## Task 1: Action Lists for Processes

**Deliverable:** A numbered list of actions for Borrowing and Returning equipment .

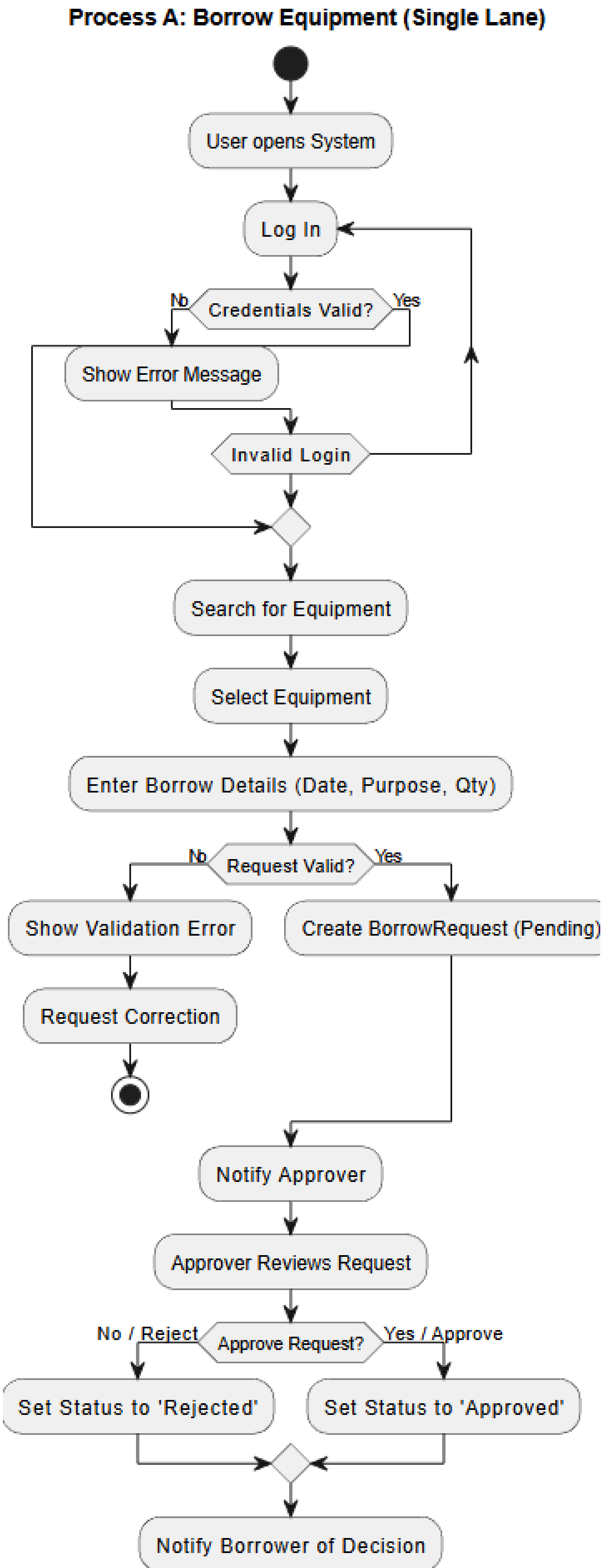
### Process A: Borrow Equipment

1. **Borrower** logs into the system.
2. **System** validates credentials.
3. **Borrower** searches for equipment.
4. **System** displays available equipment matching search criteria.
5. **Borrower** selects items and fills in borrow details (purpose, date/time, quantity).
6. **System** validates the request (checks dates and quantity limits).
    - If invalid, System shows error and asks for correction.
7. **System** creates a new BorrowRequest with status "Pending" and notifies the Approver.
8. **Approver** logs in and views pending requests.
9. **Approver** reviews the request details.
10. **Approver** approves or rejects the request.
11. **System** updates the request status to "Approved" or "Rejected".
12. **System** sends a notification to the Borrower.

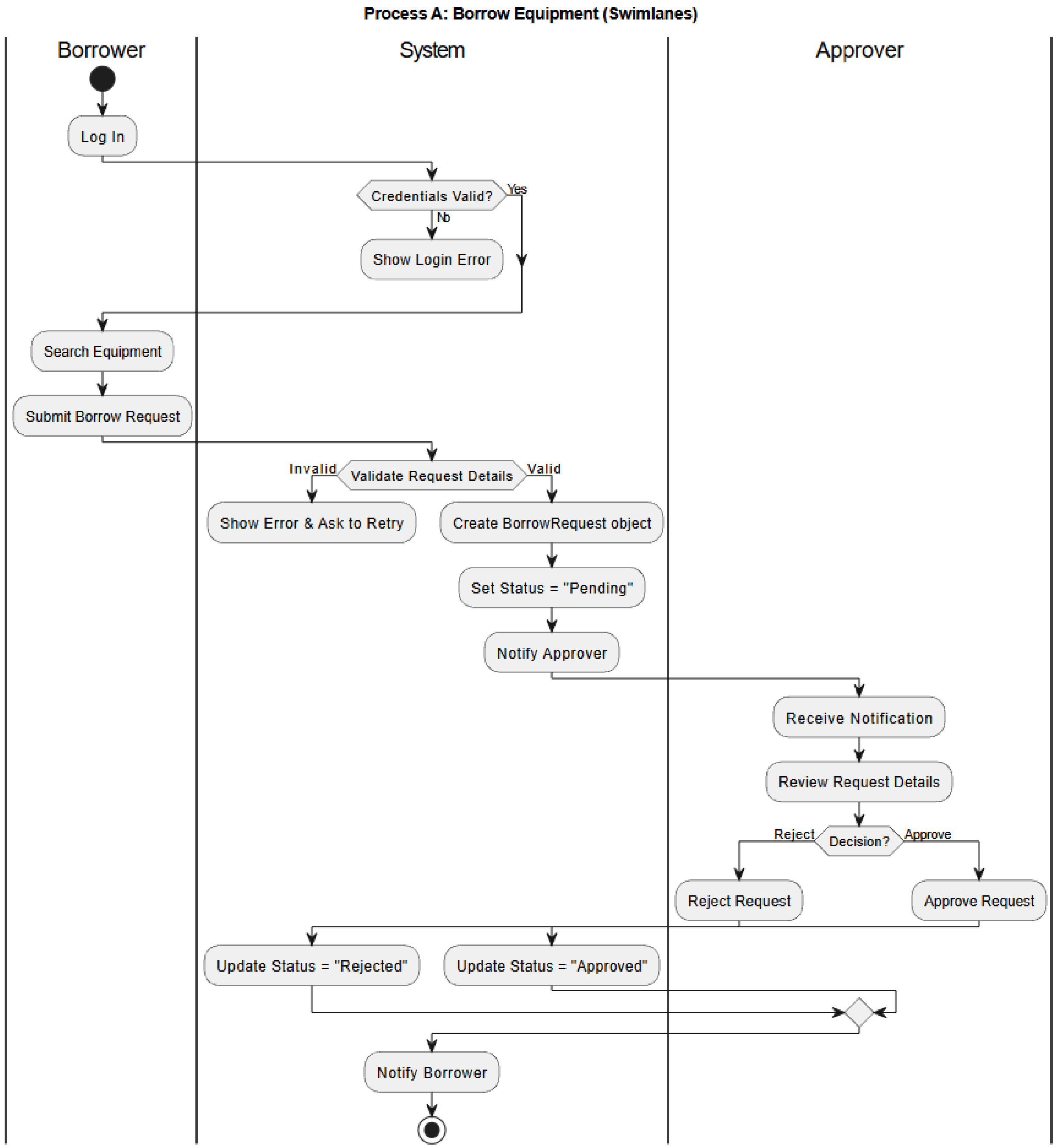
### Process B: Return Equipment

1. **Borrower** brings the equipment to the Inventory Staff.
2. **Inventory Staff** inspects the equipment condition.
3. **Inventory Staff** logs the return into the system.
4. **System** checks the return date against the due date.
    - Determines if the return is "On Time" or "Late".
5. **Inventory Staff** marks the condition.
    - If damaged, a maintenance record is created.
6. **System** (Parallel Action 1): Updates BorrowRequest status to "Returned".
7. **System** (Parallel Action 2): Updates Equipment availability to "Available" (or "Maintenance").
8. **System** logs the transaction history.
9. **System** confirms the return is complete.

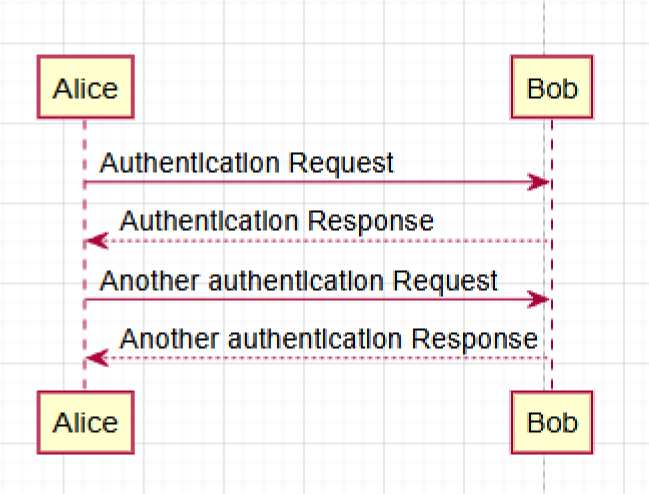
## Task 2: Basic Activity Diagram


Process A: Borrow Equipment (Single Lane)
User opens System
Log In
No
Credentials Valid?
Yes
Show Error Message
Invalid Login
Search for Equipment
Select Equipment
Enter Borrow Details (Date, Purpose, Qty)
No
Request Valid?
Yes
Show Validation Error
Create BorrowRequest (Pending)
Request Correction
Notify Approver
Approver Reviews Request
No / Reject
Approve Request?
Yes / Approve
Set Status to 'Rejected'
Set Status to 'Approved'
Notify Borrower of Decision

## Task 3: Activity Diagram with Swimlanes



## Task 4: Return Equipment (With Parallel Flows)


Alice
Bob
Authentication Request
Authentication Response
Another authentication Request
Another authentication Response
Alice
Bob

## Reflection Questions

**How do Activity Diagrams help compared to Class Diagrams?** the Class Diagram in Lab 04 showed us the static structure (what data is stored), the Activity Diagram shows the dynamic behavior (how the data changes over time). It shows the steps, which a Class Diagram cannot show

**How do Swimlanes clarify responsibility?** Swimlanes separate the actions based on who is performing them. This makes it clear that the System is responsible for validation and database updates, while the Approver is responsible for the human decision to approve or reject.

**Example of Parallel Activities (Fork/Join):** In the Return Equipment process, the use of a fork is necessary when the system updates the BorrowRequest status and updates the Equipment availability. These two database updates are independent of each other; the system does not need to wait for the history log to finish before marking the item as returned. Running them in parallel implies they happen in the same logical transaction block.