season.
- **Flow of Events**
  - Restaurant staff member opens UEats app.
  - Restaurant staff member navigates to change menu.
  - Restaurant staff member Uploads new item’s names, descriptions, pictures, seasonal tag, and prices to collection of items for sale
  - Restaurant staff Member Saves new seasonal items.
  - Restaurant staff member adds new items to active menu.
  - Restaurant staff member logins into customer side.
  - Restaurant staff member checks the active menu and sees new items with correct pictures, names, descriptions and prices, and the seasonal tag

UML Diagram:


User
+user_name
Writes
Order
+menu_id
+requests (string)
1
n
1
1
Searches
Charges
Create
n
1
1
Resteraunt
+resteraunt_name
+location
+wait_time
+reviews (string)
+rating
Sends
Payment
+Total
+Tip
1
Displays
1
Menu
+menu_items
+price

Requirements Specification:

❖Functional Requirements Specifications:

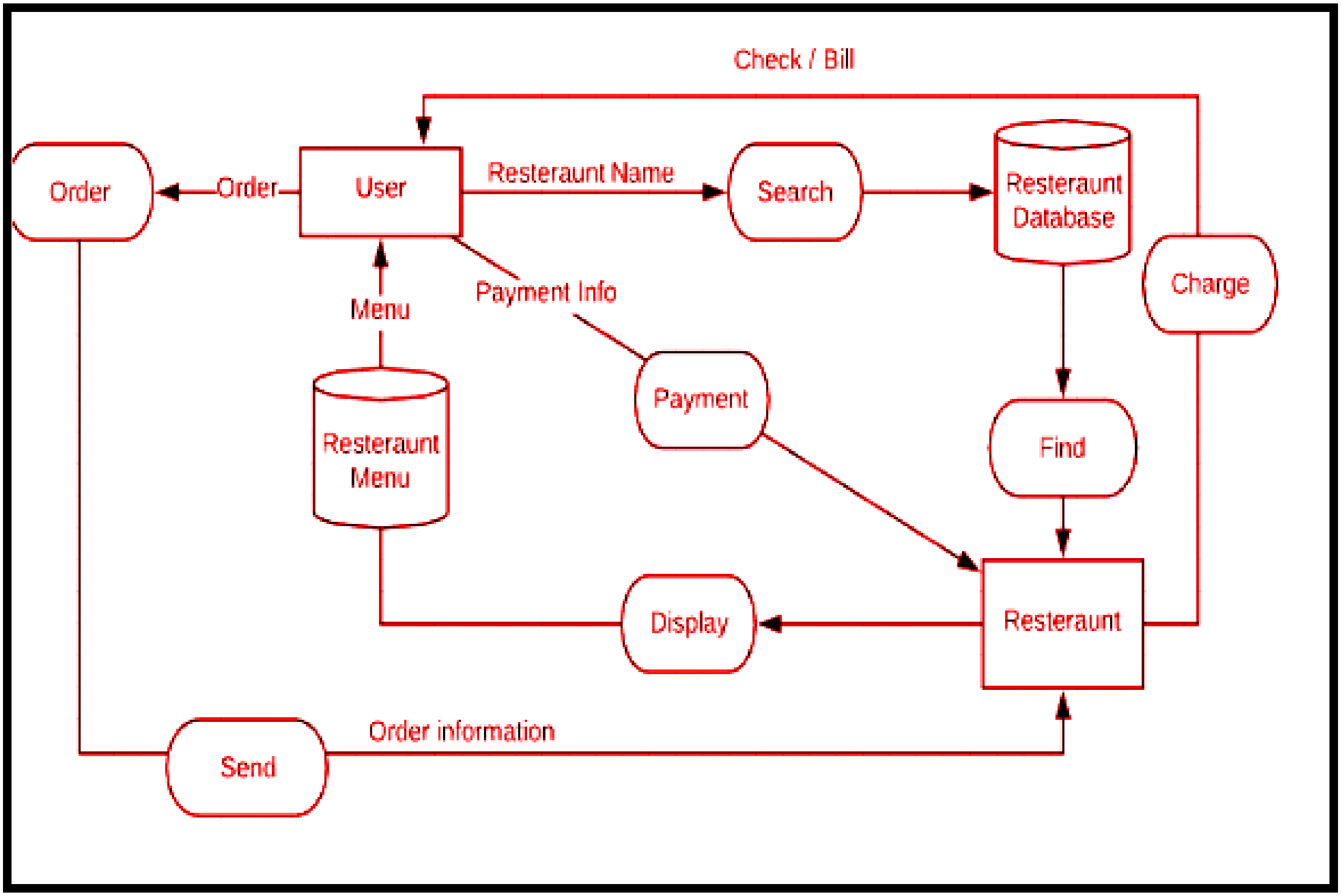
- ➢ Customer account creation will add a new entry to the database user table.
- ➢ Restaurant account creation will add a new entry to the database restaurant table.
- ➢ User login credentials will be verified against the database user table.
- ➢ User location will be obtained via location services.
- ➢ Nearby participating restaurants will be loaded from the database restaurant table.
- ➢ Nearby restaurants will be inserted into a map view.
- ➢ A restaurant search will query the database for matches or near matches and return the result.
- ➢ Customer selection of a restaurant will retrieve the restaurant's menu, max table size, and seating hours from the database.
- ➢ The system can interact with third party payment systems to complete order payment.
- ➢ After payment transaction is complete, an order is added to the database order table.
- ➢ When a customer adds items to a completed order, a new suborder is added to the database.
- ➢ When a restaurant updates the menu, seating options, or hours, the database entries are updated.
- ➢ The system will close any unclosed visits at the end of the day.

❖Non-functional Requirements Specifications:

- ➢ Customer creation will add an entry to the user table within 5 seconds.
- ➢ Restaurant creation will add an entry to the restaurant table within 15 seconds.

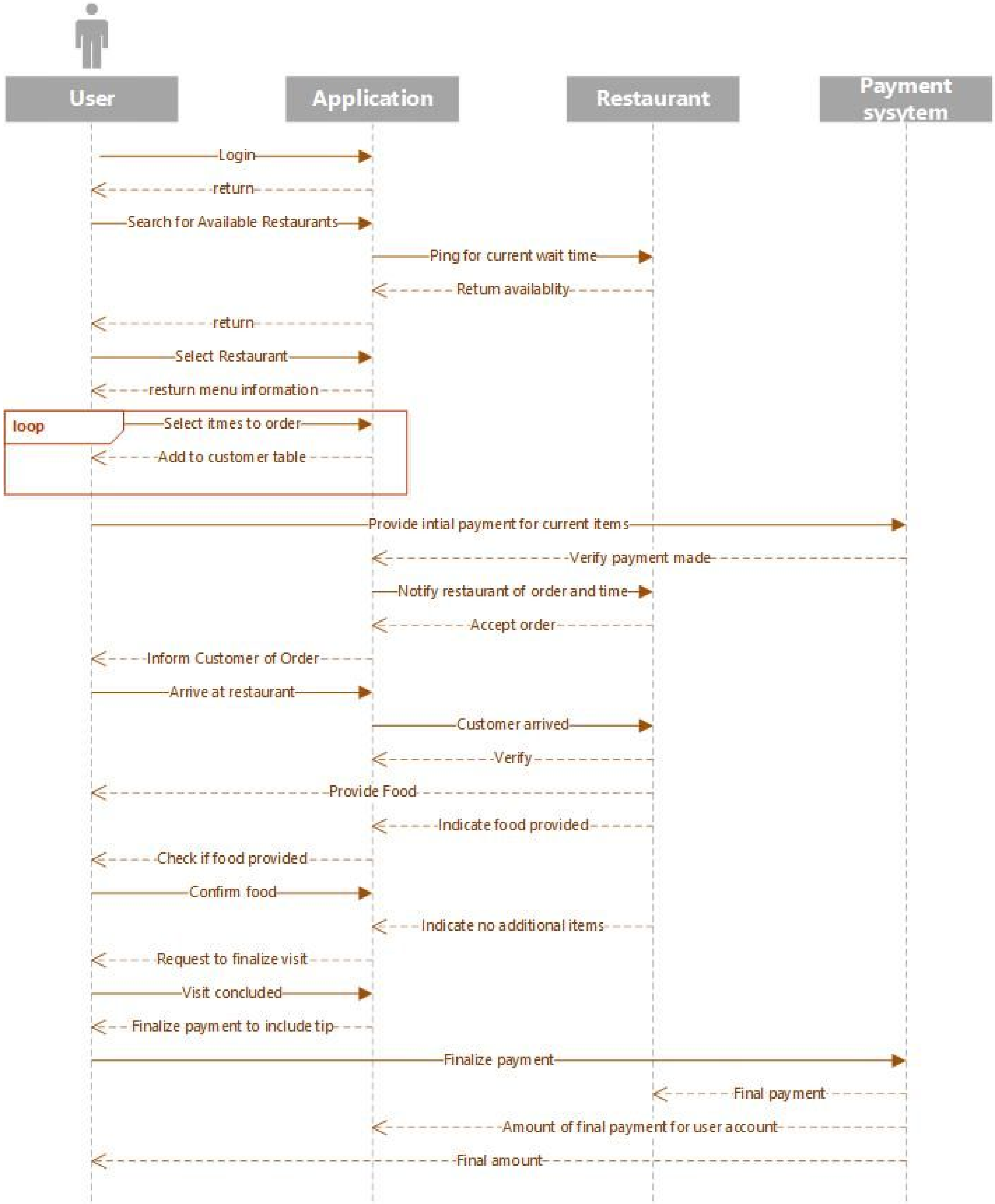
- Updating restaurant information in the database will update and return the updated restaurant information within 10 seconds.
- User login queries to the database will return within 5 seconds.
- Restaurant search queries will return within 5 seconds.
- The system will limit customer active visits to a maximum of 1.
- The system will end all of a restaurant’s active, not closed, visits 1 hour after the end of a day’s business hours.

Dataflow Diagram:

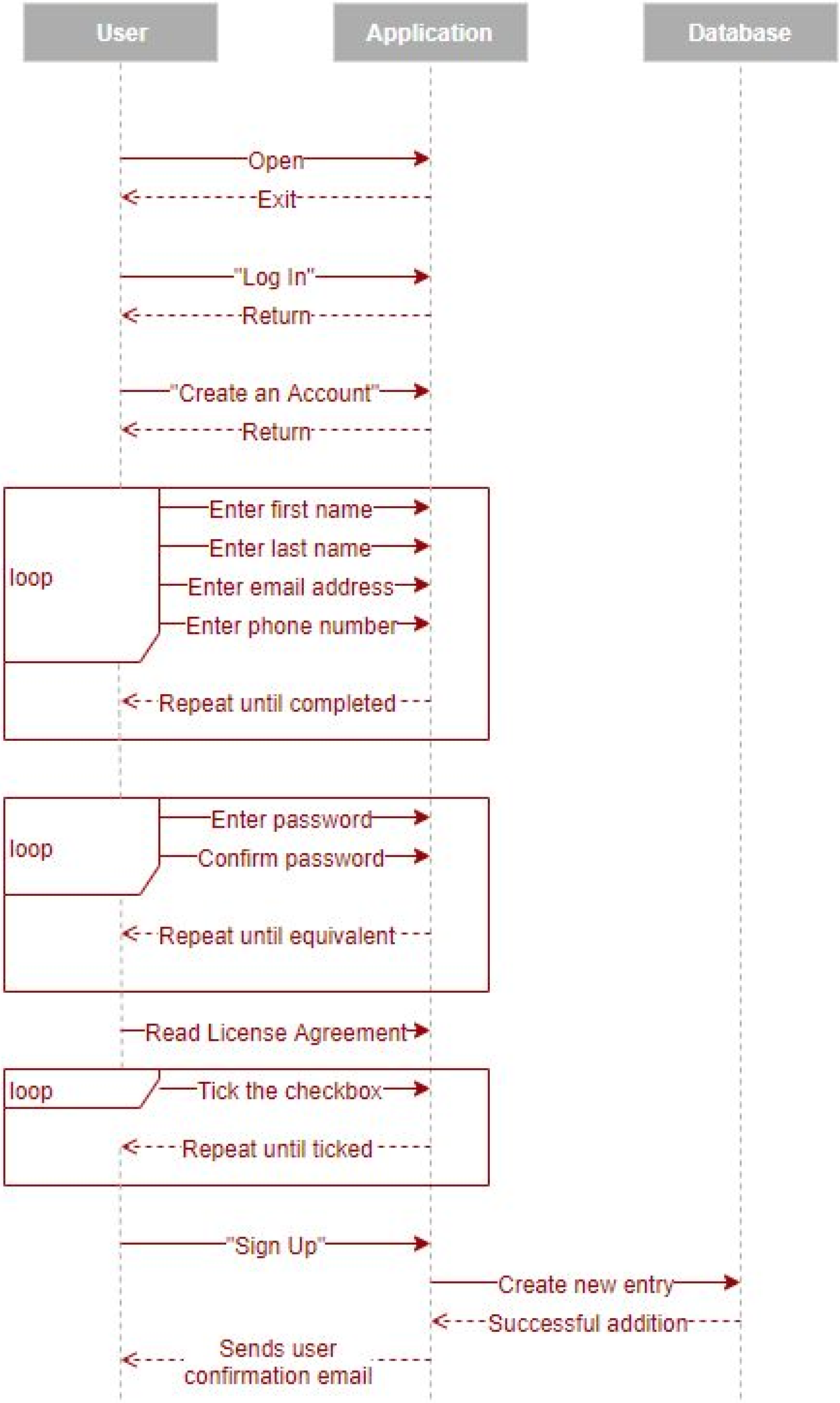


Use Case State Charts:

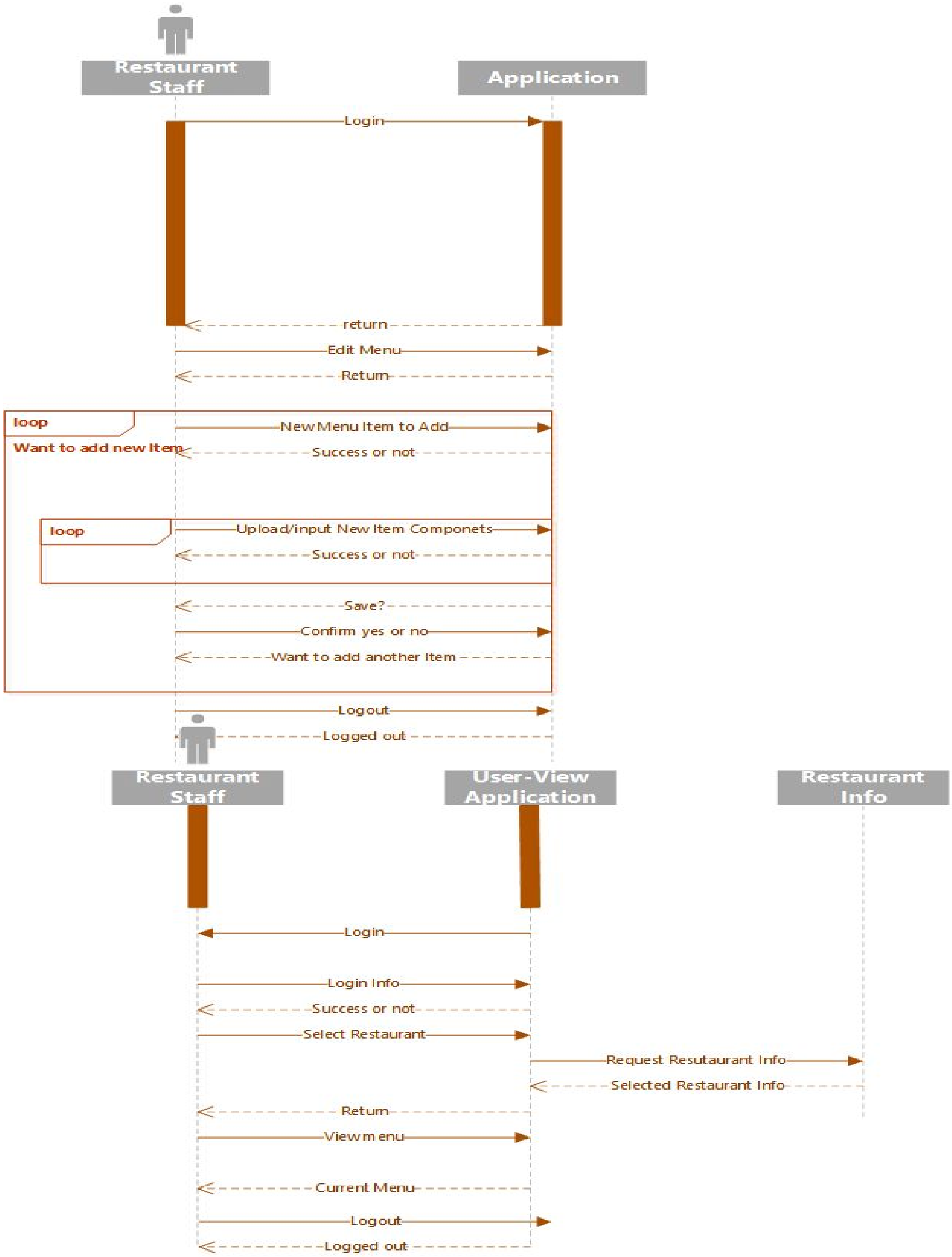
Use Case One: Standard Customer Order



Use Case Two: Account Creation



Case Three - Luke Carter: Adding New Menu Items



❖Customer Input:

- ➢We were able to have a productive, hour-long meeting with the project customer, Shannon Farazi, via Slack on 10/15/19.

❖Team Contributions:

- ➢ All: Communication via private Slack channel.
- ➢ Timothy Tseng - UML Diagram, Dataflow Diagram.
- ➢ Miles McCoy - Use Case 1, Use Case 1 State Chart, Customer Meeting planning.
- ➢ Ibrahim Mahmoud - Use Case 2, Use Case 2 State Chart.
- ➢ Luke Carter - Use Case 3, Use Case 3 State Chart.
- ➢ Mathew McDade - Shared documents setup, document formatting and submission, Customer Meeting, Functional and Non-Functional Requirements Definitions, Functional and Non-Functional Requirements Specifications.