

Parking System Using Java

Project Synopsis Presentation

Date: 30/09/2023

Mini Project (ICI552)

Degree : B.C.A.

Project Guide:
Mr. Shashank Saxena

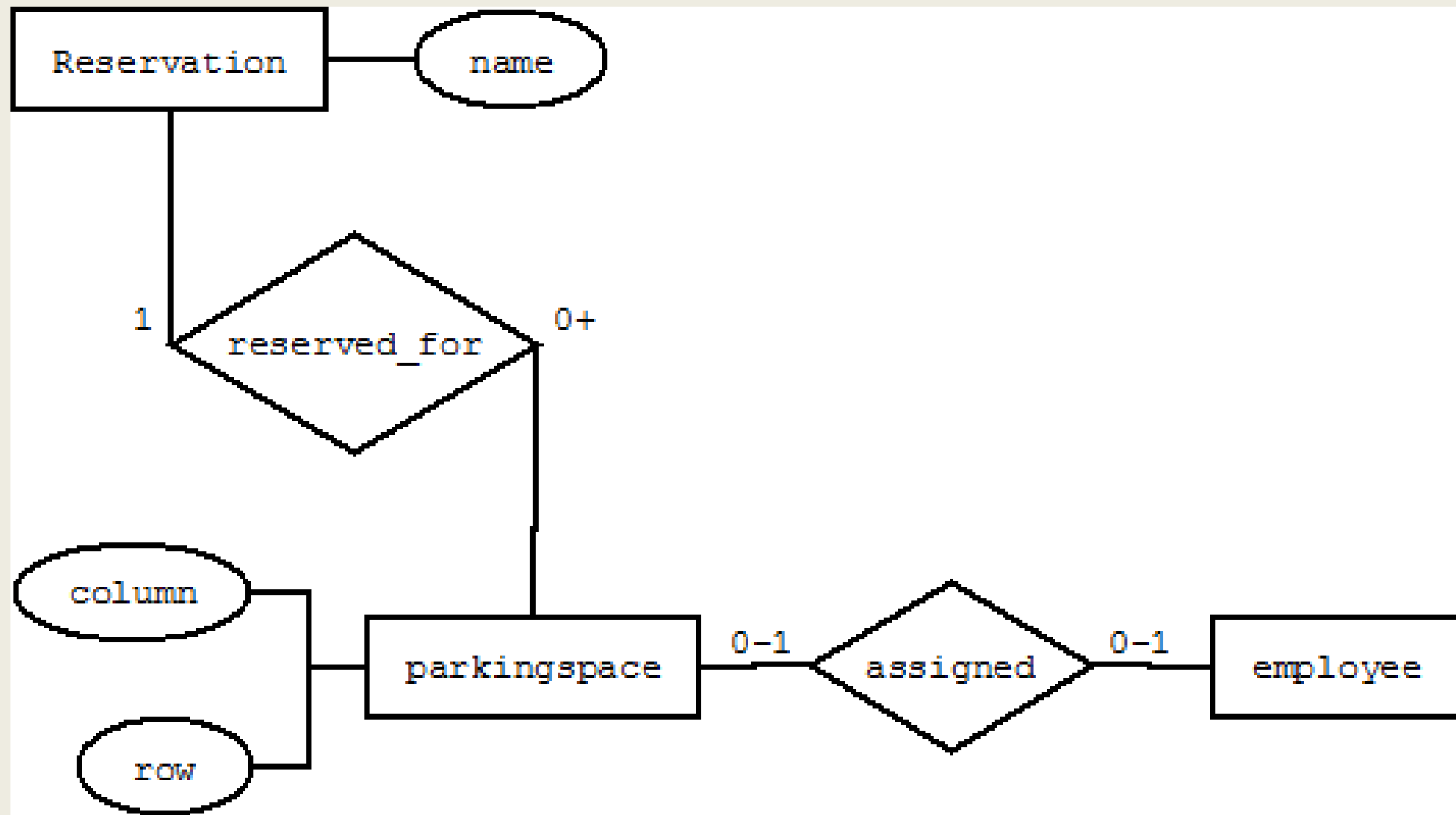
Sangeet Guatam (TCA2156024)
Sagar Rai (TCA2156023)



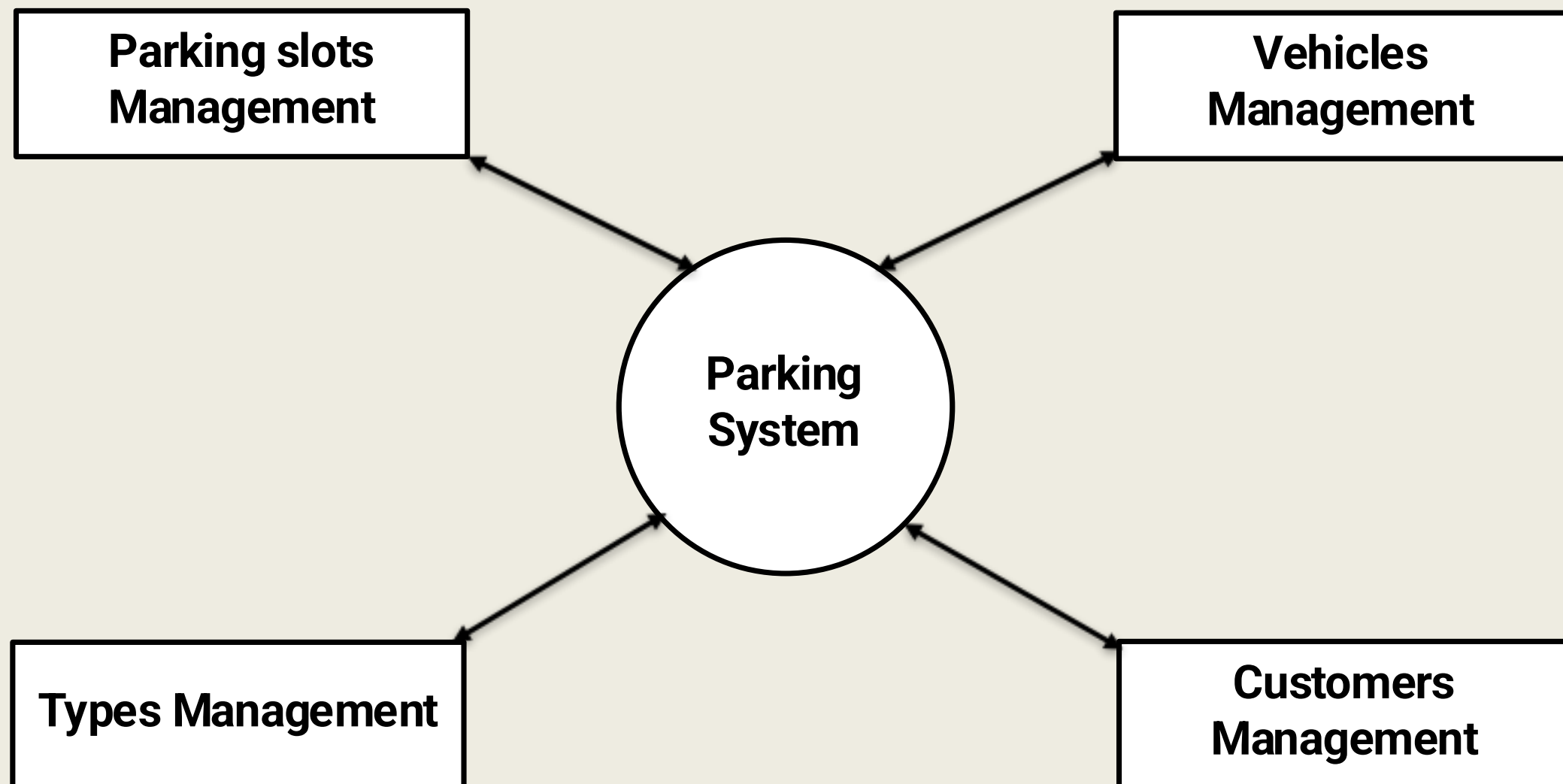
**FACULTY OF ENGINEERING & COMPUTING SCIENCES
TEERTHANKER MAHAVEER UNIVERSITY, MORADABAD**

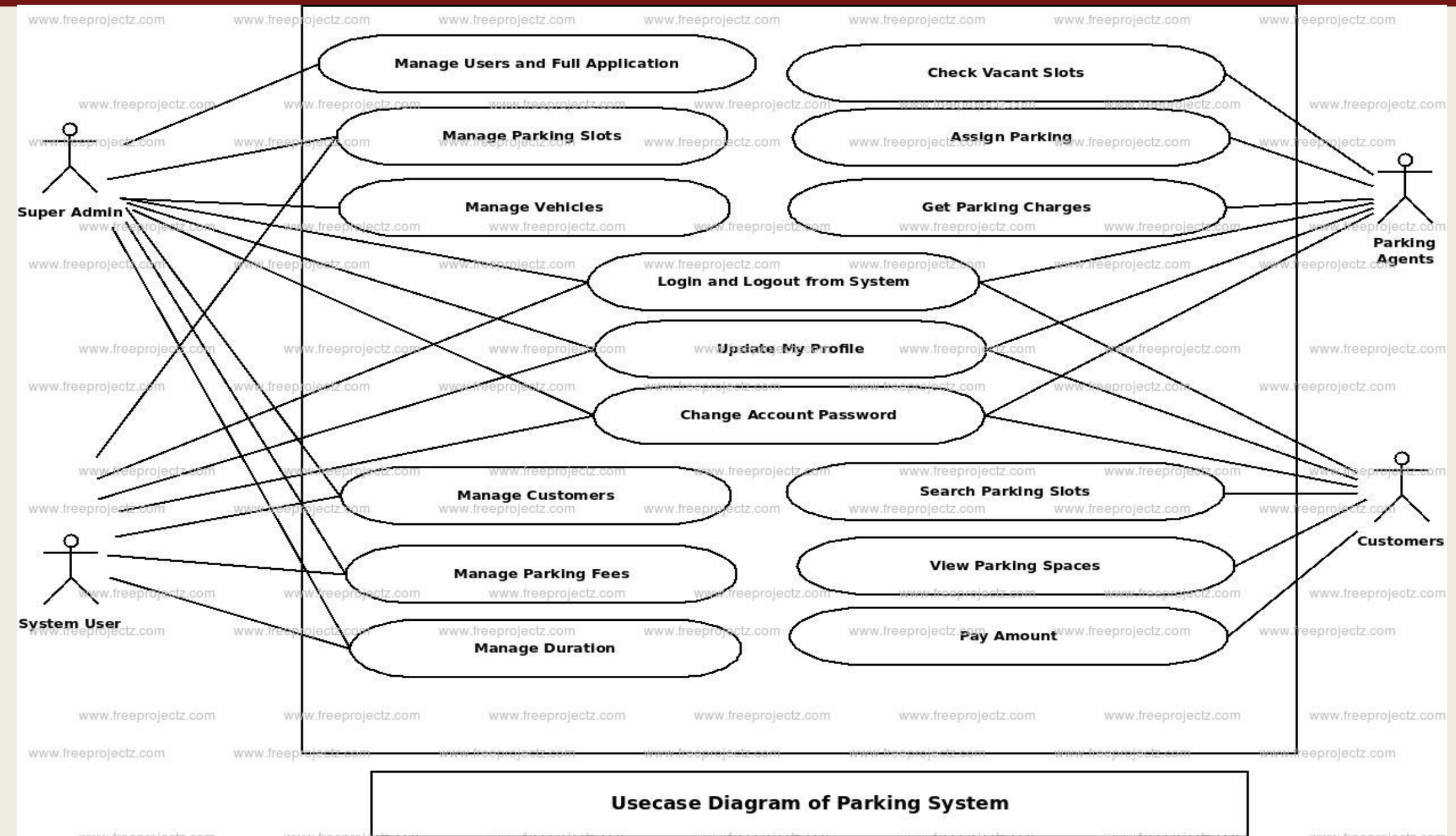
Student Name	Role
Sangeet Gautam	Developer, Testing, Researcher
Sagar Rai	Research Paper

- The Parking System we will build will be a simple console-based application that allows users to park their cars, removed their cars and view all parked cars. It presents a menu with four options: park a car, remove a car, view parked cars, and exit the program.
- It is very simple with coding point of view as it has only one java class in it. We need to import Scanner class and Array List class. Scanner class used to take input through the keyboard from user while Array List is used to make things simple for adding the cars and deleting them.
- In remove car method, we will check for number of parked cars if it appears as zero then process will not go further. By entering license number of parked car one can simply removed it. Inbuilt functions of Array List 'add' and 'remove' are used to add cars in array list and remove it from the array list, respectively.



- ZERO LEVEL DFD – PARKING SYSTEM





Coding

- `import java.util.ArrayList;`
- `import java.util.Scanner;`
- `public class ParkingSystem {`
- `static int totalSlots, availableSlots;`
- `static ArrayList<String> parkedCars = new ArrayList<String>();`
- `public static void main(String[] args) {`
- `Scanner sc = new Scanner(System.in);`
- `System.out.println("Enter the total number of parking slots:");`
- `totalSlots = sc.nextInt();`
- `availableSlots = totalSlots;`
- `while (true) {`
- `System.out.println("\nWhat would you like to do?");`
- `System.out.println("1. Park a car");`
- `System.out.println("2. Remove a car");`
- `System.out.println("3. View parked cars");`
- `System.out.println("4. Exit");`
- `int choice = sc.nextInt();`

Coding

```
• switch (choice) {  
•     case 1:  
•         parkCar()  
• parkCar();  
•                                     break;  
•  
•     case 2:  
•         removeCar();  
•         break;  
•     case 3:  
•         viewParkedCars();  
•         break;  
•     case 4:  
•         System.exit(0);  
•     default:  
•         System.out.println("Invalid choice. Please try again.");  
• }  
• }  
• }  
  
• public static void parkCar() {  
•     if (availableSlots == 0) {  
•         System.out.println("Sorry, there are no available parking slots.");  
•         return;  
•     }  
  
•     Scanner sc = new Scanner(System.in);  
•     System.out.println("Enter the license plate number of the car:");  
•     String licensePlate = sc.nextLine();  
•     parkedCars.add(licensePlate);
```


Coding

- availableSlots--;
- System.out.println("Car parked successfully. Available slots: " + availableSlots);
- }
- public static void removeCar() {
- if (availableSlots == totalSlots) {
- System.out.println("There are no parked cars.");
- return;
- }
- Scanner sc = new Scanner(System.in);
- System.out.println("Enter the license plate number of the car to be removed:");
- String licensePlate = sc.nextLine();
- if (parkedCars.contains(licensePlate)) {
- parkedCars.remove(licensePlate);
- availableSlots++;
- System.out.println("Car removed successfully. Available slots: " + availableSlots);
- } else {
- System.out.println("The car is not parked here.");
- }
- }
- public static void viewParkedCars() {
- if (availableSlots == totalSlots) {
- System.out.println("There are no parked cars.");

Coding

- `return;`
- `}`
- `System.out.println("Parked cars:");`
- `for (String licensePlate : parkedCars) {`
- `System.out.println(licensePlate);`
- `}`
- `}`
- `}`

Output

```
Enter the total number of parking slots:12
What would you like to do?
1. Park a car
2. Remove a car
3. View parked cars
4. Exit
1
Enter the license plate number of the car:1210
Car parked successfully. Available slots: 11What would you like to do?1.
    Park a car2. Remove a car3. View parked cars4. Exit|
```

Tools / Technologies Used

SOFTWARE

- Operating system : Windows XP /7/8/10
- Coding Language : Java
- Technology Tools : Java Database Connectivity, IntelliJ IDEA
- Development Kit : Java Development Kit
- IDE : Visual studio Code(VS Code)

HARDWARE

- System : PentiumIV 2.4 Ghz.
- Hard Disk : 500 GB
- RAM : 4GB
- Any desktop/ Laptop system with higher level configuration

- Parking system management utilizes technologies integrated with methodologies like data analytics and optimize space and monitor parking facilities.
- Parking system management uses like RFID, and software to automate vehicle entry, exit, and payment, improving efficiency, security, and user experience in parking facilities.
- Parking system management software for reservations, payment processing, and occupancy tracking, along with hardware like ticket dispensers, barriers, facilitating efficient parking operations

- This system allows you to make your task to park the cars easy. Java is a popular programming language that is widely used for developing parking management systems due to its simplicity, versatility, and platform independence.
- This system analyses for an available space in parking area and on the basis of it gives the output to user. If all slots are full then it will simply tell the user to not park car in that particular area.
- It increments the available slots if any user removed his/her car from parking area and decrements slots if any user park his/her car. Overall, a parking system in Java provides a flexible and powerful platform for managing parking spaces, improving traffic flow, and enhancing the overall parking experience for drivers. By leveraging the strengths of Java, developers can create robust and scalable parking systems that meet the unique needs of their users.
-

- Academic Journals and Papers: Look for research papers on parking management systems in databases like IEEE Xplore, Google Scholar, or ACM Digital Library Books "Smart Parking: The Future of Parking" by Smart Parking Ltd. "Parking Management for Smart Growth" by Richard W. Willson.
- Websites and Online Resources: Check websites of companies specializing in parking management solutions, such as ParkMobile, ParkWhiz, and T2 Systems. Explore government websites, especially those of cities and municipalities, for information on local parking management policies.
- Industry Reports and Whitepapers: Reports from market research firms like IBISWorld, Statista, and MarketResearch.com can provide insights into industry trends.
- Case Studies: Look for case studies from companies that have implemented parking management systems, as they can provide valuable real-world examples.
- Forums and Communities: Participate in online forums, such as LinkedIn groups or Reddit communities related to parking management, to connect with professionals and gain insights.
- Interviews and Surveys: Conduct interviews or surveys with parking facility managers or users to gather firsthand information and feedback.
- Government Regulations and Standards: Research local and national regulations and standards related to parking management and access control systems.
- Academic Institutions: Contact universities or research institutions that specialize in transportation or urban planning for guidance and potential collaboration.
- Technical Documentation: If you're implementing a specific technology or software, refer to their official documentation for technical details and best practices.

THANKS