Team: Landon Bedell
Deep Desai
Branden Romero
Justin Tang

Title: Massimo/MasterMouth

Project Summary: Master mouth/Massimo is an Android app the facilitates the interaction between customer and restaurant. It allows customers to find, order and pay for a meal in a seamless way. All while providing tools for the restaurant to provide a smooth and enjoyable experience.

Project Requirements:
There are no business requirements.

Priority Key: D- Desireable
M - Mandatory
O - Optional
E - Possible/Future Advancement

Complexity Key: Easy -Days
Medium - Weeks
Hard - Months
Very Hard - Non Possible

User Requirements				
ID	Requirement	Topic Area	User	Priority
UR-001	Sign in through Facebook	Authentication	Diner	D
UR-002	Sign in through Google+	Authentication	Diner	D
UR-003	Use prefered payment method	Payments	Diner	M
UR-004	Search through nearby entrees based on meal type	Search	Diner	M
UR-005	Rate specific meals	Rating	Diner	O
UR-006	Rate the restaurant	Rating	Diner	O
UR-007	Have app suggest meals for me based on what I have previously had	Curation	Diner	E
UR-008	Easily split the check between other users	Payments	Diner	D
UR-009	See what dishes are trending near me	Search	Diner	D
UR-010	Keep me logged in	Authentication	Diner	O

UR-011	Automatically charges users that forgot to pay	Payments	Restaurant	D
UR-012	Able to upload new entrees to app	Updates	Restaurant	M
UR-013	View statistical data about restaurant's entrees	Logistics	Restaurant	O
UR-014	Show total tips earned for the day so splitting between servers can be calculated	Logistics	Restaurant	D
UR-015	Know if an entree is available at restaurant	Logistics	User	D

Functional Requirements			
ID	Requirement	Complexity	Priority
FR-000	As restaurant places order, creates and updates table’s receipt	Medium	D
FR-001	Create a receipt of items a user ordered in a sitting and Display it in payment screen	Easy	D
FR-002	Auto detect accounts that user is already signed into	Easy	D
FR-003	Highlight items that the user clicks on and pop a pay for button	Easy	M
FR-004	Use GPS to determine if a user is at a restaurant	Medium	O
FR-005	Use QR Codes to determine which check is associated with the user	Medium	M
FR-006	Auto pay for forgotten tabs at midnight	Medium	E
FR-007	Store the frequency at which a specific entree is ordered	Easy	O
FR-008	Move user to rating screen after payment	Medium	O
FR-009	Display which entrees are trending based on frequency of it ordered	Medium	M
FR-010	Tag entree based on the type of food they have	Hard	O
FR-011	Store the average user rating for each entree	Medium	M
FR-012	Enable search that can be organized by distance name or rating	Medium	M

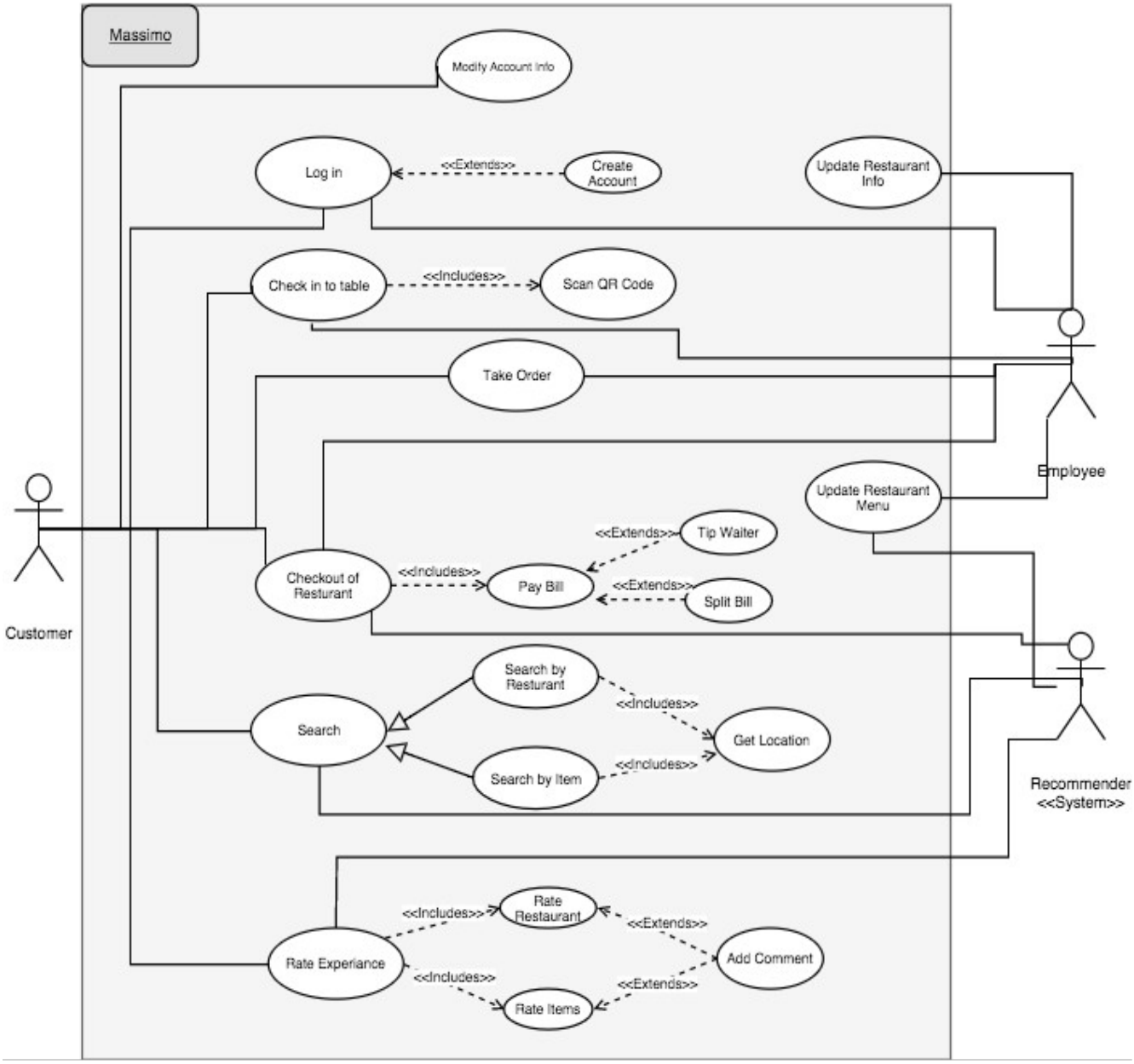
FR-013	Store pictures for entrees	Easy	D
FR-014	Display entrees' photos on search list	Easy	D
FR-015	Calculate tip and display 10% 15% and other then add selected to bill	Easy	D
FR-016	Create a homepage with a list of entrees based on location and rating	Medium	M
FR-017	Send notification to user when they enter a restaurant that support our app	Hard	O
FR-018	Have a button on home screen to bring up camera to scan QR code	Medium	M
FR-019	Once user scans QR code for table associate them with receipt	Easy	M
FR-020	If more two users scan the QR code for one sitting, enable them to select which items to pay for	Easy	M
FR-021	Update user that payment is complete after transaction goes through. Or tell them that it failed	Easy	D
FR-022	Restaurant can view a list of entries available at their restaurant and select and update which items are out of stock	Medium	O

Non-Functional Requirements				
ID	Requirement	Topic Area	Complexity	Priority
NFR-001	Signing in with other accounts takes no longer than 10 seconds	Response Time	Easy	D
NFR-002	Selecting specific items to pay for instantly updates on all users phones	Response TIme	Easy	D
NFR-003	Protect sensitive user data including payment information username password and location	Security	Medium	D
NFR-004	Search functionality should be quicker than	Response TIme	Medium	M

	30 seconds			

Use Cases:

- Use Case Overview:



- Use Case Documents:

Use Case Id:	UC-001
Use Case Name:	Log in to app (customer)
Description:	Customer can log into Massimo.

Actors:	Customers		
Pre-Conditions:	Customers must have signed up already and their unique id and password must be accessible by the system.		
Post-Conditions:	If successful the customer is directed to the search page where their personal recommendations are posted. If login fails the customer is asked to sign in again.		
Frequency of Use:	Every time the app is opened.		
Flow of Events:		Actor Action	System Response
	1	Type in username	App displays username as it is typed.
	2	Type in password	App displays stars(*) as password is typed.
	3	Click stay logged in button	Add/Remove check mark in stay logged in box.
	4	Click log-in button	Show loading icon Pass: Go to home/search screen Fail: Ask customer to log in again
Variations:	4. Click login through Facebook/Google option		

Notes and Issues:	Prioritize speedy log in because of the frequency of use.

Use Case Id:	UC-002
Use Case Name:	Log in to app (Employee)
Description:	Restaurant can log into Massimo.

Actors:	Restaurant		
Pre-Conditions:	Restaurant must have signed up already and their unique id and password must be accessible by the system.		
Post-Conditions:	If successful the restaurant is directed to the info page where menu and times open are editable . If login fails the restaurant is asked to sign in again.		
Frequency of Use:	Whenever menu updates or info updates are necessary.		
Flow of Events:		Actor Action	System Response
	1	Type in restaurant name	App displays username as it is typed.
	2	Type in password	App displays stars(*) as password is typed.
	3	Click log-in button	Show loading icon Pass: Go to home/info Fail: Ask them to log in again
Variations:	No variations.		
Notes and Issues:	The restaurant and customer sign in through the same screen but are directed to different places afterwards.		

Use Case Id:	UC-003
Use Case Name:	Rate Experience
Description:	Customer rates their most recent dining experience.

Actors:	Customers		
Pre-Conditions:	Customers have completed the transaction for a meal using the Massimo App.		
Post-Conditions:	The customer has rated their most recent Massimo meal, and future App recommendations reflect the customer's review based on how it aligns with the ratings of other Massimo users.		
Frequency of Use:	Hourly by User		
Flow of Events:		Actor Action	System Response
	1	Completes a transaction and choses to Rate Experience on Massimo App.	Displays rating screen for the preceding meal.
	2	Rates the restaurant and each item they purchased.	Fills in the selected stars with gold color.
	3	If the customer wishes to write a comment about a	Toggles textbox display when speech bubble icon is

		particular rating they have made, they can click the speech bubble icon to the right of the stars and type their comment into the text box that appears.	clicked.
	4	Clicks the “Submit” button, when finished rating each item.	If user has rated each item, screen displays a “Thanks for your Feedback!” screen for 1 second and then returns to the home screen. If user has not rated one or more items, App prompts customer to rate the remaining items before continuing.
Variations:			
Notes and Issues:	If user closes app before submitting, all ratings for the meal are discarded.		
Developer Notes:			

Use Case Id:	UC-004
Use Case Name:	Search by Restaurant
Description:	User looks for a restaurant where they would like to eat.

Actors:	Customers		
Pre-Conditions:	Customer has logged into Massimo App.		
Post-Conditions:	The customer has found a restaurant where they would like to eat.		
Frequency of Use:	Daily by Customers		
Flow of Events:		Actor Action	System Response
	1	From the Massimo home screen, the Customer selects the “Search By Restaurant” option.	Displays the Search by Restaurant screen, which features a search bar, a “By Type” button, and restaurant suggestions.
	2	Selects “By Type” and then chooses a category of food from the listed options.	Finds and then lists nearby Restaurants that fit the Customer’s selection, showing their overall rating, and their approximate distance from the user’s location.
	3	Customer selects a restaurant.	Displays the Restaurant information, including its location.
	4	Customer clicks on location	Opens map app with directions to restaurant.
Variations:	2. Searches using keyword/phrase in search bar. 2. Selects one of the recommended restaurants.		
Notes and Issues:			
Developer			

Notes:	

Use Case Id:	UC-005
Use Case Name:	Search by Item
Description:	Customer looks for a particular food item they would like to eat.

Actors:	Customers		
Pre-Conditions:	Customers has logged into Massimo App.		
Post-Conditions:	The customer has found a particular food item they would like to eat.		
Frequency of Use:	Daily by Customers		
Flow of Events:		Actor Action	System Response
	1	From the Massimo home screen, the Customer selects the “Search By Item” option.	Displays the Search by Item screen, which features a search bar, a “Master Menu” button, and Item suggestions based off of previous ratings.
	2	Selects “Master Menu”	Opens hierarchical menu containing all menu items from nearby restaurants.
	3	Customer selects an item.	Displays Item description, reviews, and a link to resaurant.
	4	Customer clicks on	Displays the Restaurant

		Restaurant link.	information, including its location.
	5	Customer clicks on location	Opens map app with directions to restaurant.
Variations:	2. Searches using keyword/phrase in search bar. 2. Selects one of the recommended items.		
Notes and Issues:			
Developer Notes:			

Use Case Id:	UC-006
Use Case Name:	Check in to Table
Description:	Customer checks into table via QR code so that the customer can see order as it is updated by employee

Actors:	Customer		
Pre-Conditions:	Customer must be logged into their account and must have their payment system linked to said account.		
Post-Conditions:	If successful the user’s app will be able to display the user’s current orders, and the user will now be able to check out.		
Frequency of Use:	Whenever user wants to pay restaurant bill using Massimo		
Flow of Events:		Actor Action	System Response
	1	Enter Check-in System	Opens phone's camera and ask user to scan QR code at table.

	2	User Scans QR Code	The app will notify employee that customer is using Massimo, and app will display user's order so far.
Variations:	No variations.		
Notes and Issues:	Customers may block the permission to use phone's camera, and we have no alternative to the scan QR code check in.		

Use Case Id:	UC-007
Use Case Name:	Update Order
Description:	Employees keep track of order in case customer signs into Massimo or has already signed in.

Actors:	Employee		
Pre-Conditions:	None		
Post-Conditions:	If the user has checked into the table then they will be able to see the current status of their order		
Frequency of Use:	Every transaction the restaurant has		
Flow of Events:		Actor Action	System Response
	1	Employee denotes table as occupied	System creates an order for that table
	2	Employee enters orders for that table	System updates order for that table
Variations:	No variations.		
Notes and	Customers do not need to be check into to table for this to		

Issues:	work, since they can later sign in or choose not to use Massimo.

Use Case Id:	UC-008
Use Case Name:	Checkout of Restaurant
Description:	Once user is done with transaction they can pay their bill via Android Pay

Actors:	Customer, Employee, and recommender		
Pre-Conditions:	Customer must be logged into their account, must have their payment system linked to said account, and have checked into a table		
Post-Conditions:	The order at that table will be emptied, the order will be saved to account and be viewable by recommender, and the payment will be processed for employee to receive		
Frequency of Use:	Whenever user wants to pay restaurant bill using Massimo		
Flow of Events:		Actor Action	System Response
	1	User clicks check out in order screen	The system takes them to the check-out screen
	2	User selects which items they want to pay for	System updates the grand total for the receipt
	3	User chooses the tip amount	System updates the grand total for the receipt
	4	User clicks on pay	Massimo will ask Android Pay for the grand total amount, send a receipt to customer,

			and label table as vacated.
Variations:	No variations.		
Notes and Issues:	Massimo facilitates splitting the fare, so the receipt is dynamic in the sense that if another person that has checked into the table has clicked on a specific item to pay for, you will be unable to click on it. Also, this allows for mixed payments, so some people may pay with cash, so the restaurant is responsible for checking if the table is actually vacated/ the total amount has been payed for.		

Use Case Id:	UC-009		
Use Case Name:	Update Restaurant information		
Description:	Once the Employee has logged in the Employee may update the restaurant's hours of operation and location.		
Actors:	Employee		
Pre-Conditions:	Employee must be logged into the restaurant account		
Post-Conditions:	The restaurant’s hours of operation and location will be updated and displayed on the app.		
Frequency of Use:	Anytime the restaurant’s hours change or when the location has been changed.		
Flow of Events:		Actor Action	System Response
	1	Employee clicks update restaurant info	The system takes them to the update restaurant info screen
	2	Employee selects what needs to be updated	The system goes to the correct editing screen

			whether that be hours of operation or location.
	3	Employee updates everything that is needed to be updated	The system updates the restaurant information.
Variations:	No variations.		
Notes and Issues:	Massimo easily allows an employee to update Restaurant information.		

Use Case Id:	UC-010		
Use Case Name:	Update Restaurant Menu		
Description:	Once the Employee has logged in the Employee may update the restaurant's menu.		

Actors:	Employee		
Pre-Conditions:	Employee must be logged into the restaurant account		
Post-Conditions:	The restaurant's menu will be updated and displayed on the app.		
Frequency of Use:	Anytime the restaurant wants to add/remove items and change prices.		
Flow of Events:		Actor Action	System Response
	1	Employee clicks update restaurant menu	The system takes them to the update restaurant menu
	2	Employee selects add/remove item or edit price.	The system goes to the correct editing screen whether that be add, remove

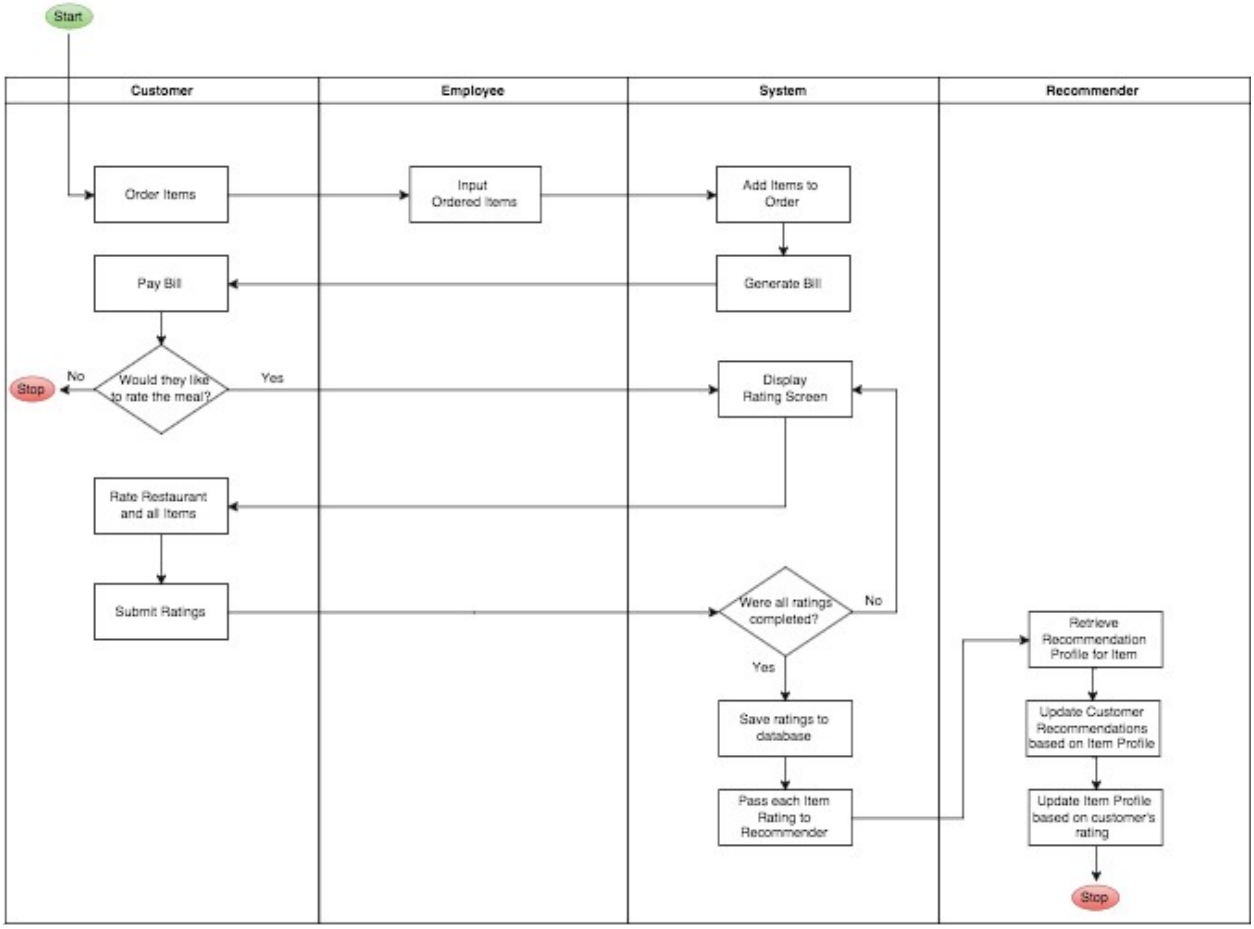
			or edit price.
	3	Employee updates everything that is needed to be updated	The system updates the restaurant menu.
Variations:	No variations.		
Notes and Issues:	Massimo easily allows employees to update the Restaurant Menu.		

Use Case Id:	UC-011		
Use Case Name:	Modify Account Info		
Description:	Customers are able to edit their username password and payment methods.		
Actors:	Customer		
Pre-Conditions:	Customer must be logged into their account and just clicked on the settings button.		
Post-Conditions:	The customer's data will be updated to the newly entered data.		
Frequency of Use:	Not very often. This use case happens when the user need to remove or add payments or needs to change password/username.		
Flow of Events:		Actor Action	System Response
	1	Customer pushes on “My settings” button.	The system pops a enterable text field and ask for the customer’s current password.
	2	Customer enters their password.	If correct the system moves on to the personal settings

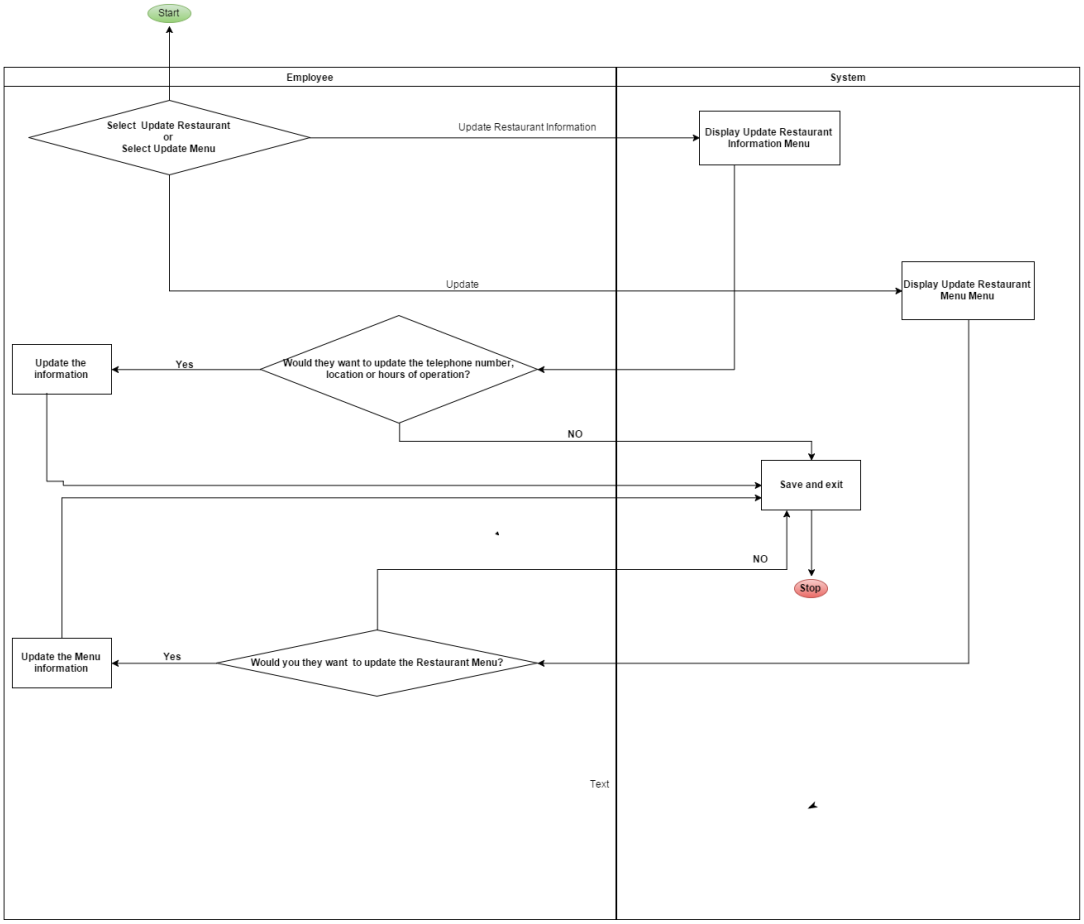
			page “my settings” if incorrect the system ask the user to enter their password again.
	3	Customer selects desired information to be changed. i.e. Username, Password, Payment method.	The system ask are you sure you want to change ___? With the desired field filled in
	4	Customer selects yes or no	If no the app moves to the “my settings page with no changes” If yes the data becomes editable.
	5	Customer edits and updates the new data.	The field is checked for validity. If valid a save changes button at the bottom becomes clickable.
	6	Customer clicks on save changes button	The app moves back to editable “my settings page”
Variations:	No variation		
Notes and Issues:	If the customer exits the app at anytime during this process re authentication is needed before continuing any of the steps.		

Activity Diagram:

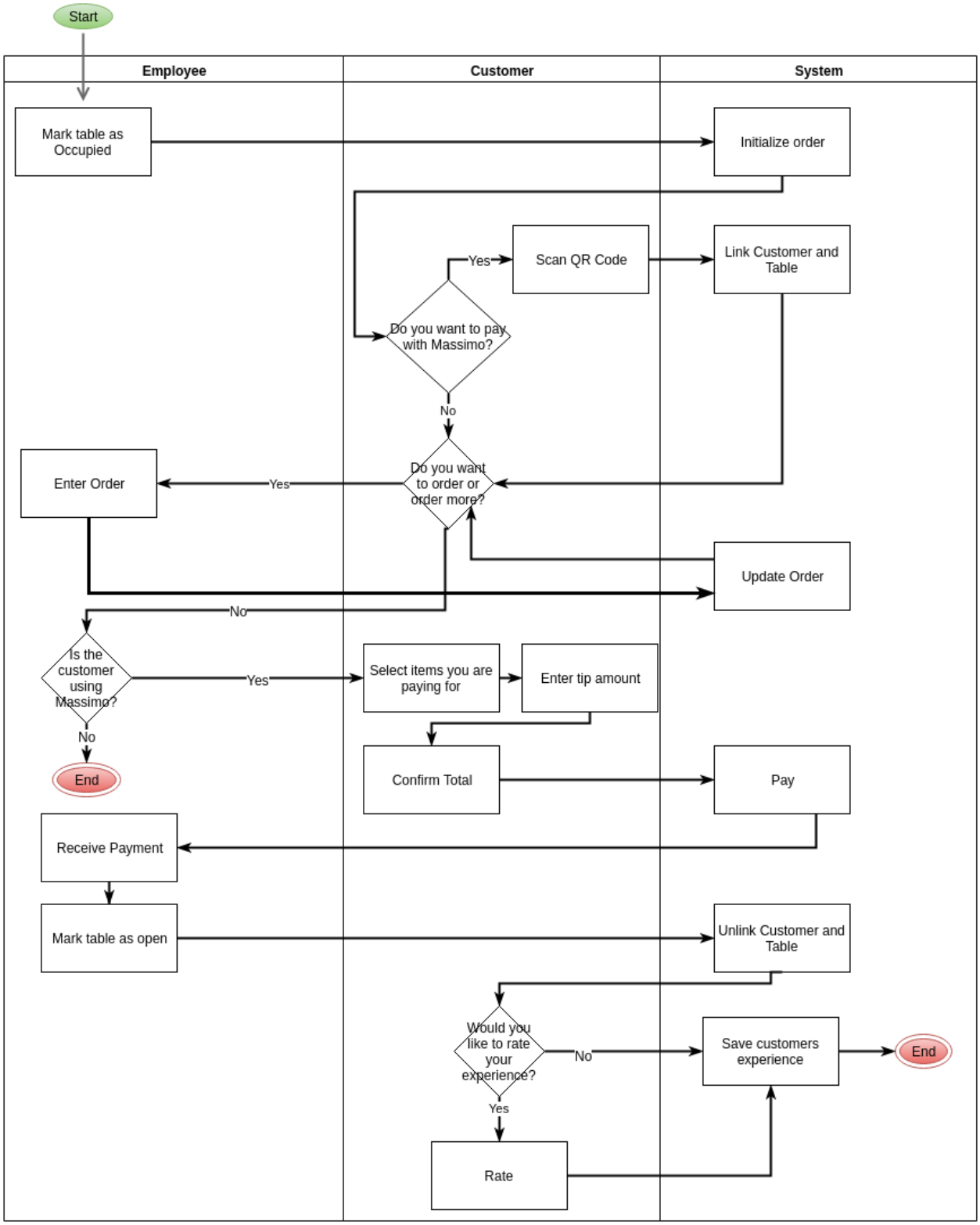
1. UC-003,UR-005,UR-006,UR-007 Landon Bedell



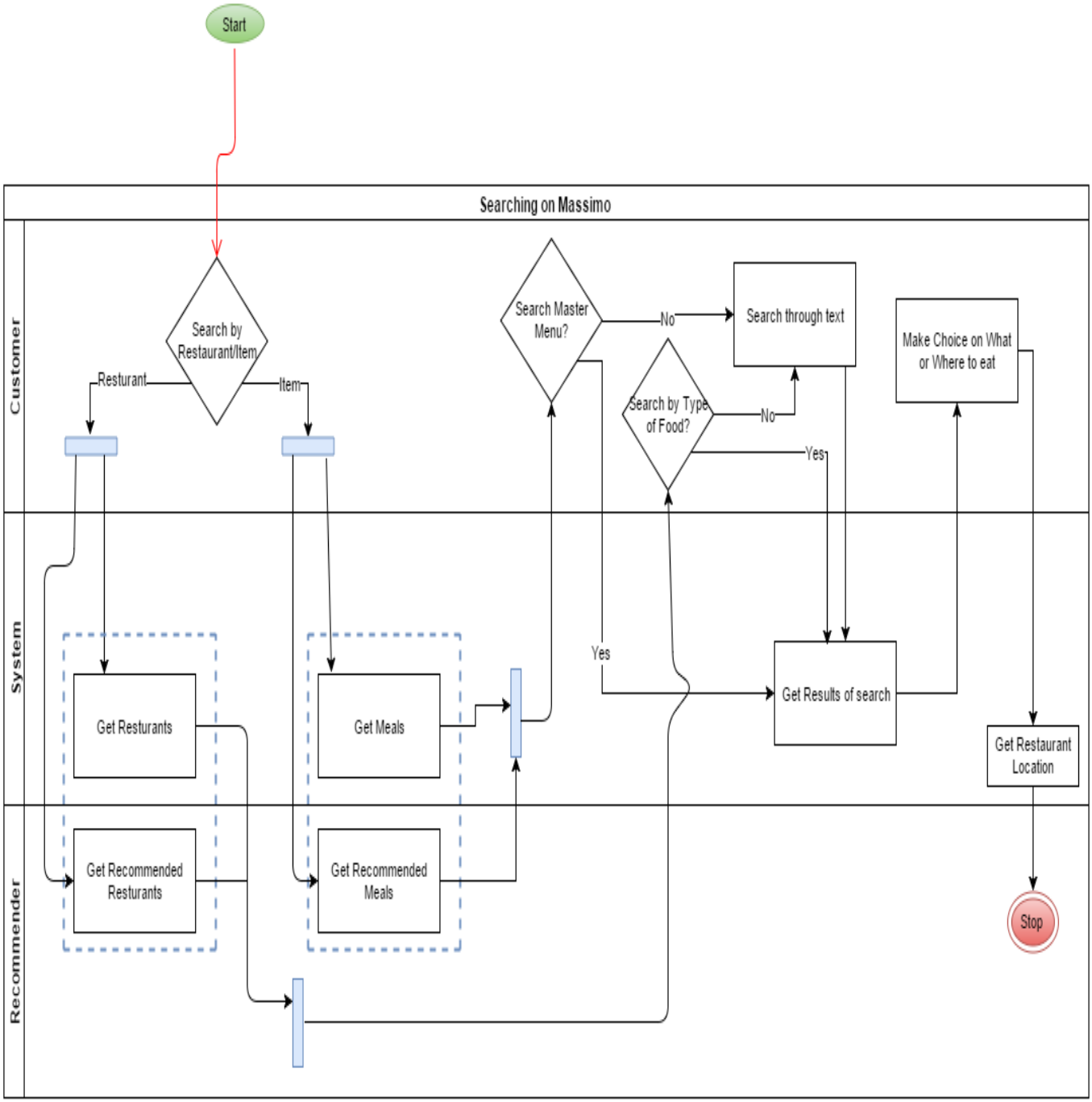
2. UR-012,UC-009 UC-010, Update Restaraunt, Deep Desai



3. Branden Romero (UC-006, UC-007, UC-008, UR-008) Checkin Checkout Update Order



4. Justin Tang (UR-004 , UC-004, UC-005, Search)

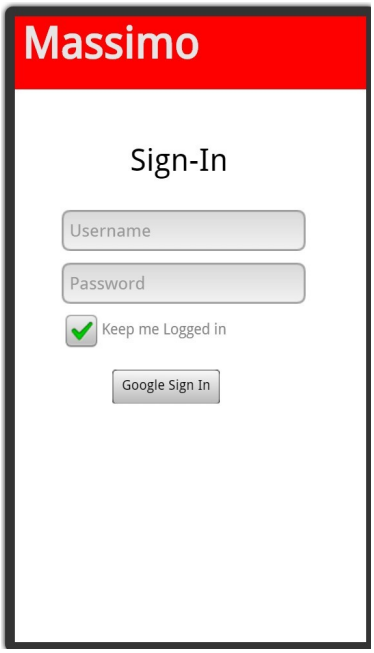


Data Storage: SQLite

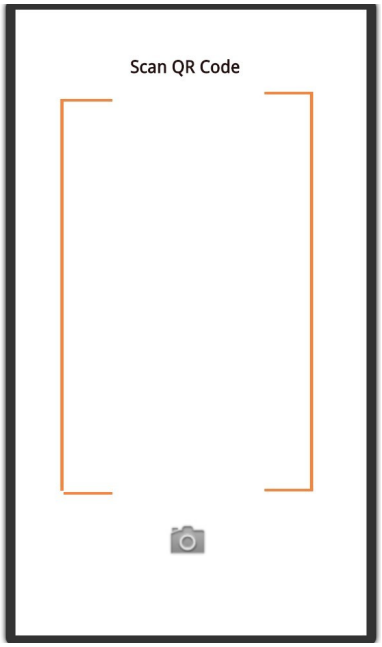
Classes:

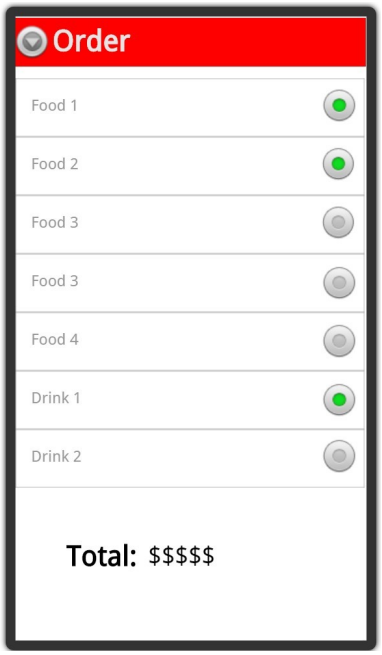
- User class to store information about employees and customers
- Restaurants class to store information about the restaurants.
- Table class to store information about the QR code and the Table ID.
- Order class to store information about the individual transactions
- Food class to store information about food items.
- Rating class to store information about customer ratings on individual items and the restaurant.

UI Mockup:


Massimo
Sign-In
Username
Password
Keep me Logged in
Google Sign In


Home
Restaurant Suggestions
Food Suggestions

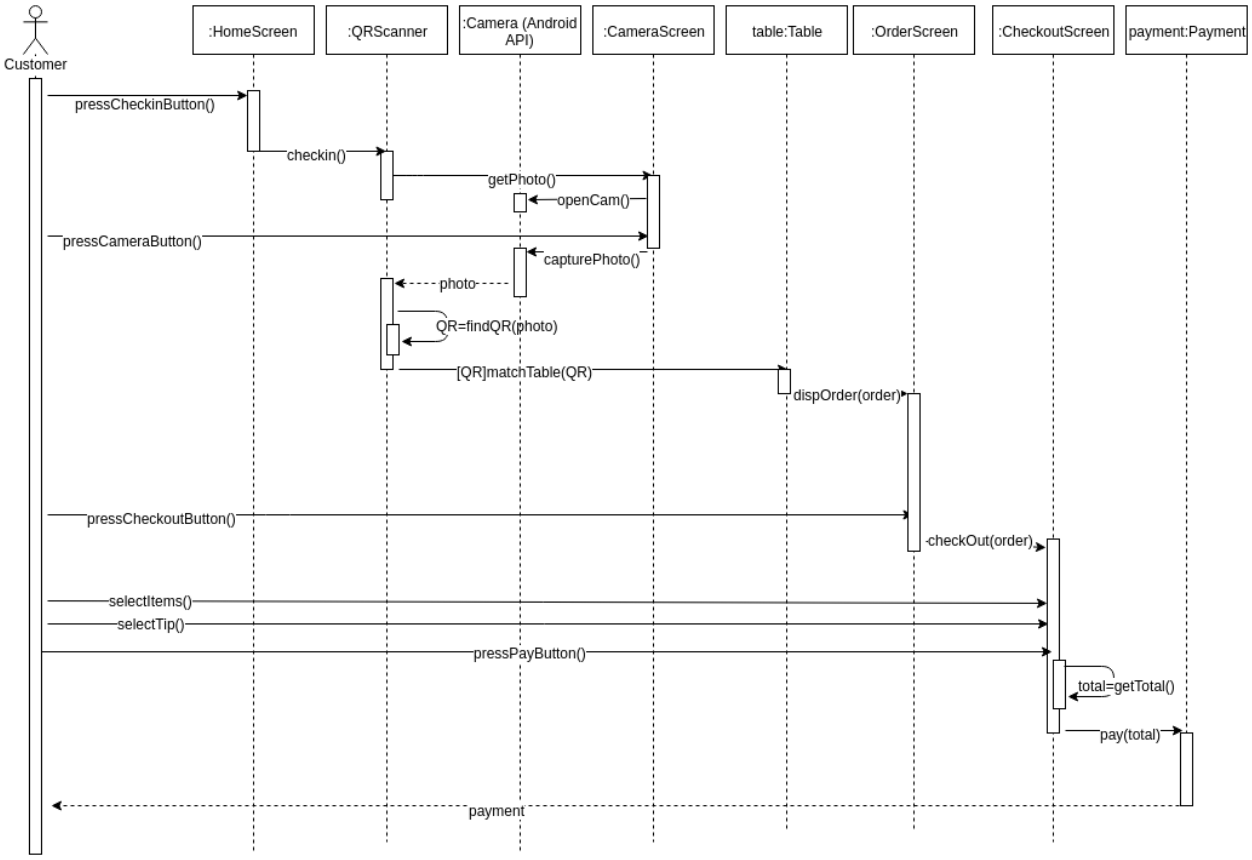

Scan QR Code


Order
Food 1
Food 2
Food 3
Food 3
Food 4
Drink 1
Drink 2
Total: $$$$$

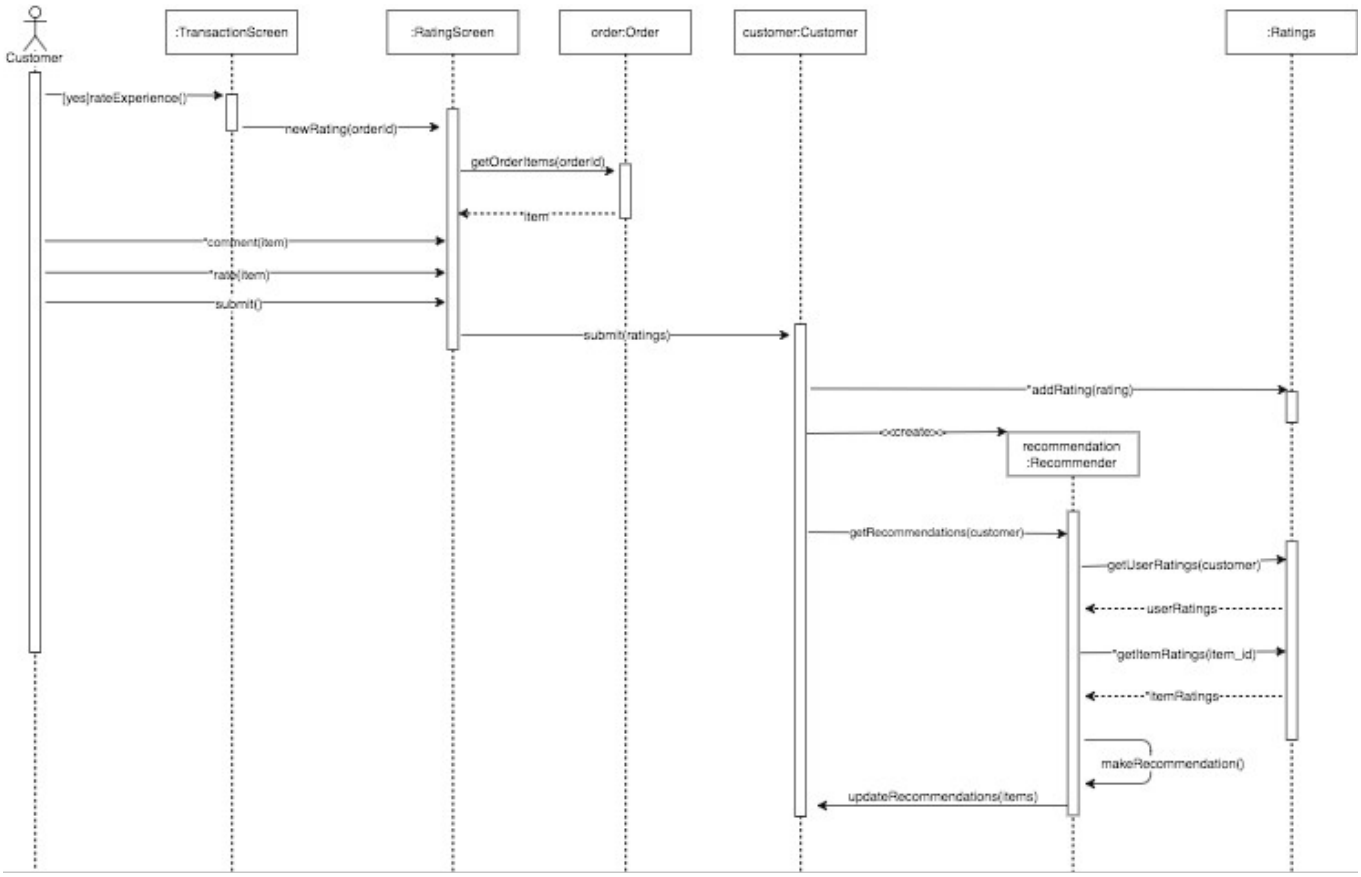

Checkout
Tip
10% $$$
15% $$$
20% $$$
Other
Total: $$$$$
Finished

User Interactions:

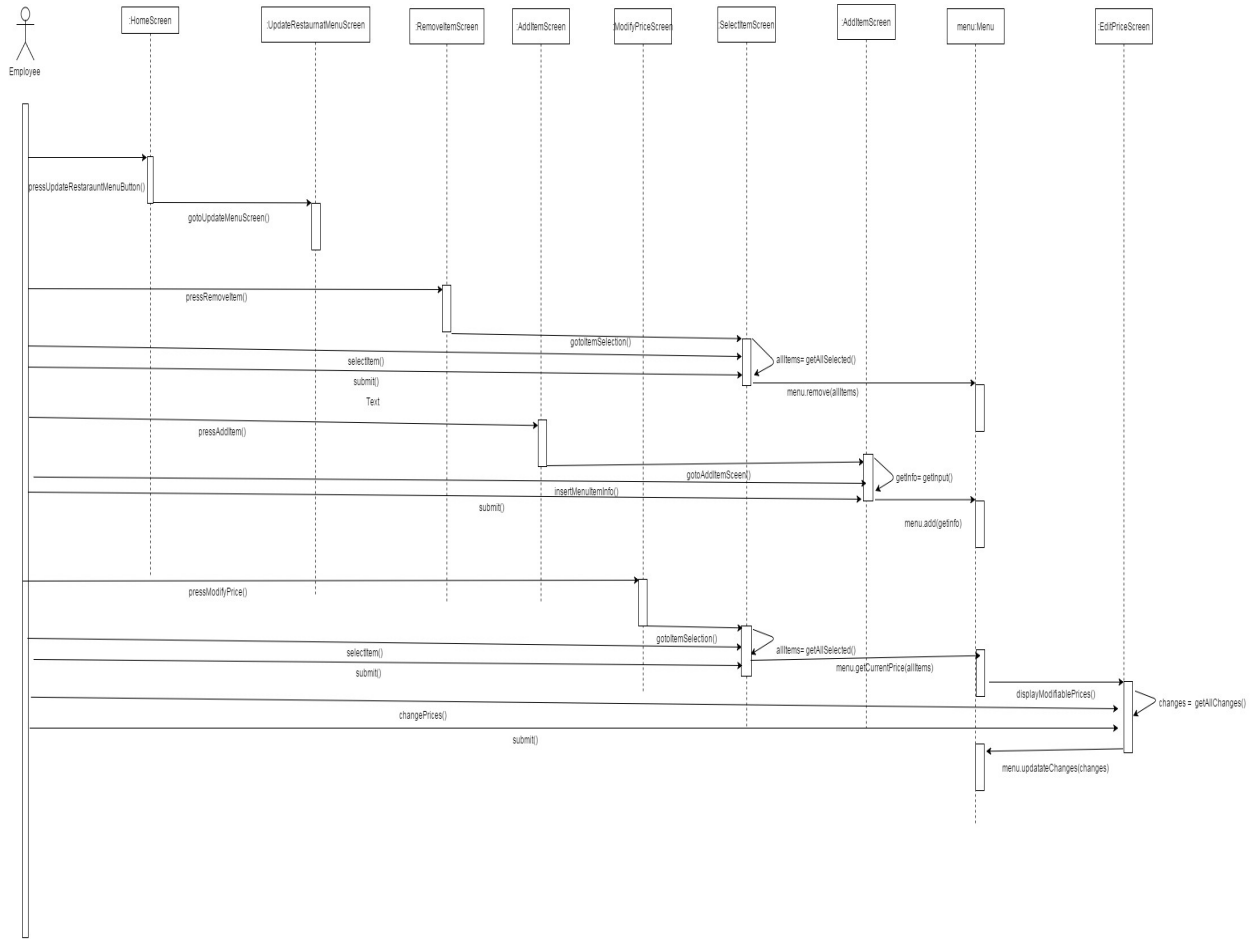
1. Branden Romero (UC-006, UC-008, UR-008) Checkin Checkout



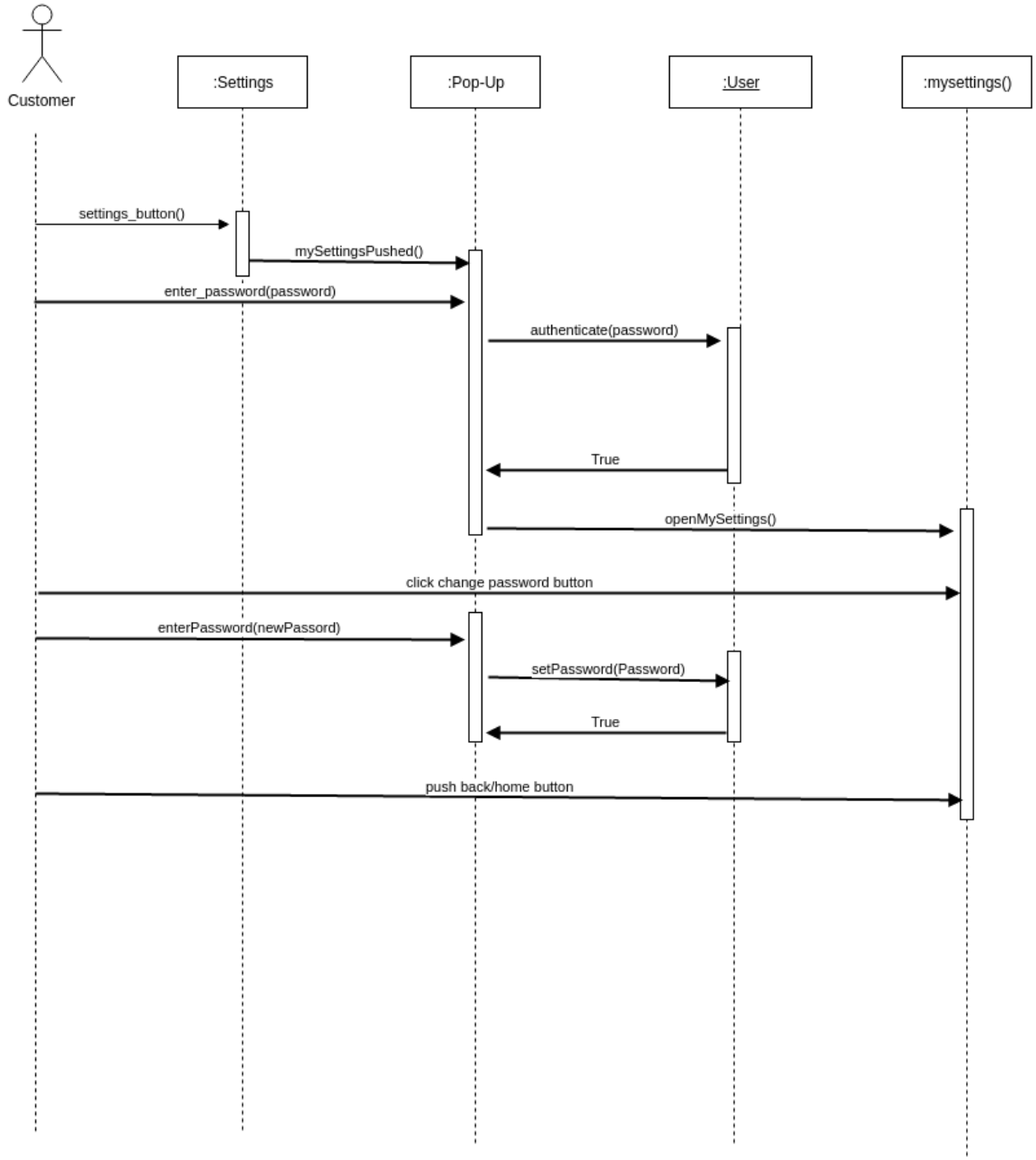
2. Landon Bedell (UC-003,UR-005,UR-006,UR-007) Rate Experience



3. UR-012,UC-010, Update Restaurant Menu, Deep Desai



4. Justin Tang (UC-004, UC-005, UR-010) Update User Settings



Class Diagram: