**4.3.5 Access control and security**

|  |  |  |  |
| --- | --- | --- | --- |
| actor/Object | Student mark | payment | School member |
| Academic user | SubmitMark() |  |  |
| Homeroom teacher | SubmitMark()  postResult() |  |  |
| Director | SubmitMark() |  | ViewMemberInfo() |
| Administrative office |  | ApproveFee() | RegisterMemebers()  UpdateMemebers() |
| Administrative head |  | ApproveFee()  ViewPaymentReport() | RegisterMemebers()  UpdateMemebers() |

|  |  |  |
| --- | --- | --- |
| Actor/Object | Student Attendance | Time Table |
| Academic user |  |  |
| Homeroom teacher | SubmitAttendance() |  |
| Director |  | PostSchedule()  updateSchedule() |
| Administrative office |  |  |
| Administrative head |  |  |

Figure Static access control for Nafyad system

Authentication mechanism that our software uses is user-name and password authentication.

The users required to fill and submit their own user-name and password and web server check from protected resources that submitted message are valid.

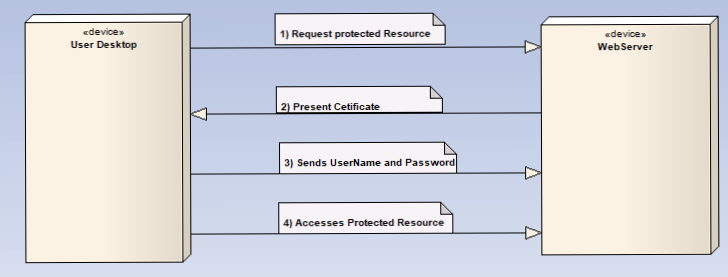


Figure user-name and password authentication mechanism

The users user-name and password using cryptographic techniques are kept as cipher text (cannot be translate directly as original word(plain text)) in database so even when attackers get cipher text they cannot know original word.

**4.3.6 Global control flow**

The sequence of action by the subsystem in our system is based on Event-driven control method that Components generate events indicating, perhaps, that some data is available for processing. The event handler detects the events, consults the event register and passes the event to those components who have declared an interest. All event from subsystem go to event and message handler to be implemented

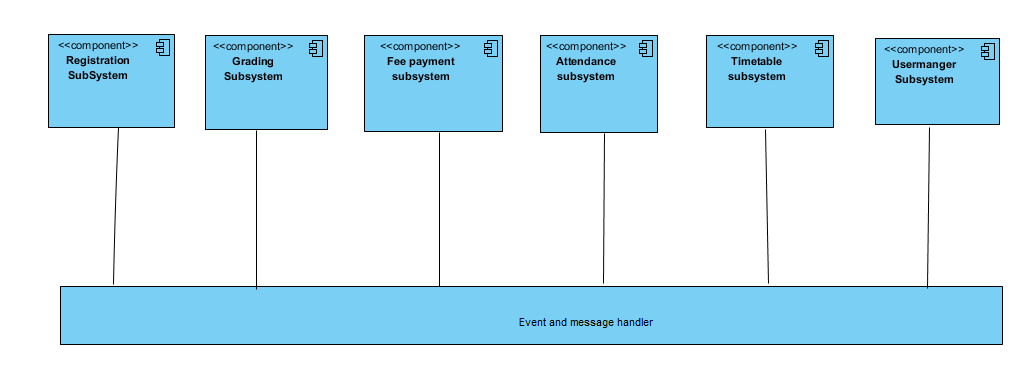


Figure event-driven control

We define the following strategy for dealing with concurrent accesses to shared data:

* Boundary objects should not define any fields. Instead, boundary objects hold temporary data associated with the current request in local variables. As boundary objects are shared among threads, this prevents concurrency hazards at that level.
* Control objects should not be shared among threads. Instead, there should be at most one control object associated with each session, and users should not be able to issue concurrent requests involving the same control object within the same session. This should especially be enforced when control objects survive the processing of a single request.
* Entity objects should not provide direct access to their fields. Instead, all changes and accesses to the object state should be done through dedicated methods.

**4.3.7 Boundary condition**

|  |  |
| --- | --- |
| Start-up server | first select safety room where web sever placed then make connection with persistent storage (can be database) and user desktop using network cable and System manager will install software to web server and start server. |
| At shutdown | When the system is shut down by system manger temporary files save to database. |

Start up and shutdown use case

|  |  |
| --- | --- |
| System failures | handling |
| Network failure between web browser and server | by notifying the user of the network failure and Re-establish connection after a network failure or to restore the state of Matches after a crash |
| A server failure | Checking the integrity of the persistent data after an unexpected termination of the server and power switch connected to sever. |

**4.4 subsystem services**

Our system is divided into 6 Subsystem

|  |  |
| --- | --- |
| subsystem | Service it provide |
| Reregistration subsystem | Responsible to upload new student information to database and updating their profiles |
| Grading subsystem | Is used to upload student mark and calculate their mark and view their final result with average and rank. |
| Fee Payment subsystem | Is used to approve student fee status |
| Attendance subsystem | Is used to take note on student present and absent condition |
| Timetable subsystem | Is used to prepare time table of school program and post Schedule other users to view its |
| User manger subsystem | Responsible to upload new user to system and update previous users status |