**3.5 ACCESS CONTROL AND SECURITY**

The application is a multiuser application so it consists of 3 types of users which are instructor, student and administrator. Because of this, the application will provide different interfaces for each user type.

First, the administrator will connect to the system with the membership interface, and will do the administrator's duties on the dating system, such as adding, updating, editing, deleting etc. By the way, registration is not necessary for the administrator through the system website, information will be entered manually into the database at the beginning of the system and the administrator will be the authority that will access the database directly. As a summary, the administrator does not have to register because it is initially registered in the database and the system. The administrator registers the instructor. Each registration process included the instructor registration made by the administrator, will be done with the user interfaces of the system. The system will store all the information in the database and in the login processes again the system will use them by collecting data from the database. The information in the database will use both the confirmation and the use of the system for users. All types of users must log in to the system with their username and password.

After these steps, the system will be ready for the appointment of the instructor. The instructors will register in the application, the system will send the request to admin and admin will approve the request of the instructor, the system will add a new user to the database. The instructor will then connect to the system with your information such as username and password. During the login procedure, the data in the user database table will be obtained and compared to the data entered by the user. Since this operation requires read-only access to the database, it can be performed from different access points simultaneously.

During registration, field filling does not require access to the database, while completion of the process requires the data to be written to the database, which requires read and write access to the database. In that case, the required database fields will be blocked and simultaneous access of multiple users will be denied.

For some situation like updating or deleting information it is necessary to update one of the tables in the database in its phase of completion and therefore must be handled with more care since several users can be the cause of updating the table at the same time. This will also be avoided by blocking.

Finally, viewing the information or lists again requires read-only access to the database. Therefore, multi-user access does not impose problems and new restrictions.

As last words, the usernames and passwords of users will be stored in the user table. No one else accepts the administrator can have access to this information. Authentication interfaces are different for each type of user and will be directed to their own main pages after the login process.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Actors/  Classes | User | Authentication | Exam | Question | Lecture |
| Admin | <<create>>  createInstructor()  updateInstructor()  deleteInsructor()  updateInfo()  listInstructor()  approveRequest() | Login()  logout() |  |  |  |
| Instructor | <<create>>  createClass()  addStudent()  deleteStudent()  listStudent() | Login()  logout()  resetPwd() | createExam()  updateExam()  removeExam() | addQuestion()  deleteQuestion()  updateQuestion()  removeQuestion() | createLecture() |
| Student | updateInfo()  requestPassRemainder() | Login()  logout()  resetPwd() | joinExam()  showExamResults() | answerQuestion()  viewAnswers() |  |

Figure 3.1 Access Control Matrix for Hospital Appointment System

Admin, User, {<<create>> createInstructor(), updateInstructor(), deleteInsructor(), updateInfo(), listInstructor(), approveRequest()}

Instructor, User, {<<create>> createClass(),addStudent(),deleteStudent(),listStudent()}

Admin, Authentication, {login(), logout()}

Student, Authentication, {login(), logout(), resetPwd()}

Instructor, Authentication, {login(), logout(), resetPwd()}

Instructor, Exam, { createExam(), updateExam(), removeExam()}

Student, Exam, { joinExam(),showExamResults()}

Instructor, Question, { addQuestion(),deleteQuestion(),updateQuestion(),removeQuestion()}

Student, Question, { answerQuestion(),viewAnswers()}

Instructor, Lecture, { createLecture()}

**3.6 GLOBAL SOFTWARE CONTROL**

External Control Flow (Between Subsystems) : ES system defined by the web application with a simple feature. Web server requests request submission of user data. Because the system is multi-user, simultaneous executions can occur. However, the control flow of a single user has a predefined form. After the login step, the system has a web page structure in the form of a tree formed by links or buttons.

Concurrent Control : Because the application is web-based, all subsystems and components can run simultaneously for different users in the application.

Internal Control (Within a Single Process) : The process control is carried out by means of the designed forms on the web. The system is based on the page structure of the page request page. This makes the designed procedures simple and mostly linear. However, procedure calls can be made to other subsystems or to the current subsystem. Threads or multiple processes can be required for a process. The system uses a database so that the response time from the database should be minimized.

User Interface : The system user interface will be made through web pages. The control of the next step depends on the user. In addition to this, the flow is implemented within the web page. Most subsystems have a different web page. Due to the system event-driven design, subsystems can not be considered to have their own event loop. However, events are controlled by web pages.

**3.7 BOUNDARY CONDITIONS**

Startup: go to system URL and login

Shut Down: click log out and close browser

Error Conditions:

* Logging in:
  + Username or password field cant be blank.
  + Username is not a 5 digit decimal number.
  + Password is not 8 characters long.
  + Password and username don’t match.
  + Username is wrong or does not exist.
  + The welcome screen does not appear after logging in.

* User settings
  + User is unable to change certain settings or changes don’t reflect.
  + Between the time of editing and updating, the system crashes.
* Data Entry
  + The system fails when the dispatcher is entering information.

* Instructor Entry
* İnstructor informations cant be exist .
* İnstructor informations cant be blank .
* Exam entry
* Exam name, date, time cant be blank.
* Date time cant be a date before today.
* Question entry
* Question or answer field cant be blank.
* Class List entry
* Same class name is exist.
* Logging out
  + Dispatcher unable to logout.