Software Requirements and Design Document

**ByteVibes**

JINX

**Prepared by:**

**Laiba Raza 22i2359\_F  
Tooba 22i1357\_F  
Tooba Ali 22i0807\_F**

**Date: 11/27/2024**

**Table of Contents**

**Table of Contents ii**

**1.** **Introduction 1**

1.1 Purpose 1

1.2 Product Scope 1

1.3 Title 1

1.4 Objectives 1

1.5 Problem Statement 1

**2.** **Overall Description 1**

2.1 Product Perspective 1

2.2 Product Functions 2

2.3 List of Use Cases 2

2.4 Extended Use Cases 2

2.5 Use Case Diagram 2

**3.** **Other Nonfunctional Requirements 2**

3.1 Performance Requirements 2

3.2 Safety Requirements 2

3.3 Security Requirements 2

3.4 Software Quality Attributes 2

3.5 Business Rules 3

3.6 Operating Environment 3

3.7 User Interfaces 3

**4.** **Domain Model 3**

**5.** **System Sequence Diagram 3**

**6.** **Sequence Diagram 3**

**7.** **Class Diagram 4**

**8.** **Package Diagram 4**

**9.** **Deployment Diagram 4**

# **Introduction**

## **Purpose**

The FAST Thesis Management System (FTMS) is designed to streamline and automate the

thesis management process for students and faculty members at FAST University.

## **Product Scope**

Previously, thesis management at FAST University has primarily been conducted manually,

relying heavily on paper-based processes and communication through emails, particularly

using Gmail. By implementing the FTMS, FAST University will support academic success and

administrative effectiveness, setting a new standard for thesis management within the institution.

## 1.3 **Title**

FAST Thesis Management System

## 1.4 **Objectives**

The primary objectives of the FAST Thesis Management System (FTMS) are to:

1. Streamline Thesis Submission and Approval Processes

2. Enhance Communication and Collaboration

3. Improve Progress Tracking and Transparency

4. Integrate with Existing University Systems

5. Provide Advanced Reporting and Analytics

6. Enhance User Experience

7. Ensure Compliance with University Guidelines

8. Centralize Thesis Documentation

## 1.5 **Problem Statement**

The primary reason for choosing the FAST Thesis Management System (FTMS) project is to

address the inefficiencies and challenges posed by the current manual thesis management

process at FAST University.

Effects: The effects are particularly felt by students, who face delays in their academic

progress, and by faculty members, who experience increased administrative burdens and

difficulties in managing and tracking thesis-related tasks.

The impact: The manual process hampers the academic performance of students, causes

administrative bottlenecks, and reduces the overall productivity of faculty members.

A successful solution would be: The implementation of the FAST Thesis Management

System (FTMS) would automate and streamline the entire thesis management process. By

providing a centralized, digital platform, FTMS would ensure efficient handling of thesis

proposals, approvals, and progress tracking. The FTMS project is feasible as it leverages

existing technology and integrates with current university systems.

# **Overall Description**

## **Product Perspective** The FAST Thesis Management System (FTMS) is a new, self-contained software product developed to digitize and streamline the thesis management process for students, supervisors, and administrative staff at FAST University. This system replaces the existing manual and semi-automated methods, addressing inefficiencies such as delays in communication, disorganized record-keeping, and lack of transparency.

## FTMS operates as a standalone application but can be extended to interface with the university's student information system (SIS) for data synchronization and authentication. Its modular architecture allows for flexible management of thesis-related workflows, with the following key components:

## Student Portal: Enables students to submit thesis topics, upload proposals, and track progress.

## Supervisor Portal: Allows supervisors to manage and evaluate thesis submissions.

## Administrative Dashboard: Provides tools for administrators to oversee and manage the overall thesis process.

## 

## **Product Functions**

The FAST Thesis Management System must perform the following major functions:

1. Student Portal:
   1. Submit thesis topics and proposals for supervisor review.
   2. Upload progress reports and final documents.
   3. View feedback and approval status.
   4. Track deadlines and receive notifications.
2. Supervisor Portal:
   1. Review and approve or reject thesis topics and proposals.
   2. Provide feedback and set expectations for progress*.*
   3. Monitor individual student progress and communicate directly with students.
3. Administrative Dashboard:
   1. Manage the allocation of supervisors to students.
   2. Track system-wide statistics (e.g., number of theses in progress, completed theses).
   3. Manage deadlines, schedules, and generate progress or completion reports.

**

**3. Other Nonfunctional Requirements**

**3.1 Performance Requirements**The Thesis Management System must efficiently handle the following scenarios:

* Support at least 100 concurrent users without a noticeable decrease in response time.
* Response time for critical actions, such as submitting a thesis or generating reports, must not exceed 3 seconds under typical load conditions.
* Handle database queries for retrieving submissions and user details in less than 1 second.

**3.2 Safety Requirements**

* The system must prevent data loss by implementing regular backups and ensuring data is saved automatically during interactions.
* Safeguards should prevent accidental deletion of critical data, such as requiring confirmation steps for deletions.
* Adhere to institutional guidelines for ethical handling and storage of sensitive academic information.

**3.3 Security Requirements**

* User authentication must be enforced, using secure methods such as password hashing and role-based access control.
* Data in transit and at rest must be encrypted to protect sensitive information, such as login credentials and thesis documents.
* Regular security audits should be performed to identify and mitigate vulnerabilities.

**3.4 Software Quality Attributes**

* **Reliability:** Ensure 99.9% uptime, with minimal disruption during updates or maintenance.
* **Maintainability:** Use a modular architecture to simplify code maintenance and feature updates.
* **Usability:** Provide an intuitive interface to accommodate users with varying technical expertise.
* **Portability:** The system should be operable on multiple platforms (e.g., Windows, macOS, and Linux).

**3.5 Business Rules**

* Only registered students can submit theses.
* Supervisors are allowed to review and comment only on theses assigned to them.
* Deadlines set by administrators must be strictly enforced, with reminders sent to users periodically.
* Only administrators can assign reviewers and manage system configurations.

**3.6 Operating Environment**

* The system must operate on standard web browsers such as Chrome, Firefox, and Edge.
* The back-end will run on a Linux server using technologies like Python and PostgreSQL.
* The environment should be compatible with cloud-hosted and on-premise deployments.

**3.7 User Interfaces**

* The system must provide a clean and intuitive dashboard for students, supervisors, reviewers, and administrators.
* Include standard navigation features, such as breadcrumbs and consistent menu placement.
* Provide feedback on user actions (e.g., "Submission successful") and clear error messages.
* Adhere to accessibility standards (e.g., WCAG 2.1) to ensure inclusivity.

This structure addresses the various nonfunctional aspects comprehensively while tailoring them for a **Thesis Management System**. Let me know if you need adjustments!

## **List of Use Cases**

ACTOR:

Student

Supervisor

panel

Examiner

Admin

Use Cases

1. Submit Thesis Proposal

2. Submit Consent Form

3. Schedule Proposal Defense

4. Evaluate Proposal Defense

5. Distribute Feedback

6. Schedule Midterm Evaluation

7. Submit Midterm Evaluation

8. Distribute Midterm Feedback

9. Announce Final Report Submission

10. Submit Final Report

11. Evaluate Final Report

12. Upload Final Grades

13. Submit Revised Thesis

14. Form MSRC Panel

1. Submit Thesis Proposal

Use Case Name:

**Thesis Proposal Submission**

Scope:

Fast Thesis Management System

Level:

User Goal

Primary Actor:

Student

Stakeholders and Interests:

Student: Wants to submit the proposal.

Supervisor: Needs to review the submitted proposal.

Panel: Needs records of the submission for evaluation.

Admin: Manages system functions and ensures proper operation.

Preconditions:

Students must be registered for the thesis.

Postconditions:

The proposal is successfully submitted and awaits supervisor approval.

Main Success Scenario:

1. Students log into the system.

2. Student navigates to the proposal submission page.

3. Student uploads the proposal document.

4. System validates the document format and content.

5. System submits the proposal and notifies the supervisor.

6. Student receives a confirmation email.

Use Case Name:

**Consent Form Submission**

Scope:

Fast Thesis Management System

Level:

User Goal

Primary Actor:

Student

Stakeholders and Interests:

Student: Wants to submit the consent form as part of the thesis process.

Supervisor: Needs to review and approve the submitted consent forms to proceed with the thesis.

Panel: May need to access records of the consent form submission for review.

Examiner: May require the consent form as part of the overall thesis assessment.

Preconditions:

The thesis proposal must be submitted before the consent form can be uploaded.

Postconditions:

The consent form is successfully submitted and linked with the thesis proposal.

The supervisor is notified for review and approval.

The form becomes part of the student’s thesis records for future review by the panel and

examiner.

Main Success Scenario:

1. Students log into the system.

2. Student navigates to the consent form submission page.

3. Student uploads the consent form document.

4. System validates the document format and content.

5. System submits the consent form and notifies the supervisor.

6. Supervisor reviews and approves the consent form.

7. Student receives a confirmation email.

3.Proposal Defense Scheduling:

Use Case Name:

**Proposal Defense Scheduling**

Scope:

Fast Thesis Management System

Level:

User Goal

Primary Actor:

Supervisor

Stakeholders and Interests:

Student: Wants to know the defense schedule to prepare for the defense.

Supervisor: Needs to attend the proposal defense and schedule the meeting.

Panel: Needs to attend the defense as evaluators.

Examiner: May participate in the defense process and should be informed of the

schedule.

Preconditions:

The thesis proposal and consent form must be submitted and approved.

Postconditions:

The defense schedule is successfully created and communicated to the student, panel,

and examiner.

The schedule is recorded in the system for future reference.

Main Success Scenario:

1. Supervisor logs into the system.

2. Supervisor navigates to the defense scheduling page.

3. Supervisor selects the proposal and sets the defense date.

4. System validates the proposed schedule for conflicts.

5. System sends notifications to the student, panel, and examiner.

6. Schedule is updated and stored in the system.  
  
  
4. Proposal Defense Evaluation

Use Case Name:

**Proposal Defense Evaluation**

Scope:

Fast Thesis Management System

Level:

User Goal

Primary Actor:

Examiner

Stakeholders and Interests:

Examiner: Evaluates the proposal defense.

Student: Wants to know the evaluation results.

Supervisor: Reviews evaluation feedback.

Panel: May need access to evaluation data.

Preconditions:

The proposal defense must be scheduled and completed.

Postconditions:

The evaluation results are recorded in the system and shared with relevant

stakeholders.

Main Success Scenario:

1. Examiner logs into the system.

2. Examiner navigates to the evaluation page.

3. Examiner selects the student’s proposal and enters evaluation comments.

4. System validates the evaluation for completeness.

5. System records the evaluation in the database.

6. Notifications are sent to the student and supervisor.

5. Feedback Distribution

Use Case Name:

Feedback Distribution

Scope:

Fast Thesis Management System

Level:

User Goal

Primary Actor:

Supervisor

Stakeholders and Interests:

Student: Wants to receive feedback.

Supervisor: Needs to review the feedback before distribution.

Panel: Needs to be aware of the feedback.

Preconditions:

The proposal defense evaluation must be completed.

Postconditions:

Feedback is successfully sent to the student and recorded in the system.

Main Success Scenario:

1. Supervisor logs into the system.

2. Supervisor navigates to the feedback distribution page.

3. Supervisor reviews the evaluation results and generates feedback.

4. System sends feedback to the student.

5. Feedback is recorded in the system.

6. Midterm Evaluation Submission

Use Case Name:

**Midterm Evaluation Submission**

Scope:

Fast Thesis Management System

Level:

User Goal

Primary Actor:

Supervisor

Stakeholders and Interests:

Supervisor: Submits the midterm evaluation.

Student: Wants to know the midterm evaluation results.

Panel: May need access to the evaluation results.

Preconditions:

The midterm evaluation must be scheduled.

Postconditions:

The midterm evaluation is submitted and recorded in the system.

Main Success Scenario:

1. Supervisor logs into the system.

2. Supervisor navigates to the evaluation submission page.

3. Supervisor selects the student and enters evaluation comments.

4. System validates the evaluation.

5. System records the evaluation in the database.

6. Notifications are sent to the student.

7. Thesis Final Submission

Use Case Name:

**Thesis Final Submission**

Scope:

Fast Thesis Management System

Level:

User Goal

Primary Actor:

Student

Stakeholders and Interests:

Student: Wants to submit the final thesis.

Supervisor: Reviews and approves the final thesis submission.

Panel: Needs to evaluate the final thesis.

Preconditions:

The thesis proposal, midterm evaluation, and final defense must be completed.

Postconditions:

The final thesis is submitted and recorded in the system.

Main Success Scenario:

1. Student logs into the system.

2. Student navigates to the final submission page.

3. Student uploads the final thesis document.

4. System validates the document for completeness and format.

5. System submits the final thesis and notifies the supervisor.

6. Student receives a confirmation email.

Use Case Name:

**Schedule Midterm Evaluation**

Scope:

Fast Thesis Management System

Level:

User Goal

Primary Actor:

Supervisor

Stakeholders and Interests:

Supervisor: Wants to schedule a midterm evaluation for students.

Student: Needs to know the evaluation schedule to prepare.

Panel: Requires the schedule for attendance and coordination.

Admin: Manages the system and ensures accurate scheduling.

Preconditions:

Supervisor must be logged into the system.

Postconditions:

The midterm evaluation is scheduled, and notifications are sent to students and panel

members.

Main Success Scenario:

1. Supervisor logs into the system.

2. Supervisor navigates to the evaluation scheduling page.

3. Supervisor selects the date and time for the midterm evaluation.

4. System validates the chosen date and time.

5. System saves the schedule and notifies all stakeholders.

Use Case Name:

**Submit Midterm Evaluation**

Scope:

Fast Thesis Management System

Level:

User Goal

Primary Actor:

Student

Stakeholders and Interests:

Student: Wants to submit the midterm evaluation.

Supervisor: Needs to review the evaluation submission.

Admin: Ensures that the submission process runs smoothly.

Preconditions:

Midterm evaluation must be scheduled.

Postconditions:

The midterm evaluation is successfully submitted and available for review.

Main Success Scenario:

1. Students log into the system.

2. Student navigates to the midterm evaluation submission page.

3. Student uploads the evaluation document.

4. System validates the document format and content.

5. System submits the evaluation and notifies the supervisor.

6. Student receives a confirmation email.

Use Case Name:

**Distribute Midterm Feedback**

Scope:

Fast Thesis Management System

Level:

User Goal

Primary Actor:

Supervisor

Stakeholders and Interests:

Supervisor: Wants to provide feedback to students.

Student: Needs feedback for improvement.

Admin: Manages the feedback distribution process.

Preconditions:

Midterm evaluations must be submitted and reviewed.

Postconditions:

Feedback is successfully distributed to all students.

Main Success Scenario:

1. Supervisor logs into the system.

2. Supervisor navigates to the feedback distribution page.

3. Supervisor selects the evaluations to provide feedback on.

4. Supervisor writes feedback comments.

5. System saves the feedback and notifies the respective students.

Use Case Name:

**Announce Final Report Submission**

Scope:

Fast Thesis Management System

Level:

User Goal

Primary Actor:

Admin

Stakeholders and Interests:

Admin: Wants to inform students about final report submission deadlines.

Student: Needs to be aware of submission requirements and deadlines.

Supervisor: Requires clear communication regarding submission timelines.

Preconditions:

Final report guidelines must be defined.

Postconditions:

Announcement is successfully made to all students.

Main Success Scenario:

1. Admin logs into the system.

2. Admin navigates to the announcement section.

3. Admin creates an announcement for final report submission.

4. System saves the announcement and distributes it to all students.

Use Case Name:

**Submit Final Report**

Scope:

Fast Thesis Management System

Level:

User Goal

Primary Actor:

Student

Stakeholders and Interests:

Student: Wants to submit the final report for evaluation.

Supervisor: Needs to review the final report.

Admin: Manages the submission process.

Preconditions:

Students must be aware of final report submission guidelines.

Postconditions:

Final report is successfully submitted and awaiting evaluation.

Main Success Scenario:

1. Students log into the system.

2. Students navigate to the final report submission page.

3. Students upload the final report document.

4. System validates the document format and content.

5. System submits the final report and notifies the supervisor.

6. Student receives a confirmation email.

Use Case Name:

**Evaluate Final Report**

Scope:

Fast Thesis Management System

Level:

User Goal

Primary Actor:

Supervisor

Stakeholders and Interests:

Supervisor: Wants to evaluate the submitted final reports.

Student: Needs feedback on their final report.

Admin: Manages the evaluation process.

Preconditions:

Final reports must be submitted and accessible.

Postconditions:

Final reports are evaluated, and feedback is recorded.

Main Success Scenario:

1. Supervisor logs into the system.

2. Supervisor navigates to the evaluation page.

3. Supervisor selects a final report to evaluate.

4. Supervisor reviews the report and writes feedback.

5. System saves the evaluation and notifies the student.

Use Case Name:

**Upload Final Grades**

Scope:

Fast Thesis Management System

Level:

User Goal

Primary Actor:

Admin

Stakeholders and Interests:

Admin: Wants to upload final grades for students.

Supervisor: Needs to verify that grades are accurately recorded.

Student: Expects to see their final grades.

Preconditions:

Final evaluations must be completed.

Postconditions:

Final grades are successfully uploaded and visible to students.

Main Success Scenario:

1. Admin logs into the system.

2. Admin navigates to the grades upload section.

3. Admin uploads the final grades file.

4. System validates the format of the uploaded file.

5. System saves the grades and notifies all students.

Use Case Name:

**Submit Revised Thesis**

Scope:

Fast Thesis Management System

Level:

User Goal

Primary Actor:

Student

Stakeholders and Interests:

Student: Wants to submit a revised thesis based on feedback.

Supervisor: Needs to review the revised thesis.

Admin: Manages the submission process.

Preconditions:

Students must have received feedback on the initial thesis submission.

Postconditions:

The revised thesis is successfully submitted and available for review.

Main Success Scenario:

1. Students log into the system.

2. Student navigates to the revised thesis submission page.

3. Student uploads the revised thesis document.

4. System validates the document format and content.

5. System submits the revised thesis and notifies the supervisor.

6. Student receives a confirmation email.

Use Case Name:

**Form MSRC Panel**

**Scope:**

Fast Thesis Management System

Level:

User Goal

Primary Actor:

Admin

Stakeholders and Interests:

Admin: Wants to form a panel for evaluations.

Supervisor: Needs to be part of the evaluation process.

Student: Requires transparency in the evaluation process.

Preconditions:

Criteria for forming the MSRC panel must be established.

Postconditions:

The MSRC panel is formed and notified of their roles.

Main Success Scenario:

1. Admin logs into the system.

2. Admin navigates to the MSRC panel formation page.

3. Admin selects members for the panel based on criteria.

4. System saves the panel formation and notifies the selected members.

## **Extended Use Cases**

**1. Submit Thesis Proposal**Extensions:

3a. Invalid Document Format: System prompts the student to upload in the correct

format.

5a. Supervisor Not Available: System queues the proposal for review when the

supervisor becomes available.

**2. Submit Consent Form**

Extensions:

3a. Invalid Document Format: System prompts the student to upload the document in

the correct format.

5a. Supervisor Not Available: System queues the consent form for review when the

supervisor becomes available.

6a. Supervisor Requests Changes: If the supervisor requests modifications, the

the student is notified to revise and resubmit the form.

**3.Proposal Defense Scheduling**Extensions:

3a. Date Conflict: The system detects a scheduling conflict (e.g., panel or examiner

availability) and prompts the supervisor to select a different date.

5a. Notification Error: The system logs the error if notifications fail and retries sending

them.

**4. Proposal Defense Evaluation**Extensions:

3a. Missing Information: System prompts the examiner to complete all required fields.

5a. Database Error: System logs the error and notifies the administrator for resolution.

**5. Feedback Distribution**  
Extensions:

4a. Email Delivery Failure: The system retries sending the feedback and logs the

error.

**6. Midterm Evaluation Submission**Extensions:

3a. Missing Information: System prompts the supervisor to complete all required

fields.

5a. Database Error: System logs the error and notifies the administrator.

**6. Midterm Evaluation Submission**Extensions:

3a. Missing Information: System prompts the supervisor to complete all required

fields.

5a. Database Error: System logs the error and notifies the administrator.

**7. Thesis Final Submission**Extensions:

3a. Invalid Document Format: System prompts the student to upload the document in

the correct format.

5a. Supervisor Not Available: System queues the thesis for review when the supervisor

becomes available.

Use Case Name:

Schedule Midterm Evaluation

Extensions:

3a. Conflicting Schedule: System prompts the supervisor to choose a different time if

there is a conflict.

4a. Invalid Date/Time: System notifies the supervisor to select a valid date/time.

Use Case Name:

Submit Midterm Evaluation

Extensions:

3a. Invalid Document Format: System prompts the student to upload in the correct

format.

5a. Supervisor Not Available: System queues the evaluation for review when the

supervisor becomes available.

Use Case Name:

Distribute Midterm Feedback

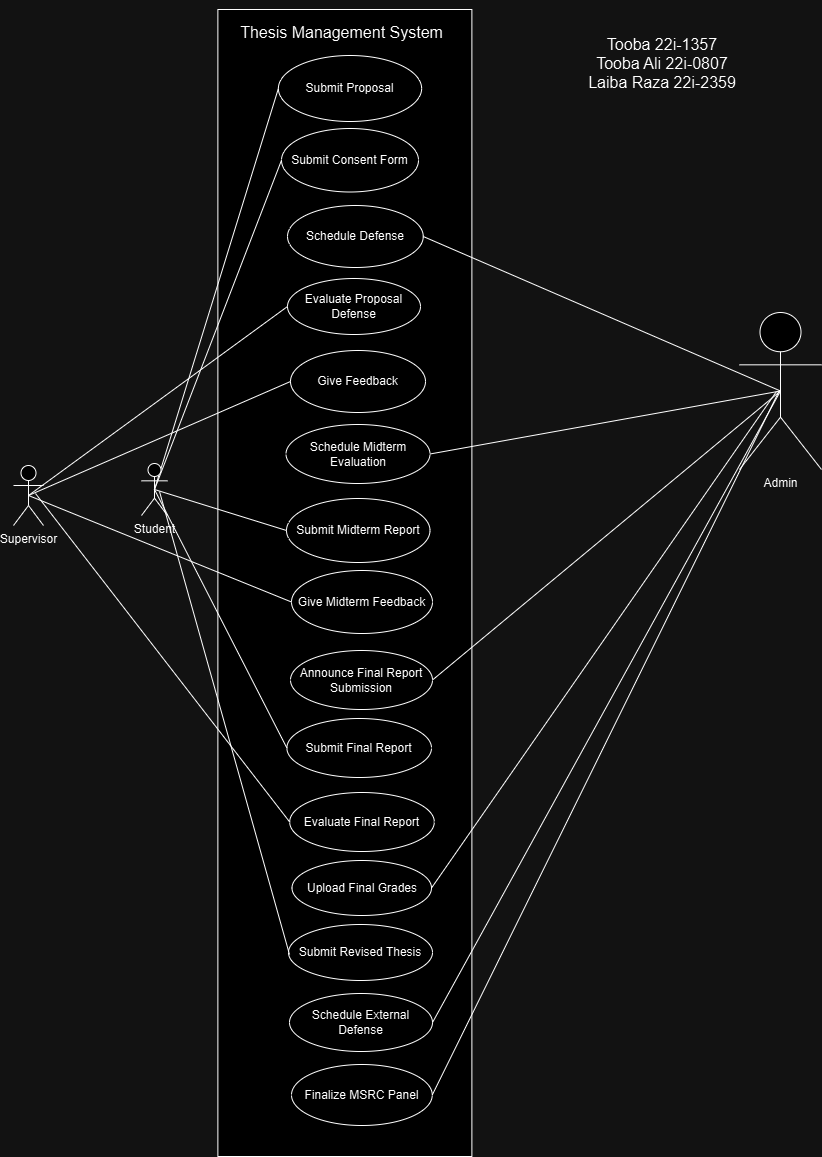
Extensions:

4a. Missing Evaluation: System notifies the supervisor if an evaluation is missing for a

student.

5a. Notification Failure: System logs an error if notifications fail to send.

## **Use Case Diagram**



# **Other Nonfunctional Requirements**

## **Performance Requirements**

* System Response Time: The system must respond to user actions (e.g., form submissions, file uploads) within 3 seconds under normal load conditions (up to 100 concurrent users).
* Concurrent Users: The system must support at least 200 simultaneous users without performance degradation.
* Data Handling: The system should process and store files (e.g., thesis documents) up to 100MB per file within 5 seconds.
* Availability: The system should maintain 99.9% uptime during academic hours (8 AM - 10 PM).
* Scalability: The system must handle a 20% increase in user load during peak periods (e.g., thesis submission deadlines) without significant performance degradation.

## **Safety Requirements**

1. Authentication: Users must log in using their university-provided credentials (e.g., email ID and password).
2. Role-Based Access Control (RBAC): The system must enforce strict role-based permissions:
   1. Students can access their submissions and supervisor feedback.
   2. Supervisors can access only the students assigned to them*.*
   3. Administrators can access and manage all system data*.*
3. Data Encryption: All sensitive data, including personal information and thesis documents, must be encrypted in transit (using TLS) and at rest.
4. Session Timeout: User sessions should automatically log out after 20 minutes of inactivity.
5. Audit Logs: The system must maintain logs of all user actions for 6 months for security audits*.*

## **Security Requirements**

1. Authentication: Users must log in using their university-provided credentials (e.g., email ID and password).
2. Role-Based Access Control (RBAC): The system must enforce strict role-based permissions:
   1. Students can access their submissions and supervisor feedback*.*
   2. Supervisors can access only the students assigned to them.
   3. Administrators can access and manage all system data.
3. Data Encryption: All sensitive data, including personal information and thesis documents, must be encrypted in transit (using TLS) and at rest.
4. Session Timeout: User sessions should automatically log out after 20 minutes of inactivity*.*
5. Audit Logs: The system must maintain logs of all user actions for 6 months for security audits.

## **Software Quality Attributes**

1. Usability: The system should have an intuitive interface to minimize the learning curve for students and supervisors, with help documentation readily accessible.
2. Reliability: The system must function without critical failures for at least 6 months between maintenance updates.
3. Maintainability: System updates and maintenance should not exceed 2 hours of downtime.
4. Interoperability: The system must integrate seamlessly with the university’s existing Student Information System (SIS).
5. Portability: The system must run on web browsers across different operating systems (Windows, macOS, Linux).
6. Testability: The system should include automated test scripts for critical functions (e.g., form submissions, report generation) to ensure seamless deployments.
7. Robustness: The system should handle incorrect user inputs gracefully and provide meaningful error messages.

## **Business Rules**

1. Only students enrolled in the thesis semester can submit topics and upload documents.
2. Supervisors can approve or reject student submissions but cannot modify the submissions directly*.*
3. Only administrators can assign students to supervisors and modify deadlines.
4. Students cannot make edits to their submissions after supervisor approval*.*
5. Deadlines are enforced strictly; late submissions require explicit administrative approval.

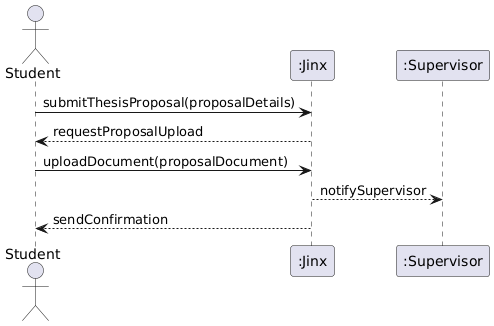
## **Operating Environment**

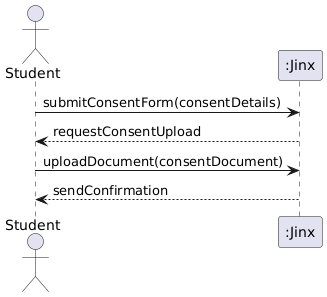
1. Hardware Platform: Local client machines with:
   1. Processor: Dual-core CPU (2 GHz or faster).
   2. RAM: 4 GB (8 GB recommended).
   3. Storage: 1 GB free disk space*.*
2. Operating System:
   1. Windows 10 or later
   2. macOS 11 (Big Sur) or later
   3. Linux (with JDK 17+)*.*
3. Development Environment:
   1. JavaFX for UI development*.*
   2. IntelliJ IDEA as the primary IDE*.*
4. Software Dependencies*:*
   1. Java Development Kit (JDK) 17 or higher.
   2. SQL for local database management.

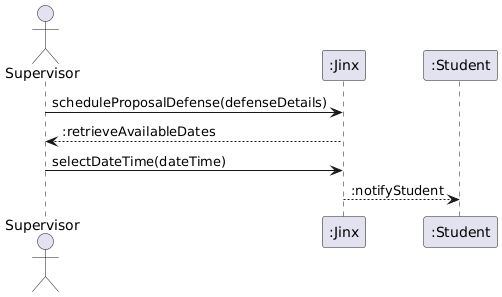
## **User Interfaces**

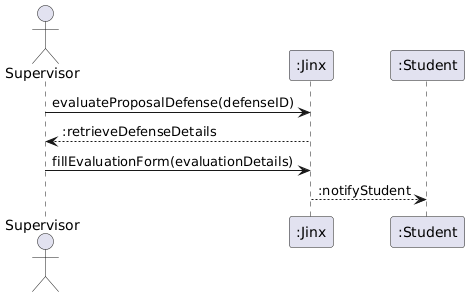
# **Domain Model**

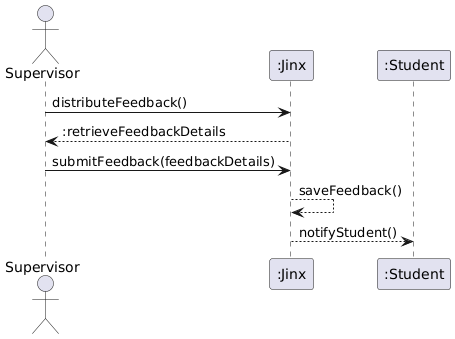
# **System Sequence Diagram**

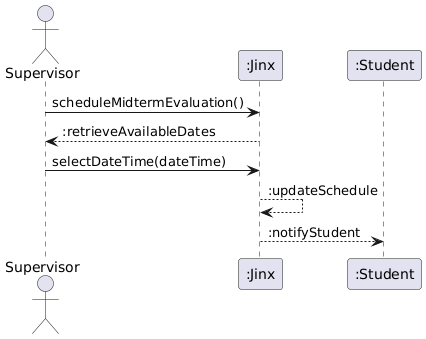


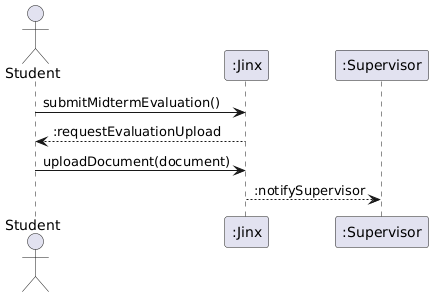


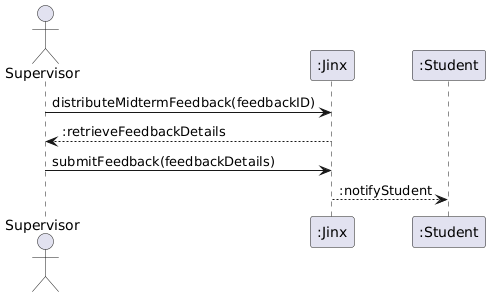


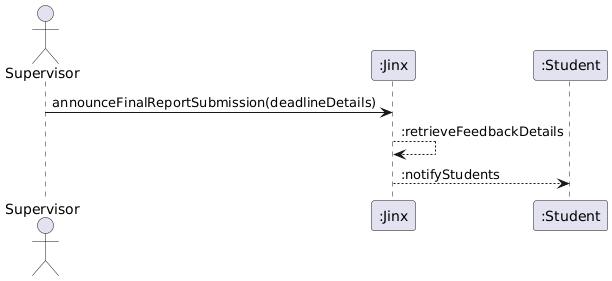


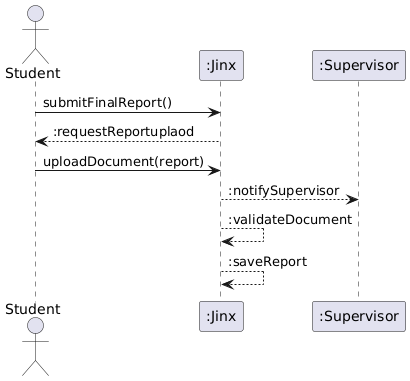


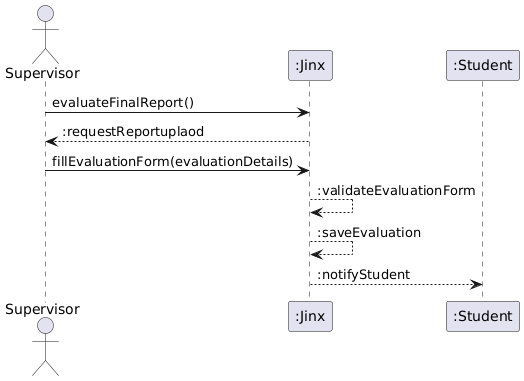


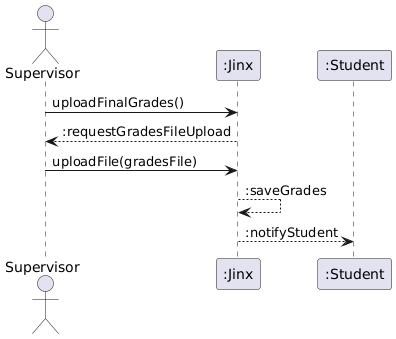


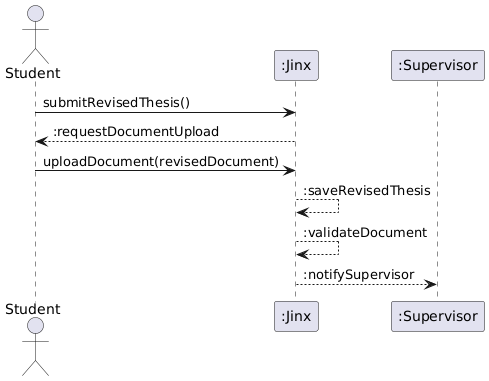


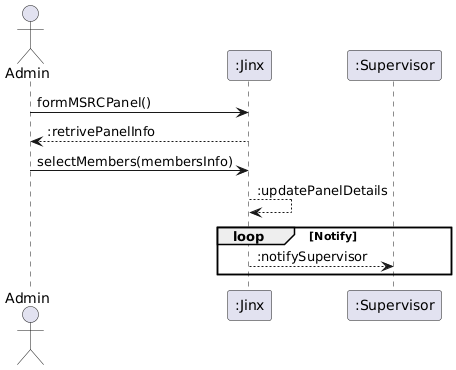




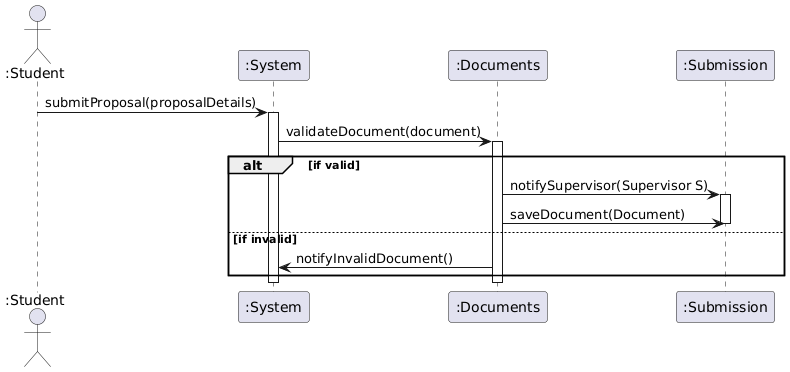


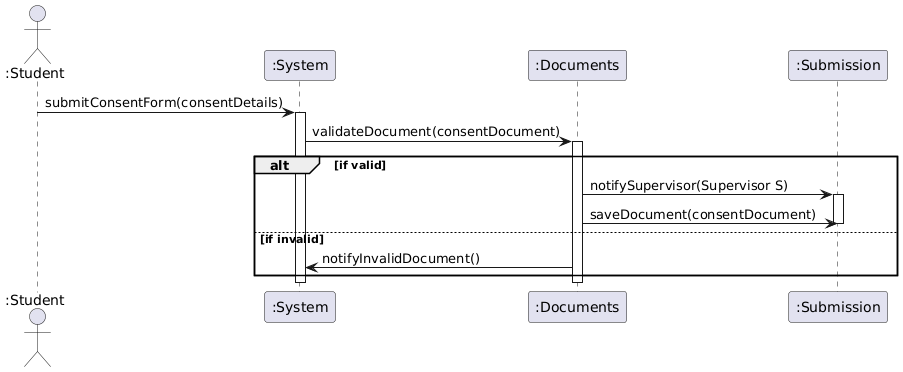


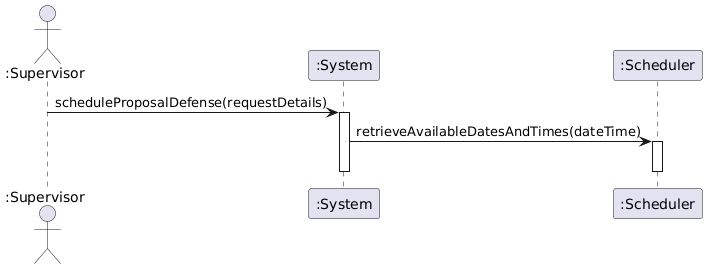


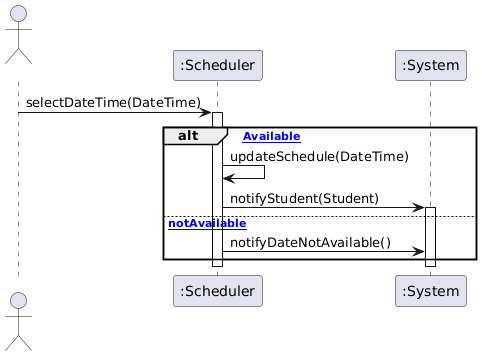


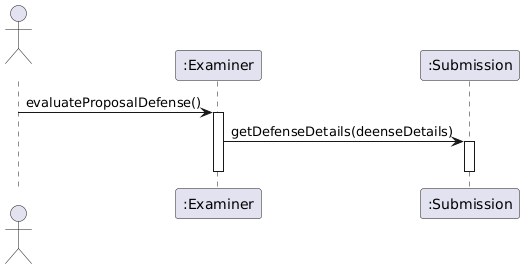
# **Sequence Diagram**

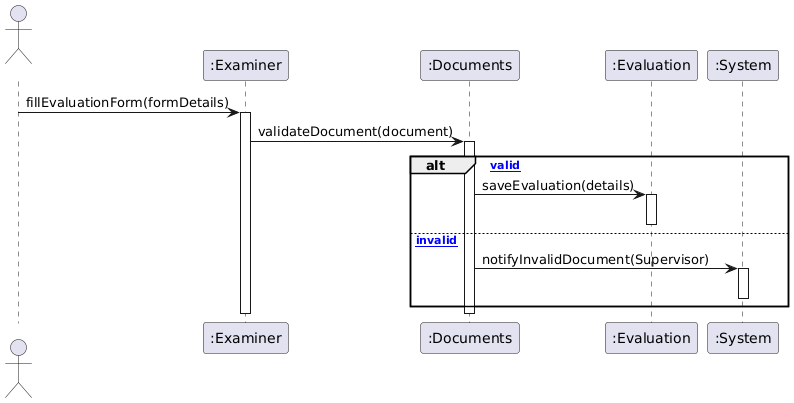


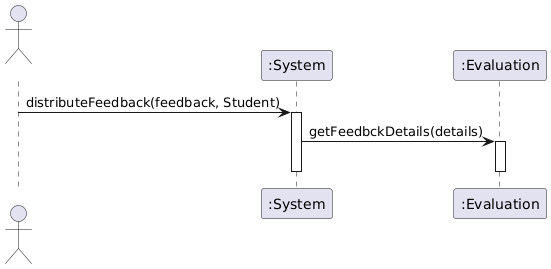


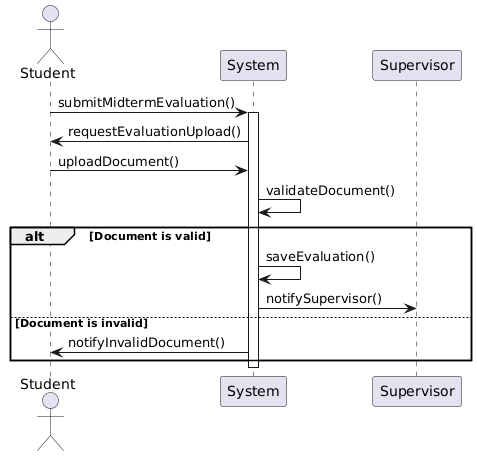


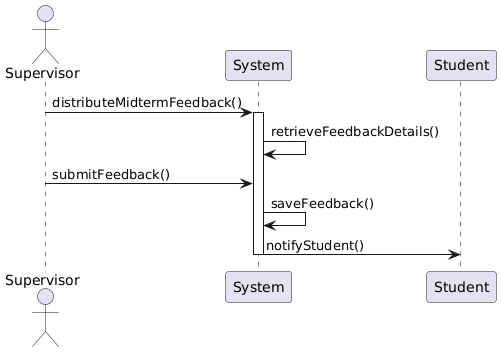


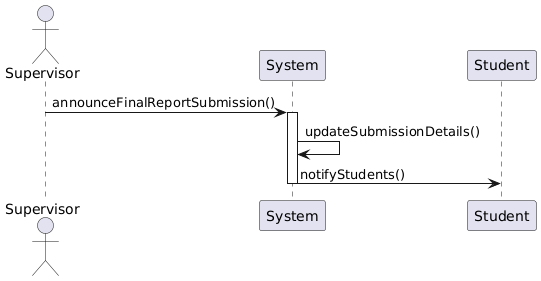


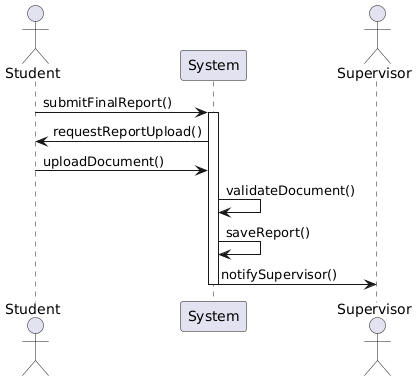


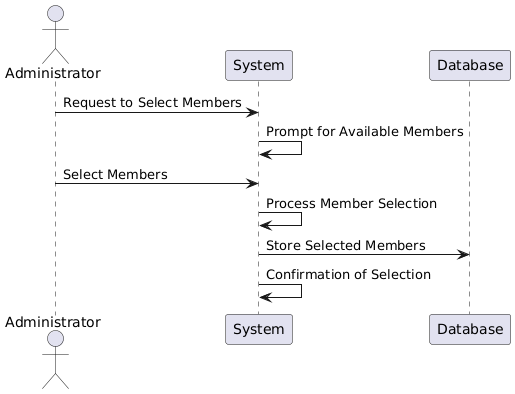


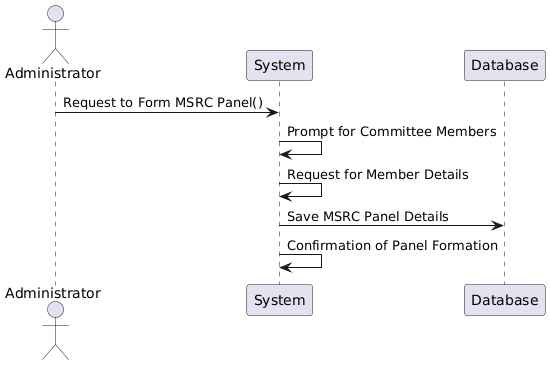


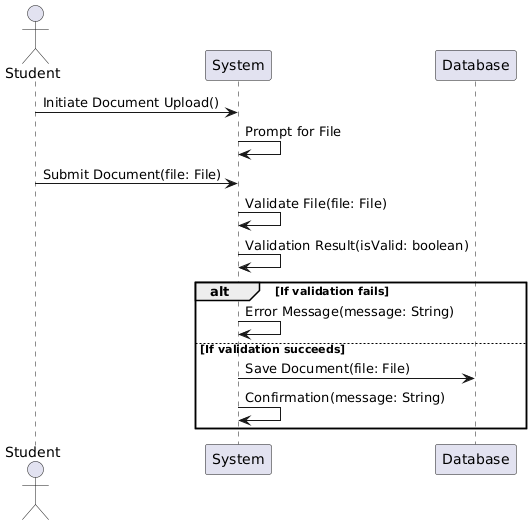


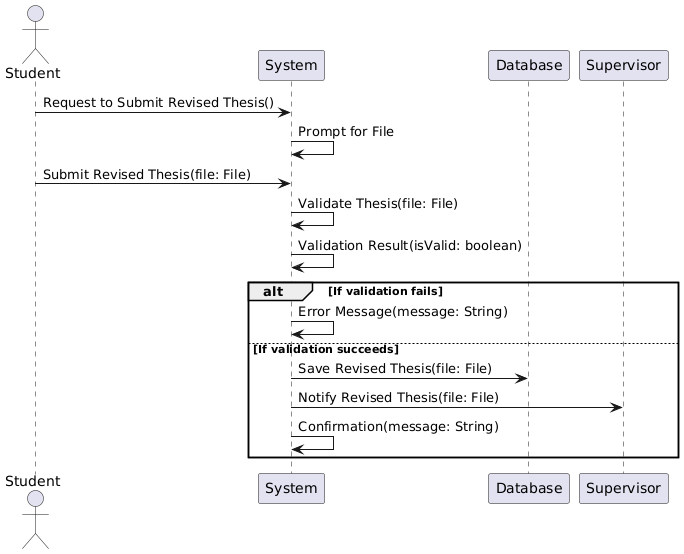


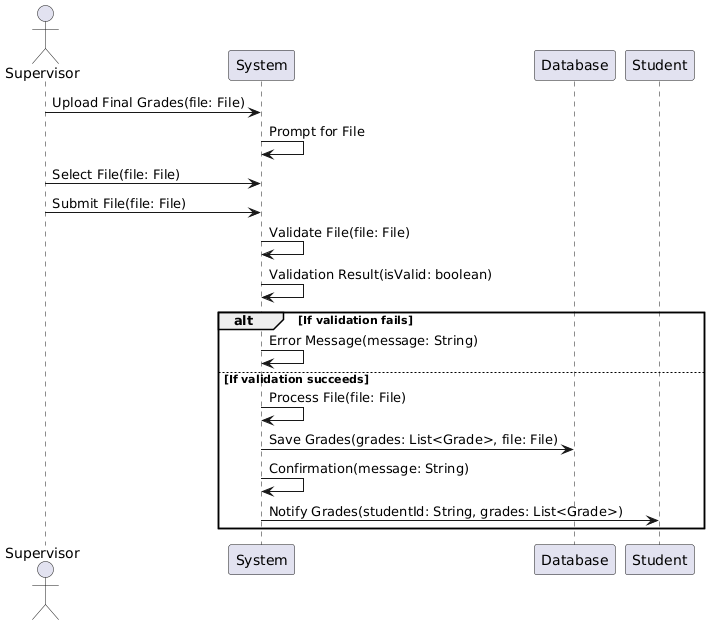












A diagram of a system

Description automatically generated

A diagram of a system

Description automatically generated

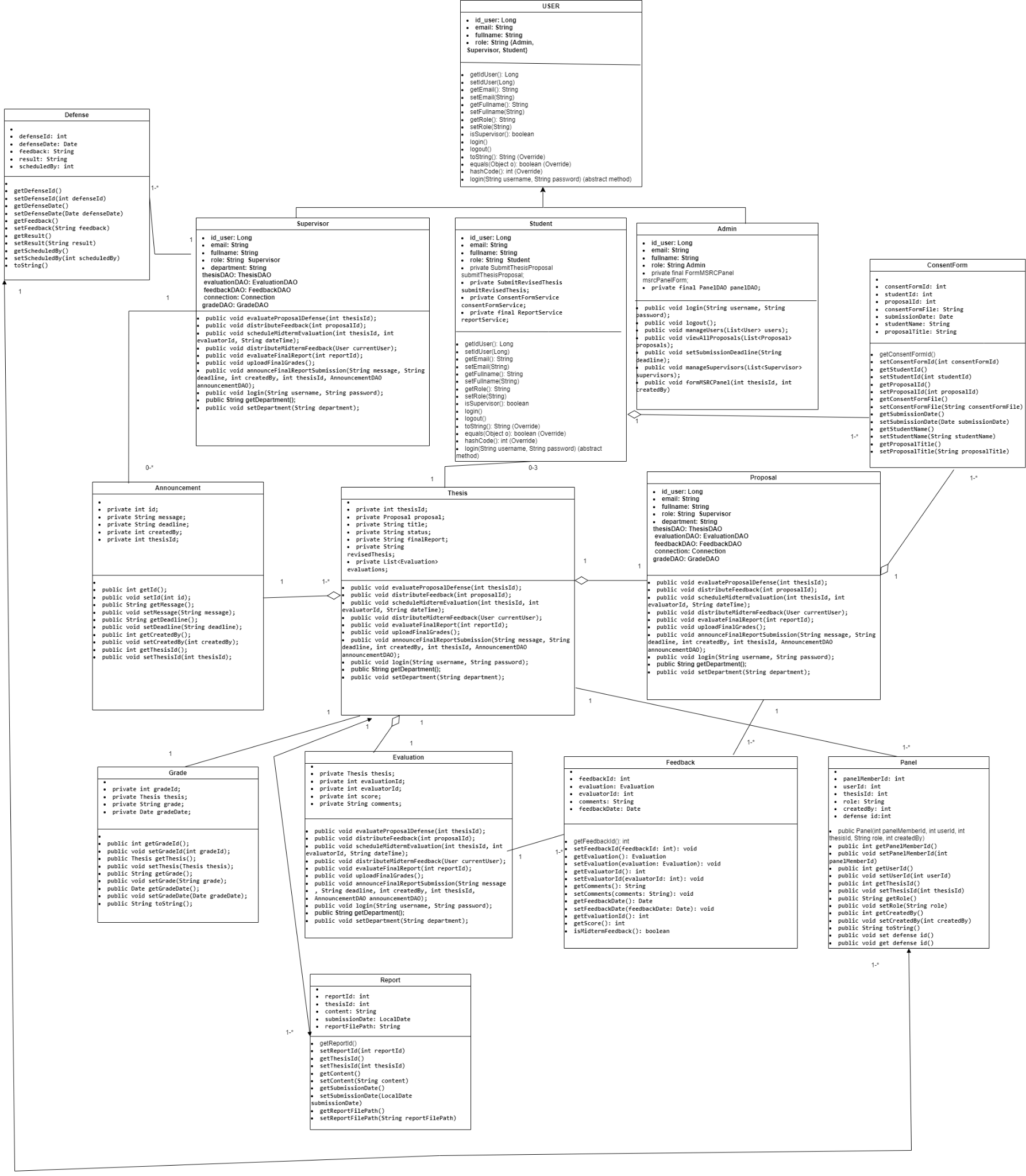
A diagram of a system

Description automatically generated

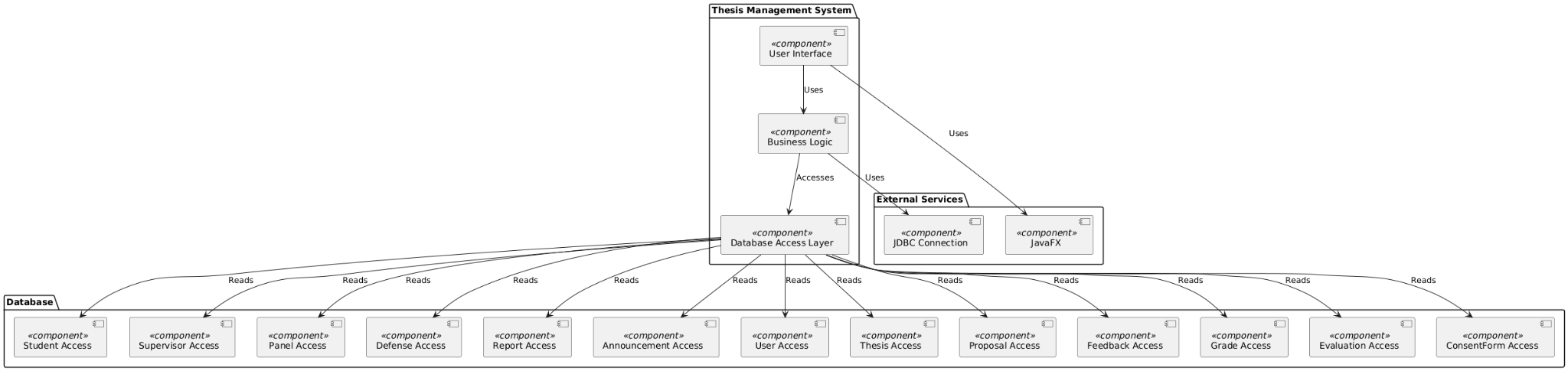
A diagram of a computer diagram

Description automatically generated

# **Class Diagram**



# **Component Diagram**

1. 

# **Package Diagram**

# **Deployment Diagram**

