# Pizza Ordering System

# Problem Statement

## Objective

Development of a commercial Pizza Ordering System which allows the users to view and order pizza.

## Abstract of the project

This project is aimed at developing a Pizza Ordering System. This is a web based application that can be accessed throughout the web. This application consists of two actors.

* Admin
* User

This application enables the admins to add, remove, modify or update pizza and store details in the database. It enables the user to perform CRUD operations (i.e., add, remove, modify, update pizza into the cart). Once the order is confirmed, the user must get a notification prompting that the order has successfully been placed.

## Functional components of the project

Following is a list of functionalities of the system. Wherever, the description of functionality is not adequate; you can make appropriate assumptions and proceed.

1. Admin should be able to
   * Add store/pizza details into the database.
   * Remove store/pizza details into the database.
   * Modify store/pizza details into the database.
   * Update store/pizza details into the database.
2. User should be able to
   * Add pizza to the cart.
   * Remove pizza to the cart.
   * Modify pizza to the cart.
   * Update pizza to the cart.

Assumptions:

1. One Store can have many pizza.
2. One User can have only one pizza.
3. Customer can book one pizza at a time.

Validations:

1. Username should contain atleast 5 digits.
2. Password must be a strong password.
3. Phone number should be of 10 characters.
4. username should be unique.

Scope**:**

Using the Pizza Ordering System, a user can make one order at a time. The user can select a store and based on that user can select a pizza and finally make a order. The admin can add, update and delete store and pizza.

Out of Scope**:**

Customer cannot make more than one order.

**Class and Methods Description:**

The application will consist of the following classes, the utility of which have been described below:

1. **Customer:** This class stores the details of Customer:

Attributes:

userId: String

userName: String

userEmail: String

userPhone: String

firstName: String

lastName: String

roles: String

active: boolean

Methods: -

*register(): Register a new user*

*Validate if all field are valid and register the user.*

*logIn()*: User will login if credentials are correct.

*logOut()*: Removes all the session related information of a user.

makeCart(username, pizza):Will first show the user a list of stores and then the user can select the pizza.

The following methods will only be allowed for user type:

1. **Administrator**. :-

*addStore(Store):* Adds a *Store*.

*removeStore (Store):* Removes a Store

*update* Store *(*Store*):* Updates a Store

*addPizza(Pizza):* Adds a *Pizza*.

*updatePizza (Pizza):* Updates *Pizza*

*deletePizza(Pizza):* Deletes *Pizza*

1. **Store:** This class stores the details of Customer:

Attributes:

storeId: String

storeName: String

storePhone: String

storeAddress: String

1. **Pizza:** This class stores the details of Customer:

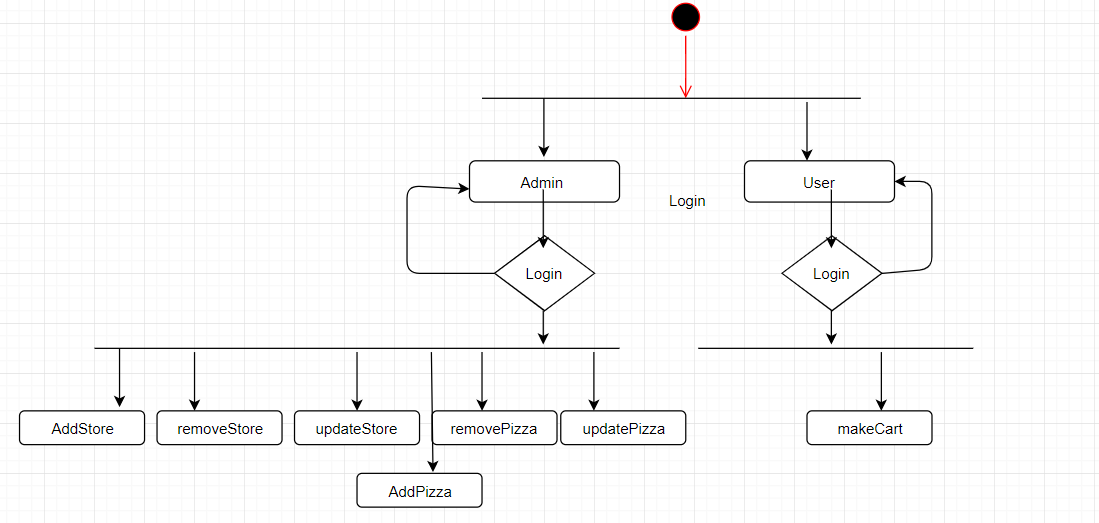
Attributes:

pizzaId: String

pizzaName: String

pizzaDescription: String

**Activity Diagram**



Class Diagram

