Use Case Name: Log into site

Short Description: When the user wishes to edit data and thus is required to log in as per the security requirement.

Actors: User

Triggers: When the user enters the login page/ opens up login prompt

Preconditions: User has access to internet, Server is online and is able to respond to requests, and Database is online and functioning correctly.

Post conditions: User’s request goes through presentation’s servlet and the login method set is used after the request is processed.

Results: User is logged in and can proceed to edit timetable (actions are dependent on if they are normal user or super user)

Outline:

Perfect Flow:

1. User is on the timetable site
2. Login prompt/page opened
3. User inputs login information
4. HTTP request is sent from client to presentation
5. Presentation looks at request and sees that the request has login information and thus the servlet uses the login method set to process the information
6. Login data is looked for and compared in the database
7. Login data is correct and an http response is created through the successful login method set
8. Response is sent back to the user
9. Web builds the successful response page based on the http response

Alternate Flow 1.1 (what the user is unable to login):

1. User is on the timetable site
2. Login prompt/page opened
3. User inputs login information
4. Information isn’t sent to presentation
5. Error prompt is sent from presentation to web
6. Web builds error response page.

Alternate Flow 1.2 (user inputs bad info)

1. User is on the timetable site
2. Login prompt/page opened
3. User inputs login information
4. HTTP request is sent from client to presentation
5. Presentation looks at request and sees that the request has login information and thus the servlet uses the login method set to process the information
6. Login data is looked for and compared in the database
7. Login data is incorrect and an http response is created through the unsuccessful login method set
8. Response is sent back to the user
9. Web builds the unsuccessful response page based on the http response

Use Case Name: Edit Data (can be to create, edit or remove as the process for the 3 are similar)

Short Description: When the user wishes to edit data and login is successful

Actors: User

Triggers: When the user edits the information on the site

Preconditions: User has access to internet, Server is online and is able to respond to requests, and Database is online and the user is logged in.

Post conditions: User’s request goes through presentation’s servlet and the editing data method sets are used after the request is processed.

Results: User is able to edit data and save it onto the database afterwards. (actions are dependent on if they are normal user or super user)

Outline:

Perfect Flow (for this example we will change the time of an exam):

1. User is on the timetable site
2. Login prompt/page opened
3. User logs in
4. User’s login in verified
5. User can choose to view, edit, or logout
6. User chooses to edit
7. Edit page is opened
8. User enters in old exam time slot, new exam time slot, course type
9. Data is sent through presentation to be processed into a request
10. Request is seen as an edit type request and pulls method sets that are used to edit the time table along with the information to change
11. Server verifies the user is still in session, if the time slot is taken, if there are any repeated exams for the same section
12. Server allows changes to database as they don’t break any of the rules (stated above are some, I assume we will allow server to see what will be restricted)
13. Database edits the timetable by moving the exam time to a different slot (or in this case, create a new exam time and then removing the old one) and saves
14. HTTP response is made from the presentation after the data has been updated
15. Web processes response and shows the updated timetable

Alternate flow 1.1(session timeout before edit is finished)

1. User sends data
2. Information is processed as a request
3. Server doesn’t allow any changes to database due to session timeout and gives presentation the response
4. Web receives session timeout page from presentation

Alternate flow 1.2 (Database isn’t found)

1. Data is sent
2. Information is processed
3. Server cannot allow changes as the database doesn’t exist/isn’t found
4. Presentation gets server’s error and invokes missing database methods to build a response
5. Web lets the user know that the database wasn’t found or is missing

Alternate Flow 1.3 (Changes break a rule, in this case we will say it has a course with a section have multiple timeslots)

1. Server stops changes
2. Presentation gets a request from server to make a response to ask the user if they wish to replace the previous timeslot
3. User is given the prompt(in this case the user will say yes)
4. Request is given to server to allow the data to be replaced
5. Database creates new timeslot for the section’s exam and removes the previous one

Use Case Name: Log off site

Short Description: When the user is finished and wishes to end session to prevent unauthorized access

Actors: User

Triggers: When the user is able to choose to view/edit/ logout the timetable site

Preconditions: User has access to internet, Server is online and is able to respond to requests, and Database is online and is logged in to begin with.

Post conditions: User’s request goes through presentation’s servlet and the log out method set is used after the request is processed.

Results: User is logged out

Outline:

Perfect Flow:

1. User is on the timetable site
2. Login prompt/page opened
3. User logs in
4. User’s login in verified
5. User can choose to view, edit, or logout
6. User chooses to edit
7. User is finished editing
8. User can choose to view, edit, or logout
9. User chooses to logout
10. HTTP request is processed and logout method set is used based on tag in the request
11. Server gets the request
12. Server removes user from active session list
13. Presentation gives web the logout response (based from the logout method set)