MARMARA UNIVERSITY

FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

Requirement Analysis Document (RAD)

Object Oriented Software Design – CSE3063

Project – Iteration 3

Group Members:

- 150121063 - Rabia İlhan
- 150121924 - Haifa Ghribi
- 150122531- Oğuz Fatih Akgemci
- 150120030 - Helin Avcı
- 150120002 - Barış Sedefoğlu
- 150123825 - Ayşenur Arıcı

1. Project Description

We are going to build the final version of the Enhanced Course Registration System, designed for the university department to facilitate course registration and scheduling. The system will accommodate different users: Students, Advisors, Scheduler, Department Head, Admin, and Student Affairs. This iteration is implemented in Python. The core functionalities are course registration, scheduling, capacity management, waitlist, notifications, and conflict management.

2. Glossary

- *Course*: an academic subject offered by the department and it has it is own prerequiste.
- *Course Section*: an instance of a course with an assigned time slot and lecturer.
- *Course Registration*: the process through which students enroll in courses each semester. Registration involves selecting courses and checking for prerequisites.
- *Student System*: sysetm provides the student with functionalities like course selection,
- *Advisor System*: sysetm allows advisors to access student registration decisions, approve or reject course selections.
- *Student*: a user register for courses.
- *Advisor*: a user who assists students in course selection, he may approve or reject student registration.
- *Classroom*: A room where courses are held. Each classroom has: capacity indicating the maximum number of students it can accommodate, timetable, and a list of courses for the day.
- *Department Scheduler (Updated)*: A user responsible for viewing classroom schedules, assigning classrooms and time slots to course sections, managing constraints such as avoiding time conflicts.
- *Mandatory Course (Added)*: A type of course that has special requirements.
- *Technical Elective (Added)*: A type of course that has special requirements.
- *Graduation Project (Added)*: A type of course that has special requirements.
- *Department Head (Added)*: A user responsible for increasing course capacity.
- *Admin (Added)*: create and delete users, courses and classrooms.
- *Student Affairs (Added)*: A user responsible for sending notifications.

3. Functional Requirements

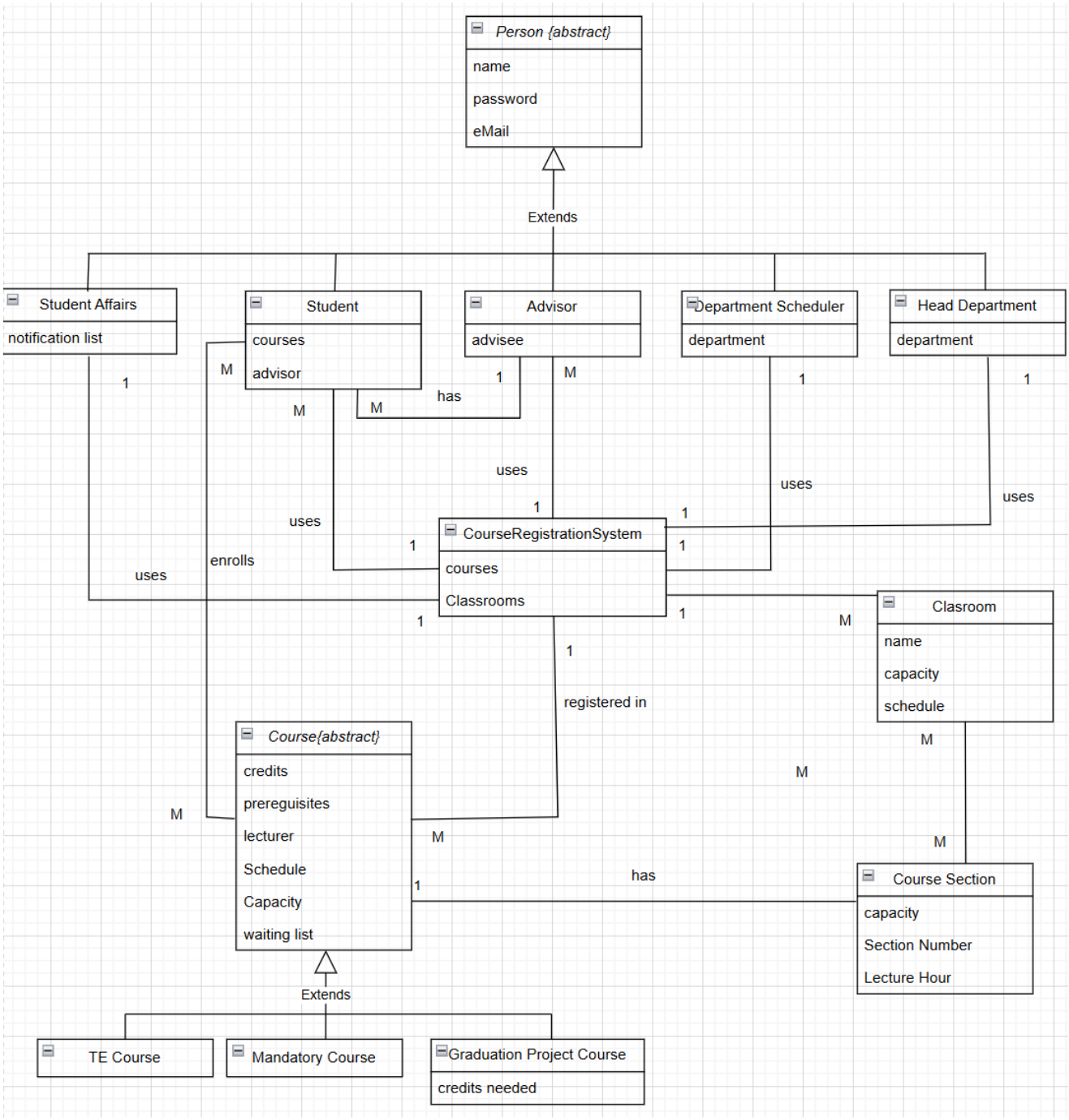
- User Authentication: The system allows students and advisors to log in using their IDs and passwords.
- Course Registration with Prerequisite Checking: Students will be able to register for courses via the system if prerequisite checking is verified.
- Advisor Approval or Rejection: Advisors will have access to view their assigned students' course selections, and they can approve or reject the selected courses.
- Data Storage: all data will be stored in Json files for an easy access.
- Terminal Based Interface: The system will be fully operational from a terminal.
- Scheduler Functionalities: Allows schedulers to assign time slots and classrooms to courses.

- Schedule Management: Students can view and manage their weekly schedules.
- Conflict Detection: Ensures no time conflicts exist for course registrations and room assignments.
- Room Capacity Management: Handles room capacities and constraints during scheduling.
- Waitlist Management: Implements waitlists for courses that reach full capacity.
- *Create and delete users (Added)*: Admin's role.
- *Notifications (Added)*: Sends updates to students about registration and waitlist status changes.
- *Head Department Functionalities (Added)*: Allows head department to increase and decrease course capacities.

4. Non-Functional Requirements

- Security: Secure login and data storage for user authentication, we are going to use different ids for students and staff. Implement role-based access control to restrict functionalities based on user roles.
- Performance (Updated): Efficient handling course selection. Conflict detection, capacity and schedule validation handled efficiently.
- Avaibility: avaible for all university members.
- Data Integrity: Ensure the accuracy of stored data, such as course schedules, waitlists, notifications, and classroom capacities
- Compliance (Updated): The system must comply with university policies on data privacy and course registration procedures. It handles al the requirements for each course type.

5. Domain Model



6. Use Cases

A. Use Case 1: Student – Course Registration

Actor: Student.

Primary Flow:

1. The student logs into the system using his username and password.
2. The system validates the login.
3. The system displays a Course Registration menu.
4. The student selects 'Select Courses' option.
5. The system displays all available courses, showing credits and capacity status.
6. The student selects a course.
7. The system checks prerequisites, capacity, time.
8. If all checks pass, the course is added to the student as selected course.
9. Repeat steps 6,7 and 8, until student clicks on 'exit' option.
10. The system displays a Course Registration menu again.
11. The student selects 'Send selected courses for advisor approval' option.
12. The system sends the selected courses to the advisor.

Alternatives:

- Step 1:
    a) The student enters incorrect credentials. Fail to login and cannot enter the system.
- Step 7:
    a) Prerequisite not met: The system denies registration and displays a message.
    b) Capacity exceeded: The system notifies the student that the course is full and place them on the waitlist.
    c) Time conflict detected: The system prevents registration.

B. Use Case 2: Advisor - Approval or Rejection

Actor: Advisor.

Primary Flow:

1. The advisor logs into the system using their username and password.
2. The system validates the login
3. The System displays a menu.
4. The advisor selects the option "List students who selected courses and sent them for approval."
5. The system displays a list of students who have submitted their course registrations.
6. The advisor selects a student from the list.
7. The system displays the student's course selections.
8. The advisor reviews, and either approves the courses or rejects the courses.

Alternatives:

- Step 1:

a) The advisor enters incorrect credentials. Fail to login and cannot enter the system.

- Step 3:
  a) No students have submitted courses for approval. Advisor returns to the main menu or exit.

C. Use Case 3: Department Scheduler - Assign Time Slots and Classrooms

Actor: Department Scheduler.

Primary Flow:

1. The scheduler logs into the system using their username and password.
2. The system validates the login.
3. The system displays a menu.
4. The scheduler selects 'Assign a new lecture hour to a course section' option.
5. The system displays courses.
6. The scheduler selects a course.
7. The system displays course sections.
8. The scheduler selects a course section.
9. The system displays classrooms.
10. The scheduler selects a classroom.
11. The system displays the schedule of the selected classroom.
12. The scheduler assigns new lecture hours.
13. The system checks and detects conflicts.
14. If all checks pass, the lecture hour is assigned to the course section and the classroom.

Alternatives:

- Step 1:
  a) The department scheduler enters incorrect credentials. Fail to login and cannot enter the system
- Step 14:
  a) Conflict detected, the system prevents the assignment and displays the specific conflict details.

D. Use Case 4: Department Head - Increase Course Capcity (Added)

Actors: Department Head

Primary Flow:

1. The department head logs into the system using their username and password
2. The system validates the login.
3. The system displays a menu.
4. The departmnet head selects 'Increase course section capacity' option.
5. The system displays cours sections.
6. The departmemt head chooses a course section and increase its capacity.
7. The capacity is increased.

8. Repeat steps 5,6 and 7, until department head clicks on ‘exit’ option.

Alternatives:

- Step 1:
    a) The department head enters incorrect credentials. Fail to login and cannot enter the system.

E. Use Case 5: Admin – Add new user (Added)

Actors: Admin

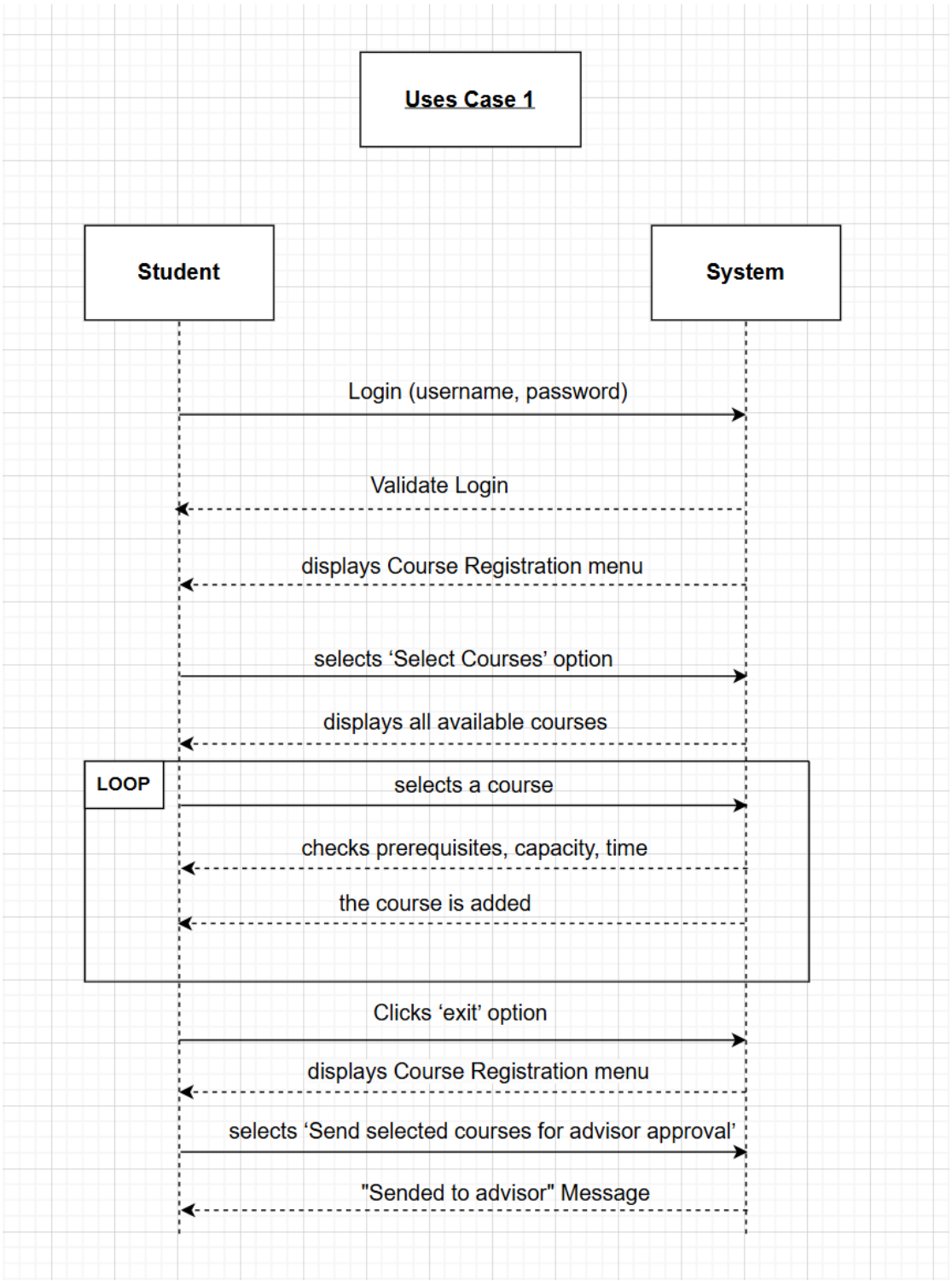
Primary Flow:

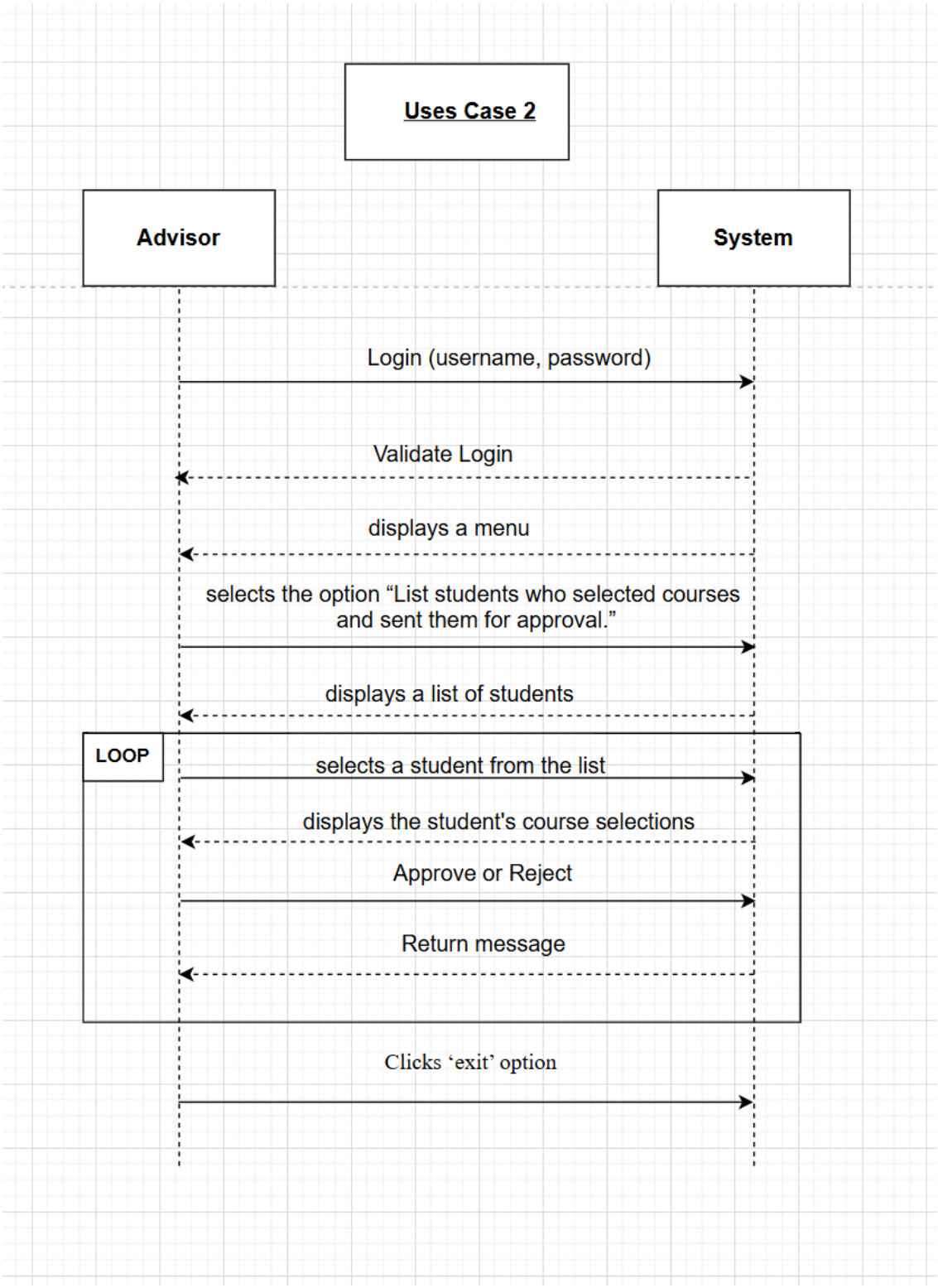
1. The admin logs into the system using their username and password.
2. The system validates the login.
3. The system displays a menu.
4. The admin selects ‘Create new user’ option.
5. The system displays users option menu.
6. The admin selects ‘Add new student’.
7. The systems dispalys a list to fill.
8. The admin adds the required informations.
9. If all checks pass, new student will be added to the system

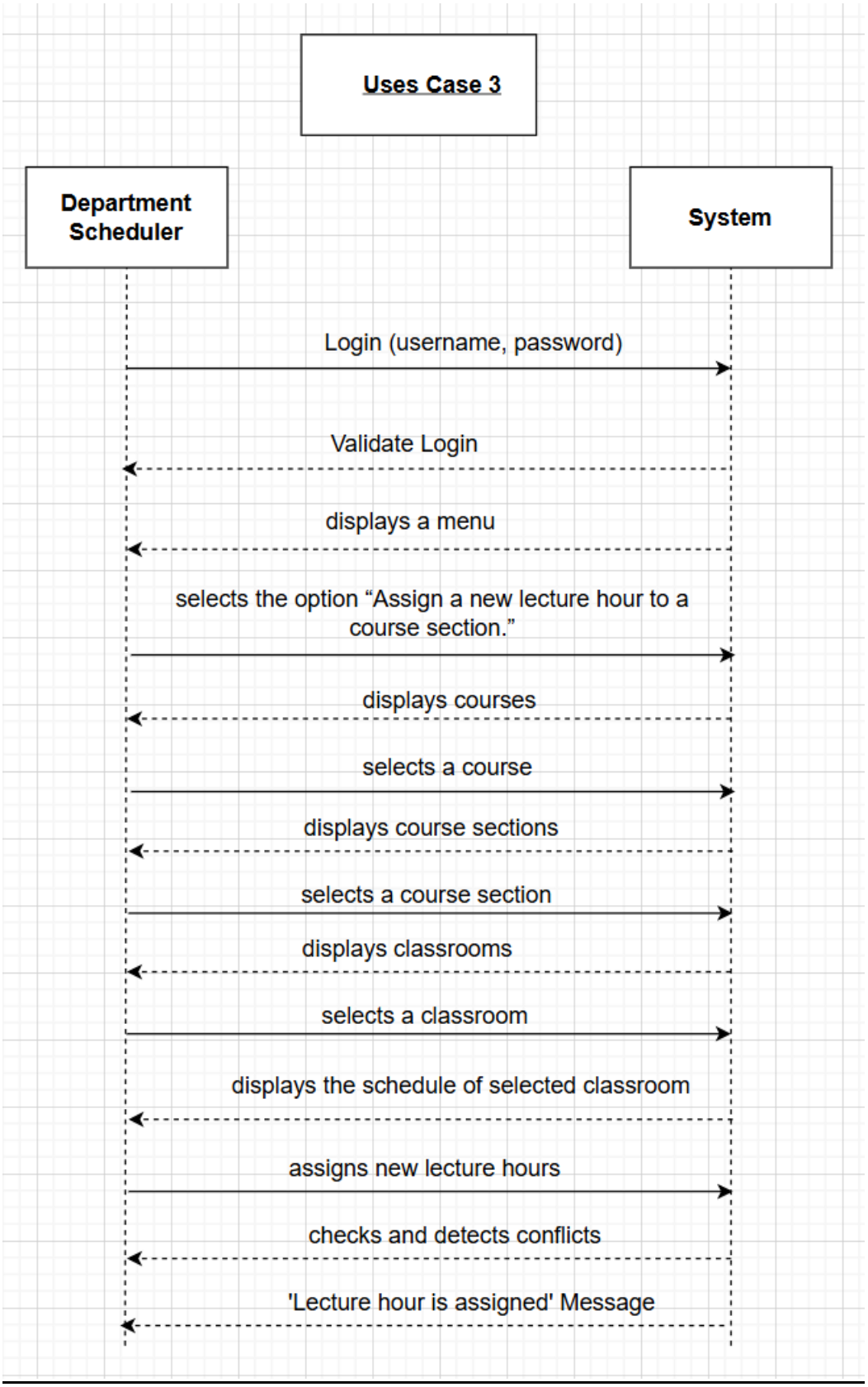
Alternatives:

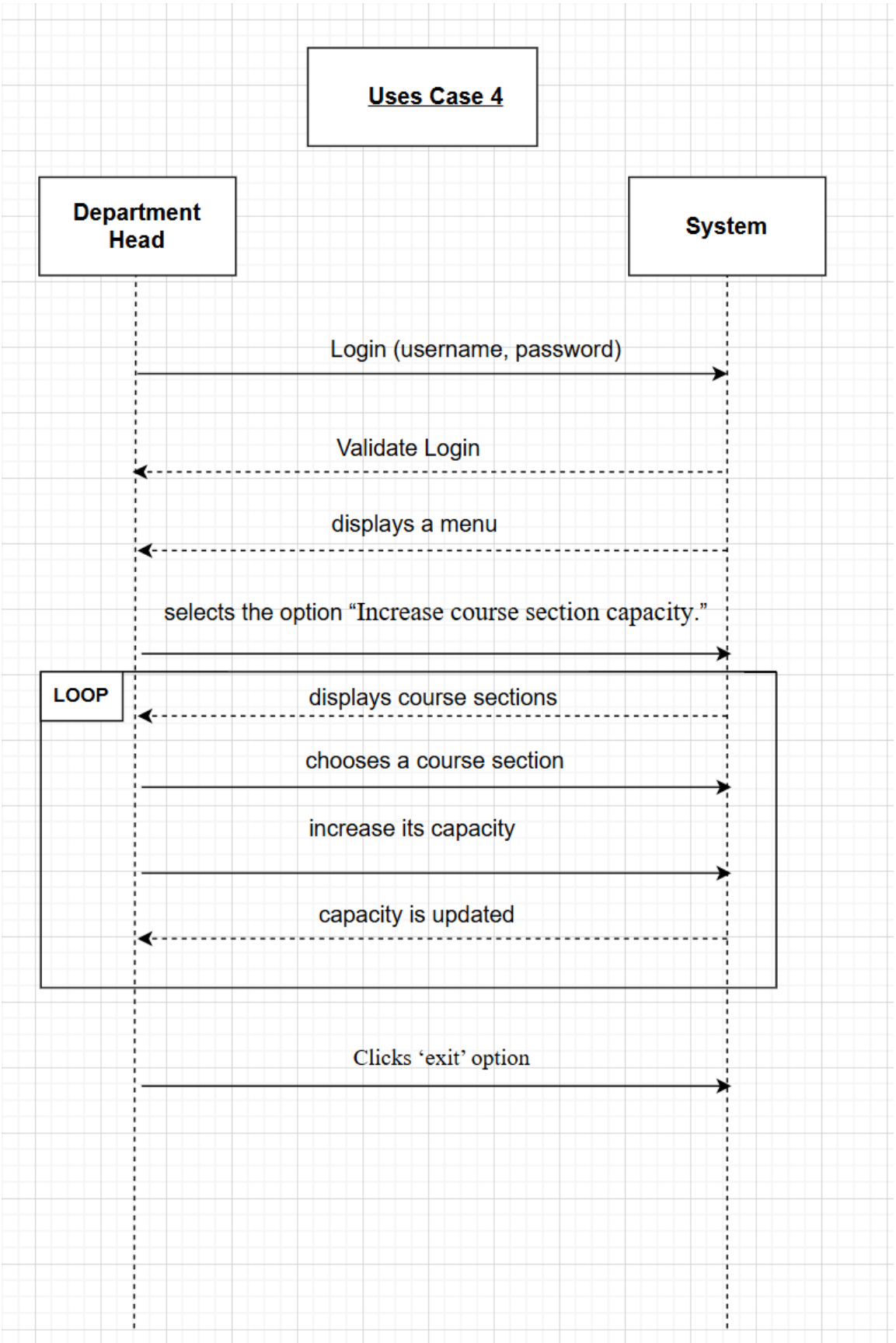
- Step 1:
    b) The admin enters incorrect credentials. Fail to login and cannot enter the system.
- Step 6:
    a) The admin selects ‘Add new advisor’
    b) The admin selects ‘Add new department scheduler’
    c) The admin selects ‘Add new department head’
- Step 9:
    a) Error occurs, the system prevents adding new user.

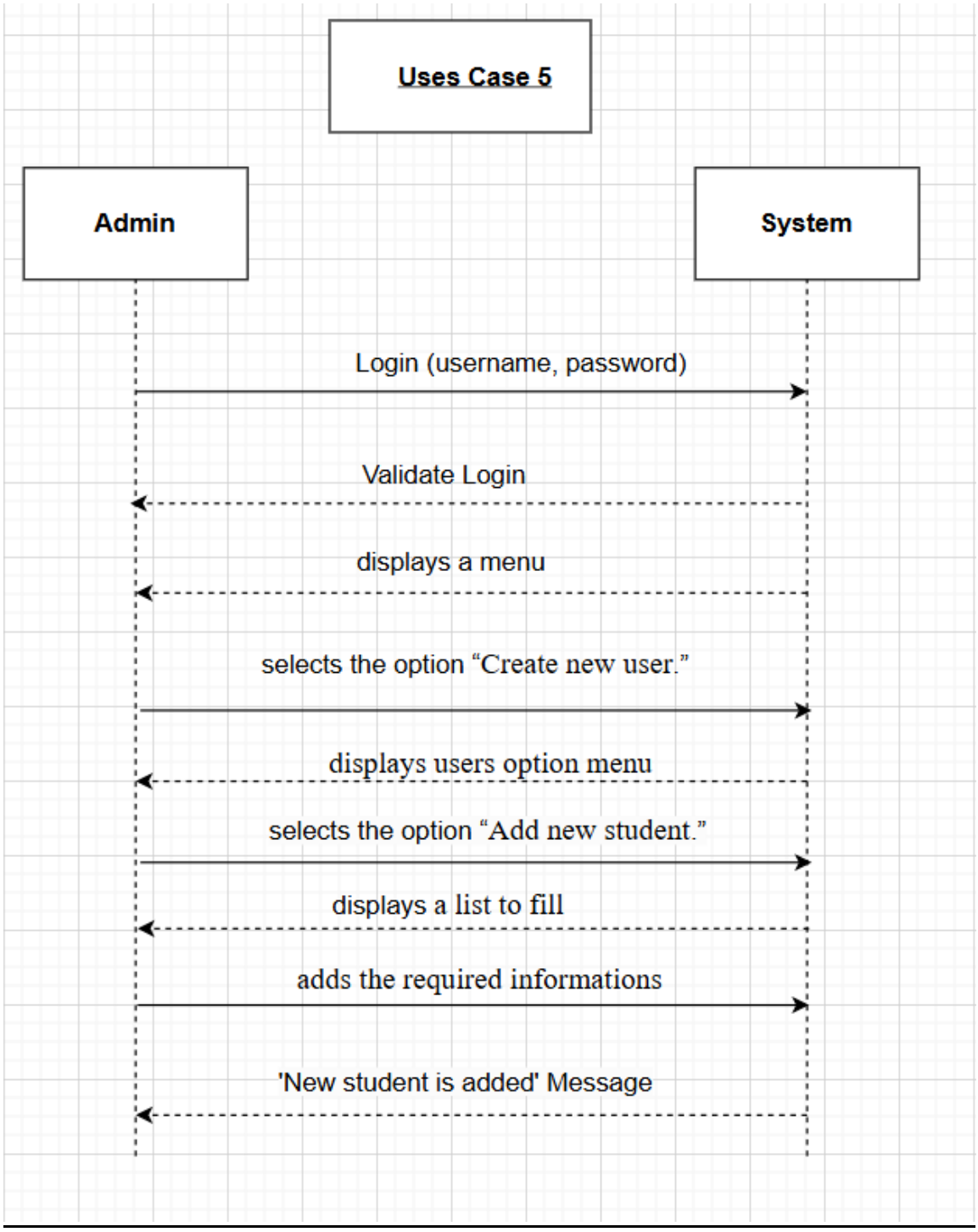
7. System Sequence Diagrams (SSDs)




Uses Case 2
Advisor
System
Login (username, password)
Validate Login
displays a menu
selects the option “List students who selected courses and sent them for approval.”
displays a list of students
LOOP
selects a student from the list
displays the student's course selections
Approve or Reject
Return message
Clicks ‘exit’ option


Uses Case 3
Department Scheduler
System
Login (username, password)
Validate Login
displays a menu
selects the option “Assign a new lecture hour to a course section.”
displays courses
selects a course
displays course sections
selects a course section
displays classrooms
selects a classroom
displays the schedule of selected classroom
assigns new lecture hours
checks and detects conflicts
'Lecture hour is assigned' Message


Uses Case 4
Department Head
System
Login (username, password)
Validate Login
displays a menu
selects the option “Increase course section capacity.”
LOOP
displays course sections
chooses a course section
increase its capacity
capacity is updated
Clicks ‘exit’ option


Uses Case 5
Admin
System
Login (username, password)
Validate Login
displays a menu
selects the option “Create new user.”
displays users option menu
selects the option “Add new student.”
displays a list to fill
adds the required informations
'New student is added' Message