Contents

[**System Design** 2](#_Toc3475172)

[4.1 INTRODUCTION 2](#_Toc3475173)

[4.1.1 Purpose of the system design 2](#_Toc3475174)

[4.1.2 Design Goals 2](#_Toc3475175)

[4.1.3 Definitions, acronyms, and abbreviations 3](#_Toc3475176)

[4.1.4 References 3](#_Toc3475177)

[4.1.5 Overview 3](#_Toc3475178)

[4.2 Current software architecture 3](#_Toc3475179)

[4.3 Proposed software architecture 5](#_Toc3475180)

[4.3.1 Overview 5](#_Toc3475181)

[4.3.2 Subsystem decomposition 6](#_Toc3475182)

[4.3.3 Hardware/ software mapping 7](#_Toc3475183)

[4.3.3 Hardware and software mapping 8](#_Toc3475184)

[4.3.4 Persistent data management 9](#_Toc3475185)

[4.3.5 Access control and security 12](#_Toc3475186)

[4.3.6 Global control flow 13](#_Toc3475187)

[4.3.7 Boundary condition 14](#_Toc3475188)

[4.4 subsystem services 15](#_Toc3475189)

**CHAPTER FOUR**

# System Design

## 4.1 INTRODUCTION

In the previous chapter we have identified the functional and non-functional requirements  
of the system and produced the analysis model. In this document the design goals, the proposed system Design and the object design (system architecture, system decomposition, deployment and database design) are included.

### 4.1.1 Purpose of the system design

Our system provide services like grading, registration and other services to academic institute using web-based system this allow institute to improve their daily activity process and organization.

### 4.1.2 Design Goals

From the nonfunctional requirements and from the application domain study of SMS we identify the following as design goals.

**Performance**

The part of the system to be used for the registration should have a fast response time (real time) with maximum throughput.

In the case of the timetabling payment subsystem, the system should be more reliable in order to satisfy the constraints than fast response time.

Furthermore, the system should not be taking up too much space in memory.

Dependability

The school needs the system to be highly dependable as it is expected to be used by non-IT professionals.

The system should be robust and fault tolerant.

As the system is handling sensitive data of the school, high emphasis should be given with regards to security.

Maintenance

Hence the proposed system did not have all features that the school wants. So, if the need is arise to add new functionality at a later stage the system should allow that. That means the system should be easily extensible to the new functionalities and should also be easily modifiable to make changes to the features and functionalities.

End User CriteriaUsability:-From the end users’ perspective the system should be designed in such a way that it is easy to learn and use it.

**Security versus response time**: Checking User-Id and Password before a member can enter to the SMS creates response time problem/overhead.

**Response time versus quality**: There is some amount of time taken by the system to generate the timetable. So the user has to wait a little after telling the system to generate the timetable and getting the result to get a quality timetable.

### 4.1.3 Definitions, acronyms, and abbreviations

|  |  |
| --- | --- |
| **abbreviations** | **Definitions** |
| IIS | Internet Information Services |
| HTTP | Hypertext Transfer Protocol |
| LAN | local area network |
| RDBMS | relational database management system |

Table 4.1 Definitions, acronyms, and abbreviations

### 4.1.4 References

*Object-Oriented Software Engineering using UML, Patterns, and Java (3rd Ed.) [Bruegge & Dutoit 2009-08-08]*

### 4.1.5 Overview

The documentation describe about current software architecture component including local server and end users.

## 4.2 Current software architecture

Architecture of the System

We use a 3 tier Client/Server Architecture where a client can use Internet browsers to access the system within the local area network of the school.

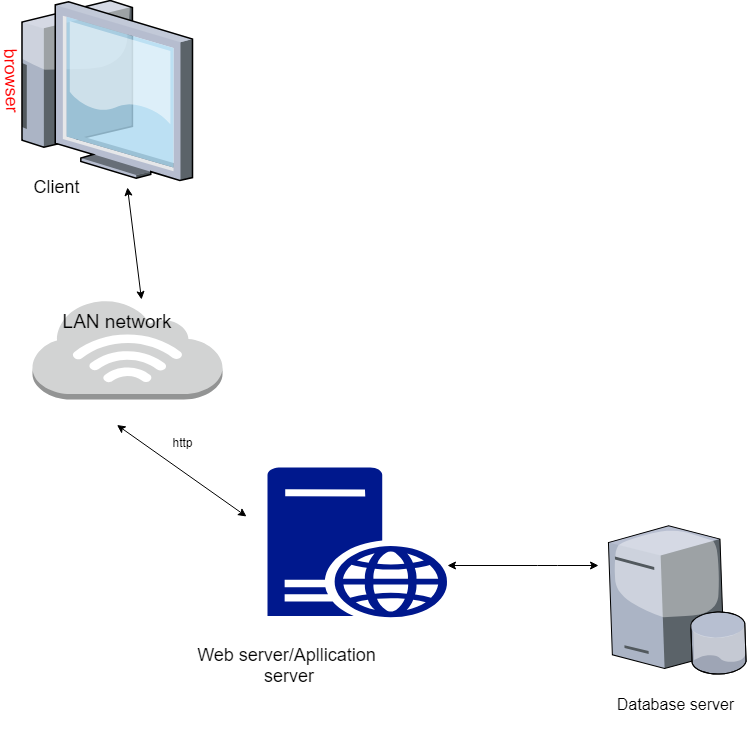


Figure 4.1 architecture of the proposed system.

The client tier is the applications user interface containing data entry forms. It displays data to the user. Users interact directly with the application through user interface.

The controller logic processes client requests such as requests to view student’s result, to record attendance or to retrieve data from the database. Business rules enforced by the business logic dictate how clients can and cannot access application data and how applications process data.  
The web server used in this system is IIS. HTTP is used to transfer data across  
an Intranet or the Internet.

The middle tier (web/application server) implements the business logic, controller logic  
and presentation logic to control the interaction between the application’s clients and  
data.  
The data tier maintains the applications data such as student data, teacher data, timetable  
data etc. It stores these data in a relational database management system (RDBMS).

The client tier interacts with the web/application server to make requests and to retrieve data from the database. It then displays to the user the data retrieved from the server.

## 4.3 Proposed software architecture

### 4.3.1 Overview

The proposed system is composed of 6 subsystem in next topic it explain about subsystem decomposition , hardware and software mapping, Persistent data management ,Access control and security ,Global software control and Boundary condition

4.3.2 Subsystem decomposition

Figure 4.2 Subsystem decomposition

### 4.3.3 Hardware/ software mapping

|  |  |  |
| --- | --- | --- |
| **subsystems** | **Purpose** | **classes** |
| Registration | * To Register students and other staff members and store their profile * Store student transcription | * Register\_Student * View\_Profile * View\_Transcript |
| Student Grading | * Submit each student’s mark and calculate their total mark | * Submit\_Mark * Calculate\_Mark * Display\_Mark |
| User Management | * Managing users of the system with different level of permission which includes: - * Adding users * Updating Permission of each user * Removing users | * Add\_User * Update\_Permission * Remove\_User |
| Attendance | * Creating attendance form for students and submitting , displaying the attendance when needed | * Take\_Attendance * Display\_attendance |
| Fee Payment | * Collecting student’s monthly payment and providing report | * Payment * Payment\_report |
| Schedule | * Creating schedule for each section of the school and displaying | * Prepare\_schedule * View\_schedule |

Table 4.2:-Subsystem purpose

### 4.3.3 Hardware and software mapping

The system will have Client (user level pc) where each user get to use the system with the given permission, LAN (Local Area Network) where each users can communicate with, Web server that handles all the Back-End operation (record request and replay) and Relational Database (mysql) to handle sql request and replay, execute transaction and store data in a tabular form.­­

Figure 4.3 Hardware and software mapping



### 4.3.4 Persistent data management

Information related to student and employee basic information, student’s attendance and mark, the timetable produced and other related information are persistent data and hence we use relational database management system.

In order to store information persistently we map objects/classes into tables and the attributes into fields to the specific table based on the objects found on the system. Therefore, we identified the major tables that will be implemented on the selected DBMS.

**Mapping**



Figure4.4: Student mark mapping

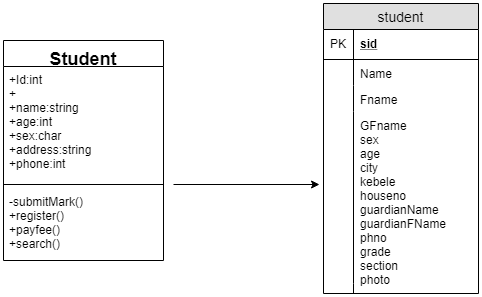


Figure4.5: student mapping

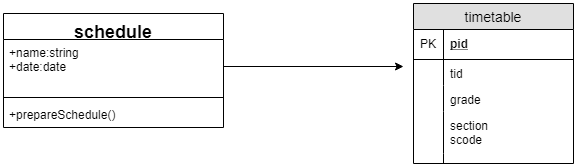


Figure4.6: schedule mapping

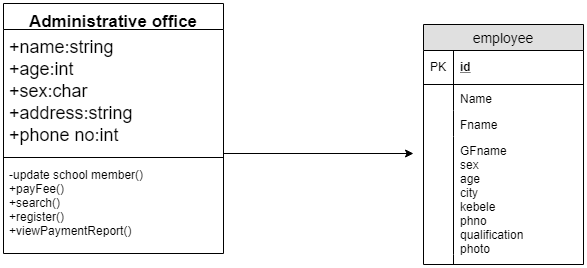


Figure 4.7: administrative office mapping

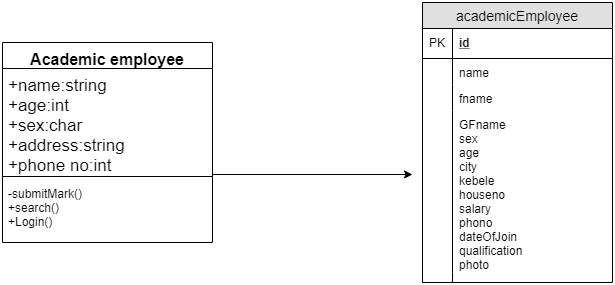


Figure 4.8: Academic employee mapping



Figure 4.9: payment mapping

**Database diagram**

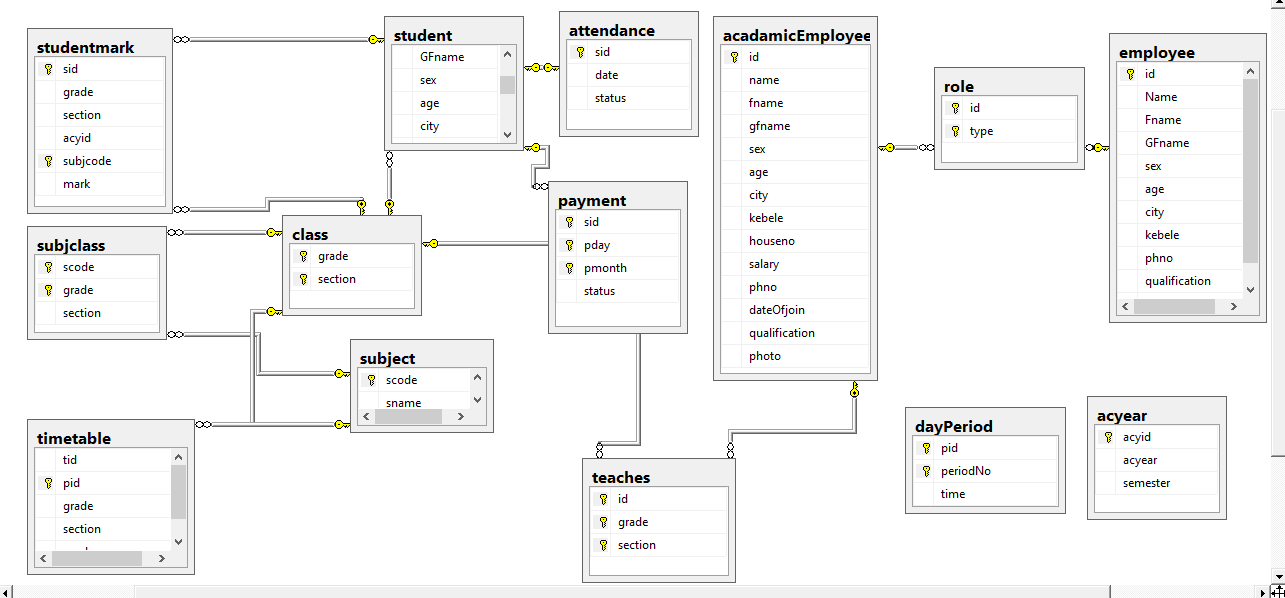
****

Figure 4.10: data base diagram

### 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 |  |  |

Table 4.3 Static access control for Nafyad system

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

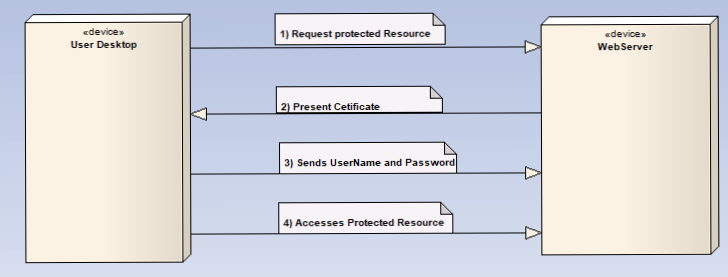
The users required to fill and submit their own user-name and password and web server check from protected resources that submitted message are valid.

Figure 4.11user-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

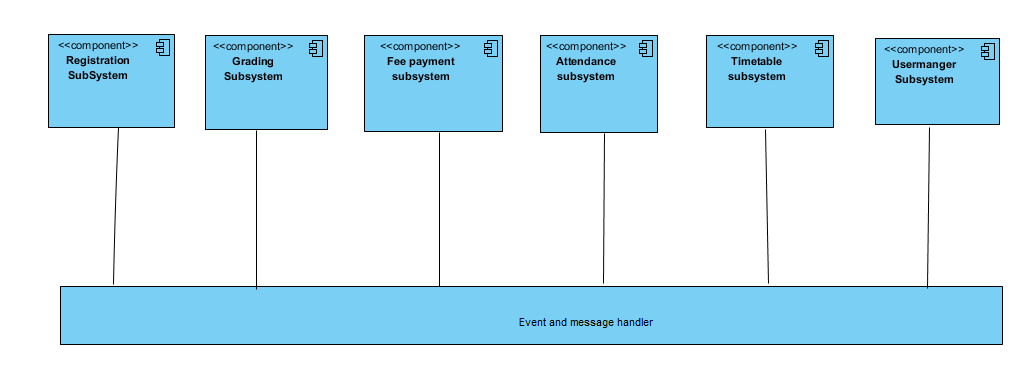
****

Figure 4.12 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 server 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. |

Table 4.4 Boundary condition

## 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 |

Table 4.5 **subsystem services**