**Use Case Name:**Edit Account Information  
**Actors:** User  
  
**Summary:** The user’s information has changed in some way. This could be a change of address (important since pizza must be delivered), has a new phone number and cannot be reached on the old number, or has a new credit card that they need to add to the system.   
  
**Preconditions:**Information was previously saved to the system and a user needs to edit the information  
  
**Post-Conditions:** The old information gets removed from the system.  
**Uses:** Record Transactions, Cancel Action  
  
**Extends:** None  
  
**Normal Course of Events:**

1. The use case starts when the user wants to edit some information that they have in the account.
2. The user selects the set of information that he/she would like to edit and directs the system to delete the information. – Exception 2, 3
3. The system responds by asking the user to confirm the new information.
4. The user confirms edits.
5. Alternative Path: Cancel Action
6. A system responds by deleting the old information and saving the new data. The system will provide the user a record of the edits made and will email the user regarding the recent changes to the account.
7. Uses: Record Transaction
8. This use case ends.

**Alternative Path - The user does not confirm Deletion**

1. If the user does not update, the new information will be discarded.
2. Uses: Cancel Action

**Exceptions:**

1. The system does not allow address update if location is not accessible via API location information.
2. The system will not allow the user to save null values for phone number and address.
3. The system will not allow foreign numbers to be saved in the system.

**Assumptions:**

1. Deleted information is not retained in the system.
2. A user can only edit information if an account has been previously created, and the user can login to this account.

**Use Case Name:**Set up new profile/account  
**Actors:** User  
  
**Summary:** The user is interested in the company or product and would like to create an account to start ordering/receiving pizza/   
  
**Preconditions:**The user has not created an account yet and does not have their email registered.   
  
**Post-Conditions:** User information gets stored in the system database.  
**Uses:** Approve and save account, Cancel Action  
  
**Extends:** None  
  
**Normal Course of Events:**

1. The use case begins when the user wants to create an account with the company.
2. The user goes through the required fields in the form and inputs their information.
3. System responds by asking the user to confirm the information. Exception can be raised if the user information has been previously used before (Email) , if password created does not match system specifications, or if address cannot be found .
4. User is successful and confirms all information
5. Alternative path: User cancels action. Chooses not to create an account.
6. System will respond by welcoming the user with a prompt. System will submit an email to user to confirm/notify the creation of the account
7. System will provide a quick tutorial of how to navigate the UI and direct user to “GET STARTED”/ start ordering
8. This use case ends.

**Exceptions:**

1. The email has been used to previously register with the system.
2. The system does not allow address update if location is not accessible via api location information.
3. The system will not allow the user to save null values for phone number and address.
4. The system will not allow foreign numbers to be saved in the system.

**Assumptions:**

1. The user does not currently have an account registered with email.

**Use Case Name:**Create Order.   
**Actors:** User  
  
**Summary:** The user has an account and has logged in. The user would like to begin a new order and have food delivered.   
  
**Preconditions:**The user has an account. The user has all information registered in the account required to make the purchase, and complete delivery (phone number, address).

**Post-Conditions:** User can make a payment.

**Uses:** User creates order with items, cancel order.   
  
**Extends:** None  
  
**Normal Course of Events:**

1. The use case begins when the user would like to create a new order.
2. User interacts with menu UI to choose items that they want to add to their cart.
3. User can edit items, add, or remove ingredients from items.
4. System calculates total, includes tax amount – Background process
5. System displays full information about order total to user. Exception 1: user cancels order
6. Alternate path: user cancels, ask user if they would like to save order, or delete. Use case ends
7. Alternate path: user wants to add more items to cart return to order.
8. Alternate path: user wants to remove items from cart. Offer quick remove item button. Recalculate total and proceed back to step 5.
9. User accepts the order total -> proceed to payment
10. Use case ends.

**Exceptions:**

1. User Cancels order

**Assumptions:**

1. The user has a credit card, debit card, or some kind of viable and accepted payment option.
2. User has an account already registered.
3. User can either pick up pizza, or has an address to deliver pizza to.

**Use Case Name:**Make Payment  
**Actors:** User  
  
**Summary:** The user has created an order with the items that they would like to purchase, before the order can be registered to a queue, the user must pay for their order.   
  
**Preconditions:**The user has an account. The user has all information registered in the account required to make the purchase, and complete delivery (phone number, address). User has payment method with available funds.

**Post-Conditions:**  the order is initiated, and the user expects food delivery.

**Uses:** Make successful payment and proceed, cancel order.   
  
**Extends:** None  
  
**Normal Course of Events:**

1. The use case begins when the user needs to pay for the order.
2. User fills out all required fields in the form, including form of payment. Exception 4: if the user would like to pay in store upon pickup, cancel delivery service option. Submit order to employees/store to begin cooking.
3. User fills out credit card information. Exception: 1, throw error and prompt, let user know what field is not complete.
4. User successfully completes all fields. Exception 2 or 3, the user’s information is not accepted by the system. Prompt user that information is either incorrect, or bank has rejected card. Allow user to clear form and start over with new payment method.
5. Alternative path: user cancels process. Use case ends
6. User is successful. Prompt user with approval and allow to submit order/ complete payment process.
7. Send Order information to restaurant to add to queue.
8. Use case ends.

**Exceptions:**

1. User does not completely fill out all payment options
2. User payment option does not exist
3. User payment option does not clear funds verification.
4. User would like to pick up food in person and pay in cash.
5. User cancels payment process.

**Assumptions:**

1. The user has a credit card, debit card, or some kind of viable and accepted payment option.
2. The user needs to have completed the create order use case, this order has a list of all items the user would like to purchase and calculates the order total with tax included.