Use-Case Descriptions

Software Engineering Course 2015

Project

“Automated Core Course Scheduling**”**

Customer

Erich Reindel

Developers

Amanuel Ghirmay, Anastasiia Izycheva, Deepti Mittal , Emamurho Ughergughe, Kemi Oladipo, Onaopepo Adekunle , Sotaya Yakubu

Tutor

Rafaella Antonyan

Submission Date

15.1.2016

First Increment (04/12/2015)

Use case	*Register*
Primary actor	Professor/Lecturer
Goal in context	To enable Lecturer to use the system; Also, to enable system get details of user. This way, only authorized personnel can have access to the system
Preconditions	System has been programmed to receive user registration details including user-name and password for the first time
Trigger	The lecturer has clicked the "register" button on the application home page

Scenario:

1. Lecturer: enters his registration details including first name, last name, email address, password, confirm password.
2. Lecturer: Clicks the Submit button
3. System: Stores lecturer data into database (if all validation is passed) and redirects to lecturer home page

Exceptions:

1. Email is incorrect: system uses regular expression to prevent this
2. One or more Registration details are incorrect or empty; Error message specifying what went wrong
3. Database error; contact admin.
4. User name exists; change username.

Priority: Essential, must be implemented

Frequency of use: Once for the system

Channel to actor: Via browser

Secondary actors: None

Channels to secondary actors: None

Open issues:

1. should we send registration confirmation email to lecturer or not
2. If we are concerned about simplicity, is it necessary for Lecturer to include his title and user-name in the registration page.
3. If, we send registration confirmation to lecturer's email, what happens if the lecturer never clicks on the confirmation email in his in-box?
4. Are there secondary actors: should the study coordinator have any authority here?
5. Can registration expire?

Use case	*Log-in*
Primary actor	Professor/Lecturer
Goal in context	To grant access to lecturer's account
Preconditions	The system already has the details of the Lecturer
Trigger	The lecturer has clicked the “log-in” button on the application home page.

Scenario:

1. Actor: Enters his log-in details including email address, password.
2. Actor: Clicks the Login button
3. System validates credentials and logs user in if everything is successful.
4. Actor: Observes that he has been redirected to his account page.

Exceptions:

1. Login details are incorrect: the lecturer receives messages saying his login details are incorrect and that he should try again
2. Login details are incorrect after three Logins: the lecturer is asked to contact system administrator
3. The Lecturer forgot his password/user-name: the Lecturer clicks “forgot password” and is asked to enter his email. Then the system reset password and sends him a new one

Channel to actor: Via browser

Secondary actors: None
Channels to secondary actors: None

Open issues:

1. Should we prevent the lecturer from using the system after more than three log-in

Priority: High [Implement with base functionality]

When Available: First Increment

Frequency: once per actor

Use case	***Logout***
Primary actor	Professor/Lecturer/Coordinator
Goal in context	Enable the Lecturer/Coordinator log out of the system
Preconditions	The lecturer/coordinator is already logged into the system
Trigger	The lecturer/coordinator has clicked the “Logout” button on his account page

Scenario:

1. Actor Clicks the logout button
2. System logs actor out and is redirected to the login page

Exceptions:

1. No session found; refresh page and try again or contact admin to close session.

Channel to actor: Via browser

Secondary actors: None

Priority: High [Implement with base functionality]

When Available: First Increment

Frequency: once per actor

Second Increment (11/12/2015)

Use case	***Register in the System***
Primary Actor	Study Coordinator
Goal in Context	Register as a user of the system
Pre-Conditions	A fully deployed core course scheduling system
Trigger	Study coordinator register as a user of the system

Scenario

1. The actor opens the webpage of the core course scheduling system and selects register
2. The system provides the form to fill to register
3. The actor fills the relevant information in the form
4. The core course scheduling system asks for a key to necessary for registering as this actor
5. The actor provides the key and submit.
6. The core course scheduling system accepts and provides feedback of successful registration

Exceptions

1. Registration fails; feedback is provided as to the errors for correction
2. Registration rejected; contact primary coordinator to get a valid key to register as a coordinator.
3. The server hosting the interface is down; manage the server hardware issue
4. The database crashes; Reload the core course scheduling system

Channel to actor: Via browser

Secondary actors: None

Priority: High [Implement with base functionality]

When Available: First Increment

Frequency: once per actor

Use case	***Log on to the System***
Primary Actor	Study Coordinator/Lecturer
Goal in Context	Gain access to the core course scheduling system
Pre-Conditions	study coordinators are registered in the system
Trigger	Study coordinator wants to access the core course scheduling system functionalities

Scenario

1. The actor opens the web page of the core course scheduling system and selects log in
2. The system provides the interface to log in
3. The actor provides the username and password
4. The core course scheduling system accepts and provide the functional index page for coordinators.
5. The actor selects or adds (if no semester have been created)
6. The core course scheduling system provides the functions for that given semester.
7. The actor selects relevant functionalities

Exceptions

1. Log in validation error as a result of wrong user name or password; The system alerts the actor of the issue and provides opportunity to keep trying as many times as possible or recover username and password based on registered email address.
2. The server hosting the interface is down; manage the server hardware issue
3. The database crashes; Reload the core course scheduling system

Channel to actor: Via browser

Secondary actors: None

Priority: High [Implement with base functionality]

When Available: Second Increment

Frequency: high

Use case	***Password Recovery***
Primary Actor	Study Coordinators/Lecturers
Goal in Context	Reset Password
Pre-Conditions	study coordinators/lecturers are registered in the system
Trigger	Study coordinator wants to reset his/her password

Scenario

1. Actor opens the webpage of the core course scheduling system and selects login
2. The system provides a link to reset password.
3. Actor selects Forgot password
4. The system provides and email field.
5. Actor fills and clicks on reset
6. System sends a link to reset password

Exceptions

1. User does not exist; user needs to register first.
2. Email not sent; user resents request or contacts admin if no email is sent.
3. database crashes; user restarts the application.

Channel to actor: Via browser

Secondary actors: None

Priority: High [Implement with base functionality]

When Available: Second Increment

Frequency: high

Use case	***Add Core Courses, Estimated students and Fields of study***
Primary Actor	Study Coordinator
Goal in Context	Have a list of core courses and the corresponding field of study they belong to be accessed by the lecturers.
Pre-Conditions	Study coordinator is registered in the system, a new semester is created or an existing semester was registered prior, core course is not available already in the database.
Trigger	Study coordinator wants to add a new core course.

Scenario

1. The study actor logs on to the core course scheduling platform
2. The system provides the interface to select/create a semester
3. The study actor chooses a semester or create one before choosing.
4. The system provides the add course functionality as one of the functions
5. The study actor selects to add a course
6. The system provides fields to create new course, field of study and estimated number of students.
7. The actor fills and saves the course.
8. The system displays registered courses.

Exceptions

1. Log in validation error; refer to the log on to the system use case
2. The server hosting the interface is down; manage the server hardware issue
3. The database crashes; Reload the core course scheduling system
4. The course already exists; Feedback is provided for the error

Channel to actor: Via browser

Secondary actors: Secondary Coordinators

Priority: High [Implement with base functionality]

When Available: Second Increment

Frequency of use: At the start of a new semester

Third Increment (18/12/2015)

Use case	***Add Semester***
Primary Actor	Study Coordinator
Goal in Context	The coordinator creates a new semester to which preferences can be set to.
Pre-Conditions	Study coordinator is registered in the system, the semester does not already exist in the database, core course is not available already in the database.
Trigger	Study coordinator clicks on “add new semester”.

Scenario

1. The study actor logs on to the core course scheduling platform
2. The system provides the interface to select/create a semester
3. The study actor selects create a semester.
4. The system provides the interface with winter and summer semester with year field.
5. The study actor selects a semester, adds a year and clicks on “Add”.
6. The system adds the semester and reroute to the home page with the semester as a select option.

Exceptions

1. Log in validation error; refer to the log on to the system use case
2. The server hosting the interface is down; manage the server hardware issue
3. The database crashes; Reload the core course scheduling system
4. The semester already exists; Feedback is provided for the error to add a new one.

Channel to actor: Via browser

Secondary actors: Secondary coordinarors

Priority: High [Implement with base functionality]

When Available: Third Increment

Frequency of use: At the start of a new semester

Use case	***Manually change Time slots***
Primary Actor	Study Coordinator
Goal in Context	Edit the time slot of core courses
Pre-Conditions	lecturers have declared a course and set time preferences, study coordinators and lecturers are registered in the system, the system has generated some schedule
Trigger	Study coordinator wants to modify the core courses time slots

Scenario

1. The study coordinator logs on to the core course scheduling platform
2. The system provides the interface to view a generated schedule
3. The study coordinator modifies the schedule and move core courses to other time slot
4. The study coordinator saves the changes made
5. The system provides a feedback to acknowledge changes

Exceptions

1. Log in validation error; refer to the log on to the system use case
2. The server hosting the interface is down; manage the server hardware issue
3. The database crashes; Reload the core course scheduling system
4. The modification is invalid; the system provides feedback on the error and provides option to try again

Channel to actor: Via browser

Secondary actors: Secondary coordinators

Priority: High [Implement with base functionality]

When Available: Third Increment

Frequency: medium

Use case	***Add Preferences***
Primary Actor	Lecturer
Goal in Context	Add preferences to timeslots for a given course based on availability
Pre-Conditions	Coordinators and lecturers are registered in the system, the coordinator have created a semester and the course the preference will be set to.
Trigger	Lecturer clicks on “Add Preferences function”

Scenario

1. The lecturer logs on to the core course scheduling system.
2. The system provides the interface to with add preferences button
3. The lecturer clicks on add preferences.
4. The system provides a calendar based preference selection function with submit and reset.
5. The lecturer selects course, number of hours and choose preferences on calendar and submits.
6. The system provides a success message and adds the preferences to the database for the given course.

Exceptions

1. Course or number of hours not selected; the system gives an alert to add this fields.
2. The actor choose to click the reset button; all options are reset and nullified.
3. The page crashes; reload and try again.
4. There are no course to select; contact coordinator to add courses.
5. Database error, values not logged; contact administrator.
6. After second login, not previous preferences are shown; contact admin.

Channel to actor: Via browser

Secondary actors: None

Priority: High [Implement with base functionality]

When Available: Third Increment

Frequency: At the beginning of each semester.

Fourth Increment (26/12/2015)

Use case	***View Preferences***
Primary Actor	Study Coordinator
Goal in Context	View for each course which preference timeslots are chosen with color codes.
Pre-Conditions	lecturers have declared a course and set time preferences, study coordinators and lecturers are registered in the system, the system has generated some schedule
Trigger	Study coordinator clicks on the preferences function.

Scenario

1. The study coordinator logs on to the core course scheduling system.
2. The system provides the interface to choose a semester.
3. The study coordinator selects the semester.
4. The system provides the coordinator menu with functionalities.
5. The study coordinator selects the “preferences” function.
6. The system displays all courses and the preferences colored in their color codes.

Exceptions

1. Log in validation error; refer to the log on to the system use case
2. The server hosting the interface is down; manage the server hardware issue
3. The database crashes; Reload the core course scheduling system
4. No preferences available; contact admin if lecturers have set a preference or ask lecturers to set preference if not set yet.

Channel to actor: Via browser

Secondary actors: Secondary coordinators

Priority: High [Implement with base functionality]

When Available: Fourth Increment

Frequency: High

Use case	***View Schedules***
Primary Actor	Study Coordinator
Goal in Context	View the suggested schedule based on the given preferences.
Pre-Conditions	lecturers have declared a course and set time preferences, study coordinators and lecturers are registered in the system, the system has generated some schedule
Trigger	Study coordinator clicks on the schedules function.

Scenario

1. The study coordinator logs on to the core course scheduling system.
2. The system provides the interface to choose a semester.
3. The study coordinator selects the semester.
4. The system provides the coordinator menu with functionalities.
5. The study coordinator selects the “schedules” function.
7. The system displays all possible schedules based on the preferences from lecturers.

Exceptions

1. Log in validation error; refer to the log on to the system use case
2. The server hosting the interface is down; manage the server hardware issue
3. The database crashes; Reload the core course scheduling system
4. No schedules available; manually change timeslots and click recalculate or ask lecturers to set preference if non is set yet.

Channel to actor: Via browser

Secondary actors: Secondary coordinators

Priority: High [Implement with base functionality]

When Available: Fourth Increment

Frequency: High

Fifth Increment (14/1/2016)

Use case	***Mark Timeslots***
Primary Actor	Study Coordinator
Goal in Context	Select timeslots to be blocked for a specific item e.g prof launch, VC courses
Pre-Conditions	lecturers have declared a course and set time preferences, study coordinators and lecturers are registered in the system, the system has generated some schedule
Trigger	Study coordinator clicks on the “mark timeslots” function.

Scenario

1. The study coordinator logs on to the core course scheduling system.
2. The system provides the interface to choose a semester.
3. The study coordinator selects the semester.
4. The system provides the coordinator menu with functionalities.
5. The study coordinator selects the “mark timeslots” function.
6. The system displays the view to block timeslots.
7. The study coordinator selects a reason, select times to block and submits.

Exceptions

1. Log in validation error; refer to the log on to the system use case
2. The server hosting the interface is down; manage the server hardware issue
3. The database crashes; Reload the core course scheduling system
4. Timeslots already marked; deselect them and select for the particular reason.

Channel to actor: Via browser

Secondary actors: Secondary coordinators

Priority: High [Implement with base functionality]

When Available: Fifth Increment

Frequency: Medium

Use case	***Update Passphrase***
Primary Actor	Study Coordinator
Goal in Context	Updates the passphrase to enable other coordinators to register with.
Pre-Conditions	Coordinator is registered in the system, a semester have been created.
Trigger	Study coordinator clicks on the “update passphrase” function.

Scenario

1. The study coordinator logs on to the core course scheduling system.
2. The system provides the interface to choose a semester.
3. The study coordinator selects the semester.
4. The system provides the coordinator menu with functionalities.
5. The study coordinator selects the “update passphrase” function.
6. The system displays the view to update passphrase.
7. The study coordinator enters new passphrase and submit.

Exceptions

1. Log in validation error; refer to the log on to the system use case
2. The server hosting the interface is down; manage the server hardware issue
3. The database crashes; Reload the core course scheduling system
4. The passphrase is same as previous; coordinator should choose a new passphrase.

Channel to actor: Via browser

Secondary actors: Secondary Coordinators

Priority: High [Implement with base functionality]

When Available: Fifth Increment

Frequency: High

Sixth Increment 21/01/2016

Use case	***Modify Constraints***
Primary Actor	Study Coordinator
Goal in Context	Set the constraints to be applied to lecturer preferences to get a schedule based on the constraints.
Pre-Conditions	Coordinator is registered in the system, a semester have been created and selected.
Trigger	Study coordinator clicks on the “Modify Constraints” function.

Scenario

1. The system provides the coordinator menu interface.
2. The study coordinator selects ‘modify constraints’.
3. The system opens the “modify constraint interface”.
4. The study coordinator sees already existing constraints, ticks current constraints to set and save.
5. The system updates the current constraints and activate them in the constraint solver.

Exceptions

1. Log in validation error; refer to the log on to the system use case
2. The server hosting the interface is down; contact system admin
3. The database crashes; Reload the core course scheduling system or logout and login again
4. Constraint the same as previous; nothing happens.

Channel to actor: Via browser

Secondary actors: Secondary Coordinators

Priority: High [Implement with base functionality]

When Available: sixth Increment

Frequency: High