Palestine Technical University - Kadoorie

College of Engineering and Technology

Computer Systems Engineering

Project title:

BUILDING A SERVICE PLATFORM FOR PEOPLE IN PALESTINE

By:

Ahmad Tomeh 202110082

Yazan husain 202110871

Yousef Jaber 202110266

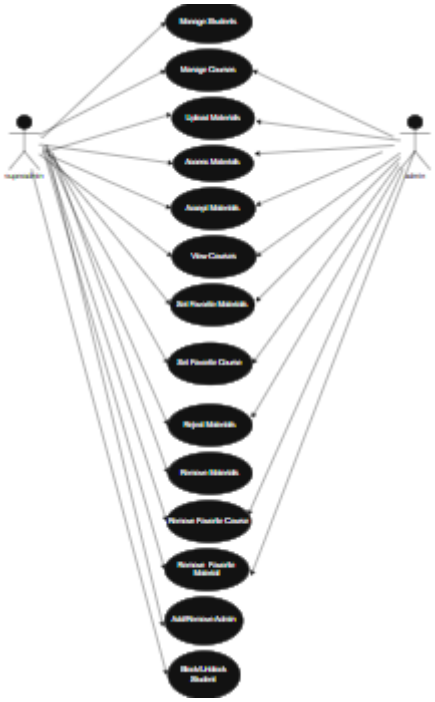
Supervisor:

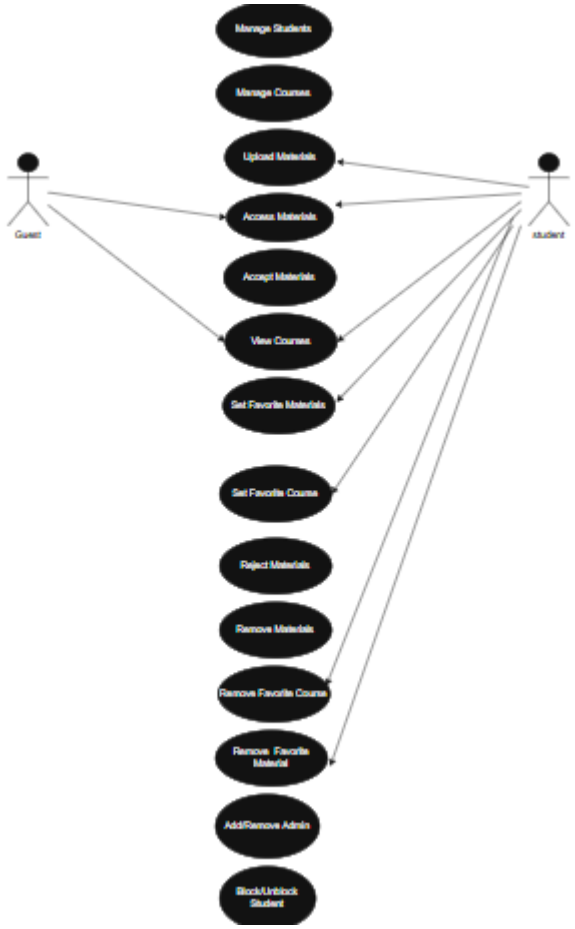
Dr. Yazeed Sleet

Tulkarm, Palestine

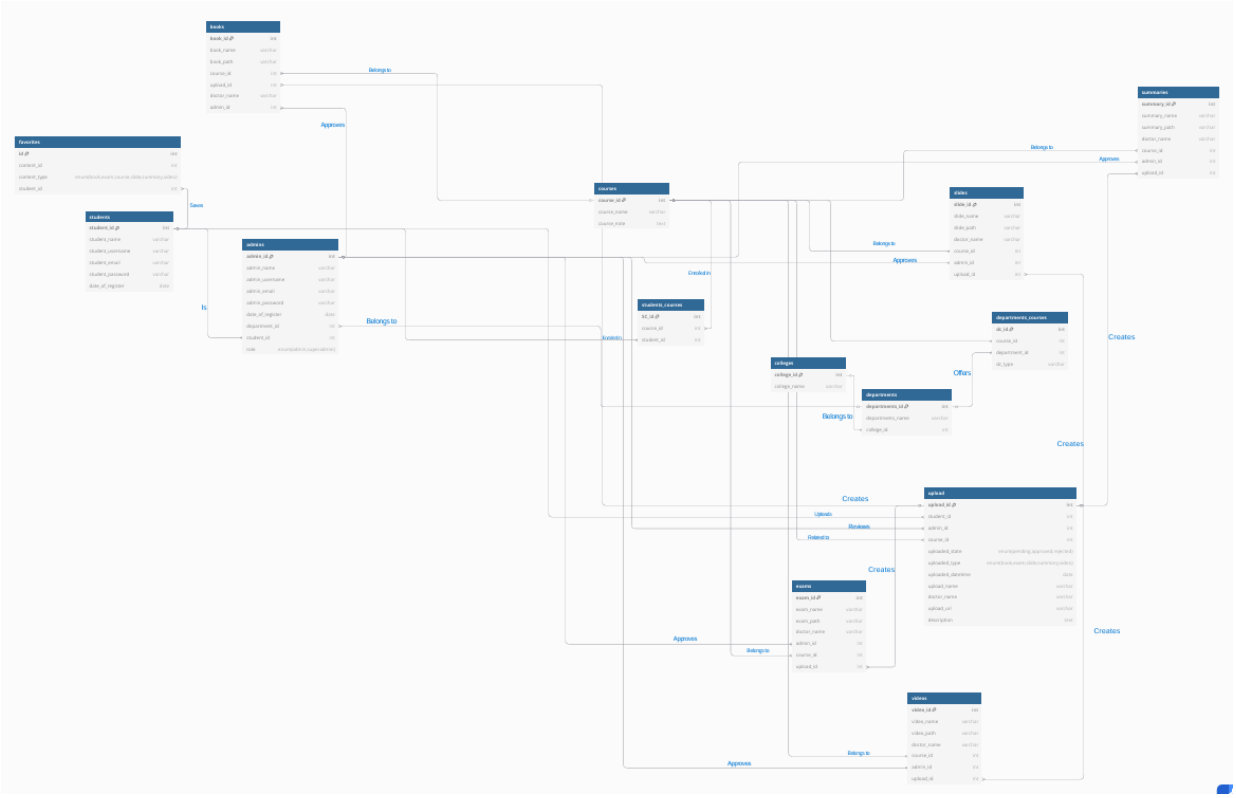
May 2025 20

1- Use Case Diagrams

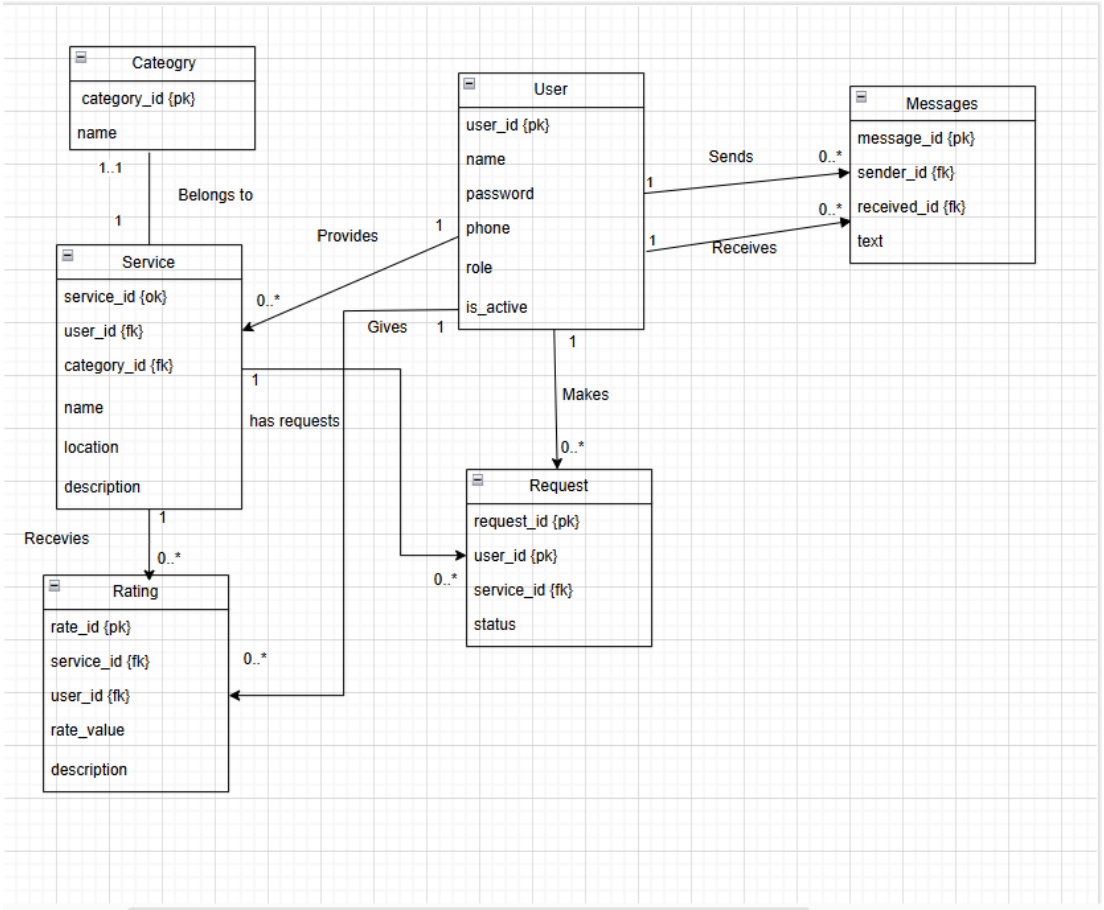




2 - Entity relationship diagram :



3 - Domain model class diagram :



Team Work:

Names	Work
Yousef Jaber	‘Log out’ diagrams
Ahmad Tomeh	‘Register for Student’ diagrams
Yazan Husain	‘Edit Service Requester data’ diagrams

4 - Fully Developed use case description :

1 . Register for Student

2 . *upload file by student*

3 . log out

<table>
<tr><td>Use case name</td><td>Register for Student</td></tr>
<tr><td>Scenario</td><td>A new student registers for the educational platform.</td></tr>
<tr><td>Triggering event</td><td>The student clicks the “Register” button on the login page.</td></tr>
<tr><td>Brief description</td><td>This use case describes the process where a student fills out a registration form with their personal information (full name, email, university ID, password), uploads an image of their student card, and submits the request. The system stores the data and sets the account status as "Pending Approval" until verified by an admin.</td></tr>
<tr><td>Actors</td><td>Student
Admin (indirectly – reviewer)</td></tr>
<tr><td>Related use cases</td><td>Admin approves student registration</td></tr>
<tr><td>Stakeholders</td><td>Student
Admin
System users (who may interact later with the student)</td></tr>
<tr><td>Preconditions</td><td>1. The student is not already registered.

All mandatory fields are correctly filled in.</td></tr>
<tr><td>Postconditions</td><td>1. The student account is created with status = “Pending Approval”.

Admin will review and approve/reject the account later.</td></tr>
<tr><td>Flow of activities</td><td>
<table>
<tr><td>System</td><td>Actor</td></tr>
<tr><td>1.1 Navigates user to the profile editing page</td><td>1. Clicks on “Register”</td></tr>
<tr><td>1.2 Displays editable fields</td><td>2. Enters full name, email, university ID, password</td></tr>
<tr><td>1.3 Validates input data</td><td>3. Uploads student card image</td></tr>
<tr><td>4.1 Validate form input</td><td>4. Clicks “Submit”</td></tr>
<tr><td>4.2 Check if email or ID already exists</td><td></td></tr>
<tr><td>4.3 Save student data with status “Pending</td><td></td></tr>
</table>
</td></tr>
</table>

	Approval”	
	4.4 Show confirmation message	
Exception conditions	• **4.1** If any required field is missing or invalid → show error message. • **4.2** If email or university ID already exists → show duplication warning. • **4.3** If student card upload fails → ask student to re-upload. 2.1	

Figure 4.1: Fully developed use case description for the " **Register for Student** " use case

<table>
<tr><td>:Use case name</td><td>Upload File by Student</td></tr>
<tr><td>Scenario</td><td>A student uploads an educational file (e.g., book, slide, exam) to the platform.</td></tr>
<tr><td>:Triggering event</td><td>The student initiates the upload process by clicking the "Upload File" button and selecting the appropriate file type (e.g., book, slide, exam).</td></tr>
<tr><td>:Brief description</td><td>This use case describes the process where a student uploads a file to the platform (e.g., book, slide), enters basic metadata (e.g., course name, doctor name), and submits the form. The file will be saved with status “Pending Approval” until reviewed by an admin.</td></tr>
<tr><td>:Actors</td><td>Student, Admin (indirectly after submission).</td></tr>
<tr><td>:Related use cases</td><td>Admin approves uploaded file.</td></tr>
<tr><td>:Stakeholders</td><td>Student, Admin, Other students.</td></tr>
<tr><td>:Preconditions</td><td>1 - The student must be logged in.
2 - File type, file name, and course info must be provided.</td></tr>
<tr><td>:Postconditions</td><td>1-File is uploaded and saved in the database.
2- File status is set to “Pending Approval.”
3-Admin is notified (optional future flow).</td></tr>
<tr><td>:Flow of activities</td><td>
<table>
<tr><td>Actor</td><td>System</td></tr>
<tr><td>1 - Clicks on "Upload File".
2 -Selects file type and enters file info.</td><td></td></tr>
<tr><td>3-Uploads the file and submits.</td><td>3.1- validate form input.

3.2 –Saves file and metadata.

3.3- Sets status to "Pending Approval".</td></tr>
</table>
</td></tr>
<tr><td>Exception :conditions</td><td>3.1 - The system shows an error message if required fields are missing or invalid. Upload cannot proceed until corrected.

3.2 -The system displays a warning if a file with the same name and type already exists. The student must rename or change the file.

3.3-The system displays a warning if a file with the same name and type already exists. The student must rename or change the file.</td></tr>
</table>

Figure 4.2: Fully developed use case description for the " **Upload File by Student**" use case

<table>
<tr><td>:Use case name</td><td>log out</td></tr>
<tr><td>Scenario</td><td>A logged-in user wants to end their session securely</td></tr>
<tr><td>:Triggering event</td><td>The user clicks the "Log Out" button</td></tr>
<tr><td>:Brief description</td><td>This use case describes the process by which a user (admin, student, .or student admin) logs out to ensure account security and privacy</td></tr>
<tr><td>:Actors</td><td>Student, admin and student admin</td></tr>
<tr><td>Related use :cases</td><td>Login use case</td></tr>
<tr><td>:Stakeholders</td><td>User</td></tr>
<tr><td>:Preconditions</td><td>.The user must be logged in</td></tr>
<tr><td>:Postconditions</td><td>.The user’s session is securely terminated - 1
.The user is redirected to the homepage or login page - 2</td></tr>
<tr><td>:Flow of activities</td><td><table><tr><td>Actor</td><td>System</td></tr><tr><td>1 - Clicks the "Log Out" button or link on the website.</td><td>1.1 Detects the user's request to log out.
1.2 Terminates the user's session and clears session data.
🎬</td></tr></table></td></tr>
<tr><td>Exception :conditions</td><td>Network error</td></tr>
</table>

Figure 4.3: Fully developed use case description for the " ***log out*** " use case

5 - Activity diagram :

1 . register new service provider

2 . *edit service requester data*

3 . log out

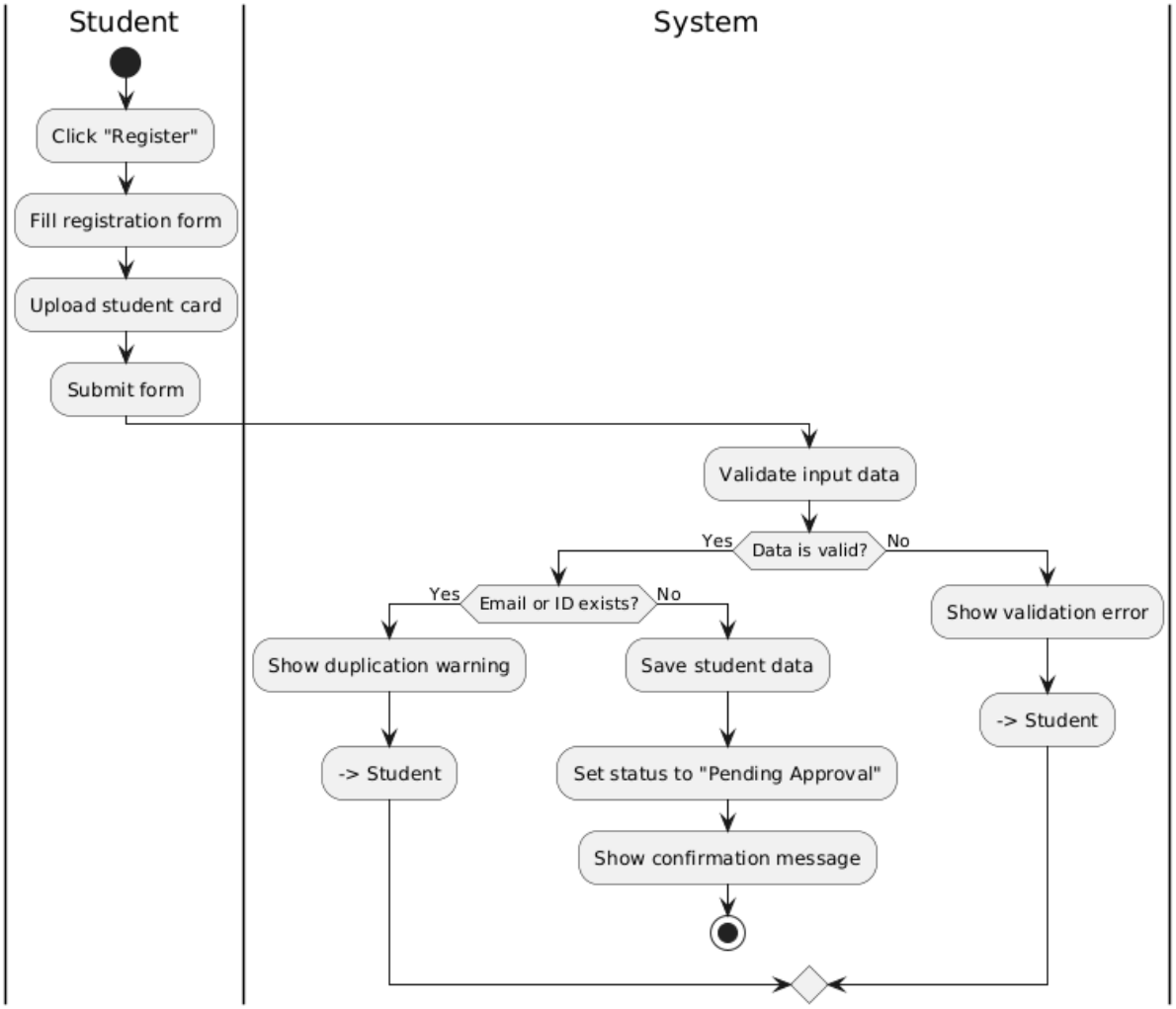


Figure 5.1: Activity diagram for " **register for student** " use case

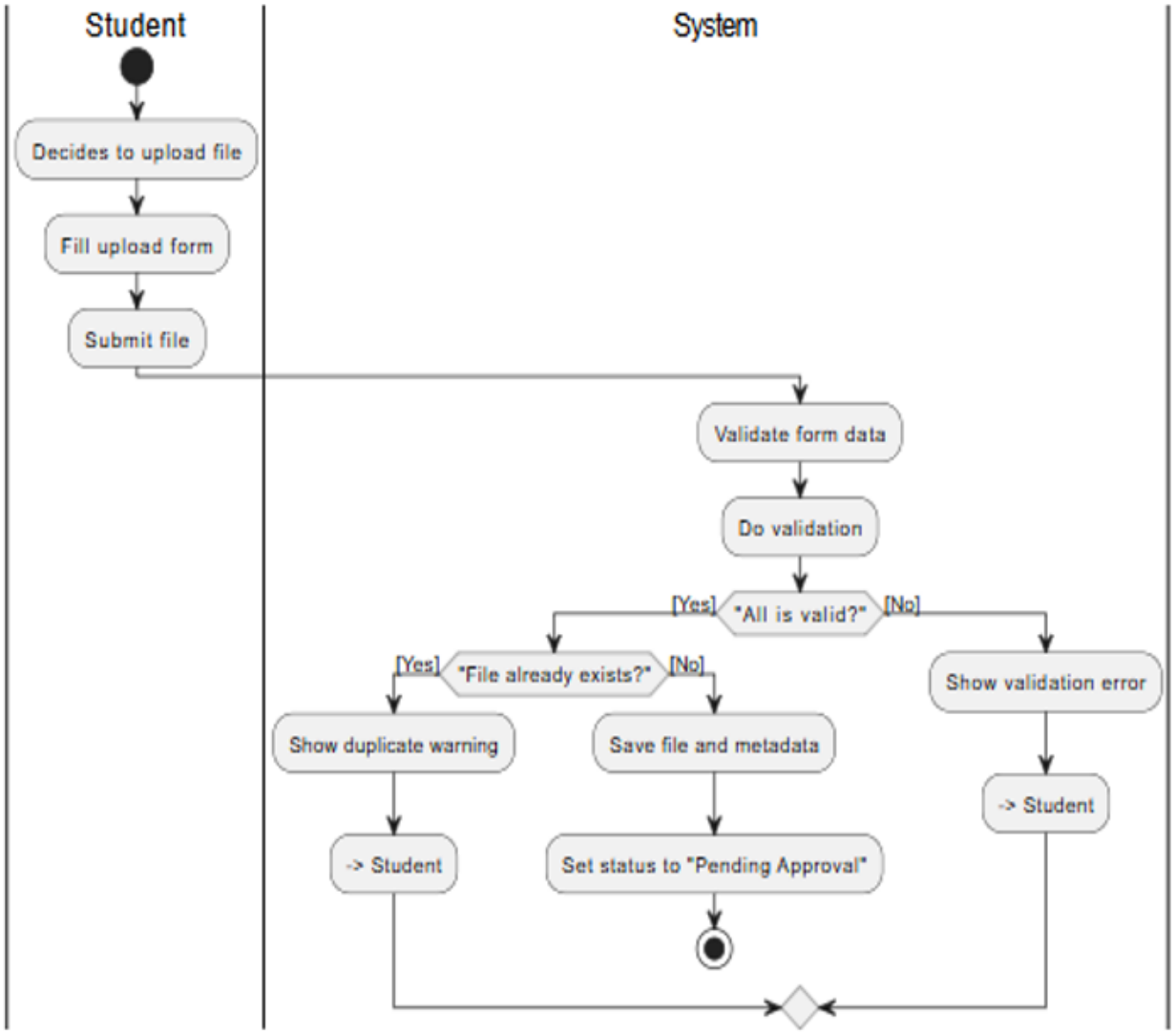


Figure 5.2: Activity diagram for " ***upload file by student***" use case

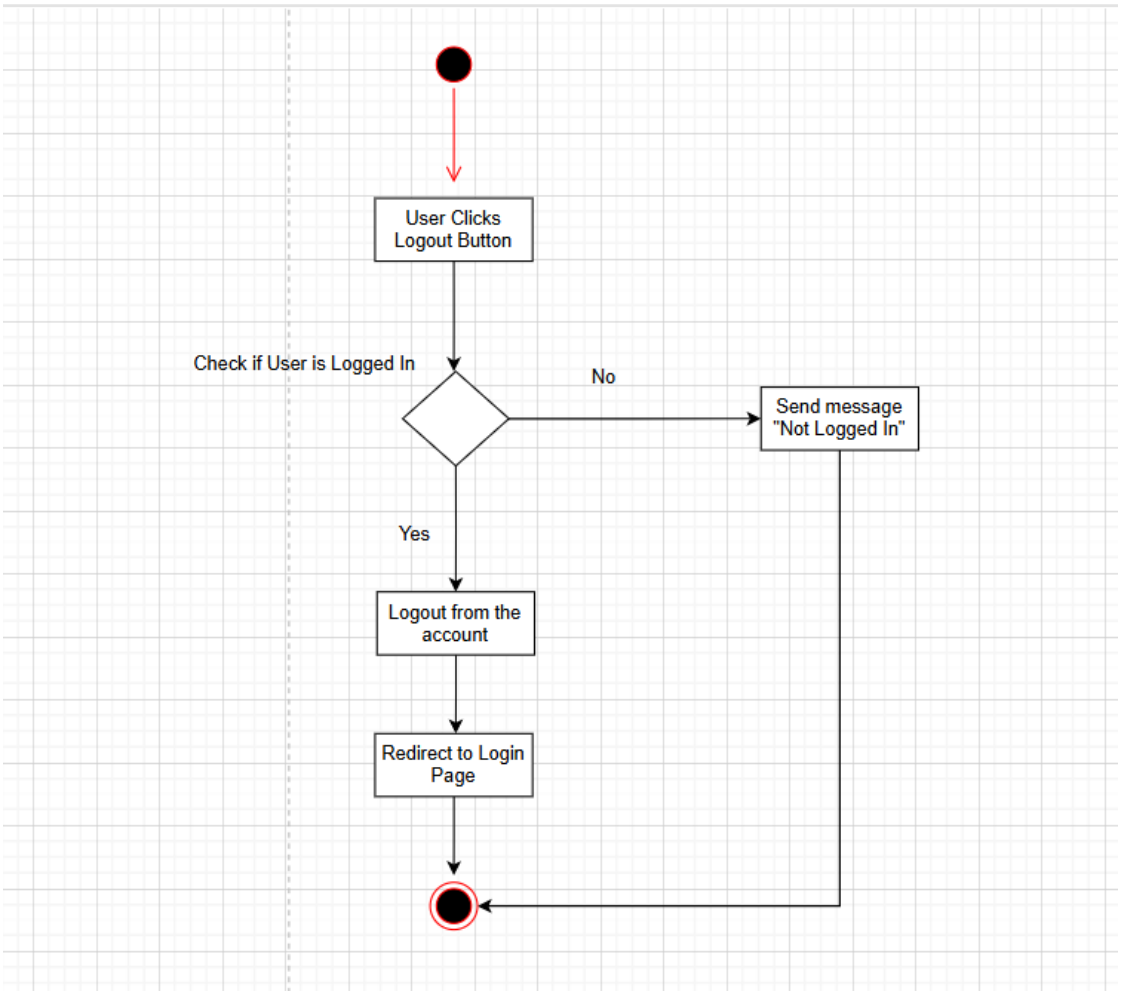


Figure 5.3: Activity diagram for " ***log out*** " use case

6 - System sequence diagram:

1 . register new service provider

2 . *edit service requester data*

3 . log out

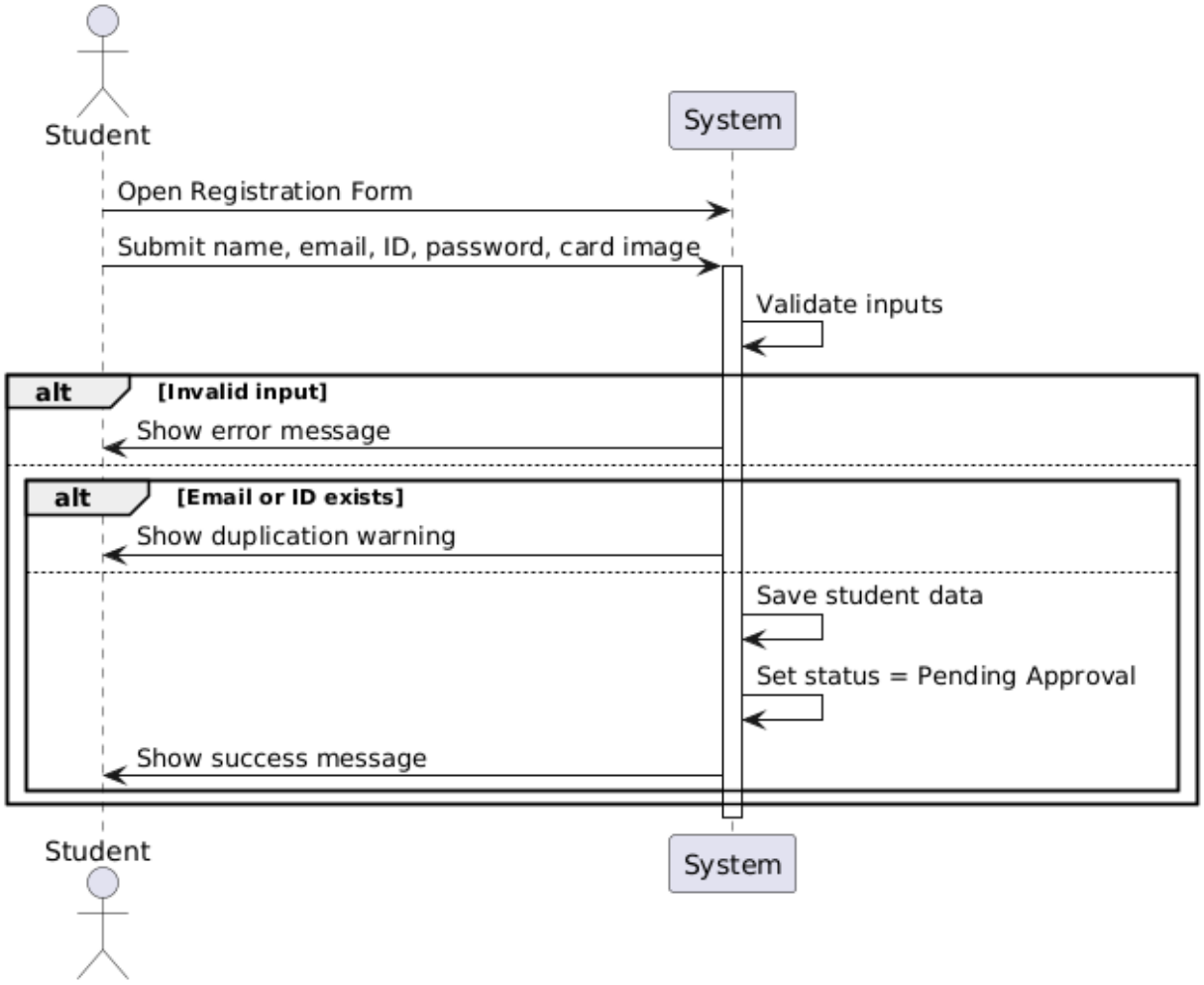


Figure 6.1: System Sequence Diagram for " **register for student** " use case

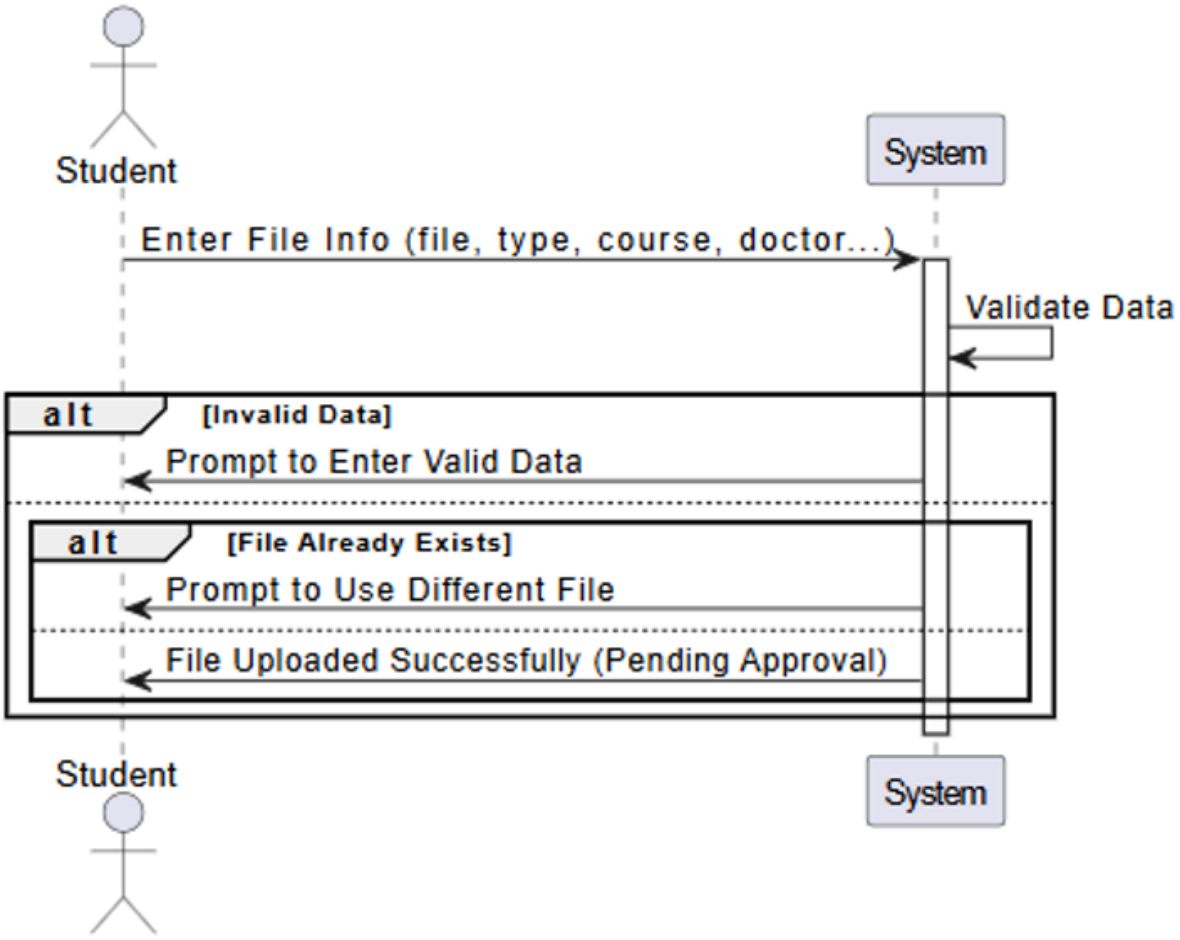


Figure 6.2: System Sequence Diagram for " ***upload file by student*** " use case

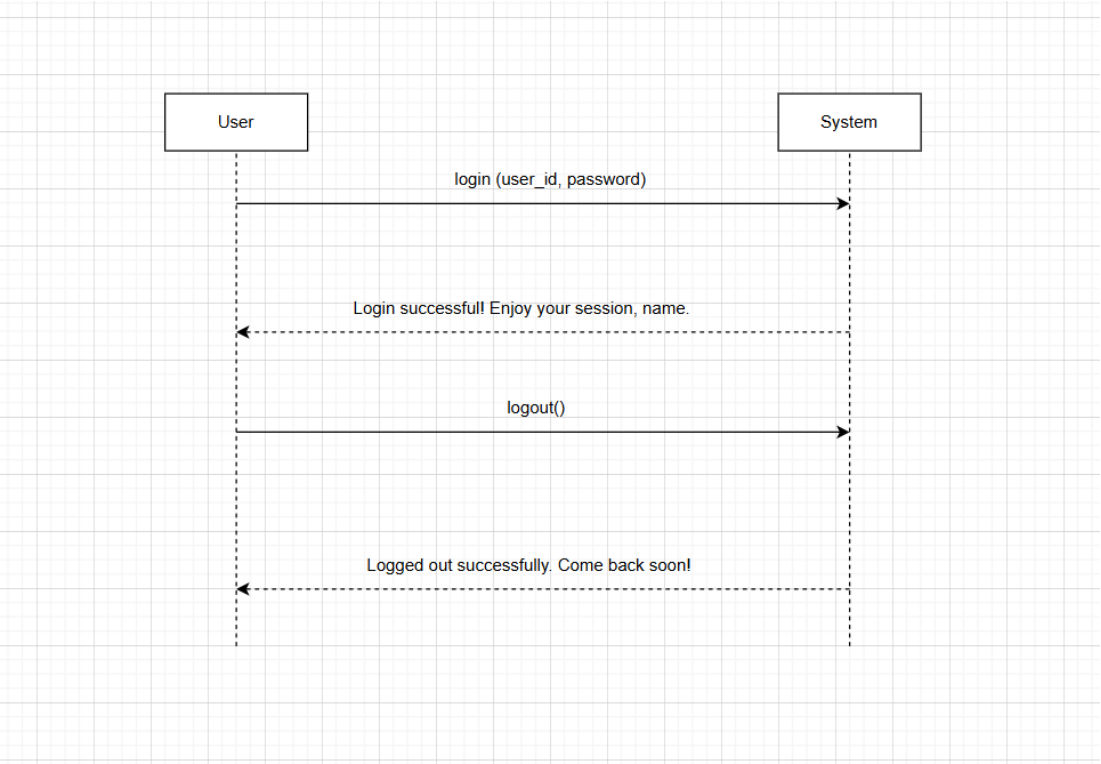


Figure 6.3: System Sequence Diagram for " ***log out***" use case

7 - State machine diagram:

1 . register new service provider

2 . ***edit service requester data***

3 . log out

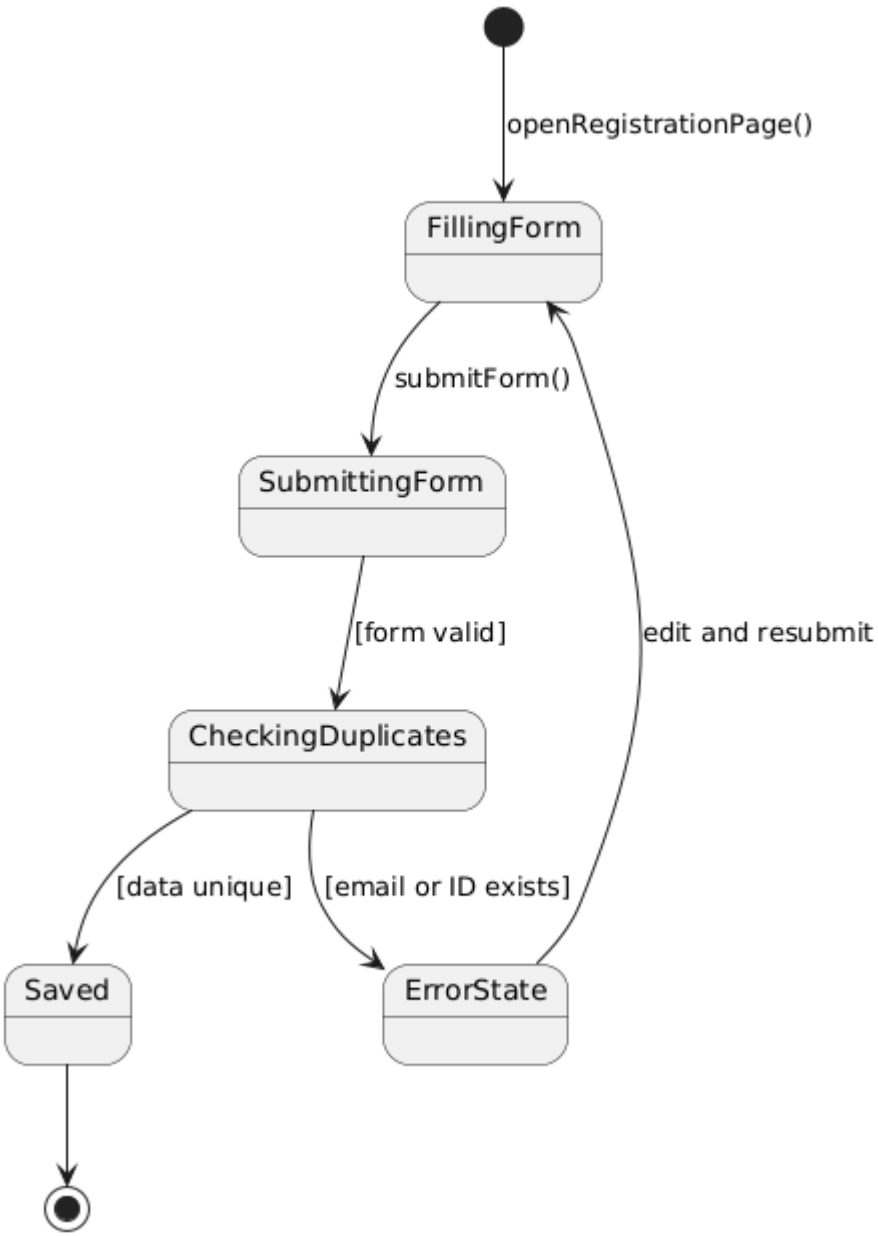


Figure 7.1: State Machine Diagram for " **register for student** " use case

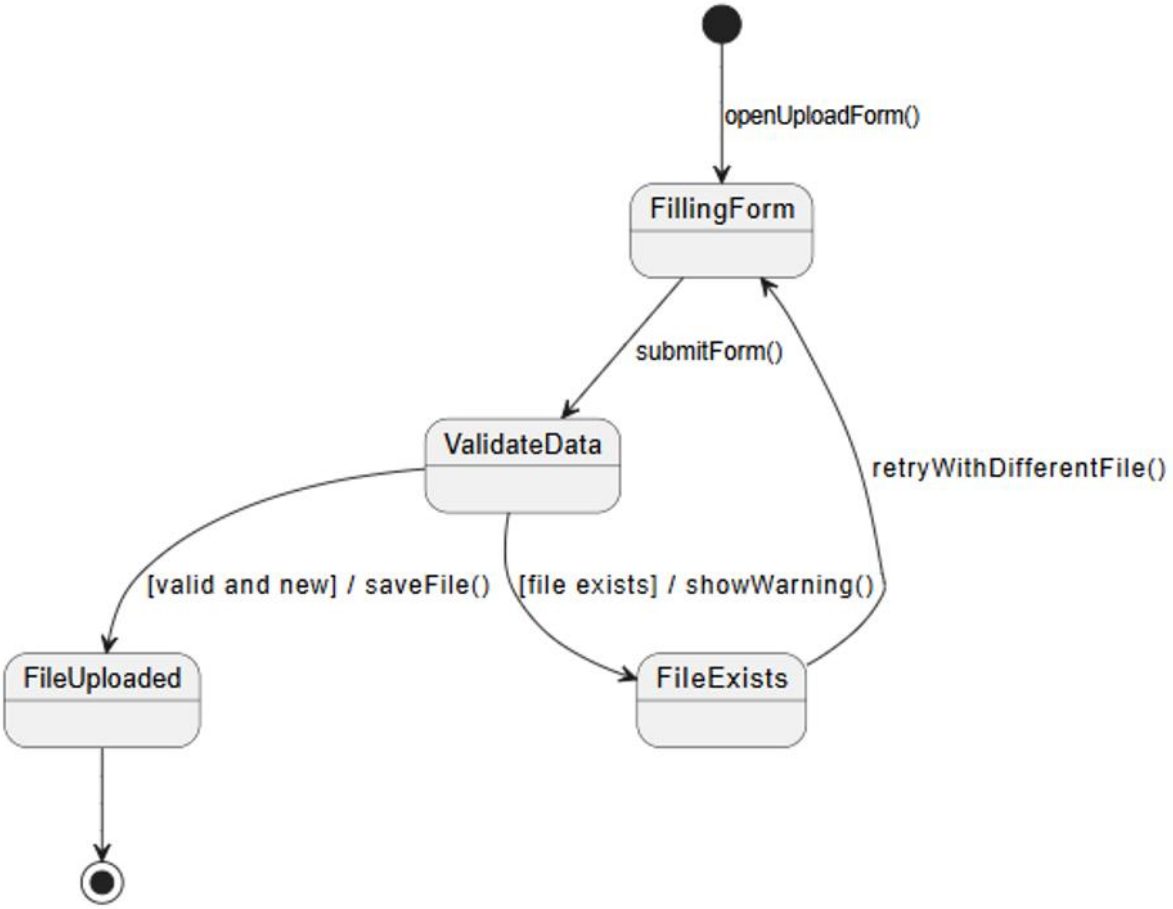


Figure 7.2: State Machine Diagram for " ***upload file by student*** " use case

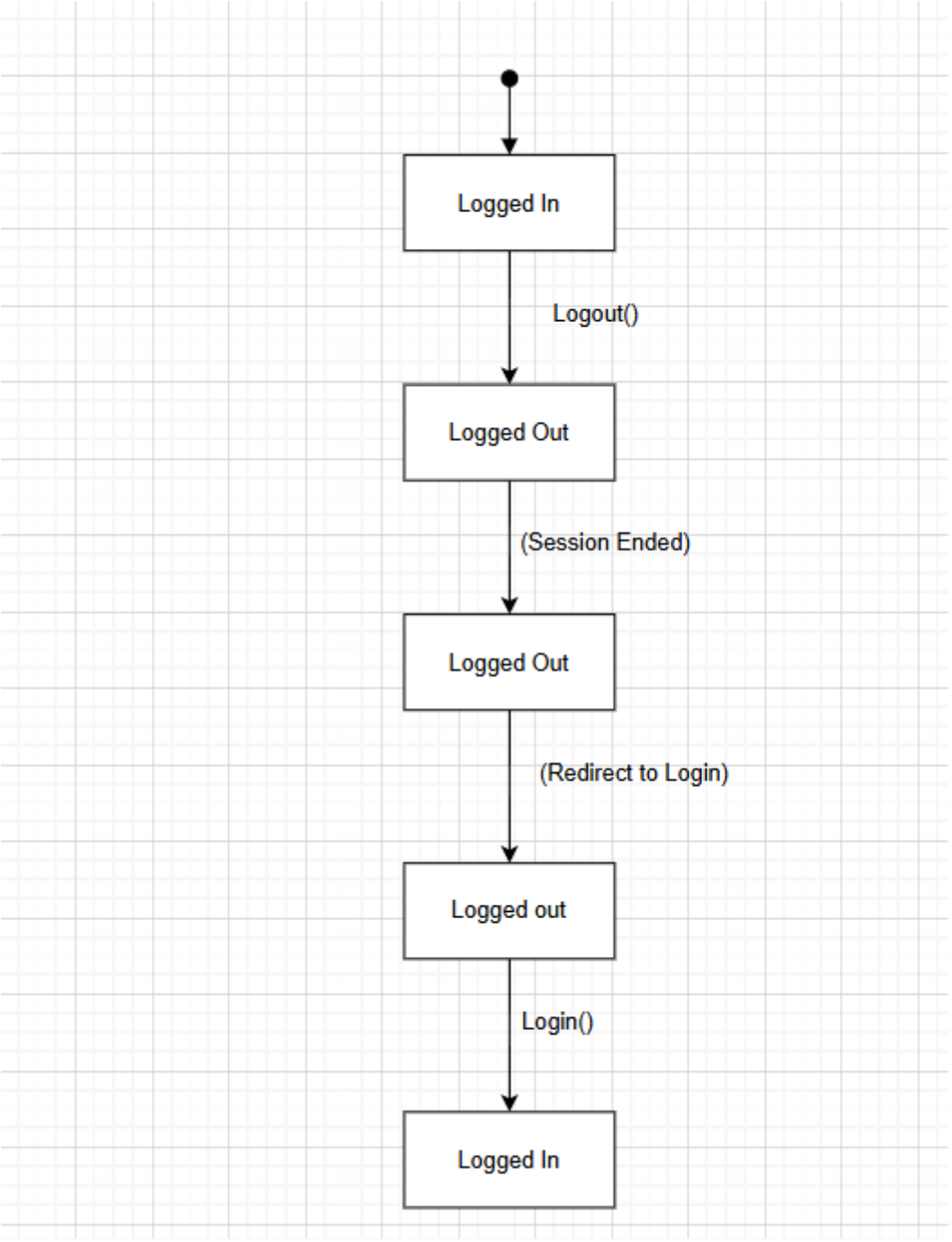


Figure 7.3: State Machine Diagram for " ***log out***" use case