

# 1. Facebook - A Social Network

## ➤ Description

- Facebook is an online social networking service where users can connect with other users to post and read messages. Users access Facebook through their website.
- **System Requirements:**
  - Each member should be able to add information about their basic profile, work experience, education, etc.
  - Any user of our system should be able to search for other members, groups or pages by their name.
  - Members should be able to send and accept/reject friend requests from other members.
  - Members should be able to follow other members without becoming their friend.
  - Members should be able to create groups and pages, as well as join already created groups, and follow pages.
  - Members should be able to create new posts to share with their friends.
  - Members should be able to add comments to posts, as well as like or share a post or comment.
  - Members should be able to create privacy lists containing their friends. Members can link any post with a privacy list to make the post visible only to the members of that list.
  - Any member should be able to send messages to other members.
  - Any member should be able to add a recommendation for any page.
  - The system should send a notification to a member whenever there is a new message or friend request or comment on their post.
  - Members should be able to search through posts for a word.
- **We have three main actors in our system:**
  - Member: All members can search for other members, groups, pages, or posts, as well as send friend requests, create posts, etc.
  - Admin: Mainly responsible for admin functions like blocking and unblocking a member, etc.
  - System: Mainly responsible for sending notifications for new messages, friend requests, etc.
- **Here are the top use cases of the Facebook System:**
  - Add/update profile: Any member should be able to create their profile to reflect their work experiences, education, etc.
  - Search: Members can search for other members, groups, or pages. Members can send a friend request to other members.
  - Follow or Unfollow a member or a page: Any member can follow or unfollow any other member or page.
  - Send message: Any member can send a message to any of their friends.

- Create post: Any member can create a post to share with their friends, as well as like or add comments to any post visible to them.
- Send notification: The system will be able to send notifications for new messages, friend requests, etc.
  - A notification can be an email, SMS, WhatsApp message (one or even a combination of any of them).

#### ➤ Group size

- 5 - 6 members.

#### ➤ Deliverables

- A System with GUI that fulfills the above requirements.
- You must apply at least 3 reasonable design patterns.
- Project documentation:
  - The project's name.
  - The team members' info (ID, name and section number).
- A UML class diagram for the project.

#### ➤ [Bonus]

- Apply more than 3 patterns.
- Build the system as a web application using .NET framework and C#.
- **Amazing** GUI.

## **2. Library Management System**

### ➤ **Description**

- A Library Management System is a software built to handle the primary housekeeping functions of a library. Libraries rely on library management systems to manage asset collections as well as relationships with their members. Library management systems help libraries keep track of the books and their checkouts, as well as members' subscriptions and profiles.
- Library management systems also involve maintaining the database for entering new books and recording books that have been borrowed with their respective due dates.
- **System Requirements:**
  - Any library member should be able to search books by their title, author, subject category as well by the publication date.
  - Each book will have a unique identification number and other details including a rack number which will help to physically locate the book.
  - The system should be able to retrieve information like who took a particular book or what are the books checked-out by a specific library member.
  - There should be a maximum limit (5) on how many books a member can check-out.
  - There should be a maximum limit (10) on how many days a member can keep a book.
  - The system should be able to collect fines for books returned after the due date.
  - The system should be able to send notifications whenever the reserved books become available, as well as when the book is not returned within the due date.
- **We have three main actors in our system:**
  - Librarian: Mainly responsible for adding and modifying books, book items, and users. The Librarian can also issue, reserve, and return book items.
  - Member: All members can search the catalog, as well as check-out, reserve, renew, and return a book.
  - System: Mainly responsible for sending notifications for overdue books, canceled reservations, etc.
- **Here are the top use cases of the Library Management System:**
  - Add/Remove/Edit book: To add, remove or modify a book or book item.
  - Search catalog: To search books by title, author, subject or publication date.
  - Register new account/cancel membership: To add a new member or cancel the membership of an existing member.
  - Check-out book: To borrow a book from the library.
  - Reserve book: To reserve a book which is not currently available.
  - Renew a book: To reborrow an already checked-out book.

- Return a book: To return a book to the library which was issued to a member.
- Send notification: The system will be able to send notifications for overdue books, canceled reservations, etc.
  - A notification can be an email, SMS, WhatsApp message (one or even a combination of any of them).

#### ➤ Group size

- 5 - 6 members.

#### ➤ Deliverables

- A System with GUI that fulfills the above requirements.
- You must apply at least 3 reasonable design patterns.
- Project documentation:
  - The project's name.
  - The team members' info (ID, name and section number).
- A UML class diagram for the project.

#### ➤ [Bonus]

- Apply more than 3 patterns.
- Build the system as a web application using .NET framework and C#.
- **Amazing** GUI.

### 3.Movie Ticket Booking System

#### ➤ Description

- An online movie ticket booking system facilitates the purchasing of movie tickets to its customers. E-ticketing systems allow customers to browse through movies currently playing and book seats, anywhere and anytime.
  
- **System Requirements:**
  - It should be able to list the cities where affiliate cinemas are located.
  - Each cinema can have multiple halls and each hall can run one movie show at a time.
  - Each Movie will have multiple shows.
  - Customers should be able to search movies by their title, language, genre, release date, and city name.
  - Once the customer selects a movie, the service should display the cinemas running that movie and its available shows.
  - The customer should be able to select a show at a particular cinema and book their tickets.
  - The service should show the customer the seating arrangement of the cinema hall. The customer should be able to select multiple seats according to their preference.
  - The customer should be able to distinguish between available seats and booked ones.
  - The system should send notifications whenever there is a new movie, as well as when a booking is made or canceled.
  - Customers of our system should be able to pay with credit cards or cash.
  - The system should ensure that no two customers can reserve the same seat.
  - Customers should be able to add a discount coupon to their payment.
  
- **We have five main actors in our system:**
  - Admin: Responsible for adding new movies and their shows, canceling any movie or show, blocking/unblocking customers, etc.
  - Front Desk Officer: Can book/cancel tickets.
  - Customer: Can view movie schedules, book, and cancel tickets.
  - Guest: All guests can search movies but to book seats they have to become a registered member.
  - System: Mainly responsible for sending notifications for new movies, bookings, cancellations, etc.
    - A notification can be an email, SMS, WhatsApp message (one or even a combination of any of them).

➤ **Here are the top use cases of the Movie Ticket Booking System:**

- Search movies: To search movies by title, genre, language, release date, and city name.
- Create/Modify/View booking: To book a movie show ticket, cancel it or view details about the show.
- Make payment for booking: To pay for the booking.
- Add a coupon to the payment: To add a discount coupon to the payment.
- Assign Seat: Customers will be shown a seat map to let them select seats for their booking.
- Refund payment: Upon cancellation, customers will be refunded the payment amount as long as the cancellation occurs within the allowed time frame.

➤ **Group size**

- 5 - 6 members.

➤ **Deliverables**

- A System with GUI that fulfills the above requirements.
- You must apply at least 3 reasonable design patterns.
- Project documentation:
  - The project's name.
  - The team members' info (ID, name and section number).
- A UML class diagram for the project.

➤ **[Bonus]**

- Apply more than 3 patterns.
- Build the system as a web application using .NET framework and C#.
- **Amazing** GUI.

## 4. Restaurant Management System

### ➤ Description

- A Restaurant Management System is a software built to handle all restaurant activities in an easy and safe manner. This System will give the Restaurant management power and flexibility to manage the entire system from a single portal. The system allows the manager to keep track of available tables in the system as well as the reservation of tables and bill generation.
- **System Requirements:**
  - The restaurant will have different branches.
  - Each restaurant branch will have a menu.
  - The menu will have different menu sections, containing different menu items.
  - The waiter should be able to create an order for a table and add meals for each seat.
  - Each meal can have multiple meal items. Each meal item corresponds to a menu item.
  - The system should be able to retrieve information about tables currently available to seat walk-in customers.
  - The system should support the reservation of tables.
  - The receptionist should be able to search for available tables by date/time and reserve a table.
  - The system should allow customers to cancel their reservation.
  - The system should be able to send notifications whenever the reservation time is approaching.
  - The customers should be able to pay their bills through credit card, check or cash.
  - Each restaurant branch can have multiple seating arrangements of tables.
- **We have six main actors in our system:**
  - Receptionist: Mainly responsible for adding and modifying tables and their layout and creating and canceling table reservations.
  - Waiter: To take/modify orders.
  - Manager: Mainly responsible for adding new workers and modifying the menu.
  - Chef: To view and work on an order.
  - Cashier: To generate checks and process payments.
  - System: Mainly responsible for sending notifications about table reservations, cancellations, etc.
- **Here are the top use cases of the Restaurant Management System:**
  - Add/Modify tables: To add, remove, or modify a table in the system.
  - Search tables: To search for available tables for reservation.
  - Place order: Add a new order in the system for a table.

- Update order: Modify an already placed order, which can include adding/modifying meals or meal items.
- Create a reservation: To create a table reservation for a certain date/time for an available table.
- Cancel reservation: To cancel an existing reservation.
- Check-in: To let the guest check in for their reservation.
- Make payment: Pay the check for the food.
- Send notification: The system will be able to send notifications about table reservations, cancellations, etc.
  - A notification can be an email, SMS, WhatsApp message (one or even a combination of any of them).

#### ➤ Group size

- 5 - 6 members.

#### ➤ Deliverables

- A System with GUI that fulfills the above requirements.
- You must apply at least 3 reasonable design patterns.
- Project documentation:
  - The project's name.
  - The team members' info (ID, name and section number).
- A UML class diagram for the project.

#### ➤ [Bonus]

- Apply more than 3 patterns.
- Build the system as a web application using .NET framework and C#.
- **Amazing** GUI.



## 5.Mailing System

### • Description

- Mailing System is a system for sending and receiving mails with some more advanced options.
- **System Requirements:**
  - Each member should be able to add information about their basic profile, work experience, education, etc.
  - Any user can add contacts with details like name, mail, phone number, relation
  - Any user can make a group of contacts with specific name and description.
  - Send mail to contact or a group
  - User should be able to receive mails.
  - Report mail as a spam or add contact as a spam list which redirect mails from this contact to spam
  - Create directories in your mail
  - Select mail and move it to specific folder
  - User can delete mail from the inbox and move directly to trash
  - Each mail is removed completely from spam or trash after 30 days
  - The message has 2 statuses read and unread
  - User can change message status from read to unread
  - User can bin mails to appear at the top of mail messages
  - mails can be flagged as important
  - System should have the search functionality with (mail address, name, date, flagged as important)
  - Send notification with the new mail
  - Message counter to the unread messages in the inbox folder
  - Automatically forward mails from specific contact to specific folder
  - Send messages should appear in sent folder
  - User can write mail and save it as a draft to edit or send it latter
  - User can schedule mail to be sent latter in specific date and time
    - System should validate the date and time before save (error appears without saving)
    - Validate that the mail has a destination the save a time before save ( error appears without saving)
    - Make sure that the mail has body (error appears without saving)

### • Group size

- 5 - 6 members.

### • Deliverables

- A System with GUI that fulfills the above requirements.

- You must apply at least 3 reasonable design patterns.
- Project documentation:
  - The project's name.
  - The team members' info (ID, name and section number).
- A UML class diagram for the project.

- **[Bonus]**

- Apply more than 3 patterns.
- Build the system as mobile with Flutter.
- **Amazing** GUI.

## 6.Shared Ride Application

### • Description

- Shared Ride is an application that is used by car owner and customers, providing delivery services by car owners, and managing the services by the system company's employees.
- Car owner add his trip start and destination location also add the start time and the expected end time.
- When the system gets a request to same locations and same time slot start so, the system when receive requests to same start point, destination and time slot send the request to the car owner who can accept or reject.
- The system send notification back to the client
- The designed system consists of modules:
- **Customer Module:** This module gives the ability for customer to
  - Register to the system
  - Login
  - Book a car by doing the following:
    - Write the pickup point
    - Write the destination place.
    - Write the time of the trip. (The time when the customer wants the car to come).
    - Confirm the trip information.
  - Update his/ her personal info.
  - View his/her previous trips' details.
  - rate the car owner.
  - Send complaints through the system. Employees can view customers' complaints.
- **Car Owner module:** This module gives the ability for a car owner to
  - Login
  - Add the trips he intends to drive.
  - Search available trips.
  - Accept a trip.
  - View his/her previous trips details.
  - Every car owner has a car, and its data should be stored on the system
- **Employee module:** This module gives the ability for an employee to
  - Login
  - View Customers complains.
  - Remove car owner with more than 10 complains.
- **Trip module:** Every trip has
  - Customer.
  - Car Owner
  - Pickup point.
  - Destination.
  - Trip time. (The time when the Car Owner intend to start the journey).

- Group size

- 5 - 6 members.

- Deliverables

- A System with GUI that fulfills the above requirements.
- You must apply at least 3 reasonable design patterns.
- Project documentation:
  - The project name.
  - The team members' info (ID, name, and section number).
- A UML class diagram for the project.

- [Bonus]

- Apply more than 3 patterns.
- Build the system as an android mobile application.
- **Amazing** GUI.

## 7.A Grocery Shop application

### ➤ Description

- The main objective of this application is to create e-commerce platform for customers to buy and sell fruit and vegs. It's an online shop where you can place your order according to your requirements.
- There are three types of actors in the application which are ADMIN, Shop owner and Customers.
- **Shop owner functionalities:**
  - Sign in with his username and password.
  - Every shop owner can manage his products.
    - Add, delete, edit items details such as prices and available quantities.
  - Can view the customers' orders and their details.
  - Can upload the image of the products and change the price at any time.
  - See reports about his best seller items.
- **Customer functionalities:**
  - Can view products and place an order to buy them.
    - To buy an item they should specify some details like the quantity of items.
    - Can see the products but cannot make orders without login.
    - The prices of the products can be changed by the city or the address of the customer.
  - Can cancel or edit their orders after have been made.
- **Admin functionalities:**
  - Login with user name and password.
  - Can check the order history of any shop in the system.
  - Can view all the transactions like pending orders or the placed orders.
  - Can remove a shop or accept the new shop owners.
  - Can add more categories of items or update the current categories.
  - Can view all the transactions like pending orders or the placed orders.

### ➤ Group size

- 5 - 6 members.

### ➤ Deliverables

- A System with GUI that fulfills the above requirements.
- You must apply at least 3 reasonable design patterns.
- Project documentation:
  - The project's name.
  - The team members' info (ID, name and section number).

- A UML class diagram for the project.

#### ➤ [Bonus]

- Apply more than 3 patterns.
- Build the system as a web application using JSP and servlet.
- Handel more than one customer at the same time.
- **Amazing** GUI.

## 8.Online Book Order System (OBOS)

### ➤ Description

- Design, develop and test an application for ordering books. The system will consist of two main subsystems: administration system and user system.
- The administration system provides: items administration, orders management, log of purchased books and support for statistics. The user system is a web application system that permits the creation of book orders. Every time a book is ordered, the order appears in the administration system so that the administrator confirms the order.
- **User functionalities:**
  - Sign in with a username, password, address and phone then login.
  - Add, edit, or delete books from his/her cart.
    - each book has its specific details, price and popularity.
  - View reviews of a specific book.
  - Add a review to the purchased books.
  - Check the order status with the ability to cancel the order.
  - Filter the books by categories or sort by popularity
    - there can be books for IT, history, classics, crimes, ...etc.
  - The user can check his previous orders that he/she made before
    - this information will be displayed to the user either after he login or by clicking a specific link in the home page.
  - The user should reach a payment page after selecting the books and place the order.
- **Admin functionalities:**
  - Add, delete books or edit their specifications in the system
    - Add, delete new category of books.
    - Change the status of books such as in stock or few numbers left.
    - Specifications of books can be: number of available units - book price – book edition- book cover that will be shown to the customers...etc.
  - Every time a user made an order it should be appeared in the administration system so that the admin can confirm or cancel the order.
  - The admin can see some statistics and figures about the type of sold books.

### ➤ Group size

- 5 - 6 members.

### ➤ Deliverables

- A System with GUI that fulfills the above requirements.
- You must apply at least 3 reasonable design patterns.

- Project documentation:
  - The project's name.
  - The team members' info (ID, name and section number).
- A UML class diagram for the project.

➤ [Bonus]

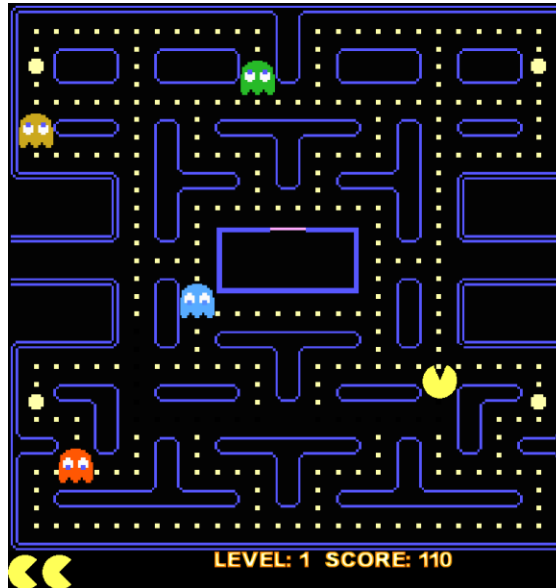
- Apply more than 3 patterns.
- Use data mining techniques to show the statistics and charts about the sold books.
- **Amazing** GUI.



## 9.Pacman Game

### ➤ Description

- Implement the well-known [Pac Man](#) game using java and any graphics library.



- Your game should be able to:
  - Build an original game world map consisting of obstacles and free roads that contains food points.
  - Move the Pac man player according to the user input by keyboard.
  - Move the enemies randomly.
  - Increment the score counter according to food points collected.
  - Detect collision between player and any enemy (handle losing case)
  - Move to the next level if the player finished all food points (handle winning case)
  - Game should have at least 2 levels of 2 different original & distinct world maps.
  - Game should have the ability to modify the configurations of the game such as the game difficulty and number of enemies (1~4).

### ➤ Group size

- 5 - 6 members.

### ➤ Deliverables

- A System with GUI that fulfills the above requirements.
- You must apply at least 3 reasonable design patterns.
- Project documentation:

- The project name.
  - The team members' info (ID, name, and section number).
- A UML class diagram for the project.

➤ **[Bonus]**

- Apply more than 3 patterns.
- Enemies are smart enough to determine the player position and move towards it (you can search for and use A\* or BFS algorithm).
- **Amazing** GUI and sound effects.

## 10.Please! Make my Pizza

### ➤ Description

- This project is designed to make it easy for users to purchase pizza online. It will simply allow a user to get his/her favorite pizzas online without any effort.
- It is a multi-role application project with a single Admin and several Users, with the admin having complete authority over the system.
- Admin will be in charge of maintaining the system as well as keeping track of orders and system administration, pizza delivery to customers, delivery workers, and their availability. Users, on the other hand, may register, log in using their login credentials, and purchase pizza online.
- Admin can categorize the pizza, add/remove/update any pizza, edit prices of the pizza, view/confirm/cancel orders placed by users, accept payments online, and so on. Admin can keep all the records of the Users.
- Users can add one or more pizzas of their choice to the cart, make payments online and get it delivered to their doorstep.
- Users will also be able to order a customized pizza i.e. users can choose pizza crust, toppings, cheese, etc of their choice.
- Two users can interact with this application 1) Admin 2) User
- **Admin functionalities:**
  - Can Log in/Log out of the system.
  - Can View/Edit/Delete pizza categories into the system.
  - Can View/Confirm/Cancel booking orders have placed by the User.
  - Can check payments done by User.
  - Can check the availability of pizza in stock.
  - Can manage Delivery staff.
  - Can change prices of pizza.
  - Can change password.
  - Can manage “My Profile”.
- **Users’ functionalities:**
  - Can Log in/Log out of the system.
  - Can Manage “My profile”.
  - Can search for pizza in categories.
  - Users can Place/Cancel an order.
  - Can make Payments in COD or Visa.
  - Can change password.
- **System Modules**
  - Admin Module- This module will allow Admin to log in to the system and manage the system and its functions. Admin can View/Confirm/Cancel orders, can manage payments, delivery status, etc.

- User Module– In this module, a user can register themselves first using their name, contact number, address and also, can manage their profile. The user module will allow users to log in to the system using their names & contact number.
- Order Module: In this module after logging in, Users can choose their favorite pizza or customize their pizza and then place an order. Pay for the confirmed order and get the pizza delivered to the registered address. In this module pizza orders, quantity, etc can be managed.
- Category: In this module, pizzas/toppings/crust can be categorized accordingly. Example- Vegetarian, Non-vegetarian, Vegan, Gluten-free, thin crust, thick crust, price range, etc
- Payment Module: In this module, Users can make payments for their orders.
- Availability: In this module, Admin can check whether the particular pizza from a specific category is available in stock or not.
- Sold Number: In this module, Admin can manage the number of pizzas sold.
- Details of User: Details of a User like a Name, Contact Number, Address can be managed by Admin in this module.
- Delivery Staff Module: In this module, details related to delivery staff can be managed.

#### ➤ Group size

- 5 - 6 members.

#### ➤ Deliverables

- A System with GUI that fulfills the above requirements.
- You must apply at least 3 reasonable design patterns.
- Project documentation:
  - The project's name.
  - The team members' info (ID, name and section number).
- A UML class diagram for the project.

#### ➤ [Bonus]

- **Apply** more than 3 patterns.
- **Build** the system as a web application using JSP and servlet.

- **Allow** multiple users to make orders at the same time (parallel processing)
- **Amazing** GUI.

## 11.Banking System

### ➤ Description

- Complete Banking system enables bank users to access their accounts and general information about bank's products and services from the comfort of your PC. This application aims at secure banking system which will be accessible to any authenticated user with valid user id and password. Through this system, user can access his account from anywhere.

### ➤ **Modules**

- Login Module
  - Open account
  - Balance Enquiry
  - Account History
  - Admin Module
  - Loan Module
  - Money Transfer
- **Login Module**
  - Users can log in with user id and password

- **Open account**

This is for new users. They can add an account. For this they need to fill a form and the system provides them with the user id and password to authenticate themselves through the email.

- **Balance Enquiry**

This module facilitates balance enquiry.

- **Account History**

User can check the account history. Account history includes account details, transaction details etc. It simulates mini transaction operation but slightly different as it gives all details.

- **Admin Module**

Complete control over all the actions of the application.

- **Loan Module**

- User can apply for loans through this module. Various loans offered by the bank like education , home, personal loans etc. are included. This section also includes documents needed for loan application, processing status and whether loan is approved or not.

- **Money Transfer**

User can transfer money from his account to other account.

- **Some important requirements**

- The Account class is the base class for all accounts, which has an instance variable balance and methods for getting and setting the balance, depositing, withdrawing, and transferring funds.
- The Account class has can be any of two types, Checking or Savings account.
- The Savings account has an instance variable interestRate.
- A Checking account has a child class MoneyMarket while the Savings account has a child class CertificateDeposit.
- The 4 specific account types (not including the base Account class) vary in the way they handle withdrawals.
- Withdrawals can be done from basic Checking and Savings accounts without penalty. However, withdrawals from MoneyMarket and CertificateDeposit accounts involve a cost.
- For MoneyMarket accounts, each withdrawal costs \$1.50.
- For CertificateDeposit accounts, there is a severe penalty for withdrawal: 20% of the amount withdrawn.

#### ➤ Group size

- 5 - 6 members.

#### ➤ Deliverables

- A System with GUI that fulfills the above requirements.
- You must apply at least 3 reasonable design patterns.
- Project documentation:
  - The project's name.
  - The team members' info (ID, name and section number).
- A UML class diagram for the project.

➤ [Bonus]

- Apply more than 3 patterns.
- Build the system as an android application.
- **Amazing** GUI.



## 12.Doctor Finder System

### ➤ Description

- This project is designed to automate the process of searching for the right doctors and booking an appointment. It will help in managing doctors, patients, appointments, etc...
- There could be hundreds of doctors available in your city, but not sure which one to go and consult with. Having a system to find the doctors for your concerns can help eliminate the confusion and pain of visiting every other hospital, saving you some time without affecting your health.
- With the help of the Doctor Finder system, Users will be able to find the doctor in their nearby area with complete details of each doctor. Once the appointment is confirmed, the User can visit the doctor and make payments accordingly.
- It is a multi-role application project i.e., single Admin and multiple User, where Admin will have the main control over the system.
- The system has two main roles i.e. Admin and User. Admin being the primary user. Admin can add/remove/update any details related to the system, update fees of the doctors & their appointments, treatments, and other services, View/Confirm/Cancel appointment booking done by Users, and so on. Admin can manage doctors' details and also, keep the records of the Users, etc.
- Admin will be responsible for keeping a check on the booking of appointments, scheduling, and management of the system. On the other hand, Users can register themselves, log in using the login credentials, search for doctors for their specific concerns, fix an appointment, and make payments online after successful consultation with the doctor.
- Admin is responsible for keeping track of doctors' accounts, availability of doctors, viewing/confirming/canceling an appointment for the patients, patient records, and maintaining a database.
- Admin can Add/Remove/Update details of doctors from the doctor's list. Admin can notify users of doctor's availability. Admin can also View/Confirm/Cancel rescheduled appointments for Users.

- Users can register themselves and then fix the appointment with the doctor they choose. Users can find detailed information about the doctors & fees on the system after logging in.

## ➤ **System Modules**

**Admin Module:** This module will allow Admin to log in to the system and manage the system and its functions. Admin can View/Confirm/Cancel appointments, manage doctors, and can check payments, etc. Admin is responsible for keeping track of doctors, and patient records, and maintaining a database.

**User Module:** In this module, the User can register themselves. Once registered, the user can log in/log out of the system. Users can manage their profiles.

**Doctor Module:** In this module, doctors can be categorized by Admin according to their specialty. Admin can manage details of all the doctors and their fees, treatments, and courses available.

**Find Doctor Module:** Users can find doctors according to their concerns in this module. Detailed information about each and every doctor will be available in this module.

**Appointment Module:** Users can book an appointment with a doctor with the help of this module. In this module, Users can clearly check the availability of doctors and make an appointment accordingly.

**Payment Module:** Once the User has consulted with the doctor, the User can make payments online in this module.

**Details of Users:** In this module, Admin will be able to keep the records of Users.

## **Users Roles**

Two users can interact with this application **1) Admin 2) User**

**Admin:**

- Can Log in/Log out of the system.

- Admin can manage doctors in the system.
- Admin can View/Confirm/Cancel appointments done by the User.
- Can check payments done by User.
- Can manage and check the availability of doctors for users.
- Can change consultation fees and other fees of doctors
- Can change password.
- Can manage “My Profiles”.

#### User:

- Can log in/log out of the system.
- Can Manage “My profile”.
- Can search for doctors.
- Users can book/cancel an appointment.
- Can make payments online.
- Can change password.

#### ➤ Group size

- 5 - 6 members.

#### ➤ Deliverables

- A System with GUI that fulfills the above requirements.
- You must apply at least 3 reasonable design patterns.
- Project documentation:
  - The project’s name.
  - The team members’ info (ID, name and section number).
- A UML class diagram for the project.

➤ [Bonus]

- Apply more than 3 patterns.
- Build the system as a web application using JSP, servlet and MYSQL.
- **Amazing** GUI.

## 13.Exams Management System

### ➤ Description

- Exams management system is a desktop application that allows teachers to do their work easily. It will be used for adding, deleting or editing exams, checking students' grades, and issuing reports for students that contain the exams they took, dates, and grades, assigning assignments to classes ...
- School has a principal, teachers, students and employees. Only principle, teachers and students have accessibility to this application. So, each one of them has a specific functionality and limited accessibility.
- School has a number of students in different grades (first, second, third...).
- **Principle functionalities:**
  - Sign in with his username and password.
  - Add new teacher, edit, and delete them.
    - A teacher has personal details, salary, and the subject he/she is teaching.
  - View already exist teachers' details.
  - View details of all students.
  - Get information about the conducted exams.
  - Prepare the time table or the schedule of classes.
- **Teacher functionalities:**
  - Add, edit, or delete exams:
    - An exam can be weekly, monthly or final
  - Checking students' grades, and issuing reports for students that contains the exams they took, dates, and grades.
  - Assigning assignments to classes.
  - Sending notifications to students telling them that their grades have been published
    - A notification can be an email, SMS, WhatsApp message (one or even a combination of any of them).
- **Student functionalities:**
  - Login with user name and password that were sent to them by the principal.
  - taking their exams according to their year grade.
  - See their grades when they are published.
  - Write a note to the principle.

### ➤ Group size

- 5 - 6 members.

### ➤ Deliverables

- A System with GUI that fulfills the above requirements.
- You must apply at least 3 reasonable design patterns.
- Project documentation:
  - The project's name.
  - The team members' info (ID, name and section number).
- A UML class diagram for the project.

### ➤ [Bonus]

- Apply more than 3 patterns.
- Build the system as a web application using JSP and servlet. Allow the students to take the exam in a limited item.
- **Amazing GUI.**

## 14. Hotel Management System

### ➤ Description

- Hotel management system is a desktop application that allows hotel workers to do their work easily. It will be used for booking rooms, checking in and out, keeping up with rooms statuses, ...
- Hotel has number of workers, like manager, receptionist, room service. Only manager and receptionist have accessibility to this application. So, each one of them has a specific functionality and limited accessibility.
- Hotel has number of rooms with different types (single, double, and triple) and different prices.
- Hotel Boarding Types are Full Board, Half Board, and Bed and Breakfast. Each boarding type has its price.
- **Manager functionalities:**
  - Sign in with his username and password.
  - Add new worker, edit, and delete them.
    - A worker has personal details, salary, and job title.
  - View workers' details.
  - View details of all residents.
  - Get information about the income (weekly, monthly, or annually)
  - Follow up with the status of rooms (available rooms and busy rooms).
    - This information will be displayed to the manager based on the type of the room.
- **Receptionist functionalities:**
  - Add, edit, and delete residents' information where:
    - Resident has his personal information.
    - Resident has information about the duration he will be spent in the hotel.
    - Resident has information about his room.
    - Also, he has information about the type of services that he gets.
  - Resident has the cost that he should pay when he will check out. This cost will be calculated based on the type of room that he chooses, number of nights that he will spend at the hotel and boarding type.
  - Assign the resident to a room based on availability of the rooms.

### ➤ Group size

- 5 - 6 members.

### ➤ Deliverables

- A System with GUI that fulfills the above requirements.
- You must apply at least 3 reasonable design patterns.

- Project documentation:
  - The project's name.
  - The team members' info (ID, name and section number).
- A UML class diagram for the project.

➤ **[Bonus]**

- Apply more than 3 patterns.
- Build the system as a web application using JSP and servlet. Allow the customers to book the rooms.
- **Amazing** GUI.



## 15. Mini Uber Application

### ➤ Description

- Mini Uber is an application that is used for booking cars by customers, providing delivery services by drivers, and managing the services by the system company's employees. The designed system consists of modules:
- **Customer Module:** This module gives the ability for customer to
  - Register to the system
  - Login
  - Book a car by doing the following:
    - Write the pickup point
    - Write the destination place.
    - Write the time of the trip. (The time when the customer wants the car to come).
    - Confirm the trip information.
  - Update his/ her personal info.
  - View his/her previous trips' details.
  - rate the driver.
  - Send complaints through the system. Employees can view customers' complaints.
- **Driver module:** This module gives the ability for a driver to
  - Login
  - Search available trips.
  - Accept a trip.
  - View his/her previous trips details.
  - Every driver has a car. If the driver doesn't have a car employees assign car to him.
- **Employee module:** This module gives the ability for an employee to
  - Login
  - Register new driver in the system
  - Add cars to the systems.
  - Assign one car to every drive if the driver doesn't have a car.
  - Change the assigned a car to the driver with another car in the system.
  - View Customers complains.
- **Trip module:** Every trip has
  - Customer.
  - Driver
  - Pickup point.
  - Destination.
  - Car's fare.
  - Trip time. (The time when the customer wants the car to come).

### ➤ Group size

- 5 - 6 members.

### ➤ Deliverables

- A System with GUI that fulfills the above requirements.
- You must apply at least 3 reasonable design patterns.
- Project documentation:
  - The project name.
  - The team members' info (ID, name, and section number).
- A UML class diagram for the project.

### ➤ [Bonus]

- Apply more than 3 patterns.
- Build the system as android mobile application.
- **Amazing** GUI.

## 16.Online Shopping System

### ➤ Description

- The online shopping system is a system that allows buyers to buy goods/products directly from the seller and it saves time and money of buyer. The system simulates the full cycle of the shopping . It begins from customer/user entrance the to the system and if it's his/her first time of using this system then he/she should fill his/her personal info, the SSN and the credit card number. Then he/she browses the various categories and adds the interesting items (clothes, appliance, accessories, ...) to his/her Cart and before confirming this request the system shall check the stock if these items are available for now or not. If it's available, then calculate the total amount price and if there is a sale for some items the system must apply it.
- The system also should calculate the estimated time to deliver these items to this customer/user.
- System shall provide option to the customer/user if he/she like to cancel the order or some items from the order within some period.
- Finally, the system should provide option for the customer/user to leave a feedback about some product.
- **Admin/System**
  - Add new category
  - Edit category
  - Delete category
  - Add new items. Each item has details and some images.
  - Check items in stock
  - Add sale for items
  - Calculate total receipt
  - Delete user
  - Check credit card
  - Show all orders in period time
- **User/Customer**
  - Register or sign in
  - Browse all categories and items.
  - Add items to cart
  - Cancel order.
  - Cancel items
  - Update personal info
  - Add feedback to items

### ➤ Group size

- 5 - 6 members.

### ➤ Deliverables

- A System with GUI that fulfills the above requirements.
- You must apply at least 3 reasonable design patterns.
- Project documentation:
  - The project name.
  - The team members' info (ID, name, and section number).
- A UML class diagram for the project.

### ➤ [Bonus]

- Apply more than 3 patterns.
- Build the system as a web application using JSP and servlet.
- **Amazing** GUI.

## 17.Travel Management System

### ➤ Description

- The travel management system enables employees in any travel agency to manage trips and tours which are organized by the company. Also, clients can access the system to browse, and book a trip.
- There are different categories of trips, like: safari holidays, adventure, cultural, religious tours, ...
- Each Tour trip consists of:
  - Trip date.
  - Trip category.
  - Trip name.
  - Tour guide.
  - Number of tourists who can join this tour.
  - Total number of tourists who joined this trip and their personal information.
  - The details of the sites that the tourists will visit.
  - Price of the trip
  - The total profit which came out of this trip.
- Each tour guide has: ID, name, age, salary, trips to that he was assigned to before.
- Each client has his personal information plus information about trips he has.
- Client's trips can be divided into previous trips, current trip (if he is in a trip now), and coming trips (trips that doesn't start yet).
- This application should enable the employee to:
  - Add, edit, or cancel any tours or trips organized by the company.
  - Add, edit, delete, and view tour guide information.
  - Assign each tour guide to specified tour based on his availability.
  - View details of clients.
  - Calculate the total income for each category of trips separately.
  - Finally, he can get the total income of all trip categories.
  - View Clients complains.
- The client can do the following:
  - Register/ login to the system.
  - Browse trips and tours in different categories.
  - book different trips.
  - Cancel his/her registration.
  - Edit his/her account.
  - View his/her previous trip.
  - View the details of his/her upcoming trips.
  - Make a complain.

### ➤ Group size

- 5 - 6 members.

### ➤ Deliverables

- A System with GUI that fulfills the above requirements.
- You must apply at least 3 reasonable design patterns.
- Project documentation:
  - The project name.
  - The team members' info (ID, name, and section number).
- A UML class diagram for the project.

### ➤ [Bonus]

- Apply more than 3 patterns.
- Build the system as a web application using JSP and servlet.
- **Amazing** GUI.

## 18.Vehicles management system

### ➤ Description

- This system should keep track of the cars, buses and bikes of a specific company and their possible components (brand - model- engine- doors - glass - color- ...) and the possible engines that can replace the old engine (as each car should have a range of the possible engines' power that can be added to it)
- Also, the system allows admin to add, delete or update vehicle objects, and allow users to search with certain specifications and gives them a list with the possible car list that matches what their preferences.
- The system should be used in UK and USA.
- **admin functionalities:**
  - Sign in with his username and password.
  - Add new vehicle, edit, and delete them.
    - A vehicle has its specific details, price.
  - View vehicles' details.
  - View details of possible engines that can replace the old engine of specific vehicle.
  - Filter the vehicles by categories
    - A category for cars, bikes and buses
  - Follow up with the status of the vehicles (available or sold out).
    - This information will be displayed to the admin based on the type of the vehicle.
- **User functionalities:**
  - Search with certain specifications:
    - A user can have an account or not.
    - If the user has an account, save his personal preferences.
    - Specifications can be: brand - model- engine...etc.
    - After the user choose a car, he can customize it as per his preferences.
  - A user can see a dash board contains a vehicle list according to his/her preferences
  - The user can see the speed of the car as per his location.
    - In UK → MPH
    - In USA → KMPH

### ➤ Group size

- 5 - 6 members.

### ➤ Deliverables

- A System with GUI that fulfills the above requirements.

- You must apply at least 3 reasonable design patterns.
- Project documentation:
  - The project's name.
  - The team members' info (ID, name and section number).
- A UML class diagram for the project.

➤ **[Bonus]**

- Apply more than 3 patterns.
- Build the system as a web application using JSP and servlet then allow the users to buy or rent cars through the system.
- **Amazing** GUI.