

Y2 SADT PROJECT

Student Hub

Group:

[C00296913] Reanielle Patrick José Ledesma Broas

[C00297987] Eric Palm

Contents

Student User Stories	5
Professor User Stories	5
System Admin User Stories	6
Log-In	9
Manage User Permissions	10
View Grades	13
View Timetable	14
Upload Notes	15
Upload Grades	16
Update Timetable	17
Login	20
Manage user perm	21
View notes	22
Upload notes	23
View Grades	24
Upload Grades	25
View Timetable	26
Update Timetable	27

Checklist

- ☒ Summary
- ☒ ~~User Stories~~
- ☒ ~~Use Case Diagram~~
- ☒ ~~Use Case Scenarios~~
 - ☒ ~~Correlate Updated Class Diagram to Use Case~~
- ☒ ~~Class Diagram~~
- ☒ ~~Sequence Diagram~~
- ☒ ~~Sequence Table~~
- ☒ ~~Test Case Scenario~~

Summary

Status: **COMPLETE**

The Student Hub is a platform designed to help students, professors, and system administrators manage important academic resources in one place. It allows users to easily access timetables, class notes, and grades, making it simpler for everyone to stay organized and up-to-date.

Students can use the platform to view their lecture and tutorial schedules, access notes for their courses, and check their grades for exams and assignments. Professors can upload materials like class notes, enter student grades, and manage schedules for the courses they teach. System administrators handle tasks like creating and updating timetables, ensuring that the information in the system is accurate and easy to find.

The Student Hub is built to make academic life smoother by automating common tasks, such as organizing timetables and sharing grades. It reduces the amount of manual work for professors and admins while giving students quick access to the information they need to succeed. The system is user-friendly and designed to keep everything up-to-date so that everyone can rely on it.

By bringing all these features together, the Student Hub improves communication and helps students and professors focus more on learning and teaching, rather than spending time on administrative tasks.

User Stories

Status: **COMPLETE**

Student User Stories

As a [Student], I need to [Log-In], so that I can access all of my course materials, grades, and timetables.

As a [Student], I need to [View Notes], so that I can look over what I have studied and know the content of a specific course I study, that way I will be better prepared for future or current exams and assignments.

As a [Student], I need to [View Grades], so that I can see how well I have performed academically in different subjects or exams.

As a [Student], I need to [View Timetable], so that I can see when I have a lecture or tutorial throughout the day or week through a schedule.

Professor User Stories

As a [Professor], I need to [Log in to the Student Hub], so that I can securely access and manage course content, grades, and student records.

As a [Professor], I need to [View Notes], so that I can check what I have uploaded and shared with my students, and to also see the content of the course I teach.

As a [Professor], I need to [View Grades], so that I can check what I have uploaded and so that I can monitor their progress and provide feedback or adjustments as needed.

As a [Professor], I need to [View Timetable], so that I can verify that the information given is correct, and so that I view my schedule of when I'll be giving a lecture or tutorial throughout the day and week.

As a [Professor], I need to [Update Timetable], so that students always have the most accurate information about when their lectures and tutorials are scheduled.

As a [Professor], I need to [Upload Notes], so that I can provide study materials and lecture content to my students.

As a [Professor], I need to [Upload Grades], so that students can track their progress in my course.

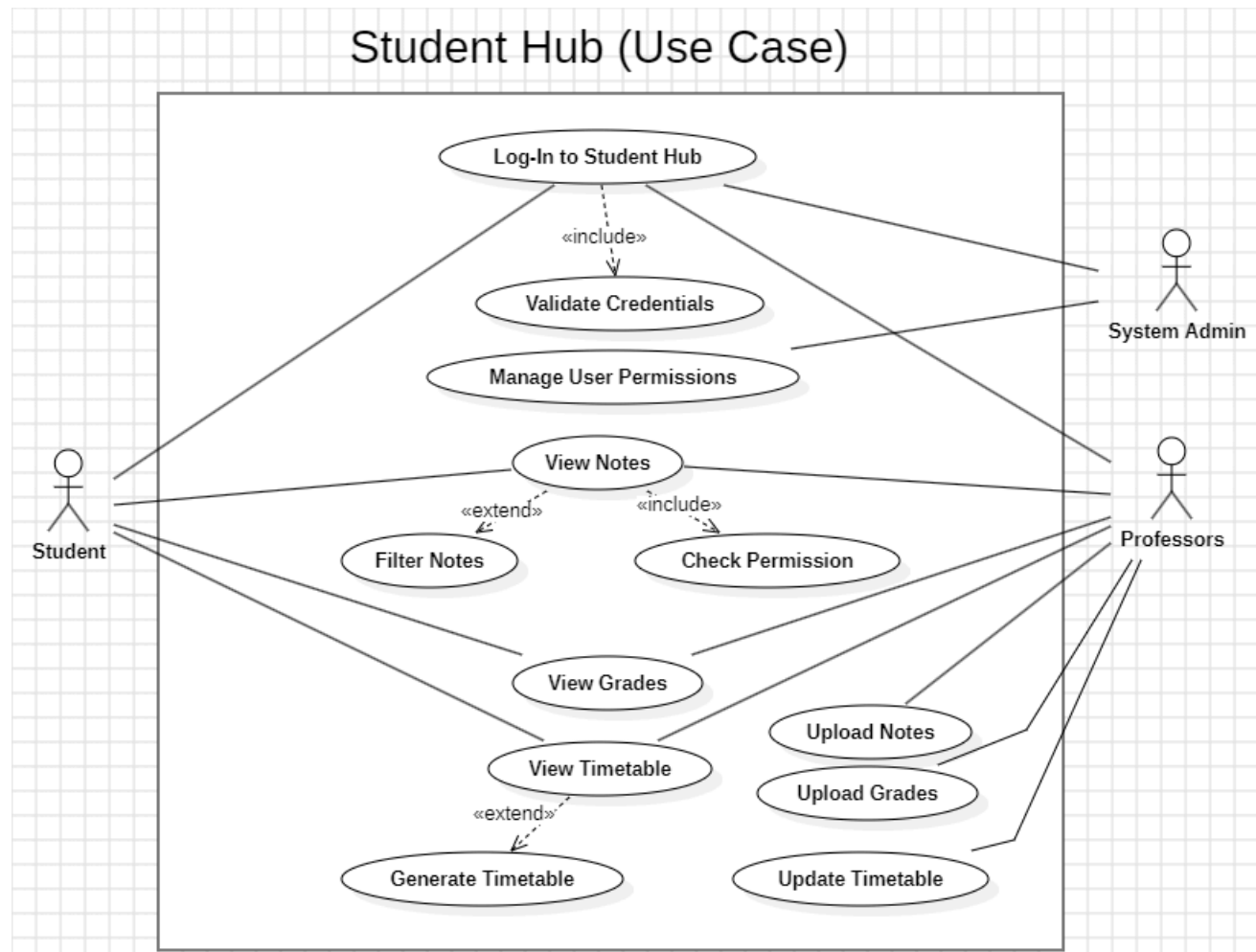
System Admin User Stories

As a [System Admin], I need to [Log in to the Student Hub], so that I can securely access the administrative features and manage user data, notes, grades, and timetables.

As a [System Admin], I need to [Manage User Permissions], so that I can control which users can access different features of the system.

Use Case Diagram

Status: **COMPLETE**



Use Case Scenario Specification

Status: COMPLETE

Log-In

As a [Student], I need to [Log in to the Student Hub], so that I can access all of my course materials, grades, and timetables.

As a [Professor], I need to [Log in to the Student Hub], so that I can securely access and manage course content, grades, and student records.

As a [System Admin], I need to [Log in to the Student Hub], so that I can securely access the administrative features and manage user data, notes, grades, and timetables.

Use case name: Log In	UniqueID: Student UC 001
Area: Student Hub System	
Actor(s): Student, Professor, Admin	
Description: Enables users to securely log in to the Student Hub.	
Triggering Event: User initiates the login process.	
Trigger type: <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal	
Steps Performed (Main Path)	Information for Steps
1. The user enters their email and password	Email, password
2. The user clicks "LogIn"	
3. The system validates for credentials	Email, password
4. If valid, access is granted and the student hub is displayed to the user.	
Preconditions: User must have a valid username/ID with corresponding password in the database.	
Postconditions: The system grants access to the user once logged in.	
Assumptions: The user is a valid student, professor, or admin in the Student Hub.	
Requirements met: As a user, I want to securely log into the Student Hub so that I can access my account and its features.	
Outstanding issues: Can a student access a professor account?	
Priority: High	
Risk: High	

Manage User Permissions

As a [System Admin], I need to [Manage User Permissions], so that I can control which users can access different features of the system.

Use case name: Manage User Permissions	UniqueID: Student UC 002
Area: Student Hub System	
Actor(s): System Admin	
Description: Allows the admin to manage user roles and permissions within the Student Hub.	
Triggering Event: Admin selects the option to manage user permissions.	
Trigger type: <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal	
Steps Performed (Main Path)	Information for Steps
1. Admin logs into the Student Hub.	Email, Password
2. The admin navigates to the “Manage User Permissions” section.	
3. The admin selects a user to modify permissions (Student or Professor).	StudentID, ProfessorID
4. The admin adjusts the user’s role or permissions.	StudentID, ProfessorID, permission, action
5. The system validates the changes and updates the user’s permissions.	
Preconditions: Admin must have valid login credentials and appropriate privileges to modify permissions.	
Postconditions: User permissions are updated successfully in the system.	
Assumptions: Admin has the necessary knowledge to assign permissions correctly.	
Requirements met: As a [System Admin], I need to [Manage User Permissions], so that I can control which users can access different features of the system.	
Outstanding issues: Ensuring proper granularity in user roles and permissions.	
Priority: High	
Risk: High (incorrect permissions could lead to unauthorized access or data exposure).	

View Notes

As a [Student], I need to [View Notes], so that I can look over what I have studied and know the content of a specific course I study, that way I will be better prepared for future or current exams and assignments.

As a [Professor], I need to [View Notes], so that I can check what I have uploaded and shared with my students, and to also see the content of the course I teach.

Use case name: View Notes	UniqueID: Student UC 003
Area: Student Hub System	
Actor(s): Student, Professor, System Admin	
Description: Allows the user to view the notes	
Triggering Event: The user selects the “View Notes” option.	
Trigger type: <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal	
Steps Performed (Main Path)	Information for Steps
1. User logs in to the Student Hub.	studentID, name, password
2. User navigates to the “View Notes” section.	
3. System checks if user has permission to view notes.	studentID, professorID, role
4. System fetches and displays available notes.	courseID, subjectID, notesID, title, uploadDate, content
Preconditions: Notes must already exist in the database.	
Postconditions: Notes are successfully displayed to the user.	
Assumptions: Notes are correctly uploaded and stored in the database. User has the necessary permission to access the notes.	
Requirements met: As a [Student], I need to [View Notes], so that I can look over what I have studied and know the content of a specific course I study, that way I will be better prepared for future or current exams and assignments. As a [Professor], I need to [View Notes], so that I can check what I have uploaded and shared with my students, and to also see the content of the course I teach.	
Outstanding issues: Notes filtering might not work for unsupported tags or metadata.	
Priority: Medium	
Risk: Low	

View Grades

As a [Student], I need to [View Grades], so that I can see how well I have performed academically in different subjects or exams.

As a [Professor], I need to [View Grades], so that I can check what I have uploaded and so that I can monitor their progress and provide feedback or adjustments as needed.

Use case name: View Grades	UniqueID: Student UC 004
Area: Student Hub System	
Actor(s): Student, System Admin	
Description: Allows students to view their grades stored in the Student Hub.	
Triggering Event: The user selects the “View Grades” option.	
Trigger type: <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal	
Steps Performed (Main Path)	Information for Steps
1. User logs in to the Student Hub	studentID, name, password
2. User navigates to the “View Grades” section.	
3. System fetches and displays the grades	gradeID, studentID, course, grade
Preconditions: Grades must already exist in the database.	
Postconditions: Grades are successfully displayed to the student.	
Assumptions: Grades are correctly uploaded and associated with the students.	
Requirements met: As a [Student], I need to [View Grades], so that I can see how well I have performed academically in different subjects or exams. As a [Professor], I need to [View Grades], so that I can check what I have uploaded and so that I can monitor their progress and provide feedback or adjustments as needed.	
Outstanding issues: Errors in the database may prevent grades from being displayed.	
Priority: Medium	
Risk: Low	

View Timetable

As a [Student], I need to [View Timetable], so that I can see when I have a lecture or tutorial throughout the day or week through a schedule.

As a [Professor], I need to [View Timetable], so that I can verify that the information given is correct, and so that I view my schedule of when I'll be giving a lecture or tutorial throughout the day and week.

Use case name: View Timetable	UniqueID: Student UC 005
Area: Student Hub System	
Actor(s): Students, Professors	
Description: Enables the user to view the timetable	
Triggering Event: The user selects the "View Timetable" option.	
Trigger type: <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal	
Steps Performed (Main Path)	Information for Steps
1. The user retrieves their courses.	studentID, professorID, courseID
2. System generates timetables.	timetableID
3. System stores the timetable in the database	timetableID, schedule
Preconditions: Students must be enrolled in courses.	
Postconditions: Timetable is successfully generated and stored.	
Assumptions: All course schedules are updated in the system.	
Requirements met: As a [Student], I need to [View Timetable], so that I can see when I have a lecture or tutorial throughout the day or week through a schedule. As a [Professor], I need to [View Timetable], so that I can verify that the information given is correct, and so that I view my schedule of when I'll be giving a lecture or tutorial throughout the day and week.	
Outstanding issues: Scheduling conflicts among courses.	
Priority: Medium	
Risk: Medium	

Upload Notes

As a [Professor], I need to [Upload Notes], so that I can provide study materials and lecture content to my students.

Use case name: Upload Notes	UniqueID: Student UC 006
Area: Student Hub System	
Actor(s): Professor	
Description: User uploads notes onto Student Hub.	
Triggering Event: Professor uploads notes, System Admin stores notes into database.	
Trigger type: <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal	
Steps Performed (Main Path)	Information for Steps
1. Professor logs in to the Student Hub.	Email, password
2. Professors select the option to Upload Notes.	professorID, courseID
3. Professor uploads notes onto the system.	note(notesID, title, content, uploadDate)
4. System stores notes into the database.	note(notesID, title, content, uploadDate)
5. System converts notes to the required format.	note(notesID, title, content, uploadDate)
Preconditions: Professor must have a valid account and access. Notes must be in supported formats.	
Postconditions: Notes become accessible to authorized users (e.g., students).	
Assumptions: Users have a stable internet connection during upload.	
Requirements met: As a [Professor], I need to [Upload Notes], so that I can provide study materials and lecture content to my students.	
Outstanding issues: Handling of unsupported file formats. Error handling for upload interruptions or database failures.	
Priority: High (critical feature for the Student Hub System).	
Risk: Medium. Medium (data loss or incompatibility with file formats)	

Upload Grades

As a [Professor], I need to [Upload Grades], so that students can track their progress in my course.

Use case name: Upload Grades	UniqueID: Student UC 007
Area: Student Hub System	
Actor(s): Professors, System Admin	
Description: User uploads grades onto the System.	
Triggering Event: The professor uploads student grades onto the Student Hub.	
Trigger type: <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal	
Steps Performed (Main Path)	Information for Steps
6. Professor logs into the Student Hub.	email, password
7. Professor selects the option to upload grades.	professorID, courseID, studentID
8. Professor inputs or uploads grades for each student.	studentID, grade, course
9. System validates the grades.	valid grade format: float 0-100.00
10. System stores the grades in the database.	grade(gradeID, studentID, courseID, grade, uploadDate)
Preconditions: Professor must have a valid account and permissions to upload grades. The grade file or manual entry must follow the system's supported formats (e.g., CSV or direct input). Students and courses must already exist in the database.	
Postconditions: Grades are successfully stored in the database. Grades are made accessible to authorized users (e.g., students, System Admin).	
Assumptions: Professors have a stable internet connection during upload. Grades entered/uploaded are accurate and complete.	
Requirements met: As a [Professor], I need to [Upload Grades], so that students can track their progress in my course.	
Outstanding issues: Handling of duplicate or conflicting grade entries.	
Priority: High	
Risk: Medium. Risks include incorrect grade inputs, technical errors during upload, and database connectivity issues.	

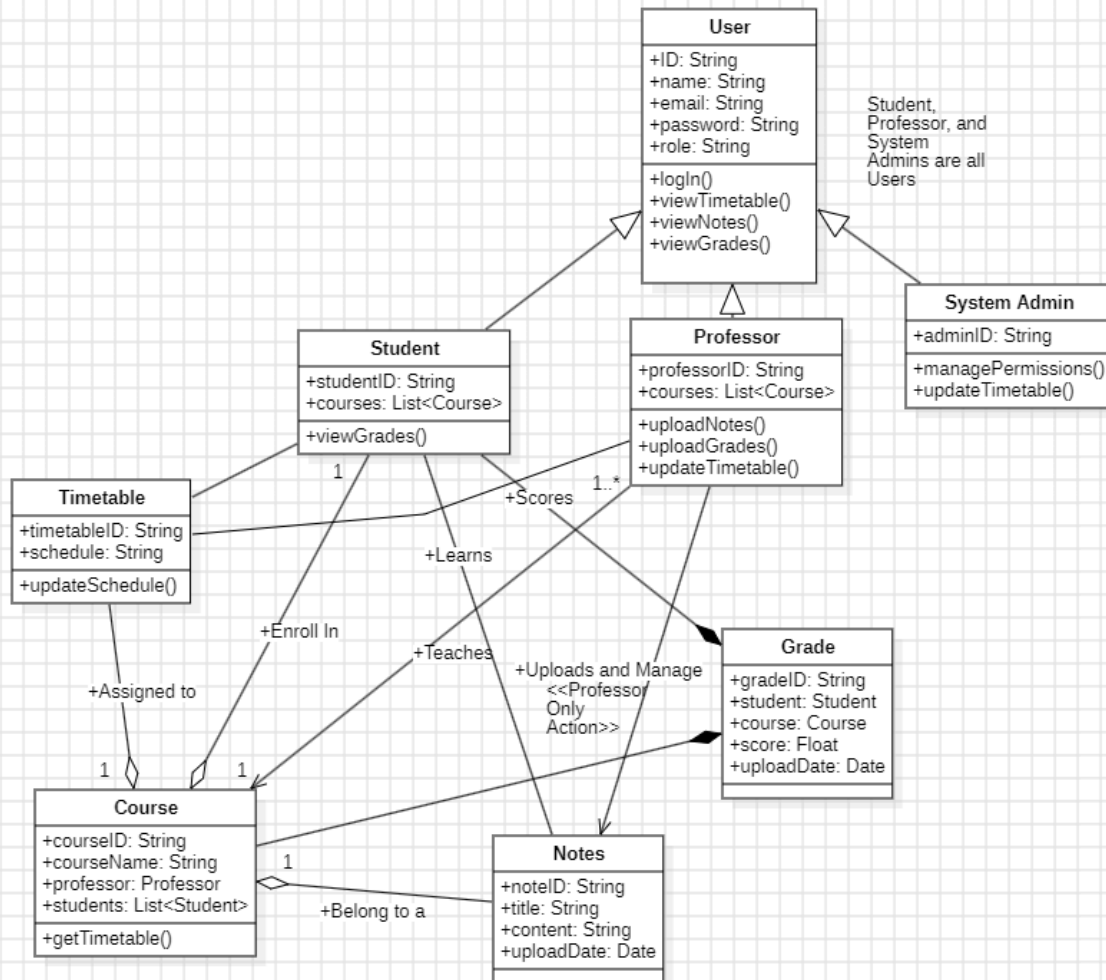
Update Timetable

As a [Professor], I need to [Update Timetable], so that students always have the most accurate information about when their lectures and tutorials are scheduled.

Use case name: Update Timetable	UniqueID: Student UC 008
Area: Student Hub System	
Actor(s): Professors.	
Description: Allows professors or system admins to update course timetables.	
Triggering Event: Professor or System Admin selects the option to update the timetable.	
Trigger type: <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal	
Steps Performed (Main Path)	Information for Steps
1. The professor logs into the student Hub.	Email, password
2. The professor navigates to the "Update Timetable" section.	professorID
3. System validates if professor has permissions	professorID
4. When valid, The professor selects the course to update.	courseID
5. The professor makes necessary changes to the timetable (e.g., date, time, or place).	timetableData(course, schedule)
6. The system validates the timetable changes.	
7. The system stores the updated timetable in the database.	timetableData
Preconditions: The professor or system admin must have valid login credentials. Courses and existing timetables must be in the system.	
Postconditions: The updated timetable is stored and accessible in the system.	
Assumptions: The user has proper permissions to update the timetable. Changes are correct and conflict-free.	
Requirements met: As a [Professor], I need to [Update Timetable], so that students always have the most accurate information about when their lectures and tutorials are scheduled.	
Outstanding issues: Handling of duplicate or conflicting grade entries.	
Priority: High	
Risk: Medium. Risks include incorrect grade inputs, technical errors during upload, and database connectivity issues.	

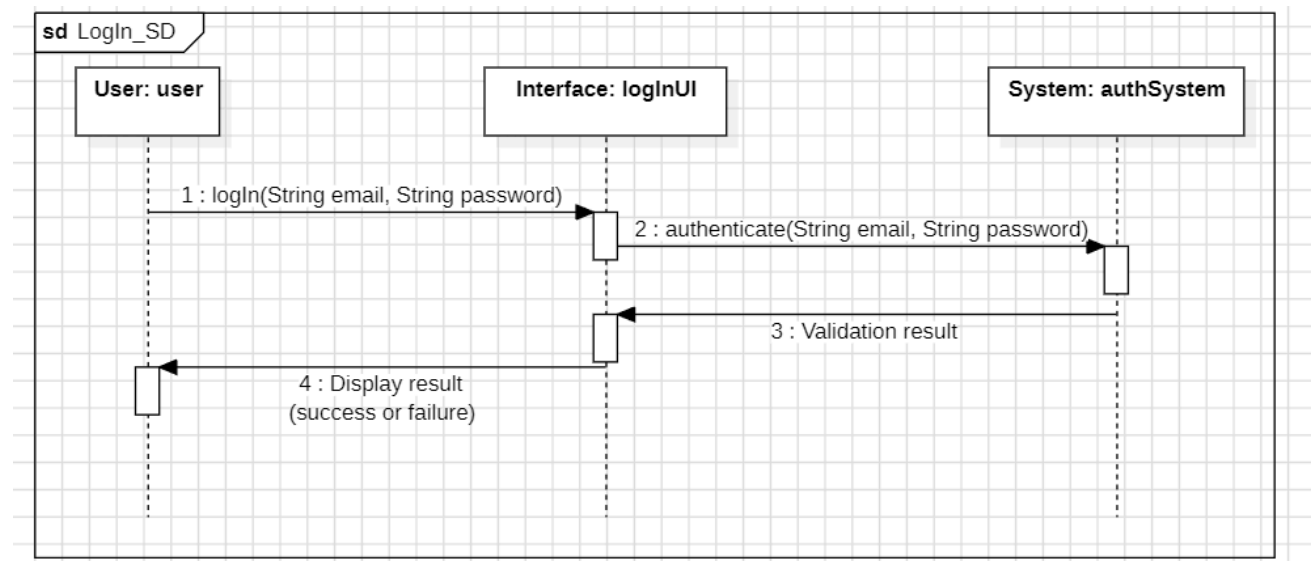
Class Diagram

Status: **COMPLETE**



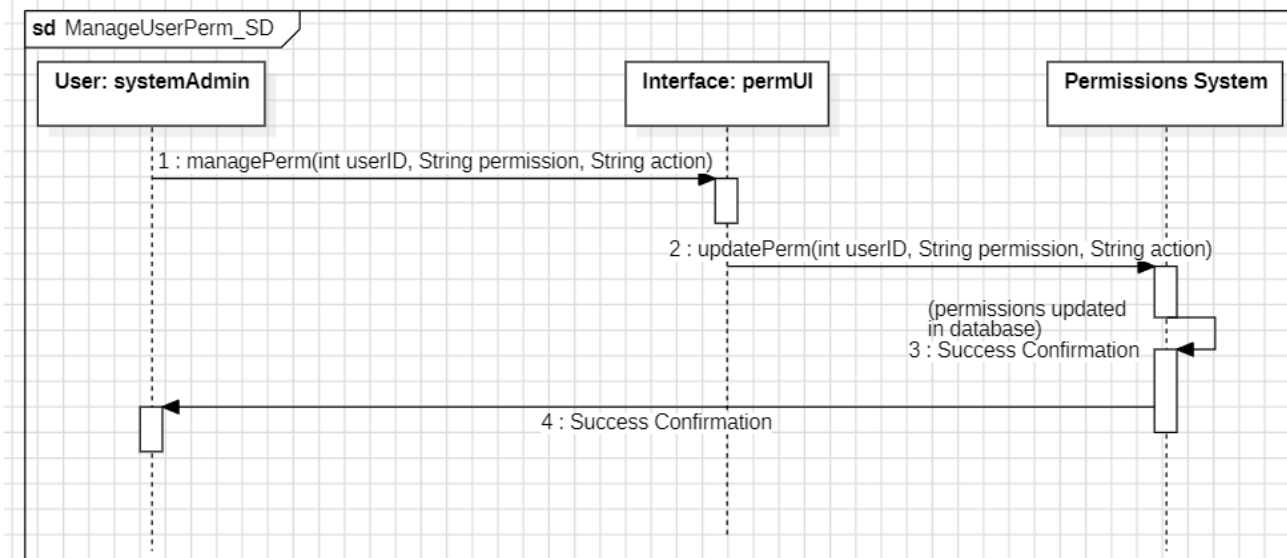
Sequence Diagrams

Status: **COMPLETE**



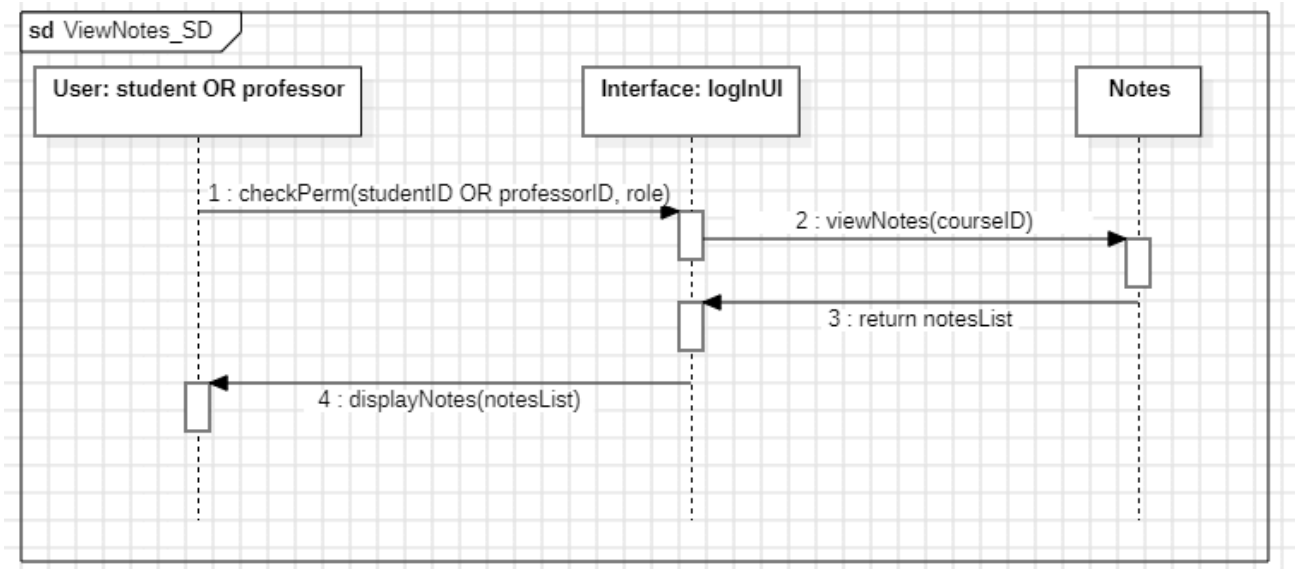
Login

Sequence Number	Caller	Callee	MSG Name	MSG Type	MSG Parameter	MSG constraint
1.	User	logInUi	checkPerm	Synchronous	Email & Password	-
2.	logInUi	authSystem	authenticate	Synchronous	Email & Password	-
3.	authSystem	logInUi	Validation Result	Synchronous	Success/Failure	-
4.	logInUi	User	Display Result	Synchronous	Success/Failure	-



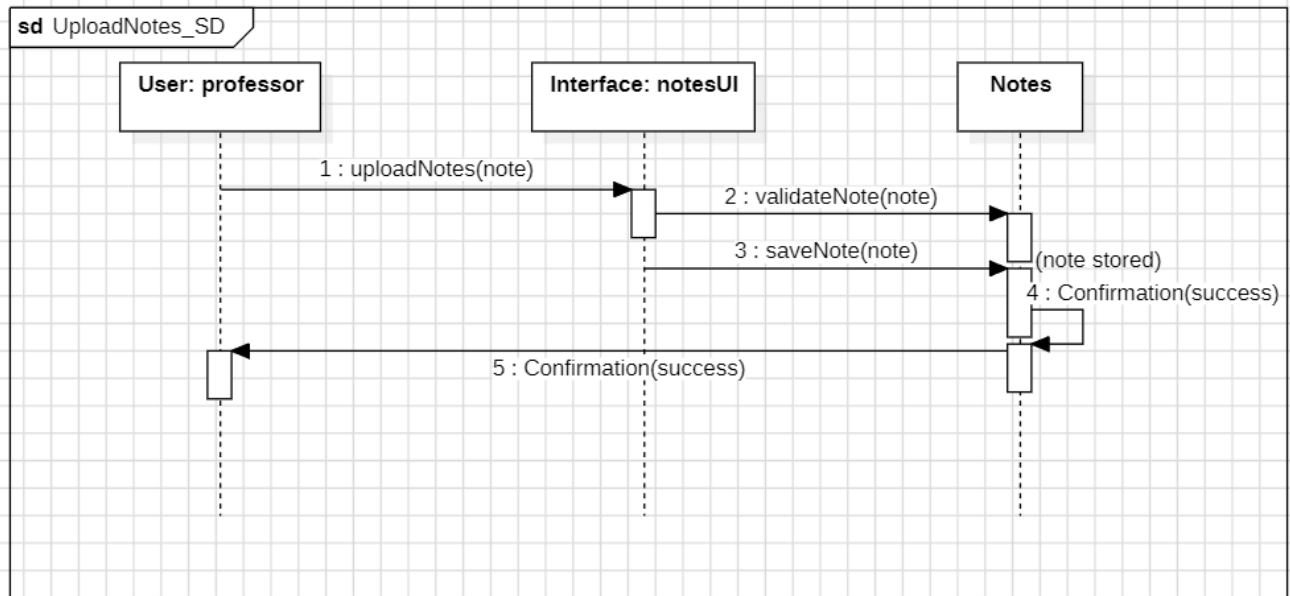
Manage user perm

Sequence Number	Caller	Callee	MSG Name	MSG Type	MSG Parameter	MSG constrain
1.	systemAdmin	permUI	managePerm	Synchronous	userID, Permission, Action	-
2.	permUI	Permissions System	updatePerm	Synchronous	userID, Permission, Action	-
3.	Permissions System	Permissions System	Validation Result	A-Synchronous	Success/Failure	-
4.	Permissions System	systemAdmin	Display Result	Synchronous	Success/Failure	-



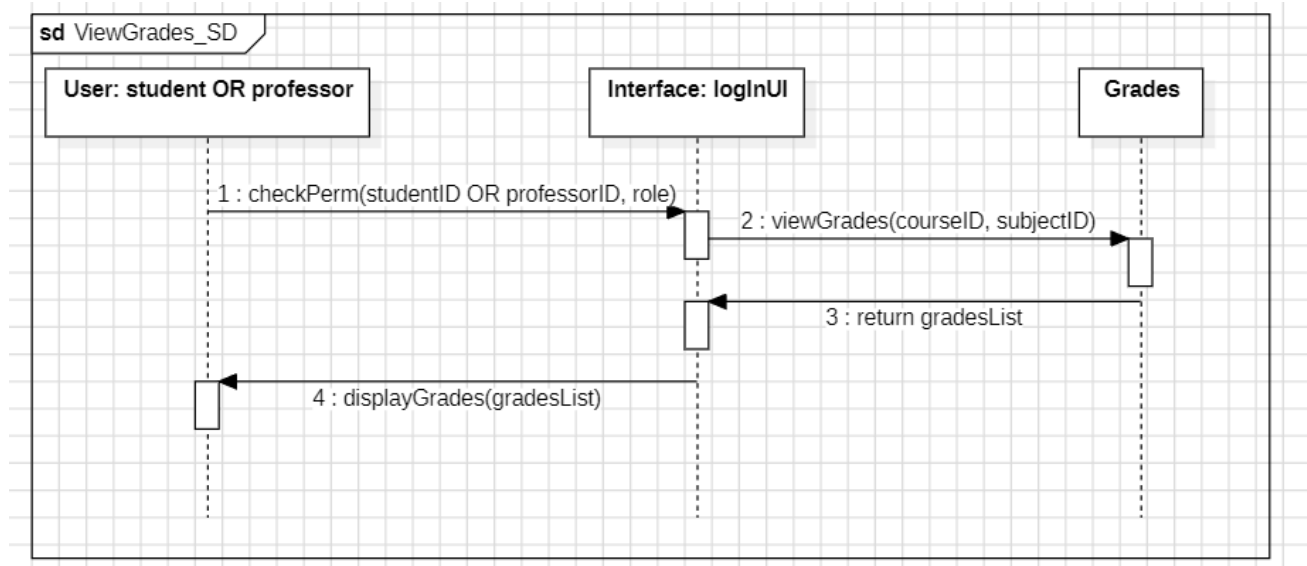
View notes

Sequence Number	Caller	Callee	MSG Name	MSG Type	MSG Parameter	MSG constrain
1.	user	logInUI	checkPerm	Synchronous	StudentID/ ProfessorID	-
2.	logInUI	Notes	viewNotes	Synchronous	courseID, SubjectID	-
3.	Notes	logInUI	return	Synchronous	notesList	-
4.	displayNotes	user	displayNotes	Synchronous	notesList	-



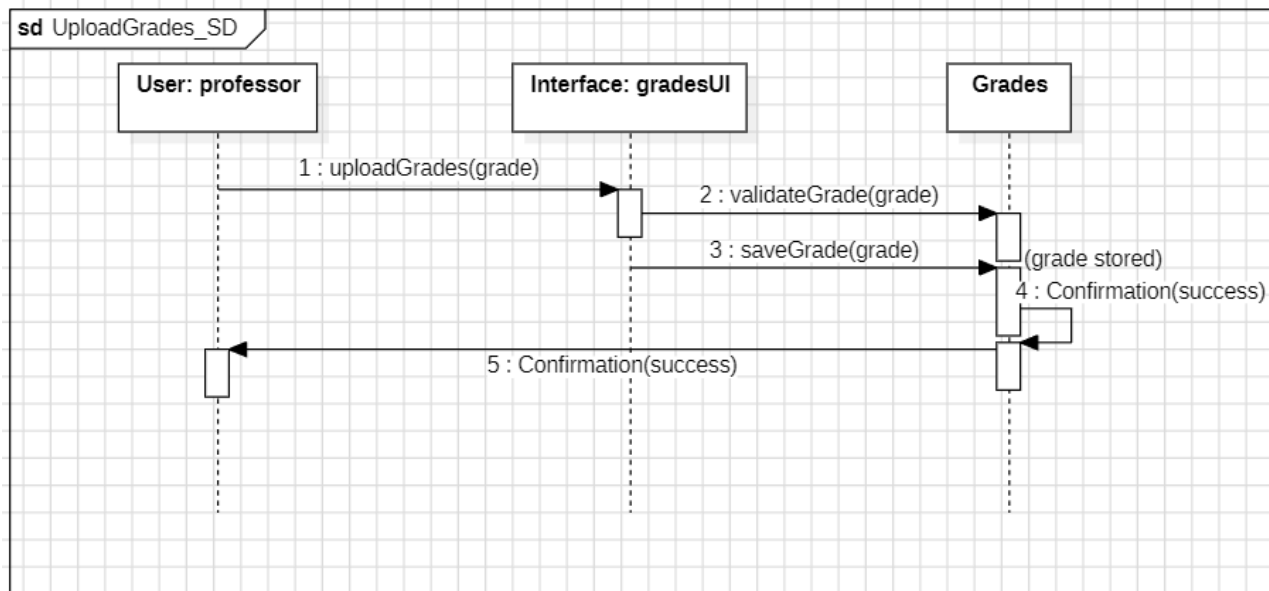
Upload notes

Sequence Number	Caller	Callee	MSG Name	MSG Type	MSG Parameter	MSG constrain
1.	professor	notesUI	uploadNotes	Synchronous	note	-
2.	notesUI	Notes	validateNote	Synchronous	note	-
3.	notesUI	Notes	saveNote	Synchronous	note	-
4.	Notes	Notes	Confirmation	A-Synchronous	Success/Failure	-
5.	Notes	professor	Confirmation	Synchronous	Success/Failure	-



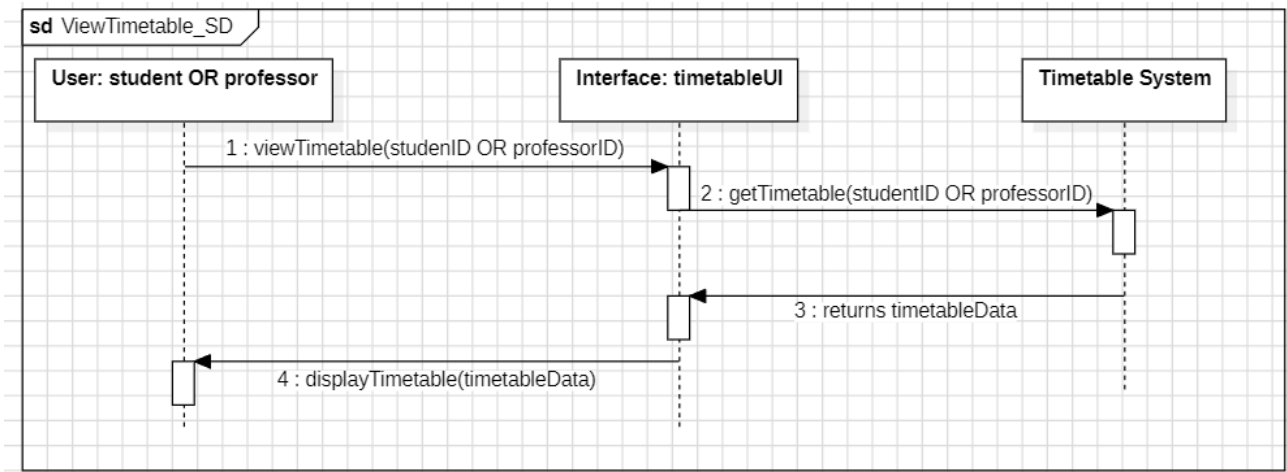
View Grades

Sequence Number	Caller	Callee	MSG Name	MSG Type	MSG Parameter	MSG constrain
1.	user	logInUI	checkPer m	Synchrono us	StudentID /Professor ID	-
2.	logInUI	Notes	viewGrade s	Synchrono us	courseID, SubjectID	-
3.	Notes	logInUI	return	Synchrono us	gradesList	-
4.	displayGr ades	user	displayGr ades	Synchrono us	gradesList	-



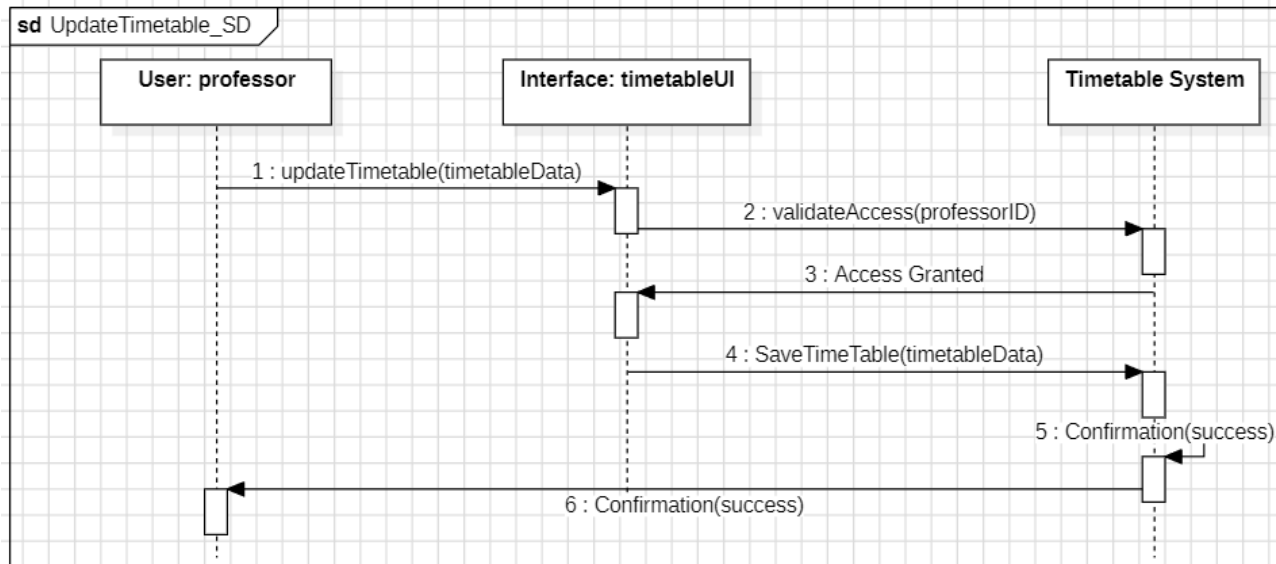
Upload Grades

Sequence Number	Caller	Callee	MSG Name	MSG Type	MSG Parameter	MSG constraint
1.	professor	gradesUI	uploadGrades	Synchronous	grade	-
2.	gradesUI	Grades	validateGrade	Synchronous	grade	-
3.	gradesUI	Grades	saveGrade	Synchronous	grade	-
4.	Grades	Grades	Confirmation	A-Synchronous	Success/Failure	-
5.	Grades	professor	Confirmation	Synchronous	Success/Failure	-



View Timetable

Sequence Number	Caller	Callee	MSG Name	MSG Type	MSG Parameter	MSG constrain
1.	user	timetable UI	viewTimetable	Synchronous	studentID /professorID	-
2.	timetable UI	Timetable System	getTimetable	Synchronous	studentID /professorID	-
3.	Timetable System	timetable UI	Returns timetableData	Synchronous		-
4.	timetable UI	user	displayTimetable	A-Synchronous	timetable Data	-



Update Timetable

Sequence Number	Caller	Callee	MSG Name	MSG Type	MSG Parameter	MSG constraint
1.	professor	timetable UI	updateTimetable	Synchronous	timetable Data	-
2.	timetable UI	Timetable System	validateAccess	Synchronous	professor ID	-
3.	Timetable System	timetable UI	Access Granted	Synchronous		-
4.	timetable UI	Timetable System	SaveTimeTable	Synchronous	timetable Data	-
5.	Timetable System	Timetable System	Confirmation	A-Synchronous	Success/Failure	-
6.	Timetable System	professor	Confirmation	Synchronous	Success/Failure	

Test Case Scenario

Status: **COMPLETE**

Test Cases (in table format) of the system that includes the following:

- (a) At least 5 test cases of the most important features of your system
- (b) A unique name and number
- (c) A requirement which this test case is exercising
- (d) Preconditions which describe the state of the software before the test case (which is often a previous test case that must always be run before the current test case)
- (e) Steps that describe the specific steps which make up the interaction
- (f) Expected Results which describe the expected state of the software after the test case is executed.

Test Case ID	Requirement	Preconditions	Steps	Expected Results
SHTC001	Users can log into the Student Hub	Users must have a valid username/ID and password.	<ol style="list-style-type: none"> 1. Enter username/ID. 2. Enter password. 3. Click "Login". 	The system validates credentials and grants access or shows an error message.
SHTC002	Admins can manage user permissions.	<ol style="list-style-type: none"> 1. User logged in as Admin. 2. Target user exists. 	<ol style="list-style-type: none"> 1. Navigate to "Manage Permissions". 2. Search and select a user. 3. Modify permissions. 4. Save changes. 	The system updates permissions and displays a success message.
SHTC004	Users can view their grades stored in the system.	<p>User must be logged into the Student Hub.</p> <p>Grades must already exist in the database and be associated with the student.</p>	<ol style="list-style-type: none"> 1. Log in to the Student Hub using valid credentials. 2. Navigate to the "View Grades" section. 3. Select the course to view grades for. 4. View the displayed grades. 	The system retrieves and displays the correct grades for the selected course.
SHTC003	Users can view uploaded notes.	<ol style="list-style-type: none"> 1. User is logged in. 2. Notes exist. 	<ol style="list-style-type: none"> 1. Navigate to "View Notes". 2. Select a note. 3. View the content. 	The system displays the selected note's content.
SHTC004	Professors can upload notes.	<ol style="list-style-type: none"> 1. User is logged in as a Professor. 2. Permissions exist. 	<ol style="list-style-type: none"> 1. Navigate to "Upload Notes". 2. Select a file. 3. Click "Upload". 4. Confirm upload. 	The system stores the note and makes it available to authorized users.

SHTC005	Users can view timetables.	1. User is logged in. 2. Timetable exists.	1. Navigate to "View Timetable". 2. Select a course/user group. 3. View timetable.	The system retrieves and displays the timetable.
SHTC006	Professors can upload grades.	1. User is logged in as a Professor. 2. Course and students exist.	1. Navigate to "Upload Grades". 2. Select course. 3. Enter grades or upload file. 4. Confirm.	The system saves grades and associates them with students. A success message is displayed.
SHTC007	Professors can update timetables.	1. User is logged in as a Professor. 2. Timetable exists.	1. Navigate to "Update Timetable". 2. Select a timetable. 3. Modify. 4. Save changes.	The system updates the timetable. Students can view updated schedules.