**NAME ANSHU GUPTA**

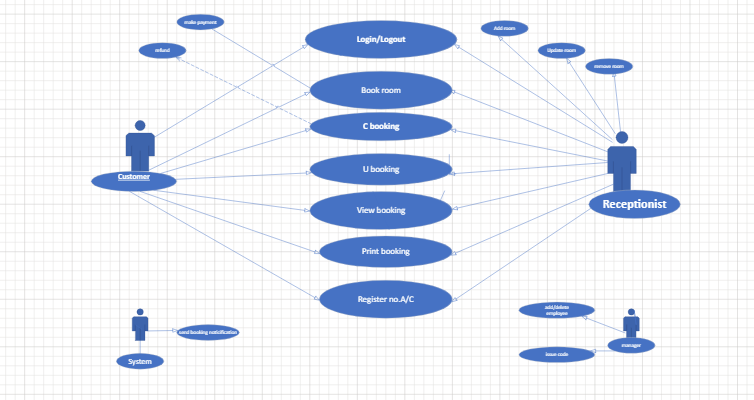
**ASSIGMENT 4**

**USE CASE DIAGRAM**

**BATCH A**

**COURSE PG-DAC**

Q1. Implement hotel management system through Use case Diagram.

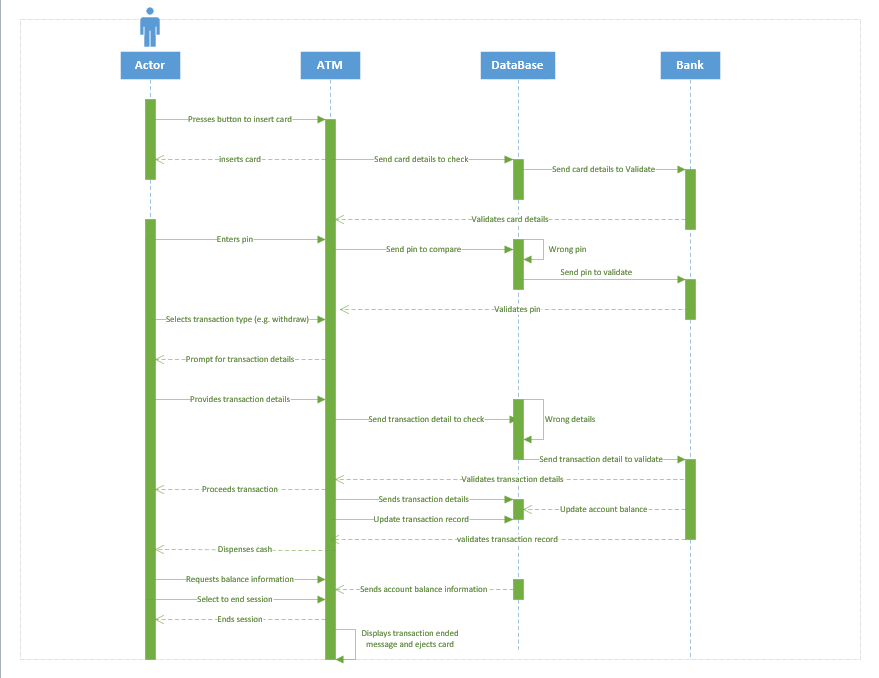


Hotel management system

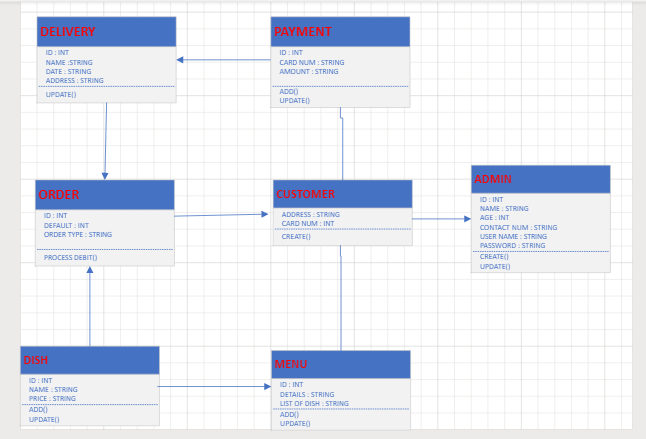
**Q2. Design the Sequence diagram for ATM System taking into consideration different**

**scenarios.**

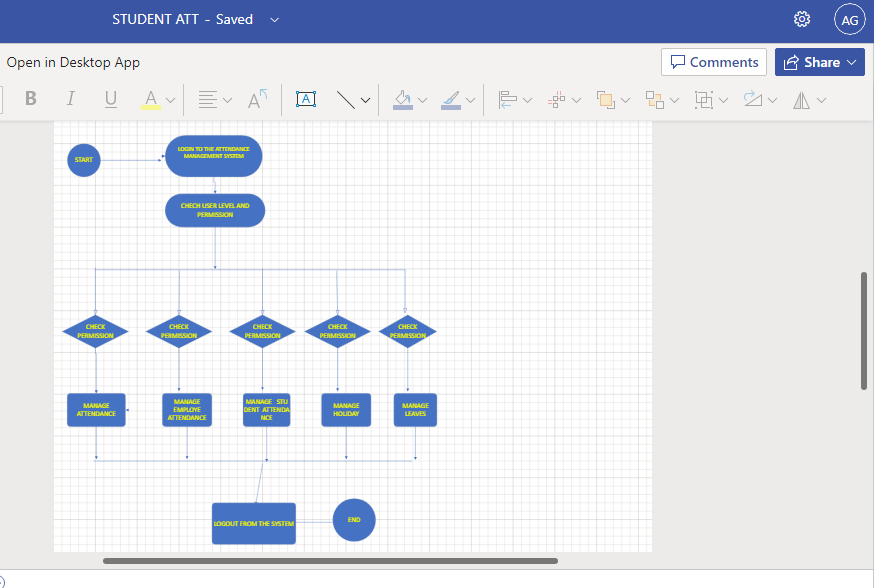
**The Sequence diagram for ATM System**



Q3. Design the class diagram for online food ordering system.

****

Online food ordering system

Q4. Design the activity diagram for student Attendance management system. 

**Q5. Which of the following design principle(s) have been violated in the following scenarios?**

**1. Abstraction**

**2. Decomposition and Modularization**

**3. Coupling &amp; Cohesion**

**4. Encapsulation**

**5. Sufficiency, Completeness and Primitiveness**

**6. All**

|  |  |  |
| --- | --- | --- |
| **No.** | **Description** | **Principle Being**  **Violated** |
| 1 | Important information of a module is directly accessible by other modules | Encapsulation |
| 2 | Too many global variables in the program after implementing the design | Decomposition and Modularization |
| 3 | Code breaks in unexpected places | Sufficiency, Completeness and Primitiveness |
| 4 | Unfulfilled requirements in the code after the design has been implemented | Abstraction |
| 5 | Cyclic dependency among classes | Coupling & Cohesion |
| 6 | Huge class doing too many unrelated operations | Cohesion |
| 7 | Several unrelated functionalities/tasks are carried out by single module | Decomposition and Modularization |
| 8 | All data of all classes in public | Encapsulation |
| 9 | Design resulting in spaghetti code | Decomposition and Modularization |
| 10 | An algorithm documented as part of design is not understandable by the programmers | Sufficiency, Completeness, and Primitiveness |