Title: Online Bank Management System

- **Subtitle:** Ex. 7 OO design – Interaction Models using ArgoUML
- **Name:** Karan Sehgal
- **Registration No:** 22BCE3939
- **Team No: 24**
- **Course/Subject:** Software Engineering Lab (BCSE301P)
- **Instructor's Name:** Dr. Mehfooza M

Date of Submission: 19/02/25

Sequence and Collaboration Diagrams for Online Bank Management System

Part 1: Sequence Diagram

1. Use Case: User Login

Step 1: Identify a Use Case

- The user logs into the system.

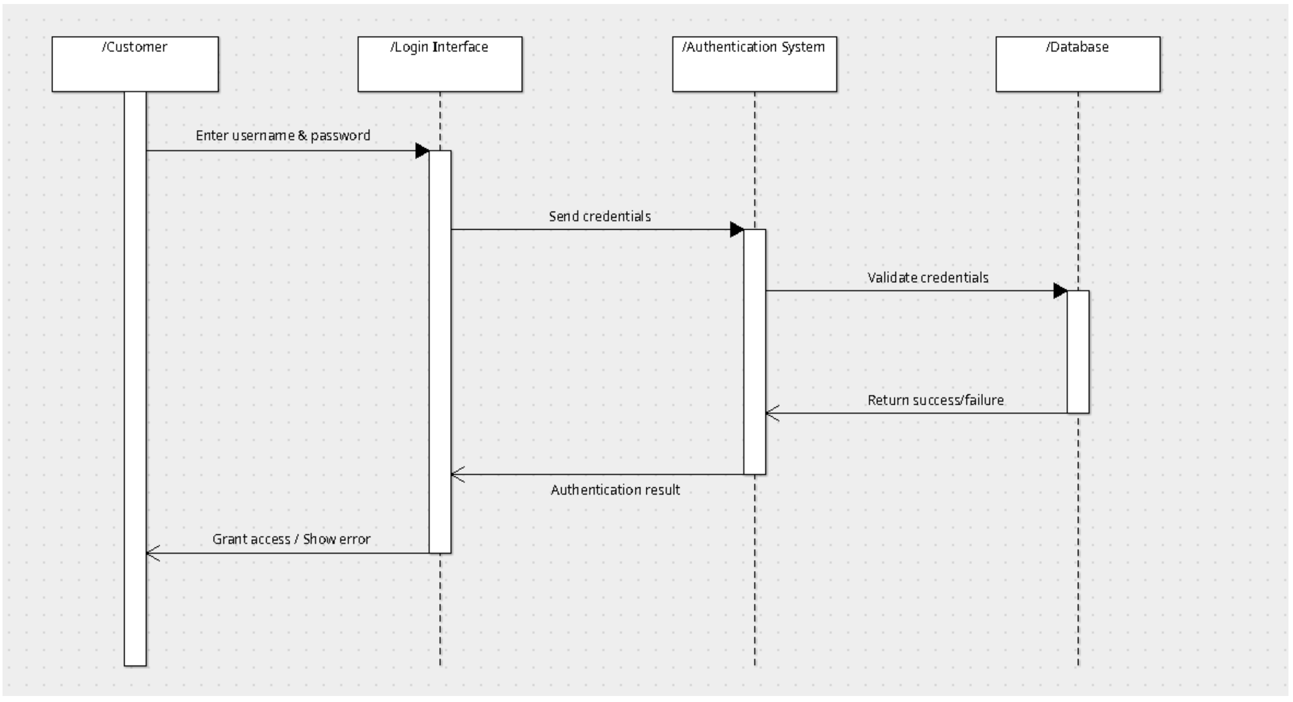
Step 2: Identify Objects

- User, Login Interface, Authentication System, Database

Step 3: Define the Sequence of Interactions

1. User enters credentials.
2. System validates credentials.
3. If valid, system grants access and displays dashboard.
4. If invalid, system shows an error message.

Sequence Diagram



Part 2: Collaboration Diagram

Step 1: Identify the Same Use Case

- The user logs into the system.

Step 2: Define Object Links

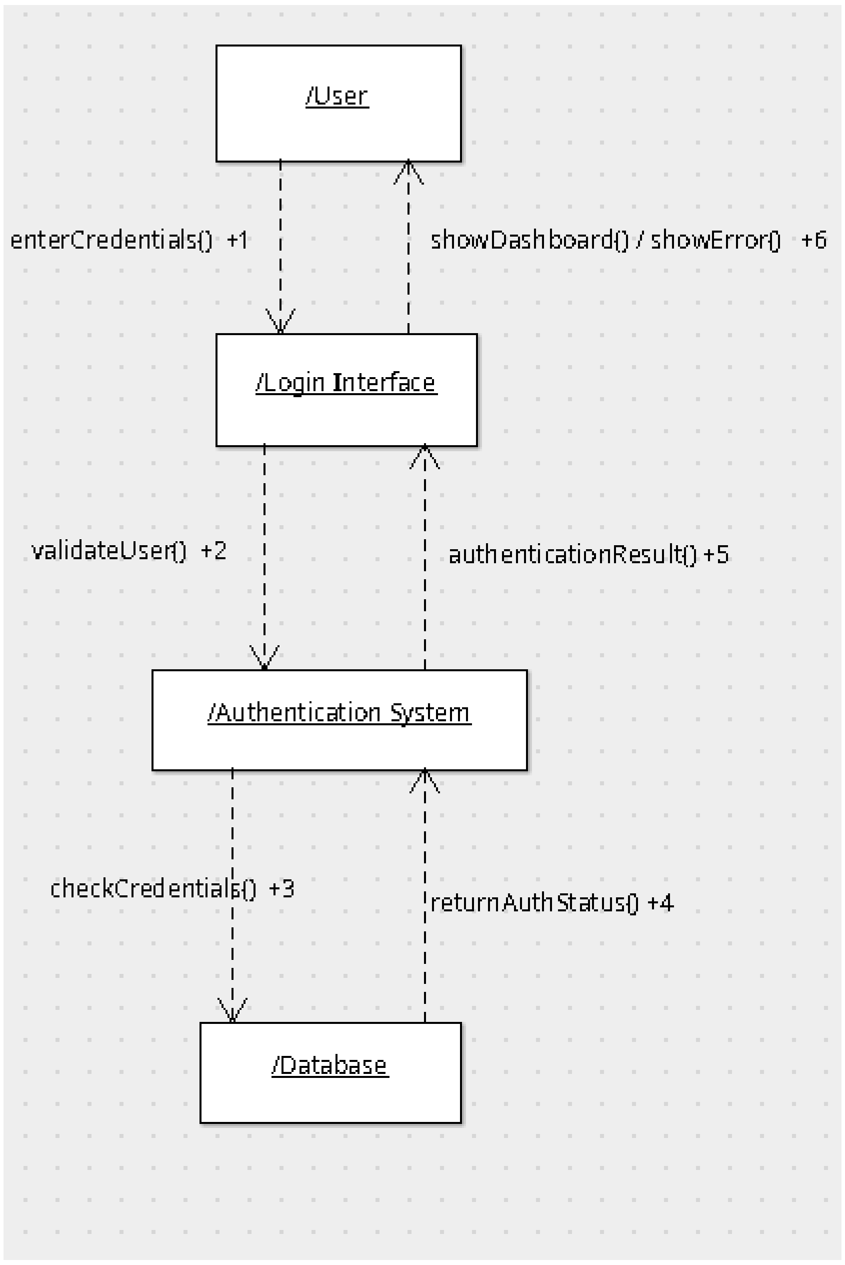
- User interacts with the Login Interface.
- Login Interface sends credentials to the Authentication System.

- Authentication System checks credentials against the Database.
- System responds with success or failure.

Step 3: Define Interactions

1. User requests login.
2. Login Interface forwards credentials to Authentication System.
3. Authentication System verifies credentials with Database.
4. System sends authentication result to User.

Collaboration Diagram



2. Use Case: Fund Transfer

Step 1: Identify a Use Case

- The user transfers funds to another account.

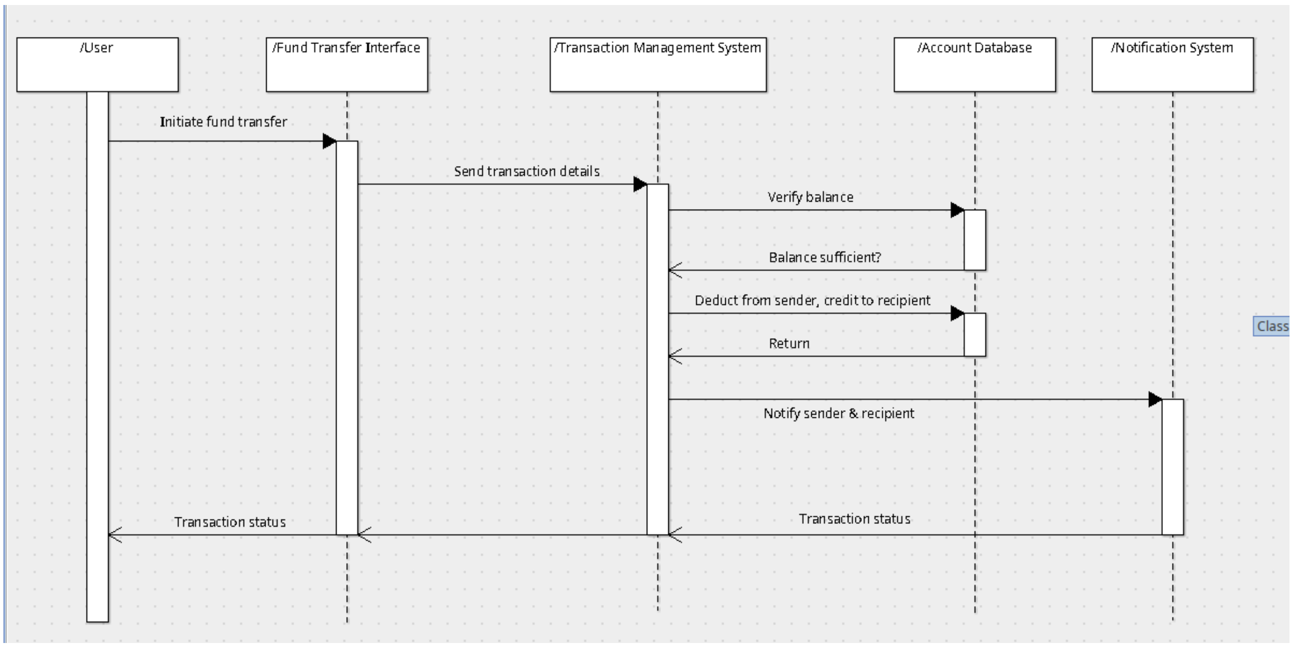
Step 2: Identify Objects

- User, Fund Transfer Interface, Transaction Processor, Account Database, Notification System

Step 3: Define the Sequence of Interactions

1. User selects fund transfer option.
2. User enters recipient details and amount.
3. System validates account balance.
4. If sufficient, system deducts amount from the user’s account and credits to the recipient.
5. System confirms the transaction and notifies the user.

Sequence Diagram



Part 2: Collaboration Diagram

Step 1: Identify the Same Use Case

- The user transfers funds to another account.

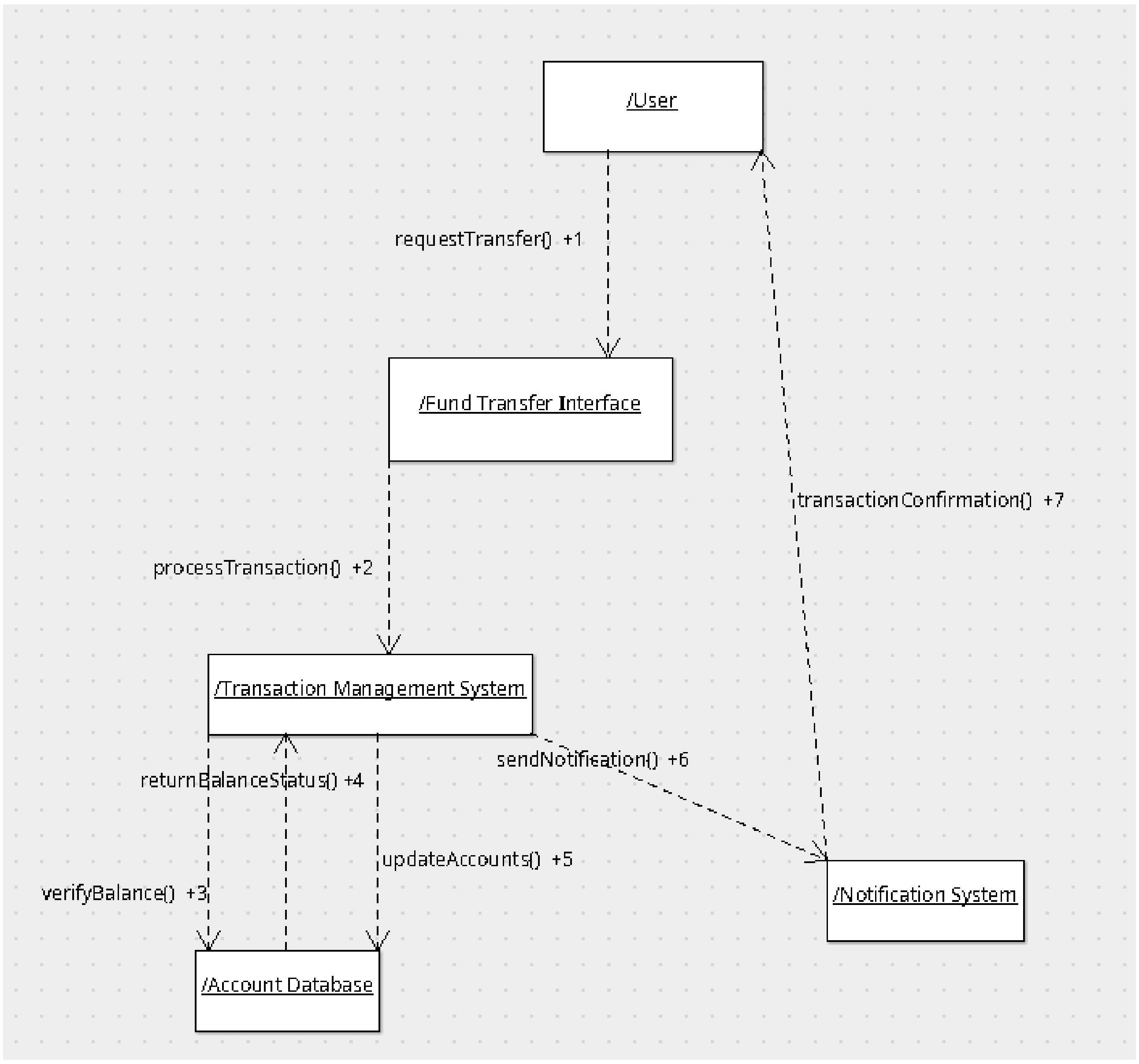
Step 2: Define Object Links

- User interacts with the Fund Transfer Interface.
- Fund Transfer Interface sends transaction details to Transaction Processor.
- Transaction Processor communicates with Account Database.
- Notification System is used to confirm the transaction.

Step 3: Define Interactions

1. User requests fund transfer.
2. Fund Transfer Interface forwards details to Transaction Processor.
3. Transaction Processor verifies balance with Account Database.
4. If sufficient, Transaction Processor updates both sender and recipient balances.
5. Transaction Processor sends confirmation to Notification System.
6. Notification System informs the User.

Collaboration Diagram



3. Use Case: Loan Application

Step 1: Identify a Use Case

- The user applies for a loan.

Step 2: Identify Objects

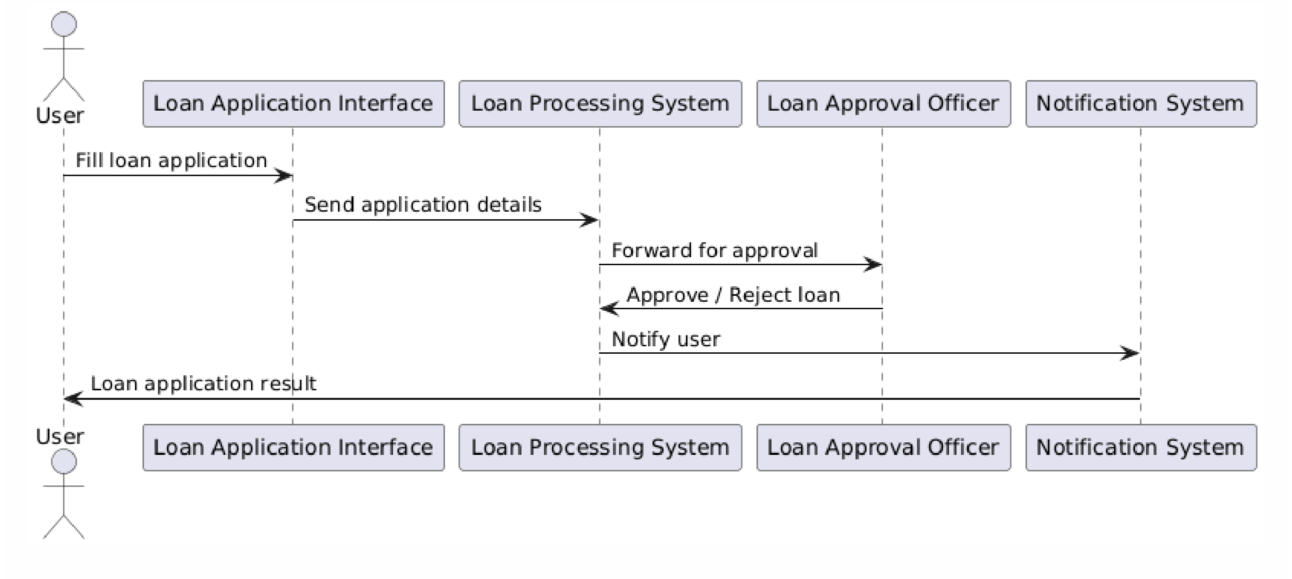
- User, Loan Application Interface, Loan Processing System, Loan Approval Officer, Notification System

Step 3: Define the Sequence of Interactions

1. User selects loan application.
2. User enters loan amount and tenure.
3. System verifies eligibility and forwards request to the Loan Approval Officer.
4. Loan Approval Officer reviews and approves/rejects the loan.
5. System notifies the user of the decision.

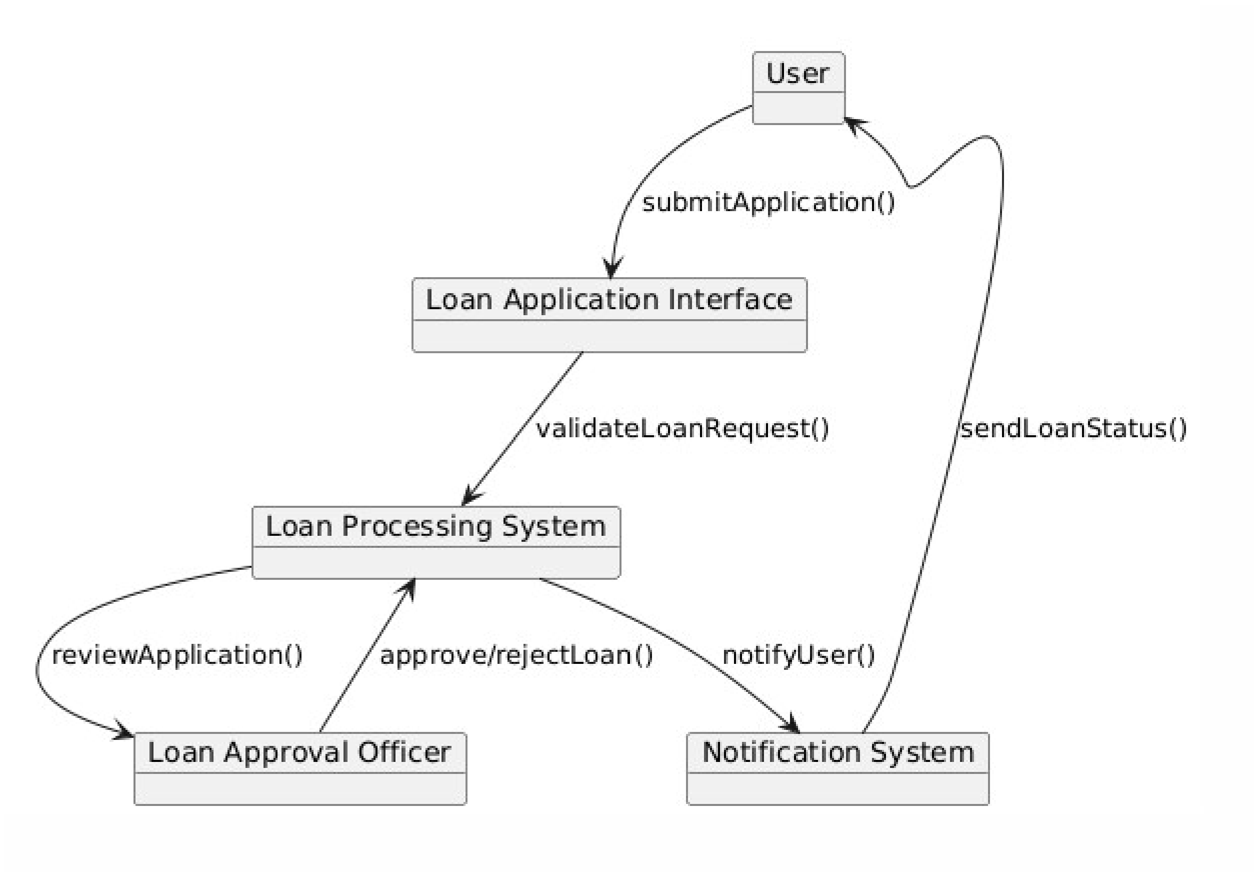
Sequence Diagram

```
@startuml
actor User
participant "Loan Application Interface" as LAI
participant "Loan Processing System" as LPS
participant "Loan Approval Officer" as LAO
participant "Notification System" as NS

User -> LAI : Fill loan application
LAI -> LPS : Send application details
LPS -> LAO : Forward for approval
LAO -> LPS : Approve / Reject loan
LPS -> NS : Notify user
NS -> User : Loan application result
@enduml
```



Collaboration Diagram

```
@startuml
object User
object "Loan Application Interface" as LAI
object "Loan Processing System" as LPS
object "Loan Approval Officer" as LAO
object "Notification System" as NS

User --> LAI : submitApplication()
LAI --> LPS : validateLoanRequest()
LPS --> LAO : reviewApplication()
LAO --> LPS : approve/rejectLoan()
LPS --> NS : notifyUser()
NS --> User : sendLoanStatus()
@enduml
```



User
submitApplication()
Loan Application Interface
validateLoanRequest()
sendLoanStatus()
Loan Processing System
reviewApplication()
approve/rejectLoan()
notifyUser()
Loan Approval Officer
Notification System