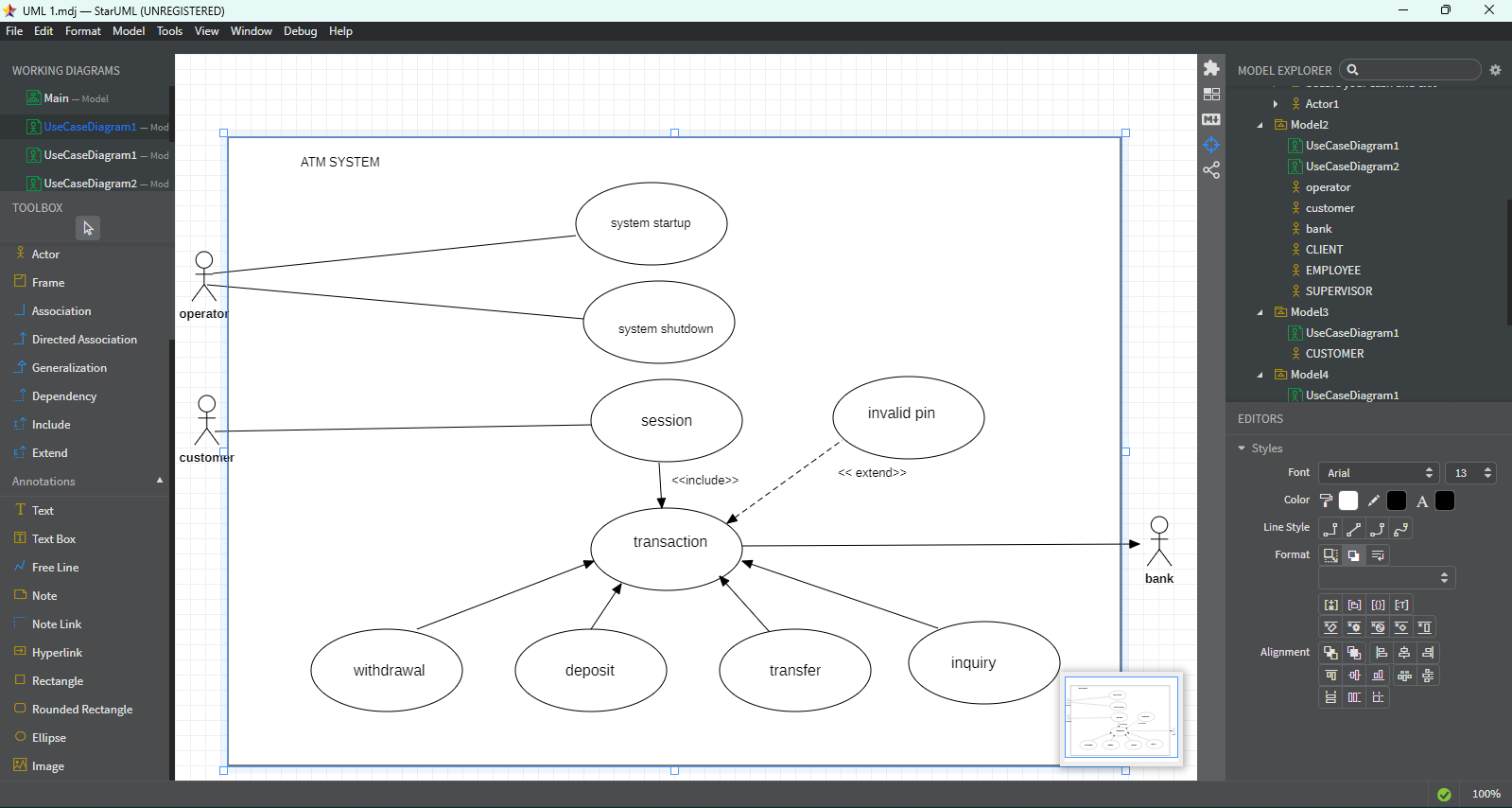
**SOFTWARE ENGINEERING LAB TASK 9**

**UML (Use Case Diagrams)**

**HU22CSEN0100999**

**Eshwar Deshmukh Chavan**

**ATM UML:**



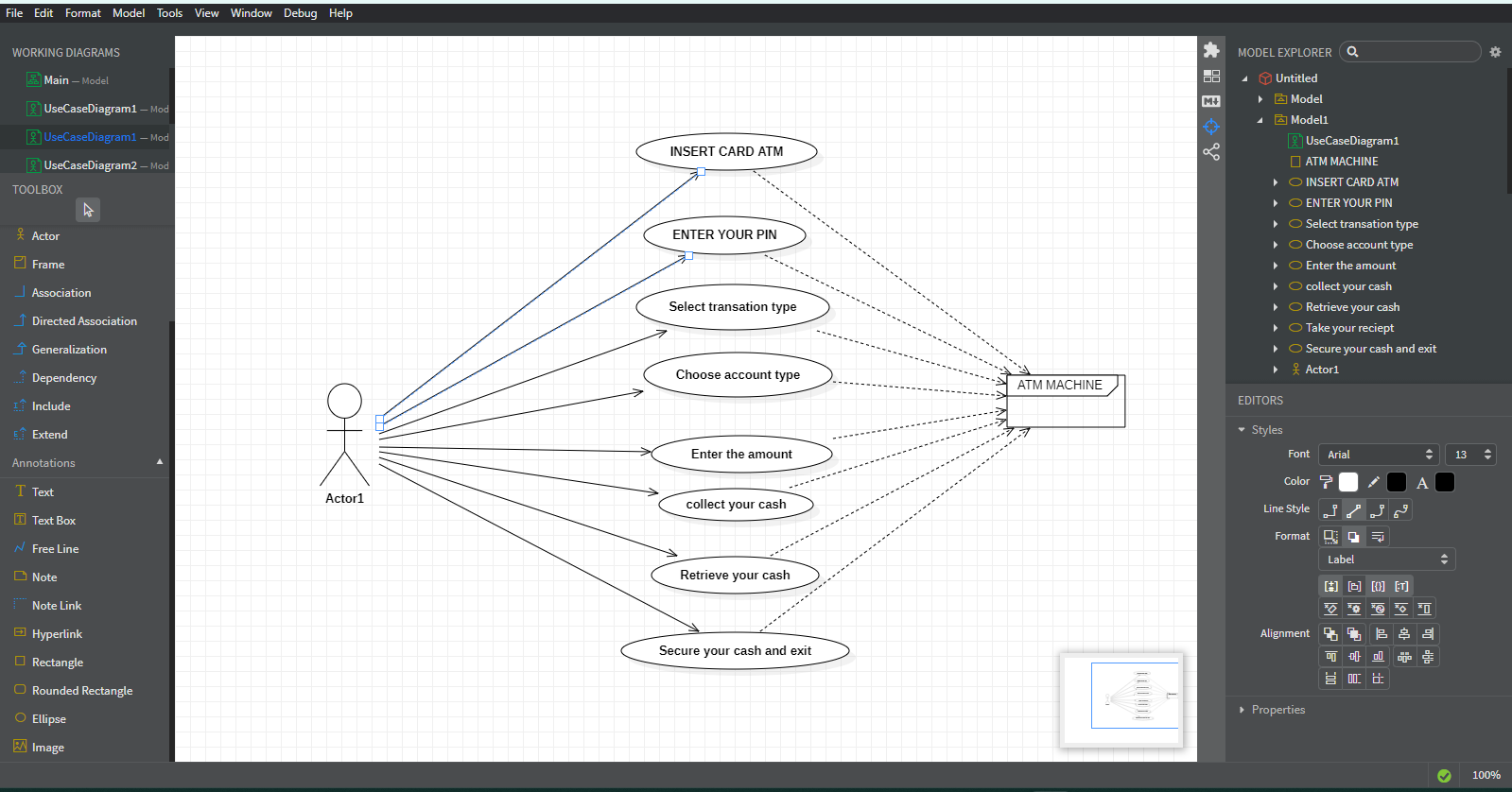
**Diagram Details:**

* **Actors**:
  + **Operator**: Responsible for system maintenance (startup, shutdown).
  + **Customer**: The user performing transactions.
* **System Boundary**: “ATM System.”
* **Use Cases**:
  + **System Startup** (Operator)
  + **System Shutdown** (Operator)
  + **Session** (Customer)
  + **Invalid PIN** (Customer)
  + **Transaction** (Customer)
    - **Withdrawal**
    - **Deposit**
    - **Transfer**
    - **Inquiry**
* **Bank** (external entity) is also shown, indicating communication with the ATM for transactions.

**Explanation**

1. **Operator** handles high-level operations:
   * **System Startup**: Initializing the ATM before customers can use it.
   * **System Shutdown**: Turning off the ATM or bringing it offline for maintenance.
2. **Customer** interactions revolve around:
   * **Session**: Logging in by inserting card and entering PIN.
   * **Invalid PIN**: An extension or alternative flow if the PIN is entered incorrectly.
   * **Transaction**: The main use case for financial operations. It includes:
     + **Withdrawal**: Taking out cash.
     + **Deposit**: Putting money into an account.
     + **Transfer**: Moving funds between accounts.
     + **Inquiry**: Checking balance or transaction history.
3. The **ATM** communicates with the **Bank** system to validate transactions and ensure real-time account updates.

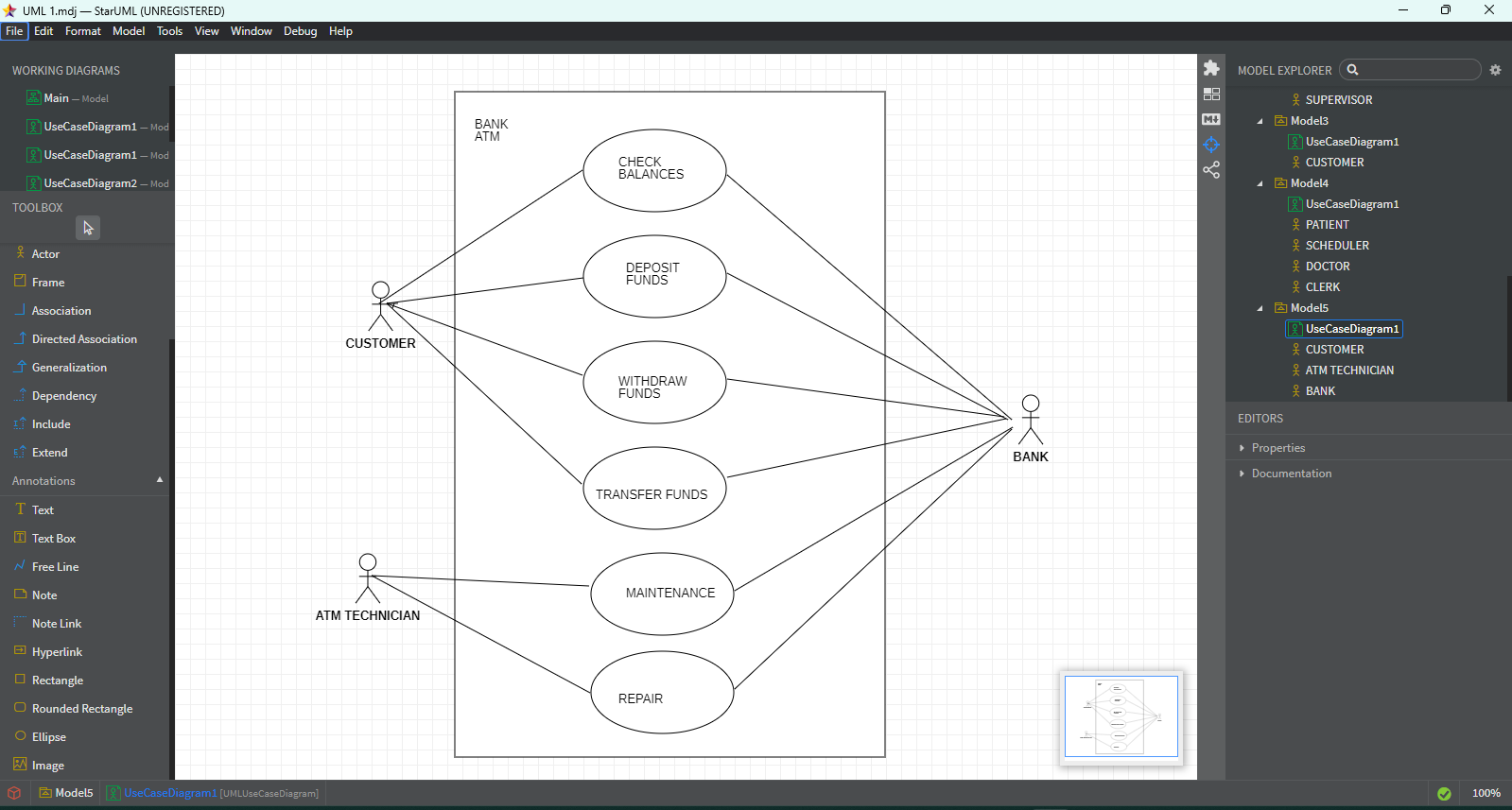
**Another UML of ATM:**



**Diagram Details:**

* **Actor**: A single user (the person using the ATM).
* **System Boundary**: Represented by the “ATM MACHINE.”
* **Use Cases** (the ovals):
  1. **Insert Card ATM**
  2. **Enter Your PIN**
  3. **Select Transaction Type**
  4. **Choose Account Type**
  5. **Enter the Amount**
  6. **Collect Your Cash**
  7. **Retrieve Your Cash**
  8. **Secure Your Cash and Exit**

**BANK ATM UML:**



**Diagram Details:**

**Actors:**

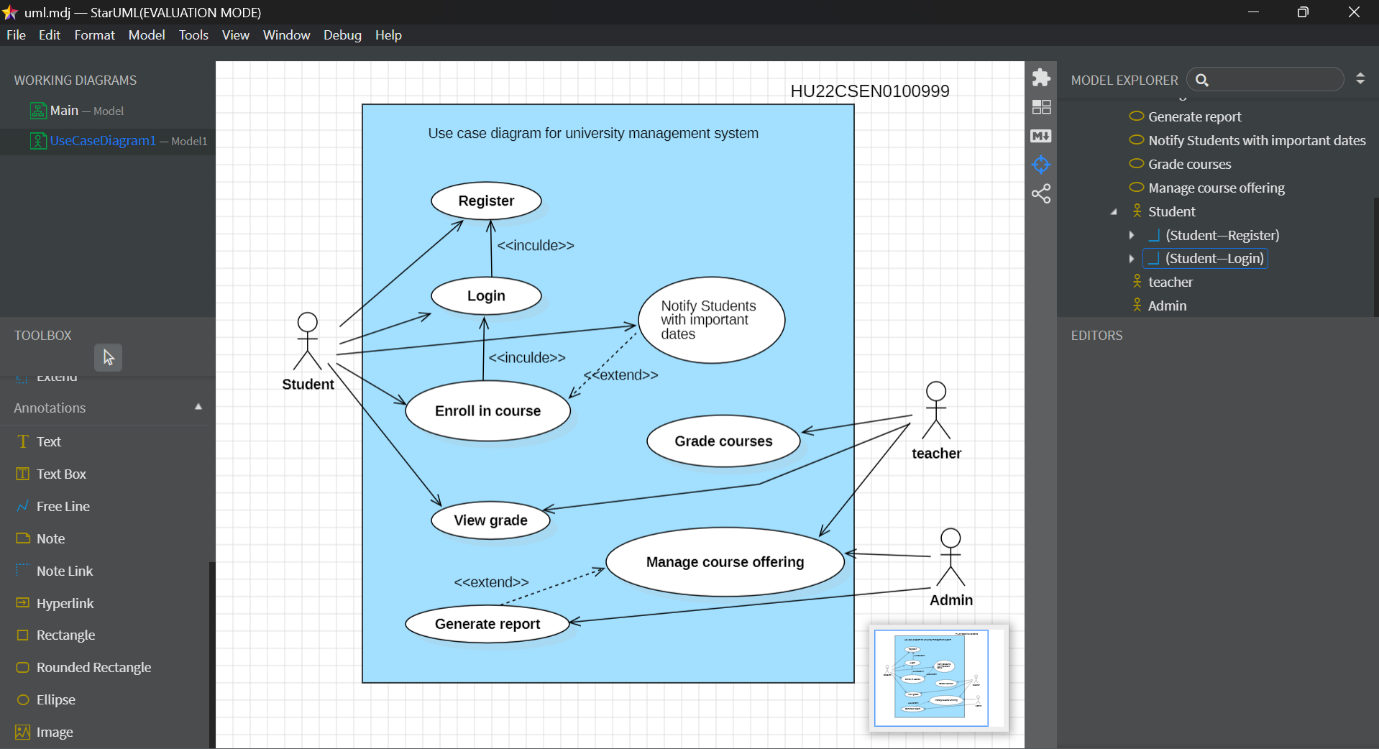
* **Customer** (on the left): checking balances, depositing, withdrawing, transferring funds
* **ATM Technician** (on the lower left): maintenance, repair
* **Bank** (on the right): validating and processing transactions

**System Boundary:** “Bank ATM”

**Use Cases** inside the system:

* **Check Balances**
* **Deposit Funds**
* **Withdraw Funds**
* **Transfer Funds**
* **Maintenance**
* **Repair**

**UNIVERSITY MANAGEMENT SYSTEM UML**



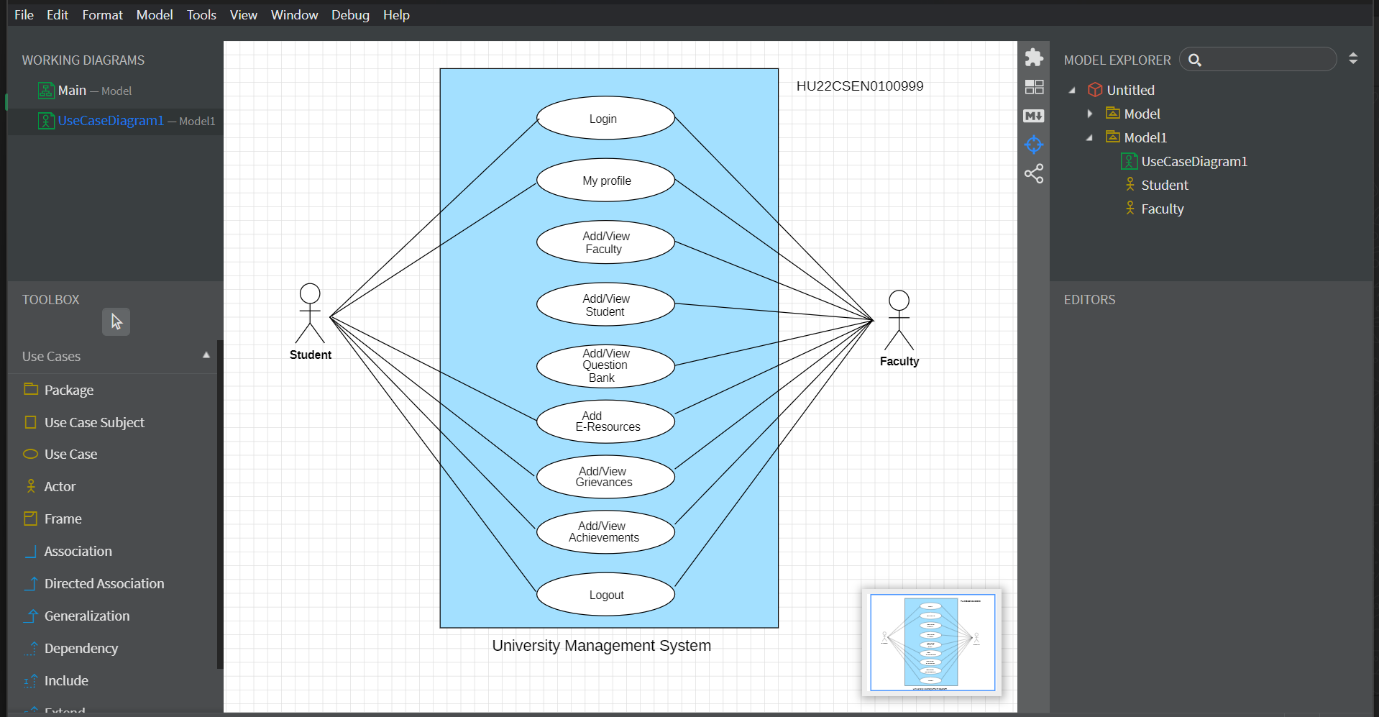
**Diagram Details:**

* **Actors**:
  + **Student**
  + **Teacher**
  + **Admin**
* **System Boundary**: “Use case diagram for university management system”
* **Use Cases**:
  + **Student** can:
    1. **Register**
    2. **Login**
    3. **Enroll in Course**
    4. **View Grade**
    5. **Generate Report** (also possibly shared with Admin)
  + **Teacher** can:
    1. **Grade Courses**
    2. **Notify Students** with important dates/announcements
  + **Admin** can:
    1. **Manage Course Offering**
    2. **Generate Report**

**Explanation**

1. **Student** tasks revolve around **registration**, logging in, and **managing academic activities** (enrolling in courses, viewing grades).
2. **Teacher** is responsible for **grading** and **sending notifications** to students about important events or deadlines.
3. **Admin** oversees the **course offerings** (creating, modifying, or removing courses) and can also **generate institutional reports**.
4. The **system boundary** shows that all these use cases happen within the same “University Management System.”

**Another University UML :**



**Diagram Details:**

* **Actors**:
  + **Student** (on the left)
  + **Faculty** (on the right)
* **System Boundary**: “University Management System’’
* **Use Cases** inside the system:
  + **Login**
  + **My Profile**
  + **Add/View Faculty**
  + **Add/View Courses**
  + **Add/View Resources**
  + **Add/View Achievements**
  + **Logout**