# Use Case UC1: Sign In

**Primary Actor:** Student, Lecturer.

**Stakeholders and Interests:**

* Student: Wants simple user interface, fast response, no system errors.
* Lecturer: ?

**Preconditions:**

* Student is registered.

**Success Guarantee (Postcondition):**

* Student is logged in.

**Main Success Scenario (or Basic Flow):**

1. Student visits system home page.

2. System shows home page with login form and sign up button.

3. Student enters his/her username and password then click login button.

4. System shows Student’s home page.

**Extensions (or Alternative Flows):**

\*a. At any time, system fails, to support recovery, ensure all transaction sensitive state and events can be recovered from any step of the scenario.

1. Student restarts System and requests recovery of prior state.

2. System reconstructs prior state.

2a. System detects anomalies preventing recovery.

1. System signals error to the Student, records the error, and enters a clean state.

3a. Student enters invalid username or password.

1. System shows errors and request to Student to retry.

2. Student enters his/her username and password.

4a. System detects failure to communicate with server.

1. System signals error and rejects the request.

# Use Case UC2: File Download

**Primary Actor:** Student

**Stakeholders and Interests:**

* Student: Wants simple user interface, fast response, no system errors.

**Preconditions**: Student is identified and authenticated.

**Success Guarantee (Postcondition):** File is downloaded.

**Main Success Scenario (or Basic Flow):**

1. Student visits file download page.

2. System lists available files.

3. Student clicks the file that she/he is wanted to download.

4. System starts the download process.

**Extensions (or Alternative Flows):**

\*a. At any time, System fails, to support recovery, ensure all transaction sensitive state and events can be recovered from any step of the scenario.

1. Student restarts System and requests recovery of prior state.

2. System reconstructs prior state.

2a. System detects anomalies preventing recovery.

1. System signals error to the Student.

2. System records the error.

3. System enters a clean state.

4a. System detects failure to communicate with server.

1. System signals error and rejects the request.

# Use Case UC3: File Upload

**Primary Actor**: Student

**Stakeholders and Interests:**

* Student: Wants simple user interface, fast response, no system errors.

**Preconditions:** Student is identified and authenticated.

**Success Guarantee (Postcondition):** File is uploaded.

**Main Success Scenario (or Basic Flow):**

1. Student visits file upload page.

2. System opens file browser dialog.

3. Student chooses the file that she/he is wanted to upload.

4. System starts the upload process.

**Extensions (or Alternative Flows):**

\*a. At any time system fails, to support recovery, ensure all transaction sensitive state and events can be recovered from any step of the scenario.

1. Student restarts System and requests recovery of prior state.

2. System reconstructs prior state.

2a. System detects anomalies preventing recovery.

1. System signals error to the Student, records the error, and enters a clean state.

3a. Invalid file.

1. System shows the error and returns the file upload page.

4a. System detects failure to communicate with server.

1. System signals error and rejects the request.

# Use Case UC4: Course Analysis Report

**Primary Actor**: Instructor

**Stakeholders and Interests:**

* Instructor: Wants easily understandable questions, fast response, no system errors.

**Preconditions**: Instructor is identified and authenticated.

**Success Guarantee (Postcondition):** Report is generated.

**Main Success Scenario (or Basic Flow):**

1. Instructor enters his/her profile page.

2. System shows Instructor’s profile page.

3. Instructor clicks My Courses button.

4. System lists Instructor’s courses.

5. Instructor selects the course that she/or is wanted to inspect.

6. System asks for report type.

7. Instructor selects report type.

8. System shows the analysis report.

**Extensions (or Alternative Flows):**

\*a. At any time, system fails, to support recovery, ensure all transaction sensitive state and events can be recovered from any step of the scenario.

1. Instructor restarts System and requests recovery of prior state.

2. System reconstructs prior state.

2a. System detects anomalies preventing recovery.

1. System signals error to the Instructor, records the error, and enters a clean state.

\*b. System detects failure to communicate with server.

1. System signals error and rejects the request.

# Use Case UC5: Adding Lecture

**Primary Actor:** Instructor

**Stakeholders and Interests:**

* Instructor: Wants to add new lecture quickly.

**Preconditions**: Instructor is identified and authenticated.

**Success Guarantee (Postcondition):** Lecture is added.

**Main Success Scenario (or Basic Flow):**

1. Instructor clicks on Add Lecture button on his/her profile page.

2. System returns the information page about the new lecture.

3. Instructor fills out fields about the lecture then clicks on Submit button.

4. System adds the new lecture and saves changes.

**Extensions (or Alternative Flows):**

\*a. At any time, System fails, to support recovery, ensure all transaction sensitive state and events can be recovered from any step of the scenario.

1. Instructor restarts System and requests recovery of prior state.

2. System reconstructs prior state.

2a. System detects anomalies preventing recovery.

1. System signals error to the Instructor, records the error, and enters a clean state.

# Use Case UC6: Deleting Lecture

**Primary Actor:** Instructor

**Stakeholders and Interests:**

* Instructor: Wants to delete lecture quickly.

**Preconditions**: Instructor is identified and authenticated.

**Success Guarantee (Postcondition):** Lecture is deleted.

**Main Success Scenario (or Basic Flow):**

1. Instructor clicks on My Lectures button on his/her profile page.

2. System returns his/her lectures.

3. Instructor choses the lecture that he/she wants to delete.

4. System returns lecture’s information.

5. Instructor clicks on the Delete Lecture button.

6. System deletes the lecture and sends the message to all student who enrolls this lecture.

**Extensions (or Alternative Flows):**

\*a. At any time, system fails, to support recovery, ensure all transaction sensitive state and events can be recovered from any step of the scenario.

1. Instructor restarts System and requests recovery of prior state.

2. System reconstructs prior state.

2a. System detects anomalies preventing recovery.

1. System signals error to the Instructor, records the error, and enters a clean state.

# Use Case UC7: Lecture Listing

**Primary Actor:** Student, Instructors

**Stakeholders and Interests:**

* Student: Wants look up his/her lecture that he/she had chosen.
* Instructor: Wants look up his/her lecture

**Preconditions:** Student is identified and authenticated.

**Success Guarantee (Postcondition):** Lectures that students chose are listing well organized. Student has enrolled lectures that she/he wants. Student deletes his/her enrollment of lectures that he/she had chosen.

**Main Success Scenario (or Basic Flow):**

1. Student clicks my Lectures button.

2. System presents Lectures that was chosen by student.

3. Student views his/her lectures.

**Extensions (or Alternative Flows):**

\*a. At any time, system fails, to support recovery, ensure all transaction sensitive state and events can be recovered from any step of the scenario.

1. Student restarts System and requests recovery of prior state.

2. System reconstructs prior state.

2a. System detects anomalies preventing recovery.

1. System signals error to the Instructor.

2. System records the error.

3. System enters a clean state.

# Use Case UC8: Enrolling to a Lecture

**Primary Actor:** Student

**Stakeholders and Interests:**

* Student: Wants to enroll new lecture easily.

**Preconditions**: Student is identified and authenticated. There must be at least one lecture that student can select.

**Success Guarantee (Postcondition):** Student has enrolled lectures that she/he wants. Student deletes his/her enrollment of lectures that he/she had chosen.

**Main Success Scenario (or Basic Flow):**

1. Student clicks the Lecture Selection button.

2. System presents lectures that student can select.

3. Student selects a lecture.

4. System wants an enrollment key for the lecture from the student.

5. Student enters the keyword for the enrollment and clicks “enroll me” button.

**Extensions (or Alternative Flows):**

\*a. At any time, System fails, to support recovery, ensure all transaction sensitive state and events can be recovered from any step of the scenario.

1. Student restarts System and requests recovery of prior state.

2. System reconstructs prior state.

2a. System detects anomalies preventing recovery.

1. System signals error to the Instructor, records the error, and enters a clean state.

2a. If there is no lecture to select, system gives a message to the student.

5a. If student enters a wrong enrollment key, system gives an error message.

1. System returns to step 4.

# Use Case UC9: Unenrolling a Lecture

**Primary Actor:** Student

**Stakeholders and Interests:**

* Student: Wants delete his/her enrollment of a lecture.

**Preconditions**: Student is identified and authenticated.

**Success Guarantee (Postcondition):** Student deletes his/her enrollment of lectures that he/she had chosen.

**Main Success Scenario (or Basic Flow):**

1. Student clicks My Lectures button.

2. System presents lectures that was chosen by student.

3. Student selects lecture that she/he wants to delete.

4. System removes student from student list of this lecture and takes out that lecture from students lecture list.

**Extensions (or Alternative Flows):**

\*a. At any time, system fails, to support recovery, ensure all transaction sensitive state and events can be recovered from any step of the scenario.

1. Student restarts System and requests recovery of prior state.

2. System reconstructs prior state.

2a. System detects anomalies preventing recovery.

1. System signals error to the Instructor, records the error, and enters a clean state.

3a. If there is no lecture student’s list system gives error message to the student.

# Use Case UC10: New Forum Post

**Primary Actor**: Student

**Stakeholders and Interests:**

* Student: Wants to create a new post easily.

**Preconditions:** Student is identified and authenticated.

**Success Guarantee (Postcondition**): Post is saved.

**Main Success Scenario (or Basic Flow):**

1. Student clicks on New Post button to post a new topic to the topic page.

2. System returns topic editor page.

3. Student fills out the required fields.

4. System checks the fields.

5. Student clicks on the Post button.

6. System records transaction and.

**Extensions (or Alternative Flows):**

\*a. At any time, system fails, to support recovery, ensure all transaction sensitive state and events can be recovered from any step of the scenario.

1. Student restarts System and requests recovery of prior state.

2. System reconstructs prior state.

2a. System detects anomalies preventing recovery.

1. System signals error to the Student, records the error, and enters a clean state.

# Use Case UC11: Reply Forum Post

**Primary Actor:** Student

**Stakeholders and Interests:**

* Student: Wants to reply a post easily.

**Preconditions:** Student is identified and authenticated.

**Success Guarantee (Postcondition):** Post is saved.

**Main Success Scenario (or Basic Flow):**

1. Student clicks on a topic to view it.

2. System returns a page which contains view of topic.

3. Student clicks on Reply button to post an entry to the topic.

4. System returns entry editor page.

5. Student fills out fields.

6. System checks out the fields.

7. Student clicks on the Post button.

8. System records transaction.

**Extensions (or Alternative Flows):**

\*a. At any time, System fails, to support recovery, ensure all transaction sensitive state and events can be recovered from any step of the scenario.

1. Student restarts System and requests recovery of prior state.

2. System reconstructs prior state.

2a. System detects anomalies preventing recovery.

1. System signals error to the Student

2. System records the error

3. System enters a clean state.