**Office Seat Booking Application – Capstone**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-Requirements**

1. Problem Statement
2. Objects Identification
3. Relation between Objects
4. Class Diagram
5. **Problem Statement**

* To book a seat by an employee
* Two slots available (Slot I: 8:30 AM-5:30 PM & Slot II: 11:00 AM- 8:30 PM)
* Two slots for each location: Gurgaon, Noida, Bangalore, Hyderabad
* 4 sites for each location

1. **Requirements**

* Registration Form (For new employee)
* Login Form (Existing employee)
* Searching availability of sites based on slot, date and location
* Booking window: availability only for next 7 days after the current date
* Seat unavailable for a particular location in both slots-provide another location and the slot accordingly.
* Mail/Message for booking confirmation
* Cancellation available: prior to 12 hours of booking date.

1. **Features of the Application**

* Client-Server Architecture
* Registration Form consisting of: Employee ID, Phone Number, Email ID and Password
* Password minimum of 8 characters
* Login Form for registered employees- Employee ID and Password
* Token based security approach implemented to secure API end point which would expire after 20 minutes and re-login/authorization would be required
* List of only available sites w.r.t selected location/date/slots to be populated for booking by the user
* Successful Mail/Message for booking confirmation
* On User’s Profile page, his information along with booking and cancellation options should be displayed.

1. **Use Case Diagram**

**Diagram

Description automatically generated**

1. **Activity Diagram**

**Graphical user interface, text, application

Description automatically generated**

1. **Entities Identification**

* Employee, System (Actors)
* Seat
* Location
* Booking

1. **Relations between Entities**

* Employee can book one seat (to one)
* Employee can book at one location (to one)
* For one date, employee can have one booking (to one)
* System(Booking) can have many Employees (to many)
* System can have multiple Bookings

1. **Class Diagram**

**Diagram

Description automatically generated**