UNIVERSITÉ Concordia UNIVERSITY

COEN 6312

Model Driven Software Engineering

Project Deliverable 4

State diagram & action specification

Course Instructor

Dr. Wahab Hamou-Lhadj

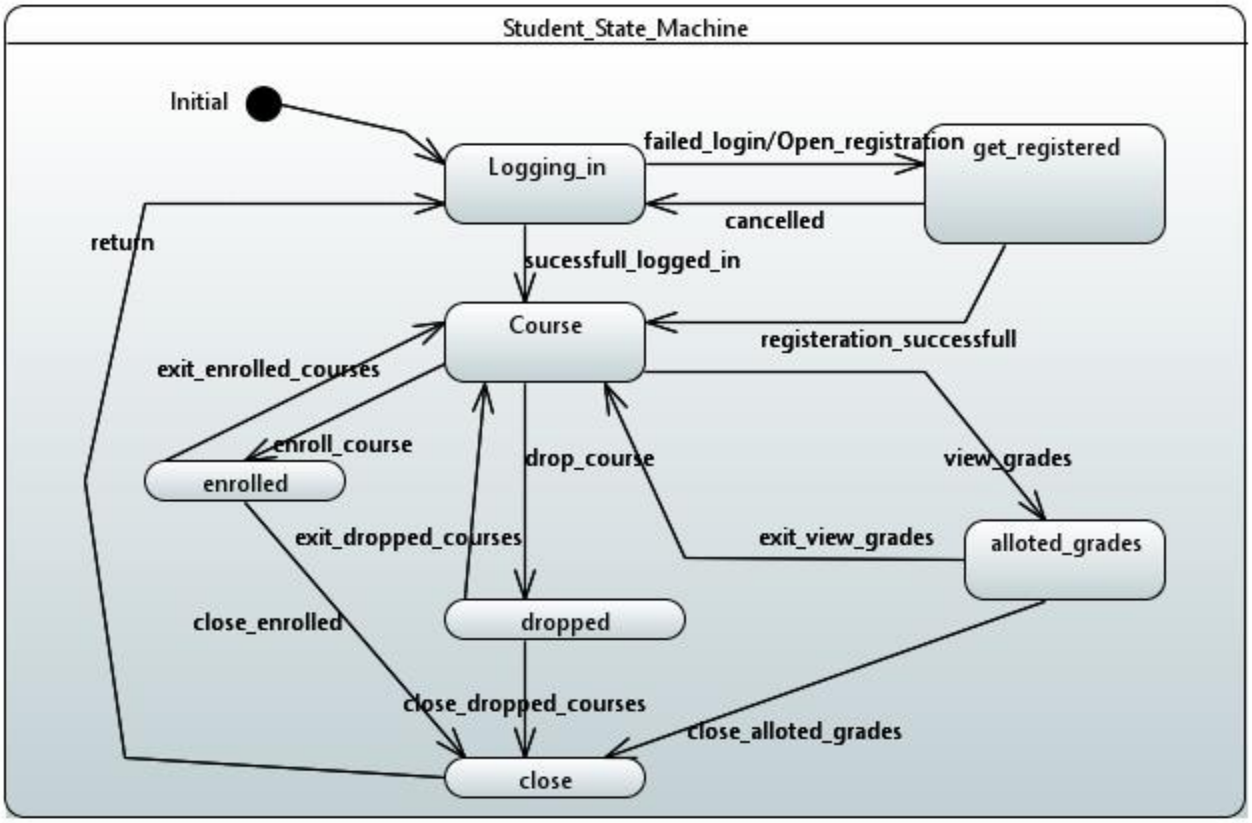
Team: Techno_reg
Project Title: Course Registration System

Submitted by

Farhan Saeed	40039670
Dinesh Pattapu	40070809
Nikitha Papani	40070806
Mohammed Hakeemuddin	40059007
Sai Abishek Thorikonda	40071051

Table of Contents

1. Student State Diagram



Student Class State Machine Diagram (Papyrus)

1.1. State Description

The student class consists of 7 states namely Logging_in state, get_register, Course, Enrolled, alloted_grades, Dropped and Close state. Every state has a transition relationship and will be able to enter the other state only when specified set of events and conditions are satisfied.

Logging_in

It is the student authentication phase i.e. if he/she is a registered student they will be forwarded to the course state, else they need to register themselves with valid details to enter the course state.

Get_registered

If the students are not registered or enters invalid credentials they will enter this state.

Course

After the students have successfully passed the authentication state of previous state they will be able to enter the course state. This state contains list of courses that students are offered.

Enrolled

After students have selected the course of their interest they will be able to enter in the enrolled state. This state provides access to student to enroll in the course of their choice and also contains the list of enrolled courses.

Allotted_grades

Only the enrolled students who have written the final exams will be able to enter this state. This state contains the grades of the students uploaded by the faculty after examination.

Dropped

If the student does not intend to continue the course, he/she can enter the drop state where they can drop the course.

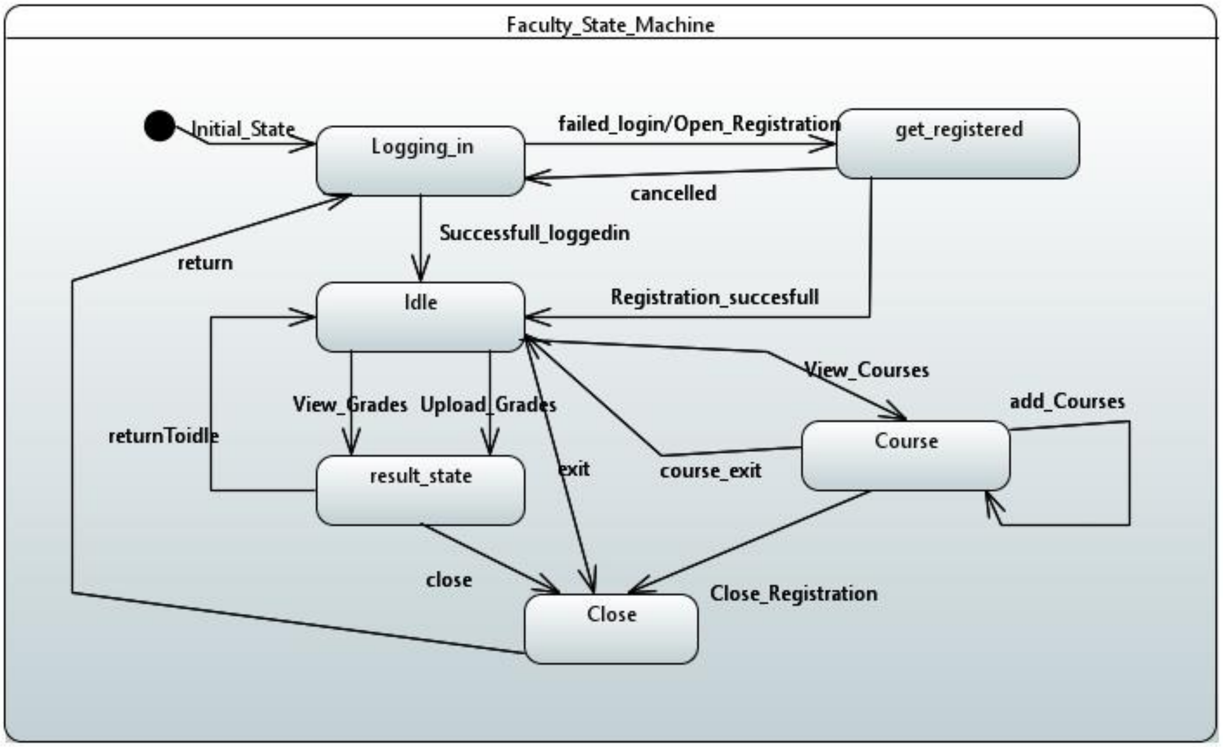
close

Finally, If student want to close their registration module they will enter this state. It will redirect them to start of the start of the module to again re-enter if required or provide access to other students to use the module as intended.

1.2. Action Description

Transition Name	Description
Failed_login/Open_registration	In case of failed login use this transition to get registered
Cancelled	In case of cancelled operation go to start of logging in state
Registration_successfull	To check whether a new student is registered or not
Successfull_logged_in	After successful authentication go to course state
Enroll_course	To get enrolled for a course
Exit_enroll_courses	To exit from enroll courses tab
Drop_course	To drop a course that was enrolled previously
Exit_drop_course	To exit from drop course tab
View_grades	To view grades of past examination
Exit_view_grades	To exit from view grades tab
Close_enrolled	To finish enrolling for a course
Close_dropped_courses	To finish removing courses
Close_alloted_grades	To finish view of past grades
return	To proceed to start of module

2. Faculty State Diagram



Faculty Class State Machine Diagram (Papyrus)

2.1. State Description

The state diagram shows the authentication process and the functionalities of the faculty at an institute. This state diagram consists of 6 states namely Logging_in, Register_Faculty, Idle, result_state, Course and Close state. Every state has a transition relationship and will be able to enter the other state only when specified set of events and conditions are satisfied.

Logging_in

Faculty need to pass the authentication phase by entering their valid credentials. If the faculty is not registered member he/she is not allowed to enter the next states.

Get_registered

If the faculty is not registered they will enter this state. This state contains the list of faculties who have registered to the university.

Idle

After successful authentication faculty will enter this state, from here they can perform transition to state of their choice.

Course

Faculties can enter this state from idle state if they wish to view course list.

Result_state

Faculties can enter this state, to either upload the grades or modify the grades.

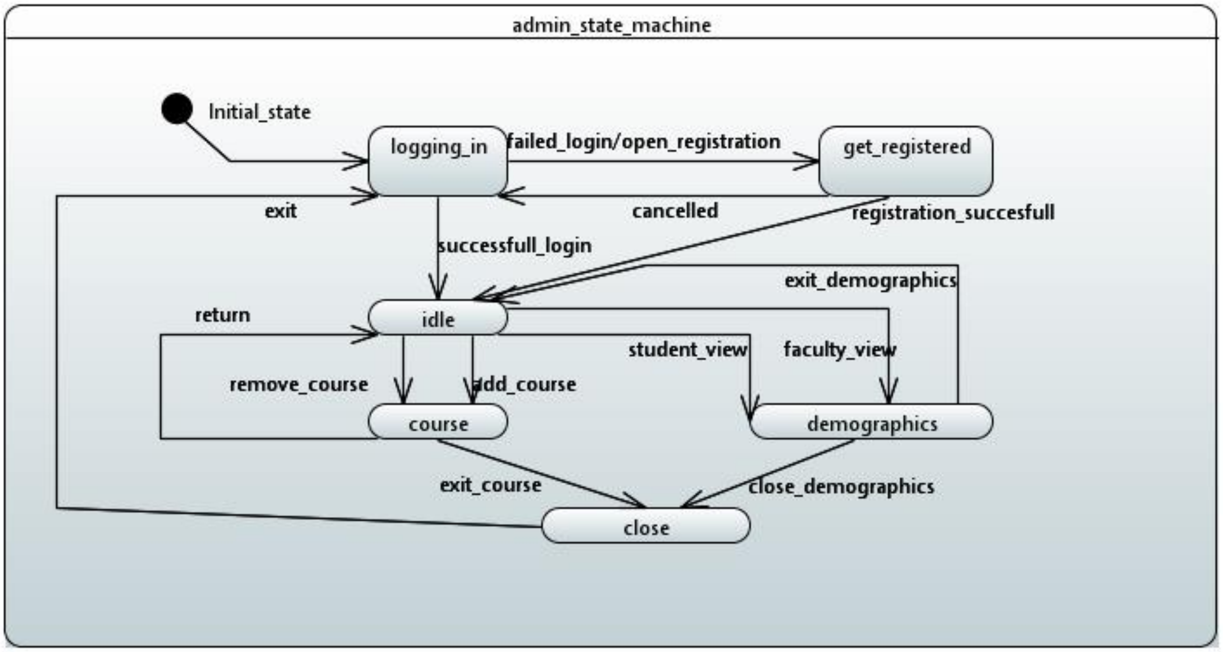
Close state

If the faculty, choose to log out from the module he/she enters close state.

2.2. Action Description

Transition Name	Description
Failed_login/Open_registration	In case of failed login use this transition to get registered
Cancelled	In case of cancelled operation go to start of logging in state
Registration_successfull	To check whether new faculty member is registered or not
Successfull_loggedin	After successful authentication go to idle state
view_courses	To view list of courses available
Course_exit	To exit from courses tab
add_courses	Add courses to be taught, self-transition
Upload_grades	To upload student grades
View_grades	To view grades of students
ReturnToidle	To exit from view grades tab
Close_registration	To finish enrolling for a course
exit	To exit from idle state
Close	To exit from result state
return	To proceed to start of module

3. Admin State Diagram



Admin Class State Machine Diagram (Papyrus)

3.1. State Description

The state diagram shows the authentication process and functionalities of the admin. Every state has a transition relationship and will be able to enter the other state only when specified set of events and conditions are satisfied. The state diagram has 6 states namely Logging_in state, get_registered state, Idle state, Demographics state, Courses state and close state.

Logging_in

The Admin have to pass the authentication phase. If the admin is not registered or entered invalid credentials he/she enters the registered state. After successful login admin enters the idle state.

Get_registered

If the admin is not registered or enters invalid details he/she enters this state. This state contains the details of registered admin.

Idle

After successful login admin enters idle state, from here they can perform transition to state of their choice.

Demographics

Admin can enter this state and view the details of Faculties and Students.

Course

Admin can enter the course state where he/she can either add or remove the courses depending on situation.

Close state

Admin choose to log out from the module he/she enters close state.

3.2. Action Description

Transition Name	Description
Failed_login/Open_registration	In case of failed login use this transition to get registered
Cancelled	In case of cancelled operation go to start of logging in state
Registration_successfull	To check whether new admin member is registered or not
Successfull_loggedin	After successful authentication go to idle state
add_courses	To add new courses to list of courses available
Remove_courses	To remove courses
Return	To exit from courses tab
Student_view	To view student details
Faculty_view	To view faculty details
Exit_demographics	To exit from demographic view and return to idle
Close_demographics	To finish faculty and student detailed view and exit
Exit_course	To finish adding or removing of course and exit
exit	To exit from system

4. Action Specification

4.1. Pseudocode for “logging_in” module of Admin State Diagram

```
If “logging_in” status is true
    Print “logged in successfully”
    Go to idle
Else
    Print “failed_login”
    And If “open_registration” is true
    Go to “get_registered”
End if
```

4.2. Pseudocode for “Enroll_course” module of Student State Diagram

```
If “enroll_course” is true and “course” is not full
    Course must be enrolled
Else if course is full
    Print “course is full”
Else if course is not available
    Print “course not available”
End if
```

4.3. Pseudocode for “add_course” module of Admin State Diagram

```
If “add_course” is true
    Enter course id
    Enter course name
    Enter department name
    Enter class size
    Print “A new course is available to enroll”
Else do nothing
End if
```

4.4. Switch case statement of complete Admin State Diagram

```
Switch (state)
{
case LOGGING_IN:
if (keysuccessful_login) State = IDLE
if(failed_login/open_registration) State = GET_REGISTERED
case IDLE:
if (keystudent_view) State= DEMOGRAPHICS
if (keyfaculty_view) State= DEMOGRAPHICS
if (keyadd_course) State= COURSE
if (keyremove_course) State = COURSE
case GET_REGISTER:
if (keycancelled) State = LOGGING_IN
if (keyregistration_successsfull) State = IDLE
case DEMOGRAPHICS:
if (keyexit_demographics) State = IDLE
if (keyclose_demographics) State = CLOSE
case COURSE:
if (keyreturn) State = IDLE
if (keyexit_course) State = CLOSE
case CLOSE:
if (keyexit) State= LOGGING_IN
}
```