**A UML Documentation for**

**Learning Management System (LMS)**

**ELEC5620**

**A UML Documentation for Learning Management System (LMS)**

1. **Introduction**

The Learning Management System (LMS) is designed by Perfect Software Pty Ltd. for Aussie College to manage students and units of study.

Students in the University of Sydney use the LMS system to manage their units of study. Learning Management System is the system that mainly includes two parts: essential requirements and desirable features. Essential requirements manage students’ course description and selection, enrollment, assignment submission, manage teachers’ handouts, assignment viewing and marking, and notification, and also manage administrators’ course allocation. Desirable features are divided into two parts: major features and minor features. Major features include program, credit management and timetable management, while minor features include discussion board, finance management and online e-learning and quiz system. The LMS system has good functions for Sydney students, teachers and administrators. The following content will demonstrate the system’s analysis.

1. **Requirements**
   1. **Login**

The user should input the username and password, and click the Login button to log into the Learning Management System.

* 1. **View enrolled units of current semester**

The student user should click the My Studies button. The enrolled units of current semester should be shown on the page.

* 1. **View unit descriptions**

At the My Studies page and the Add Units page, the student user should view the unit descriptions by clicking the unit name.

**2.4 Add units**

At the My Studies page, the student user should click the Add Units button to go to the Add Units page.

At the Add Units page, the student user should select the available units from the Units drop-down box and click the Confirm Add Units button to add the units. The student user also should view the unit descriptions by clicking the unit name.

After the student user click the Confirm Add Units button, the result should be shown on the page.

* 1. **Drop units**

At the My Studies page, the student user should click the Drop Units button. The check box should be shown before each unit name. The student user should check the checkboxes of the units that user wants to drop and click the Confirm Drop Units button. The user also should view the unit descriptions by clicking the unit name.

After the student user click the Confirm Drop Units button, the result should be shown on the My Studies page.

* 1. **Manage unit information**
     1. **Edit unit description**

The administrator user should click the Manage Units button to go to the Manage Units page. At the Manage Units page, all the units should be shown on the page. The admin user should click the unit name that the user wants to edit. The unit descriptions and current lecturer should be shown on Unit Info page and the content is editable. The admin user should click the Confirm button to finish the edit of unit description. The result should be shown on the page.

* + 1. **Allocate lecturer**

The administrator user should click the Manage Units button to go to the Manage Units page. At the Manage Units page, all the units should be shown on the page. The admin user should click the unit name that the user wants to allocate. The unit descriptions and current lecturer should be shown on Unit Info page and the content should be editable. If the lecturer has not been allocated, the content of that column should be empty. The user should allocate lecturer and click the Confirm button to finish allocating. The result should be shown on the Unit Info page.

* 1. **Manage units**
     1. **Add units (admin)**

The administrator user should click the Manage Units button to go to the Manage Units page. At the Manage Units page, all the units will be shown on the page. The admin user should click the Add Units button to go to the Add Units page. At the Add Units page, the admin user should input the unit information and then click the Confirm button to add unit. After the admin user click the Confirm button, the result will be shown on the page.

**2.7.2 Drop units (admin)**

The administrator user should click the Manage Units button to go to the Manage Units page. All the units will be shown on the Manage Units page. The admin user should click the Drop Units button. The checkbox of each unit will be shown before unit name. The admin user should check the checkboxes of the units that the user wants to drop, and then click the Confirm button. After the admin user click the Confirm button, the result will be shown on the page.

* 1. **Enrol students to units**

The administrator user should click the Manage Enrol button to go to the Manage Enrol page. At the Manage Enrol page, the admin user should input the student ID to search student, choose units for student, and click the Confirm button to finish the manual enrol. The result should be shown on the page.

* 1. **Lecture materials and announcement**

The lecture materials and announcement should be viewed and downloaded by both teacher and students who has enrolled in the specific course and can only be deleted, edit, uploaded by teacher.

**2.10 Assignment submission**

Students should submit assignments by upload files from computer and can delete the uploaded file or upload more files before submit. As long as students submit the assignment successfully, the teacher and the student can download the files which are submitted. The students and teachers should only enter the assignment submission function of the courses that he or she has enrolled in.

* 1. **Marking**

The teacher user should click “Marking” button to go to the Marking page. At the Marking page, the teacher user should select the unit and which assignment of this unit to mark. After teacher user selects the unit and assignment, all enrolled student should be shown on the page. The user should input the marks for students and click the “Submit” button. The result should be shown on the page.

The student user should click “My Grades” button to go to the View Grade page. At the View Grade page, the user should select the unit and view the assignment grades of the selected unit.

* 1. **Send emails**

The teacher user should click the Send Email button to open the send email page. The teacher user should select enrolled students, input the message and click the send button to inform student the class updates.

* 1. **Manage program**
     1. **Set units for program**

The administrator user should click the Manage Program button to go to the Manage Program page. At the Manage Program page, the administrator user should select the program and set the units for this program, which all the students who enrolled in this program can choose these units to study.

* + 1. **Set credit for program**

The administrator user should click the Manage Program button to go to the Manage Program page. At the Manage Program page, the administrator user should select the program and set the credit for this program, which all the students who enrolled in this program should complete to get the degree.

* + 1. **Set the number of units for program**

The administrator user should click the Manage Program button to go to the Manage Program page. At the Manage Program page, the administrator user should select the program and set the number of units for this program in a semester, which all the students who enrolled in this program should take in a semester.

**2.14 Feature Diagram (see in attaches)**

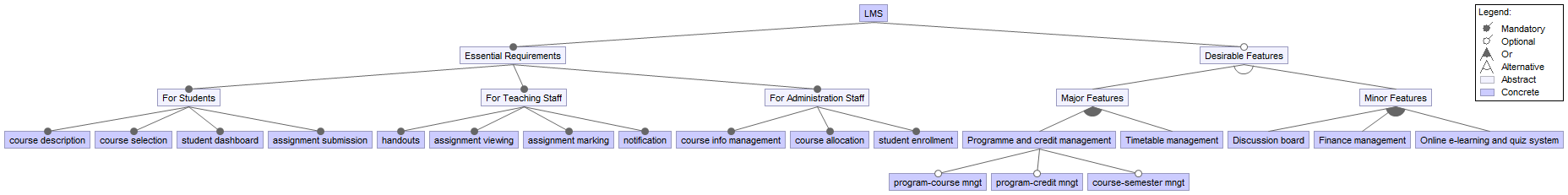


Figure1: Feature Diagram

1. **Modeling the static aspects of the system**
   1. **A snapshot of the LMS**

The learning management system designed as our class project is a system with Sydney Student and eLearning merged. For our system, the users are classified into student, teacher and administrator. The student can apply for the course and change the course, view and choose units after enrolment, download learning material, submit assignments and view their grades in the system. The teachers can release announcements and materials to the student as well as mark the quiz into the system. Both teachers and students can see their personal information including the course, unit they study or teach. The administrator can change the information of faculty, course, unit and student. They also have rights to help students enroll in units and allocate units to lectures.

* 1. **Use Case Templates**

|  |  |  |
| --- | --- | --- |
| Name | Login | |
| Goal | User logs into the Sydney Student System | |
| Level | System | |
| Precondition | User has Sydney Student’s account | |
| Success Condition | User logs into the Learning Management System | |
| Failure Condition | Fail to log into the Learning Management System | |
| Actors | Administrator, teacher, student | |
| Trigger | Using the Learning Management System | |
| Description | Step | Actions |
| 1 | Input username |
| 2 | Input password |
| 3 | Press Login button |
| 4 | User logs into the Learning Management System |
| Extends | -- | |
| Includes | -- | |

Table1: Login use case

|  |  |  |
| --- | --- | --- |
| Name | View Enrolled Units of Current Semester | |
| Goal | Show the enrolled units of current semester | |
| Level | System | |
| Precondition | User logs into the Learning Management System | |
| Success Condition | User’s enrolled units are shown on the page | |
| Failure Condition | Fail to get user’s enrolled units | |
| Actors | Student | |
| Trigger | Press the My Studies button | |
| Description | Step | Actions |
| 1 | User logs into Learning Management System |
| 2 | User clicks My Studies button |
| 3 | User’s enrolled units are shown on the page |
| 4 | User logs out |
| Extends | -- | |
| Includes | Login | |

Table2: View Enrolled Units of Current Semester use case

|  |  |  |
| --- | --- | --- |
| Name | View Unit Descriptions | |
| Goal | View the descriptions of units | |
| Level | System | |
| Precondition | User at the My Studies page or Add Units page | |
| Success Condition | the descriptions of the chosen unit are shown on the page | |
| Failure Condition | Fail to get the unit’s descriptions | |
| Actors | Student | |
| Trigger | Click the unit name | |
| Description | Step | Actions |
| 1 | User logs into Learning Management System |
| 2 | User clicks My Studies button |
| 3 | User clicks unit name |
| 4 | User logs out |
| Extends | -- | |
| Includes | Login | |

Table3: View Unit Descriptions use case

|  |  |  |
| --- | --- | --- |
| Name | Add Units | |
| Goal | Add units of the current semester | |
| Level | System | |
| Precondition | User at the My Studies page | |
| Success Condition | User’s selected units are added to the current semester | |
| Failure Condition | Fail to add user’s selected units to the current semester | |
| Actors | Student | |
| Trigger | Click the Add Units button | |
| Description | Step | Actions |
| 1 | User logs into Learning Management System |
| 2 | User clicks My Studies button |
| 3 | User clicks Add Units button |
| 4 | User selects unit from the drop-down box |
| 5 | User clicks the Confirm Add Units button |
| 6 | User logs out |
| Extends | -- | |
| Includes | Login, View Enrolled Units of Current Semester | |

Table4: Add Units use case

|  |  |  |
| --- | --- | --- |
| Name | Drop Units | |
| Goal | Drop units of the current semester | |
| Level | System | |
| Precondition | User at the My Studies page | |
| Success Condition | User’s selected units are dropped from the current semester | |
| Failure Condition | Fail to drop user’s selected units from the current semester | |
| Actors | Student | |
| Trigger | Click the Drop Units button | |
| Description | Step | Actions |
| 1 | User logs into Learning Management System |
| 2 | User clicks My Studies button |
| 3 | User clicks Drop Units button |
| 4 | User checks the check box of each selected unit |
| 5 | User clicks the Confirm Drop Units button |
| 6 | User logs out |
| Extends | -- | |
| Includes | Login, View Enrolled Units of Current Semester | |

Table5: Drop Units use case

|  |  |  |
| --- | --- | --- |
| Name | Edit Unit Descriptions | |
| Goal | Update the unit descriptions | |
| Level | System | |
| Precondition | User at the Manage Units Page | |
| Success Condition | The unit descriptions are updated | |
| Failure Condition | Fail to update the unit descriptions | |
| Actors | Administrator | |
| Trigger | Press the unit name | |
| Description | Step | Actions |
| 1 | User logs into Learning Management System |
| 2 | User clicks Manage Unit button |
| 3 | User clicks the unit name that the use wants to edit |
| 4 | User inputs the unit descriptions |
| 5 | User clicks the confirm button |
| 6 | User logs out |
| Extends | -- | |
| Includes | Login | |

Table6: Edit Unit Descriptions use case

|  |  |  |
| --- | --- | --- |
| Name | Allocate Lecturer | |
| Goal | Allocate lecturers for units | |
| Level | System | |
| Precondition | User at the Manage Units Page | |
| Success Condition | The lecturer is allocated to the unit | |
| Failure Condition | Fail to allocate lecturer to the unit | |
| Actors | Administrator | |
| Trigger | Press the unit name | |
| Description | Step | Actions |
| 1 | User logs into Learning Management System |
| 2 | User clicks Manage Unit button |
| 3 | User clicks the unit name that the user wants to allocate |
| 4 | User inputs the lecturer information |
| 5 | User clicks the confirm button |
| 6 | User logs out |
| Extends | -- | |
| Includes | Login | |

Table7: Allocate Lecturer use case

|  |  |  |
| --- | --- | --- |
| Name | Add Units (admin) | |
| Goal | Add units to the system | |
| Level | System | |
| Precondition | User at the Manage Units page | |
| Success Condition | The unit is added to the system | |
| Failure Condition | Fail to add the unit to the system | |
| Actors | Administrator | |
| Trigger | Click the Add Units button | |
| Description | Step | Actions |
| 1 | User logs into Learning Management System |
| 2 | User clicks Manage Unit button |
| 3 | User clicks Add Units button |
| 4 | User input the unit information |
| 5 | User clicks the Confirm button |
| 6 | User logs out |
| Extends | -- | |
| Includes | Login | |

Table8: Add Units (admin) use case

|  |  |  |
| --- | --- | --- |
| Name | Drop Units (admin) | |
| Goal | Drop units from the system | |
| Level | System | |
| Precondition | User at the Manage Units page | |
| Success Condition | The unit is deleted from the system | |
| Failure Condition | Fail to delete the unit from the system | |
| Actors | Administrator | |
| Trigger | Click the Drop Units button | |
| Description | Step | Actions |
| 1 | User logs into Learning Management System |
| 2 | User clicks Manage Unit button |
| 3 | User clicks Drop Units button |
| 4 | User checks the checkboxes of the units that want to delete |
| 5 | User clicks the Confirm button |
| 6 | User logs out |
| Extends | -- | |
| Includes | Login | |

Table9: Drop Units (admin) use case

|  |  |  |
| --- | --- | --- |
| Name | Enrol Students to Units | |
| Goal | Manually enrol students to units | |
| Level | System | |
| Precondition | User at the Manage Units page | |
| Success Condition | The student is enrolled to selected units | |
| Failure Condition | Fail to enrol student to selected units | |
| Actors | Administrator | |
| Trigger | Click the Manage Enrol button | |
| Description | Step | Actions |
| 1 | User logs into Learning Management System |
| 2 | User clicks Manage Enrol button |
| 3 | User inputs student id to search student |
| 4 | User selects units for the student |
| 5 | User clicks the Confirm button |
| 6 | User logs out |
| Extends | -- | |
| Includes | Login | |

Table10: Enrol Students to Units use case

|  |  |  |
| --- | --- | --- |
| Name | View Lecture Materials | |
| Goal | See specific lecture materials | |
| Level | System | |
| Precondition | User has a student ID and password and have enrolled in this course. User logs in. | |
| Success Condition | Materials of this course can be viewed by user | |
| Failure Condition | Materials cannot be viewed | |
| Actors | Students | |
| Trigger | User clicks the lecture material button | |
| Description | Step | Actions |
| 1 | User logs into Learning Management System |
| 2 | User selects material sharing function |
| 3 | User selects lecture materials function |
| 4 | User selects a material by name |
| 5 | User logs out |
| Extends | -- | |
| Includes | Login | |

Table11: View Lecture Materials use case

|  |  |  |
| --- | --- | --- |
| Name | Edit Lecture Materials | |
| Goal | Edit specific lecture materials | |
| Level | System | |
| Precondition | User has a Teacher ID and password and have enrolled in this course. User logs in. | |
| Success Condition | Materials of this course can be edited by user | |
| Failure Condition | Materials can’t be edited | |
| Actors | Teacher | |
| Trigger | User clicks the lecture material button | |
| Description | Step | Actions |
| 1 | User logs into Learning Management System |
| 2 | User selects material sharing function |
| 3 | User selects lecture materials function |
| 4 | User selects a material by name |
| 5 | Specified material is displayed |
| 6 | Select edit function |
| 7 | User edits the name and description of the material |
| 8 | Save and quit |
| 9 | User logs out |
| Extends | -- | |
| Includes | Login | |

Table12: Edit Lecture Materials use case

|  |  |  |
| --- | --- | --- |
| Name | Delete Lecture Materials | |
| Goal | Delete specific lecture materials | |
| Level | System | |
| Precondition | User has a Teacher ID and password and have enrolled in this course. User logs in. | |
| Success Condition | The Specific material is deleted and the name is not in the name list of lecture materials. | |
| Failure Condition | The specific material can’t be deleted | |
| Actors | Teachers | |
| Trigger | User clicks the lecture materials button | |
| Description | Step | Actions |
| 1 | User logs into Learning Management System |
| 2 | User selects material sharing function |
| 3 | User selects lecture materials function |
| 4 | User selects a material by name |
| 5 | Specified material is displayed |
| 6 | Select delete function |
| 7 | User logs out |
| Extends | -- | |
| Includes | Login | |

Table13: Delete Lecture Materials use case

|  |  |  |
| --- | --- | --- |
| Name | Upload Lecture Materials | |
| Goal | Upload specific lecture materials | |
| Level | System | |
| Precondition | User has a Teacher ID and password and have enrolled in this course. User logs in. | |
| Success Condition | Materials of this course can be edited by user | |
| Failure Condition | Materials can’t be edited | |
| Actors | Teachers | |
| Trigger | User clicks the lecture material button | |
| Description | Step | Actions |
| 1 | User logs into Learning Management System |
| 2 | User selects material sharing function |
| 3 | User selects lecture materials function |
| 4 | User selects “new” function |
| 5 | User edits the name and description of the material |
| 6 | User uploads the material |
| 7 | Save and quit |
| 8 | User logs out |
| Extends | -- | |
| Includes | Login, Upload files | |

Table14: Upload Lecture Materials use case

|  |  |  |
| --- | --- | --- |
| Name | View Unit Announcement | |
| Goal | See specific announcement in announcement page | |
| Level | System | |
| Precondition | User has a student ID and password and have enrolled in this course. User logs in. | |
| Success Condition | Announcements of this course can be viewed by user | |
| Failure Condition | Announcements can’t be viewed | |
| Actors | Students | |
| Trigger | User clicks the announcement button | |
| Description | Step | Actions |
| 1 | User logs into Learning Management System |
| 2 | User selects material sharing function |
| 3 | User selects announcement function |
| 4 | User selects a material by name |
| 5 | User logs out |
| Extends | -- | |
| Includes | Login | |

Table15: View Unit Announcement use case

|  |  |  |
| --- | --- | --- |
| Name | Edit Unit Announcement | |
| Goal | Edit announcement materials | |
| Level | System | |
| Precondition | User has a Teacher ID and password and have enrolled in this course; User logs in. | |
| Success Condition | Announcement can be edited by user | |
| Failure Condition | Announcement can’t be edited | |
| Actors | Teachers | |
| Trigger | User clicks the announcement button | |
| Description | Step | Actions |
| 1 | User logs into Learning Management System |
| 2 | User selects material sharing function |
| 3 | User selects announcement function |
| 4 | User selects a material by name |
| 5 | Specified material is displayed |
| 6 | Select edit function |
| 7 | User edits the name and description of the material |
| 8 | Save and quit |
| 9 | User logs out |
| Extends | -- | |
| Includes | Login | |

Table16: Edit Unit Announcement use case

|  |  |  |
| --- | --- | --- |
| Name | Delete Unit Announcement | |
| Goal | Delete announcement materials | |
| Level | System | |
| Precondition | User has a Teacher ID and password and have enrolled in this course User logs in. | |
| Success Condition | The announcement is deleted and not in the list. | |
| Failure Condition | Announcements can’t be deleted | |
| Actors | Teachers | |
| Trigger | User clicks the announcement button | |
| Description | Step | Actions |
| 1 | User logs into Learning Management System |
| 2 | User selects material sharing function |
| 3 | User selects announcement function |
| 4 | User selects a material by name |
| 5 | Specified material is displayed |
| 6 | Select delete function |
| 7 | User logs out |
| Extends | -- | |
| Includes | Login | |

Table17: Delete Unit Announcement use case

|  |  |  |
| --- | --- | --- |
| Name | Upload Unit Announcement | |
| Goal | Upload announcement materials | |
| Level | System | |
| Precondition | User has a Teacher ID and password and have enrolled in this course. User logs in. | |
| Success Condition | materials of this course can be edited by user | |
| Failure Condition | materials can’t be edited | |
| Actors | Teachers | |
| Trigger | User clicks the announcement button | |
| Description | Step | Actions |
| 1 | User logs into Learning Management System |
| 2 | User selects material sharing function |
| 3 | User selects announcement function |
| 4 | User selects new material function |
| 5 | User edits the name and description of the material |
| 6 | User uploads the material |
| 7 | Save and quit |
| 8 | User logs out |
| Extends | -- | |
| Includes | Login, Upload files | |

Table18: Upload Unit Announcement use case

|  |  |  |
| --- | --- | --- |
| Name | Submit Assignment | |
| Goal | Assignment is submitted to e-learning system | |
| Level | System | |
| Precondition | User has a Student ID and password and have enrolled in this course. User logs in. | |
| Success Condition | Assignment of this course can be submitted by user to system | |
| Failure Condition | Assignment cannot be submitted | |
| Actors | Students | |
| Trigger | User clicks the assignment button | |
| Description | Step | Actions |
| 1 | User logs into Learning Management System |
| 2 | User selects assignment submission function |
| 3 | User selects the name of specific assignment |
| 4 | User uploads the file(s) |
| 5 | User submits the files |
| 6 | The name of submitted files is displayed |
| 7 | User logs out |
| Extends | -- | |
| Includes | Login, Upload files | |

Table19: Submit Assignment use case

|  |  |  |
| --- | --- | --- |
| Name | Download Assignment | |
| Goal | Set a new entrance for student to submit the specific assignment | |
| Level | System | |
| Precondition | User has a teacher ID and password and has enrolled in the specific course. User logs in. | |
| Success Condition | The specific assignment can be downloaded | |
| Failure Condition | The specific assignment can’t be downloaded | |
| Actors | Students or teachers | |
| Trigger | User clicks the assignment button | |
| Description | Step | Actions |
| 1 | User logs into Learning Management System |
| 2 | User selects assignment download function |
| 3 | User selects the name and SID of students |
| 4 | User downloads the assignments of these students |
| 5 | User logs out |
| Extends | -- | |
| Includes | Login, Download files | |

Table20: Download Assignment use case

|  |  |  |
| --- | --- | --- |
| Name | Upload Files | |
| Goal | The specific file is uploaded | |
| Level | System | |
| Precondition | User has a student/teacher/admin ID and password and a specific file in computer. User logs in. | |
| Success Condition | Specific file is uploaded | |
| Failure Condition | Specific file cannot be uploaded | |
| Actors | Users | |
| Trigger | User clicks the upload button | |
| Description | Step | Actions |
| 1 | User selects upload function |
| 2 | The file browser is open |
| 3 | User selects the name of specific file(s) |
| 4 | The name of specific file(s) is displayed when they upload completely |
| Extends | -- | |
| Includes | -- | |

Table21: Upload Files use case

|  |  |  |
| --- | --- | --- |
| Name | Download Files | |
| Goal | The specific file is downloaded | |
| Level | System | |
| Precondition | User has a student/teacher/admin ID and password and a specific file in computer. User logs in. | |
| Success Condition | Specific file is downloaded | |
| Failure Condition | Specific file cannot be downloaded | |
| Actors | Users | |
| Trigger | User clicks the download button | |
| Description | Step | Actions |
| 1 | User selects download function |
| 2 | The file browser is open |
| 3 | User selects the file to preserve the file(s) |
| 4 | Notify the user when they are downloaded completely |
| Extends | -- | |
| Includes | -- | |

Table22: Download Files use case

|  |  |  |
| --- | --- | --- |
| Name | Mark Assignment | |
| Goal | Teacher mark unit assignments for students | |
| Level | System | |
| Precondition | User has logged into system | |
| Success Condition | Teacher marked the unit assignments for students | |
| Failure Condition | Fail to mark the unit assignment for students | |
| Actors | Teacher | |
| Trigger | Click the “Marking” button | |
| Description | Step |  |
| 1 | User logs into Learning Management System |
| 2 | User clicks “Marking” button |
| 3 | User selects the unit |
| 4 | User selects the assignment |
| 5 | User inputs the marks for students |
| 6 | User clicks the “Submit” button |
| 7 | User logs out |
| Extends | -- | |
| Includes | Login | |

Table23: Mark Assignment use case

|  |  |  |
| --- | --- | --- |
| Name | View grade | |
| Goal | Student view their marks of unit | |
| Level | System | |
| Precondition | User has logged into system | |
| Success Condition | The marks shown on the page | |
| Failure Condition | Fail to view the marks of units | |
| Actors | Student | |
| Trigger | Click the “My Grades” button | |
| Description | Step | Actions |
| 1 | User logs into Learning Management System |
| 2 | User clicks the “My Grades” button |
| 3 | User selects the unit and views the marks of that unit |
| 4 | User logs out |
| Extends | -- | |
| Includes | Login | |

Table24: View Grade use case

|  |  |  |
| --- | --- | --- |
| Name | Send Emails | |
| Goal | Send emails to students to inform the class updates | |
| Level | System | |
| Precondition | User is allocated to the unit | |
| Success Condition | Send emails to selected students | |
| Failure Condition | Fail to send emails to selected students | |
| Actors | Teacher | |
| Trigger | Click the "Send Email" button | |
| Description | Step | Actions |
| 1 | User logs into the Learning Management System |
| 2 | User clicks the “Send Email” button |
| 3 | User selects the enrolled students that wants to inform |
| 4 | User inputs the massage that wants to send |
| 5 | User clicks the “Send” button |
| 6 | User logs out |
| Extends | -- | |
| Includes | Login | |

Table25: Send Emails use case

|  |  |  |
| --- | --- | --- |
| Name | Manage Program | |
| Goal | Set units, credits and the number of units for the selected program | |
| Level | System | |
| Precondition | Administrator user logs into the system | |
| Success Condition | Set the properties for the program | |
| Failure Condition | Fail to set the properties for the program | |
| Actors | Administrator | |
| Trigger | Click the "Manage Program" button | |
| Description | Step | Actions |
| 1 | User logs into the Learning Management System |
| 2 | User clicks the “Manage Program” button |
| 3 | User selects the program that wants to set |
| 4 | User selects the units for the selected program |
| 5 | User sets the credit for this program that enrolled students have to complete to get the degree |
| 6 | User sets the number of units that enrolled students can choose in a semester |
| 7 | User clicks the “Confirm” button |
| 8 | User logs out |
| Extends | -- | |
| Includes | Login | |

Table26: Manage Program use case

* 1. **Use Case Diagram (see in attaches)**

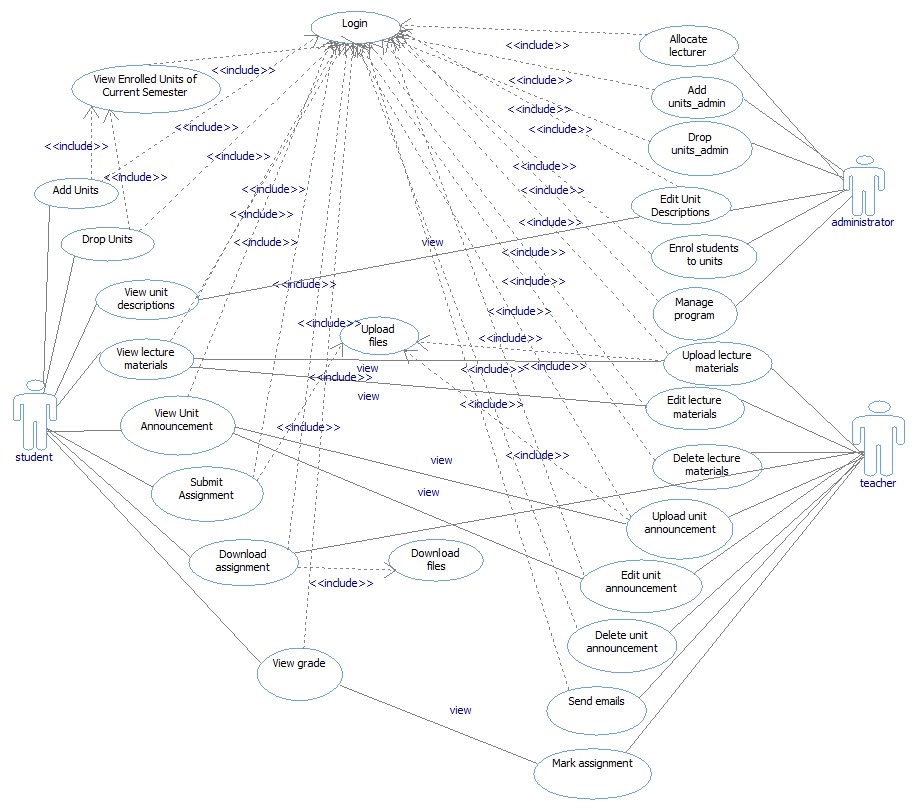


Figure2: Use Case Diagram

* 1. **Class Diagram (see in attaches)**

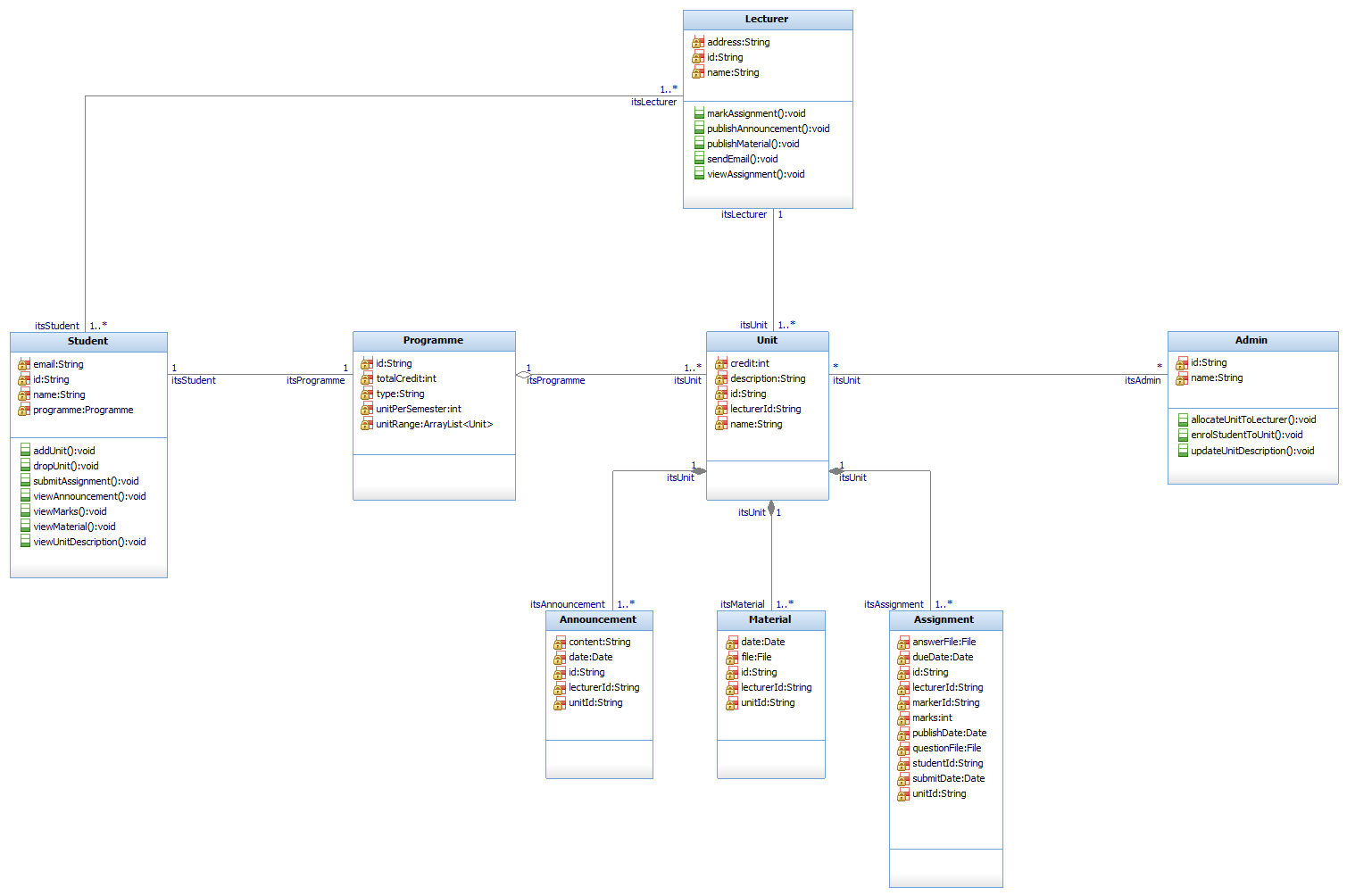


Figure3: Class Diagram

|  |  |
| --- | --- |
| Student | |
| id:String | Primary key |
| name:String | Student’s name |
| email:String | Student’s email |
| programme:Programme | Programme the student enrolled |
| viewUnitDescription() | Students can view the description of all available units of study. |
| addUnit() | Students can add units of study. |
| dropUnit() | Students can drop units of study. |
| viewAnnouncement() | After enrolment of a unit of study, students can view announcement. |
| viewMaterial() | After enrolment of a unit of study, students can view teaching materials. |
| viewMarks() | After enrolment of a unit of study, students can view marks. |
| submitAssignment() | Students can submit their assignments via the system. |

Table27: Student class

|  |  |
| --- | --- |
| Lecturer | |
| id:String | Primary key |
| name:String | Lecturer’s name |
| address:String | Lecturer’s office |
| publishAnnouncement() | Lecturers can publish announcements via the system. |
| publishMaterial() | Lecturers can publish learning materials via the system. |
| viewAssignment() | Lecturers can view the assignments of their units. |
| markAssignment() | Lecturers can mark assignments in the system. |
| sendEmail() | Lecturers can send group/individual emails to notify enrolled students about any class updates. |

Table28: Lecturer class

|  |  |
| --- | --- |
| Admin | |
| id:String | Primary key |
| name:String | Admin’s name |
| updateUnitDescription() | Admin can update the description of all units of study. |
| allocateUnitToLecturer() | Admin can allocate units of study to lecturers. |
| enrolStudentToUnit() | Admin can manually enrol students to units of study. |

Table29: Admin class

|  |  |
| --- | --- |
| Programme | |
| id:String | Primary key |
| type:String | Type of the programme |
| unitRange:ArrayList<Unit> | Range of the units of study that a student can enrol |
| totalCredit:int | Total credits that a student has to complete to get the degree |
| unitPerSemester:int | Number of units a student can enrol in a semester |
|  |  |

Table30: Programme class

|  |  |
| --- | --- |
| Unit | |
| id:String | Primary key |
| name:String | Name of the unit of study |
| credit:int | Credit of the unit of study |
| description:String | Description of the unit of study |
| lecturerId:String | ID of the lecturer who teaches the unit of study |
|  |  |

Table31: Unit class

|  |  |
| --- | --- |
| Announcement | |
| id:String | Primary key |
| unitId:String | ID of the unit of study the announcement belongs to |
| lecturerId:String | ID of the lecturer who publishes the annoucement |
| content:String | Content of the announcement |
| date:Date | Date of the announcement is published |
|  |  |

Table32: Announcement class

|  |  |
| --- | --- |
| Material | |
| id:String | Primary key |
| unitId:String | ID of the unit of study the material belongs to |
| lecturerId:String | ID of the lecturer who publishes the material |
| file:File | File of the material |
| date:Date | Date of the material is published |
|  |  |

Table33: Material class

|  |  |
| --- | --- |
| Assignment | |
| id:String | Primary key |
| unitId:String | ID of the unit of study the assignment belongs to |
| lecturerId:String | ID of the lecturer who publishes the assignment |
| publishDate:Date | Date of the assignment is published |
| questionFile:File | Question of the assignment |
| dueDate:Date | Due date of the assignment |
| studentId:String | ID of the student who submits the assignment |
| answerFile:File | Answer to the assignment |
| submitDate:Date | Date of submission |
| markerId:String | ID of the lecturer who marks the assignment |
| marks:int | Marks of the answer |
|  |  |

Table34: Assignment class

1. **Modeling the dynamic aspects of the system**
   1. **State Chart Diagrams**
      1. **State Chart for Lecture Materials Management**

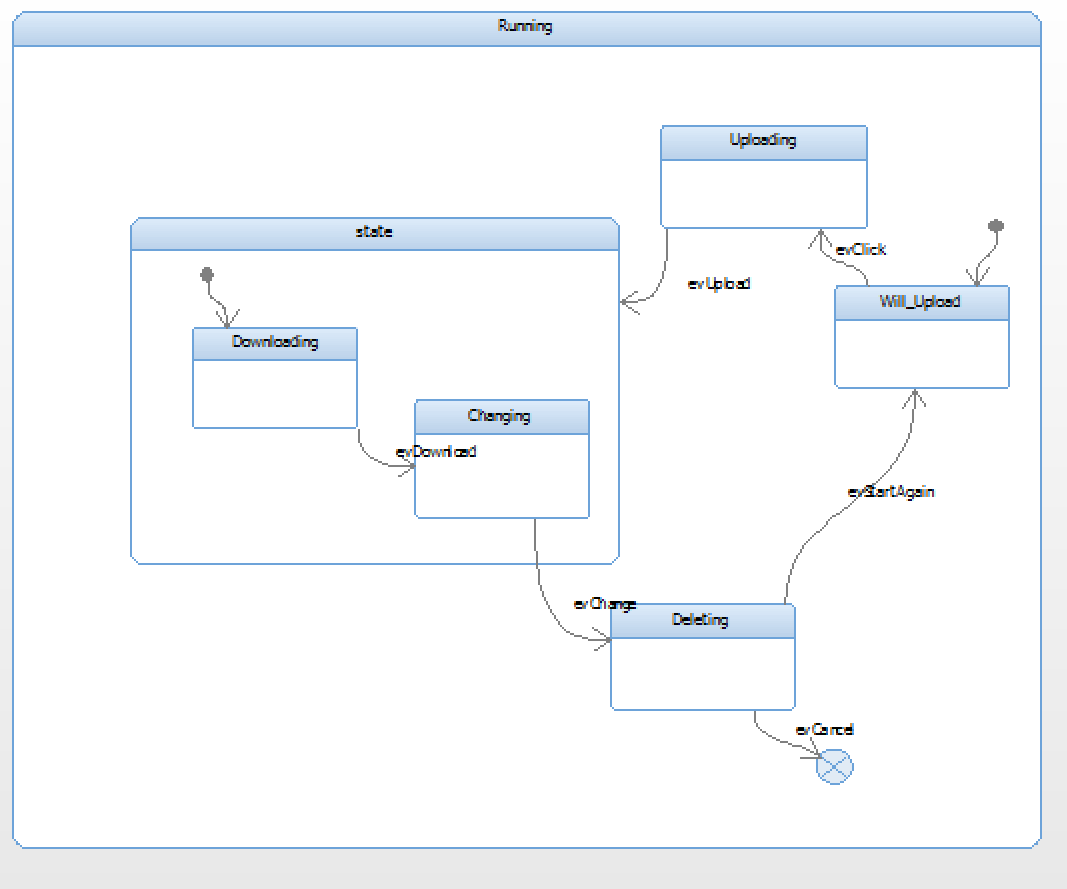


Figure4: State Chart Diagram for Lecture Materials Management

This state chart for lecture materials management is divided into four parts: will upload, uploading, downloading, changing and deleting. The first state is that lecture materials will upload to the learning management system. The second state is that Sydney students have already viewed these uploaded lecture materials through clicking the button. Then the third state should happen concurrently, that means, Sydney students should download or change these uploaded lecture materials concurrently through triggering different buttons. The last state is that Sydney students should delete these uploaded lecture materials. Then Sydney students should exit the LMS system or start these steps again.

* + 1. **State Chart for Unit Management**

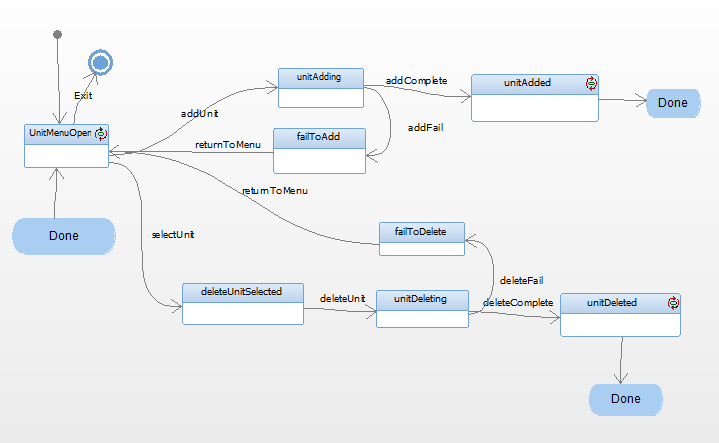


Figure5: State Chart Diagram for Unit Management

To begin with, at the UnitMenuOpen state, choosing addUnit or selectUnit trigger to transit into unitAdding state or deleteUnitSelected state. At unitAdding state, if the adding process is completed, through addComplete trigger transit into unitAdded state and then the process finish. If the adding process is failed, through addFail trigger transit into failToAdd state, and through returnToMenu state return back to UnitMenuOpen state. For delete unit process, from deleteUnitSelected state, when deleteUnit is triggered, the process will transit to unitDeleting state. If the deleting process is succeeded, the process will transit to unitDeleted state through deleteComplete trigger. If the deleting process is failed, the process will transit to failToDelete through failToDelete state and return to UnitMenuOpen state through returnToMenu state.

* + 1. **State Chart for Assignment Submission**

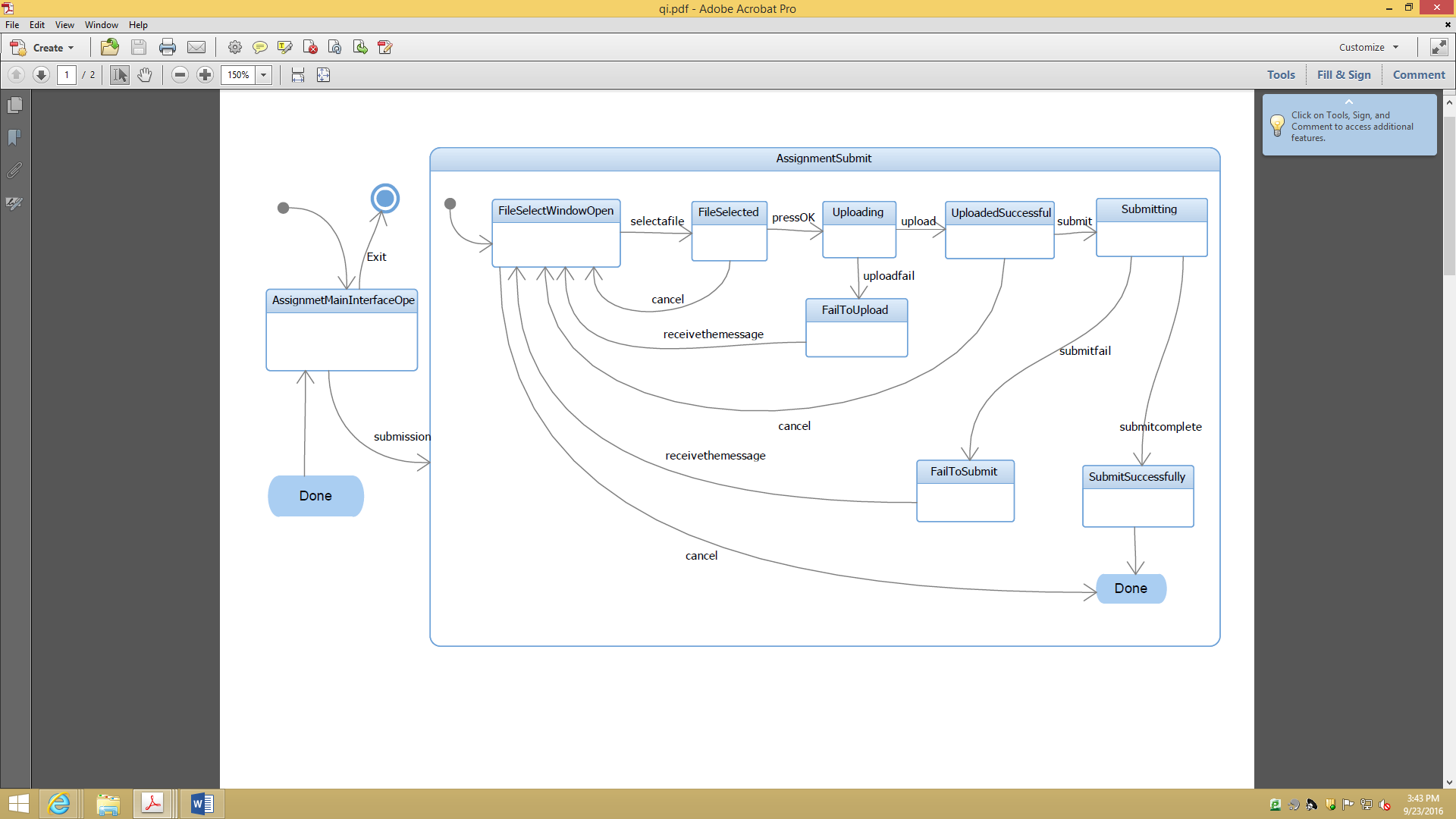


Figure6: State Chart Diagram for Assignment Submission

The state chart is for assignment submission function. Users can start with assignment main interface to enter the function by clicking submission button. After that, the state goes to assignment submit state. The first step is to open file select window for users to select the file which need to be upload. Then, the state goes to file selected. After pressing OK button by users, the system goes to uploading state and then will goes to uploaded success state. After pressing submit button by users, it is submitting and then submit successfully to jump out of assignment submit state and jump to assignment main interface open state. All interrupts in assignment submit state will lead to first state of assignment submit as shown in the chart.

* + 1. **State Chart for Marking**

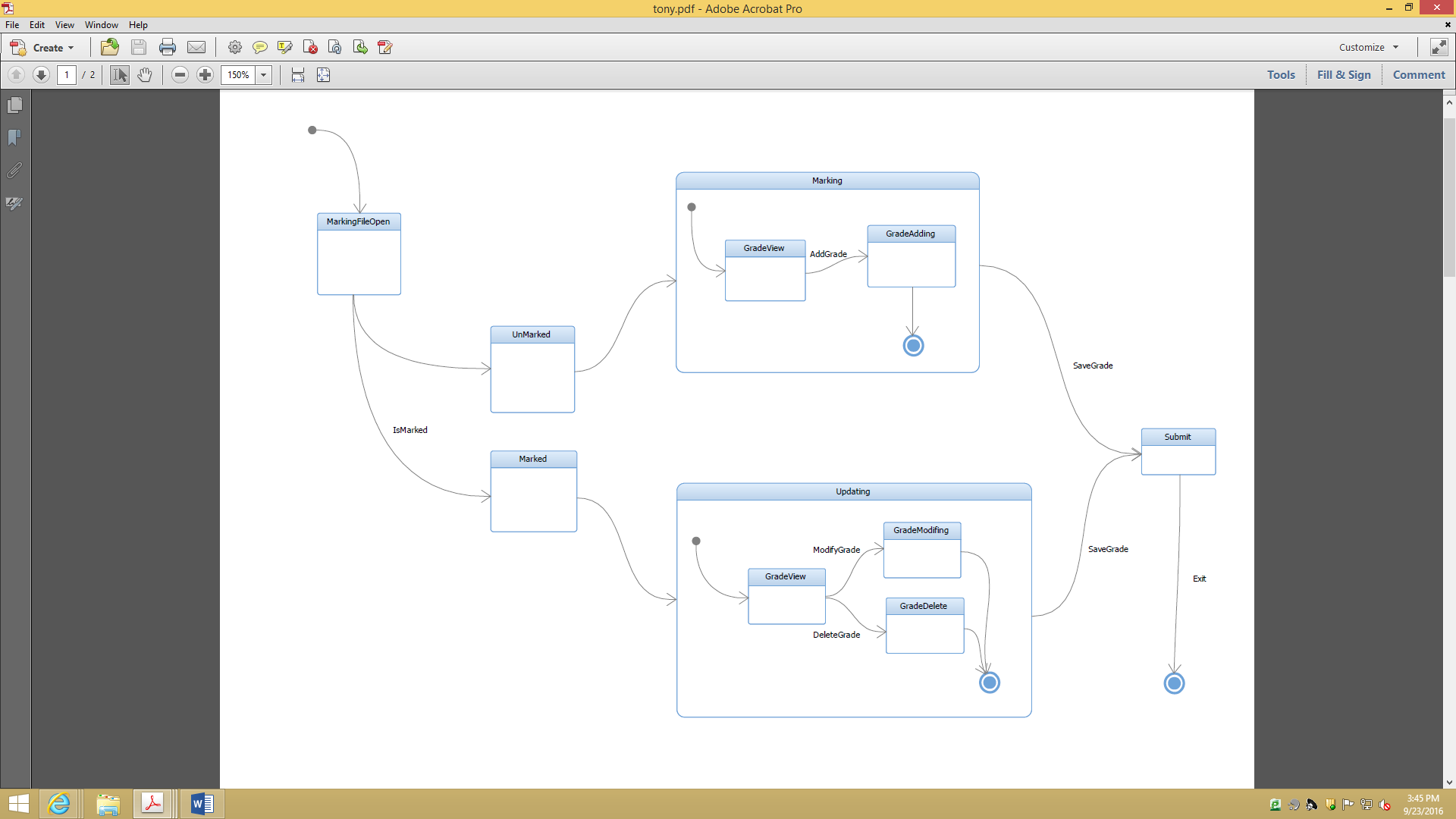


Figure7: State Chart Diagram for Marking

This state diagram describes the states of the marking operation of teachers. The first state is MarkingFileOpen. Teacher open the marking file by clicking the mark label. Then divide into unmarked state and marked state. For the Unmarked state, here we go into the marking state, follow the add grade operation and go to the GradeAdding state. For the marked state, there are two different state base on two different operations, modify and delete. Then, for both marking and updating, click save to submit the new grade into the system.

* + 1. **State Chart for Manually Enrollment**

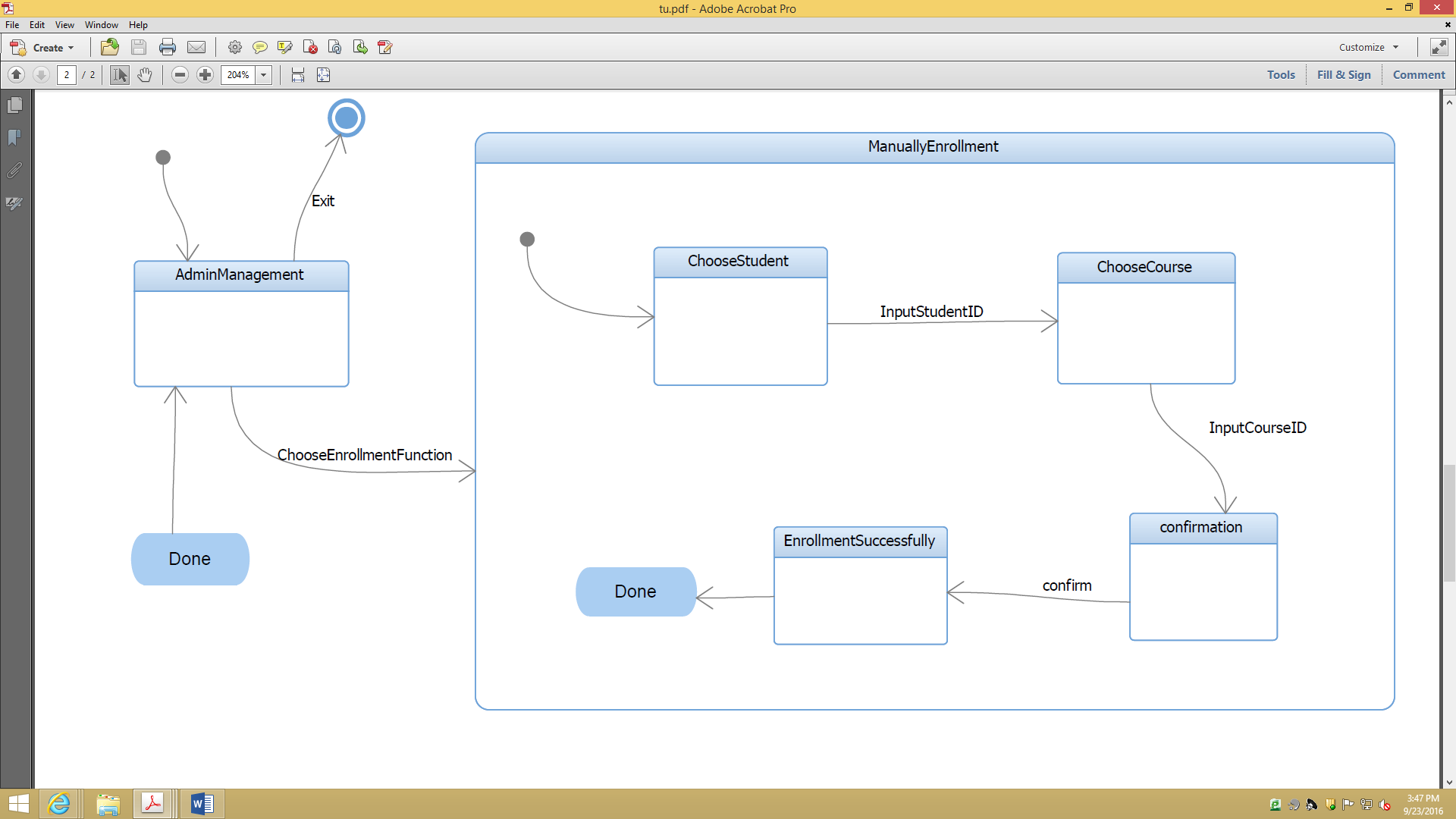


Figure8: State Chart Diagram for Manually Enrollment

The first state is admin management interface open. After clicking enrollment button by users, the manually enrollment state start. The first state in manually enrollment state is student choosing state, the user need to input StudentID to motivate it goes to course choosing state. After CourseID input by users, the confirmation state start and user need to confirm to make it goes to enrollment success state.

* 1. **Activity Diagrams**
     1. **Activity Diagram for Lecture Materials Management**

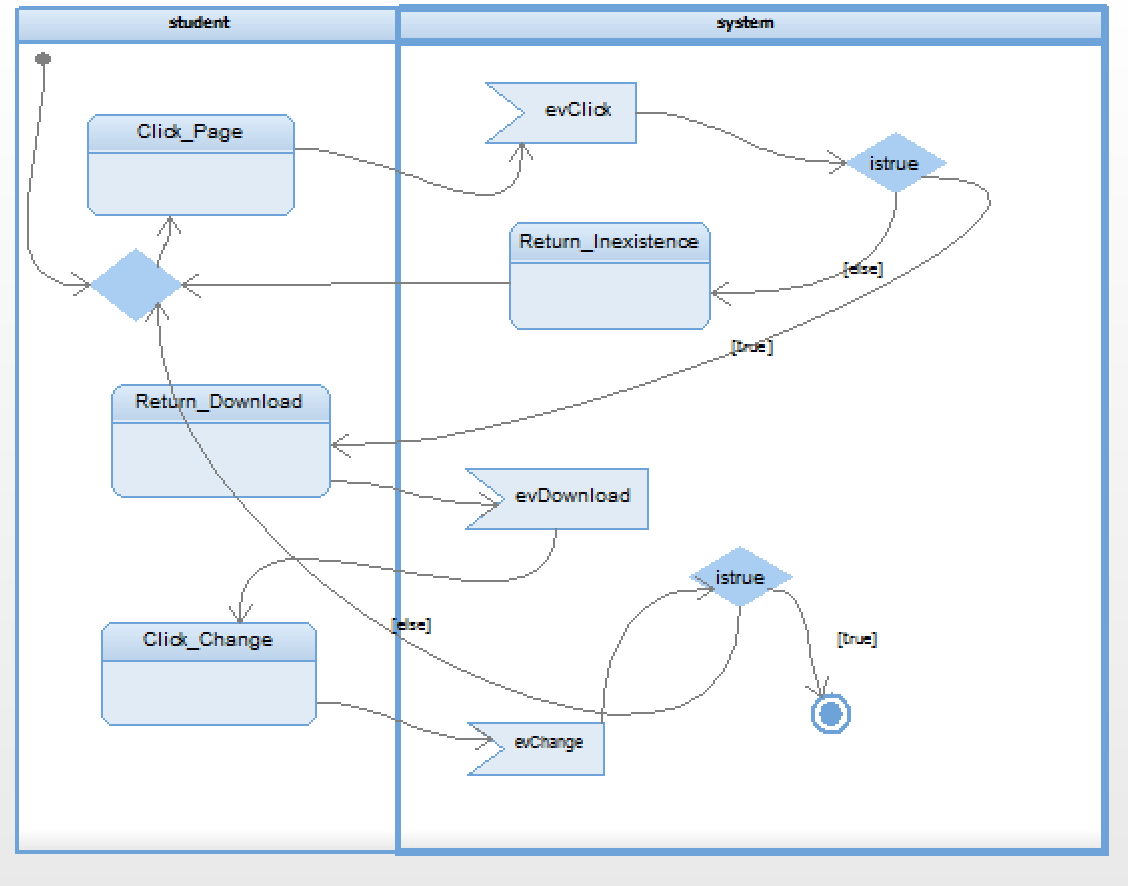


Figure9: Activity Diagram for Lecture Materials Management

This activity diagram describes the interaction for lecture materials management between Sydney students and the LMS system. The first step is that Sydney students should click page to download the uploaded lecture materials through triggering the download button, and if this trigger is true, Sydney students would download lecture materials. Then the second step is that Sydney students should change these downloaded lecture materials through triggering the change button, and if this trigger is true, Sydney students would change these downloaded lecture materials, and then Sydney students should exit the learning management system. However, if all of triggered activities are false to the system, the system would return the original stage.

* + 1. **Activity Diagram for Unit Management**

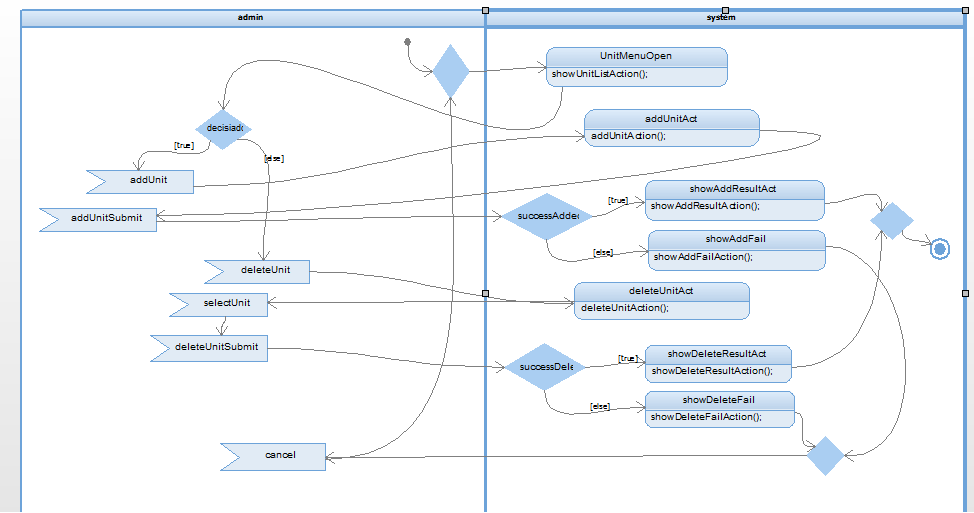


Figure10: Activity Diagram for Unit Management

First after the unitMenuOpen action, if addUnit is triggered, then addUnitAct action is processed. If addUnitSubmit is triggered, and the result is succeeded, the showAddResultAct is processed, and the process is completed. If the result is fail, the showAddFail is processed and back to the beginning.

After the unitMenuOpen action, if deleteUnit is triggered, then deleteUnit action is processed. After that, if the selectUnit and deleteUnitSubmit are triggered, and the process is successfully completed, the showDeleteResultAct will be processed and the process is completed. If the process is failed, the showDeleteFail will be processed and return to the beginning.

* + 1. **Activity Diagram for Assignment Submission**

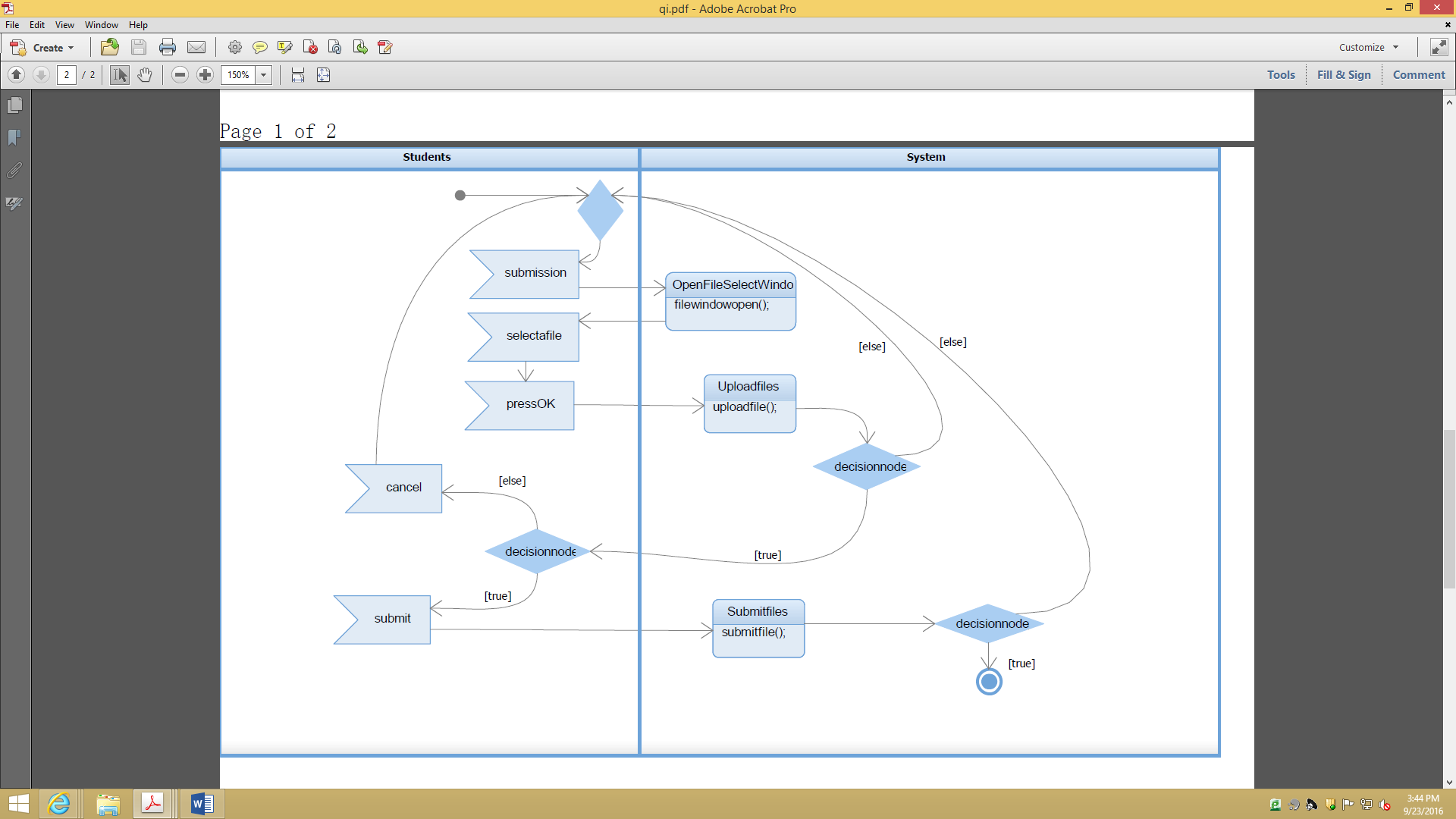


Figure11: Activity Diagram for Assignment Submission

The activity diagram is for assignment submission function. Users can start by clicking submission button. After that, the system will conduct an activity which is to open file select window for users to select the file which need to be upload. Then, users take an action to select files. After pressing OK button by users, the system will upload files. Users can press cancel button to jump to the first action or press submit button to make system goes into the activity of submission. If files are submitted successfully, the activity will be end or it will jump to beginning.

* + 1. **Activity Diagram for Marking**

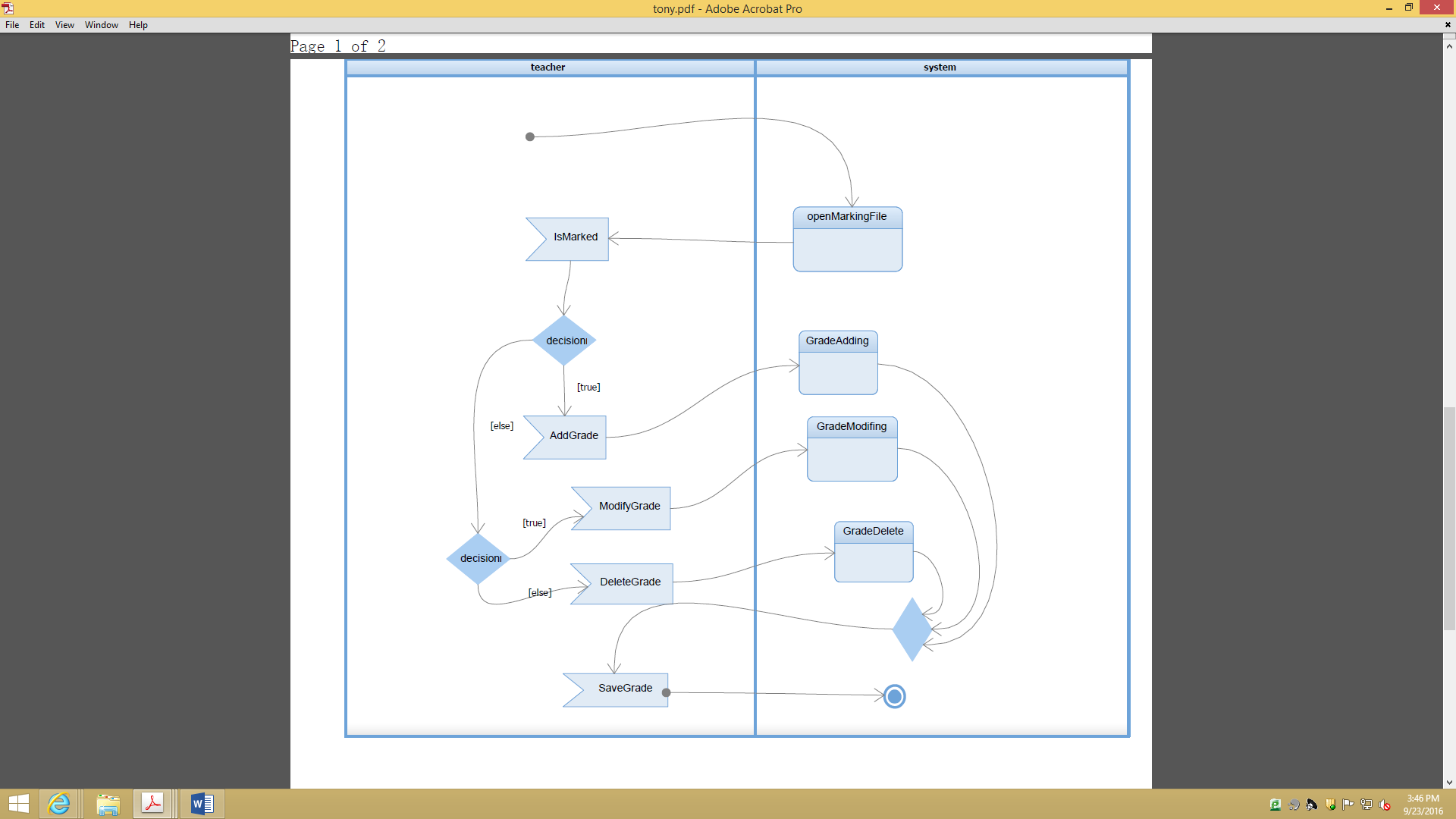


Figure12: Activity Diagram for Marking

This state diagram describes the states of the marking operation of teachers. The first state is MarkingFileOpen. Teacher open the marking file by clicking the mark label. Then divide into unmarked state and marked state. For the Unmarked state, here we go into the marking state, follow the add grade operation and go to the GradeAdding state. For the marked state, there are two different state based on two different operations, modify and delete. Then, for both marking and updating, click save to submit the new grade into the system.

* + 1. **Activity Diagram for Manually Enrollment**

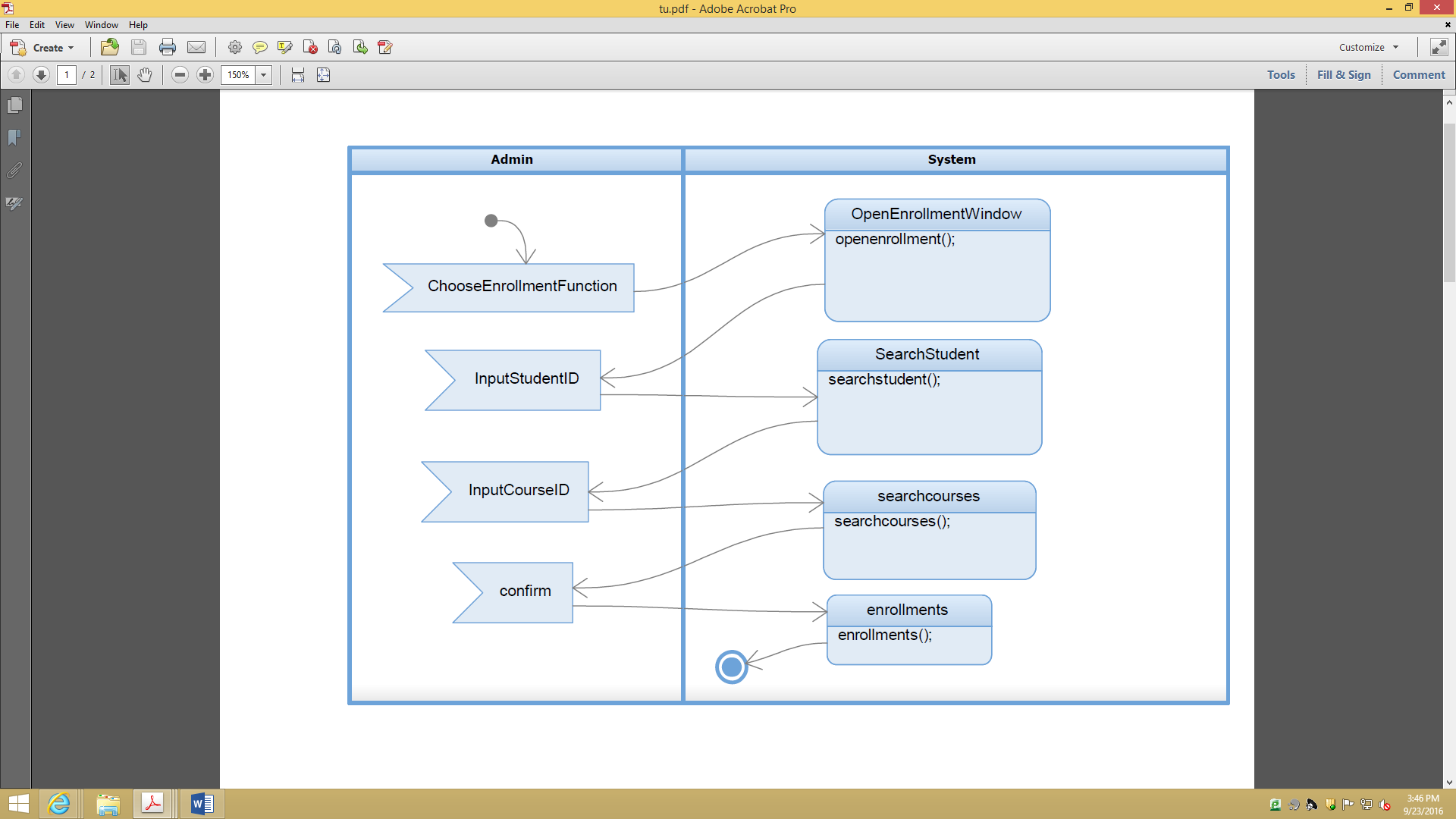


Figure13: Activity Diagram for Manually Enrollment

The first action is to choose enrollment function by users. Then system will take open enrollment function window activity. And user need to input StudentID to motivate system to search student and input CourseID to make system to search courses. Finally, user need to press confirm button to invoke enrollment operation.

1. **Conclusion**

This report introduces the design of Learning Management System’ structure and functionalities through use cases, feature diagram, class diagrams, state diagrams and activity diagrams. It can help reason about the system and answer questions of the system’s structure and model. It also can assist the further implementation.