# 3 System Analysis and Design(Record Management System)

System analysis is the method of getting the answer of what a system needs and what it can't do and then figuring out what those facts mean. This step creates sure that the customer gets high-quality application. It is more of a method that uses graphical tools to analyze and improve the goals of a present structure and create a new complete system that people can acknowledge.

* It breaks the process down into its basic steps so that you can see how the system works.
* It works from a big picture view down to specifics.
* In contrast to the physical, it is rational.

# 3.1 Preliminary Design Stages

During this phase, customer needs and use cases are used to guide the creation of system design. When compared to the analysis stage, the design phase lays out exactly how the framework will achieve its stated goals and priorities. New systems are designed and analyzed in great detail during the design phase.

It's common for the design phase to end with an operating prototype of the system. The design phase is among the most important stages of software advancement because it will have a direct impact on the final product.

# 3.1.1 Textual Analysis

Identifying courses in the scheme can be done using textual analysis. Work with the applicant checklist and their responsibilities can be done from this framework. According to the characterizations, there are a few lexical items inside the configuration that are recognized as applicant classes.

|  |  |
| --- | --- |
| Candidate Class | Candidate class responsibilities |
| Student | View assessment, view courses, view grades, view attendance, view timetable, submit assignment, view module, do attendance |
| Staff | Add staff, edit staff, display staff, archive staff, delete staff |
| Course | Add course, archive course, display , delete course, edit course |
| Module | Add module, edit module, archive module, delete module, display module. |
| Assignment | Add assignments, edit assignments, archive assignments, delete assignments, display assignments, upload assignments, and download assignments. |
| Attendance | Add attendance, edit attendance, delete attendance, archive attendance, display attendance, remark attendance. |
| Personal Tutor | Add personal tutor, edit personal tutor, delete personal tutor, display personal tutor**.** |
| Timetable | Add timetable, edit timetable, archive timetable, delete time table, display time table. |
| Diary | Add diary, edit diary, delete diary, display diary. |
| Report Generation/ | Create report, display report, print report. |

# 3.1.2 Significant Event Analysis

In the context of significant event analysis, this means analyzing the system's different events and functions.

In needed to execute an occasion, a performer must have entry to the occurrence, and a candidate attribute must be present. As shown in the following table, there are a variety of different occurrences in the framework and their corresponding performers and candidates.

|  |  |  |
| --- | --- | --- |
| **Event** | **Performers** | **Candidate Attributes** |
| **Login** | * Admin * Module Leader * Student | * Admin\_id * Staff\_id * FullName * Email * Address * Date\_of\_join * Password * Staff\_id * Name * Surname * Email * Address * Date\_of\_join * Course\_id * Module\_id * Role * Password * student\_id * course\_id * FullName * Email * password * address * phone * gender * date\_of\_birth * registration\_year |
| **Logout** | * Admin * Module Leader * Student | * Admin\_id * Staff\_id * FullName * Email * Address * Date\_of\_join * Password * Staff\_id * FullName * Email * Address * Date\_of\_join * Course\_id * Module\_id * Role * Password * student\_id * course\_id * FullName * Email * password * address * phone * gender * date\_of\_birth * registration\_year |
| **Add/edit/delete Admin** | * Admin | * Admin\_id * Staff\_id * Name * Surname * Email * Address * Joining\_date * Salary * Password |
| **Add/edit/delete Module**  **Leader** | * Admin | * Admin\_id * Name * Surname * Email * Address * Joining\_date * Password * Staff\_id * role |
| **Add/edit/delete assignments** | * Module Leader | * Staff\_id * Name * Surname * Email * Address * Date\_of\_join * Course\_id * Module\_id * Salary * Role * Password * Assignment\_id |
| **Add/edit/delete grades** | * Module Leader | * Staff\_id * FullName * Email * Address * Course\_id * Module\_id * Password |
| **Add Course** | * Admin | * Admin\_id * FullName * Email * Address * Joining\_date * Password * Staff\_id * role * module\_id * course\_id |
| **Add Module** | * Admin | * Admin\_id * FullName * Email * Address * Joining\_date * Password * Staff\_id * role * module\_id |
| **Add/edit/delete attendance** | * Module Leader | * Staff\_id * FullName * Email * Address * Joining\_date * Course\_id * Module\_id * Role * Password * Assignment\_id |
| **Add/edit/delete routines** | * Module Leader | * Staff\_id * FullName * Email * Address * Joining\_date * Course\_id * Module\_id * Role * Password * Assignment\_id |
| **Add submissions** | * Module Leader | * Staff\_id * FullName * Email * Address * Joining\_date * Course\_id * Module\_id * Role * Password * Assignment\_id |
| **Add/view/edit diaries** | * Student | * student\_id * course\_id * FullName * email * password * address * phone * gender * date\_of\_birth * registration\_year |
| **View Assignments** | * Student | * student\_id * course\_id * FullName * email * password * address * phone * gender * date\_of\_birth * registration\_year |
| **View Routines** | * Student | * student\_id * course\_id * FullName * email * password * address * phone * gender * date\_of\_birth * registration\_year |
| **View Grades** | * Student | * student\_id * course\_id * FullName * email * password * address * phone * gender * date\_of\_birth * registration\_year |

# 3.1.3 Command Queries and Constraints

# 3.2 Detailed Static System Designs

Design patterns are a valuable tool for putting the focus on structure, catching design expertise, and making it possible to change how a software system is working. Patterns are most often used with an object-oriented computer language, and that they are set up so that template participants are instances of objects that are created and linked at runtime. In this research, which is an extension of design patterns, a lot of the trend participants match up with statically instantiated and linked components.

# 3.2.1 First Draft BON System Architecture Diagram

System architecture is a set of ideas that describe the structure, behavior, and other views of a system. An architecture characterization is a formal expression and system architecture is the conceptual that is set up in a way that makes it possible to think on how the system operates and how it is put together.

A system architecture could be composed of parts of the system and advanced sub-systems that work together just to run the whole system. A class, a root class, and a cluster make up this architecture.

# 3.2.2 BON System Chart

|  |  |  |  |
| --- | --- | --- | --- |
| SYSTEM | RECORD MANAGEMENT SYSTEM | | PART 1/1 |
| PURPOSE: To maintain all the data related to the website | | Indexing: | |

|  |  |
| --- | --- |
| **CLUSTER** | **DESCRIPTION** |
| STUDENT | To maintain the data of student |
| STAFF | To maintain all the data of staff and work |
| COURSE | To maintain the course of the student |
| MODULE | To maintain all the module of the courses |
| ASSIGNMENT | To maintain the assignment of the student |
| ATTENDANCE | To maintain the attendance |
| PERSONAL TUTOR | To maintain personal tutor |
| TIMETABLE | To maintain the timetable of the class |
| DIARY | To maintain the diary of the students |
| REPORT | To maintain and display report to the students | |

# 3.2.3 BON Cluster Chart

|  |  |  |  |
| --- | --- | --- | --- |
| CLUSTER | STUDENTS | | PART 1/1 |
| PURPOSE: To maintain student data | | Indexing: | |

|  |  |
| --- | --- |
| **CLUSTER** | **DESCRIPTION** |
| Student\_Id | To know about student ID |
| Student\_FullName | To get student Full Name |
| Student\_Email | To get student email |
| Student\_address | To get student address |
| Student\_date\_of\_birth | To get student date of birth |
| Student\_registration\_year | To get student registration year |

d

|  |  |  |  |
| --- | --- | --- | --- |
| CLUSTER | STAFFS | | PART 1/1 |
| PURPOSE: To maintain staff data | | Indexing: | |

|  |  |
| --- | --- |
| **CLUSTER** | **DESCRIPTION** |
| Staff\_Id | To know about staff ID |
| staff\_FullName | To get staff Full Name |
| staff\_Email | To get staff email |
| staff\_address | To get staff address |
| Staff\_role | To get staff roles |

d

|  |  |  |  |
| --- | --- | --- | --- |
| CLUSTER | COURSES | | PART 1/1 |
| PURPOSE: To maintain course on the website | | Indexing: | |

|  |  |
| --- | --- |
| **CLUSTER** | **DESCRIPTION** |
| Course\_Id | To know about course ID |
| Course\_Name | To get course name |
| Course\_Detaiil | To get course details |

d

|  |  |  |  |
| --- | --- | --- | --- |
| CLUSTER | MODULES | | PART 1/1 |
| PURPOSE: To maintain module on the website | | Indexing: | |

|  |  |
| --- | --- |
| **CLUSTER** | **DESCRIPTION** |
| Module\_Id | To know about module ID |
| Module\_Name | To get module name |
| Module\_details | To get module details |

d

|  |  |  |  |
| --- | --- | --- | --- |
| CLUSTER | ASSIGNEMTS | | PART 1/1 |
| PURPOSE: To maintain assignments on the website | | Indexing: | |

|  |  |
| --- | --- |
| **CLUSTER** | **DESCRIPTION** |
| Assignment\_Id | To know about Assignment ID |
| Assignment\_Name | To get Assignment name |
| Assignment\_details | To get Assignment details |

d

|  |  |  |  |
| --- | --- | --- | --- |
| CLUSTER | ATTENDANCES | | PART 1/1 |
| PURPOSE: To maintain attendance on the website | | Indexing: | |

|  |  |
| --- | --- |
| **CLUSTER** | **DESCRIPTION** |
| Attendance\_Id | To know about attendance ID |
| Attendance\_details | To get attendance details |

d

|  |  |  |  |
| --- | --- | --- | --- |
| CLUSTER | PERSONAL TUTOR | | PART 1/1 |
| PURPOSE: To maintain personal tutors on the website | | Indexing: | |

|  |  |
| --- | --- |
| **CLUSTER** | **DESCRIPTION** |
| PersonalTutor\_Id | To know about Personal Tutor ID |
| PersonalTutor\_FullName | To get Full name of personal tutor |
| PersonalTutor\_Address | To get address of personal tutor |
| PersonalTutor\_Email | To get email of personal tutor |
| PersonalTutor\_role | To get role of personal tutor |

d

|  |  |  |  |
| --- | --- | --- | --- |
| CLUSTER | TIMETABLES | | PART 1/1 |
| PURPOSE: To maintain TimeTable on the website | | Indexing: | |

|  |  |
| --- | --- |
| **CLUSTER** | **DESCRIPTION** |
| TimeTable\_Id | To know about TimeTable ID |
| TimeTable\_Details | To get TimeTable Details |

d

|  |  |  |  |
| --- | --- | --- | --- |
| CLUSTER | DIARYS | | PART 1/1 |
| PURPOSE: To maintain diary on the website | | Indexing: | |

|  |  |
| --- | --- |
| **CLUSTER** | **DESCRIPTION** |
| Diary\_Id | To know about diary ID |
| Diary\_Name | To get diary name |
| Diary\_details | To get diary details |

d

|  |  |  |  |
| --- | --- | --- | --- |
| CLUSTER | REPORTS | | PART 1/1 |
| PURPOSE: To maintain reports on the website | | Indexing: | |

|  |  |
| --- | --- |
| **CLUSTER** | **DESCRIPTION** |
| Report\_Id | To know about report ID |
| Report\_Name | To get report name |
| Report\_details | To get report details |

d

# 3.2.4 BON Class Chart

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | STUDENTS | | PART 1/1 |
| TYPES OF OBJECTS: Student’s details to the website | | Indexing: | |

|  |  |
| --- | --- |
| Queries | Student\_id, student\_fullName, student\_birthdate, student\_address, student\_email, module\_id, course\_id, student\_registration\_year |
| Commands | Add, remove, display, edit |
| Constraints | Students’ id and name must be remain constant  Two Course details cannot be given  Different module can be assigned |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | STAFFS | | PART 1/1 |
| TYPES OF OBJECTS: staff’s details to the website | | Indexing: | |

|  |  |
| --- | --- |
| Queries | Staff\_id, staff\_fullname, staff\_role, staff\_address, staff\_email |
| Commands | Add, remove, display, edit |
| Constraints | Staff id must be unique  Staff role can or cannot be changed  Staff full must cannot be null |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | COURSE | | PART 1/1 |
| TYPES OF OBJECTS: Courses details to the website | | Indexing: | |

|  |  |
| --- | --- |
| Queries | Course\_id, course\_name, course\_details |
| Commands | Add, remove, display, edit |
| Constraints | Course id must be unique  Course name and details must be different |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | MODULE | | PART 1/1 |
| TYPES OF OBJECTS: Modules details to the website | | Indexing: | |

|  |  |
| --- | --- |
| Queries | Module\_id, module\_name, module\_details |
| Commands | Add, remove, display, edit |
| Constraints | Module id must be unique  Module name and details cannot be same |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | ASSIGNMENTS | | PART 1/1 |
| TYPES OF OBJECTS: Assignmets details to the website | | Indexing: | |

|  |  |
| --- | --- |
| Queries | Assignment\_id, assignment\_name, assignment\_details, assignment\_deadline |
| Commands | Add, remove, display, edit |
| Constraints | Assignment id must be unique  Assignment deadline can or cannot be changed  Assignment details must be given |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | ATTENDANCES | | PART 1/1 |
| TYPES OF OBJECTS:Attendances details to the website | | Indexing: | |

|  |  |
| --- | --- |
| Queries | Attendance\_id, attendance\_details, student\_id |
| Commands | Add, remove, display, edit |
| Constraints | Attendance id must be unique  Student must be assigned with each attendance id |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | PERSONAL TUTOR | | PART 1/1 |
| TYPES OF OBJECTS: Personal Tutor details to the website | | Indexing: | |

|  |  |
| --- | --- |
| Queries | Staff\_id, staff\_fullname, staff\_role, course\_id, module\_id, assignment\_id |
| Commands | Add, remove, display, edit |
| Constraints | Each personal tutor is given with staff id  Must have own course and module id |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | TIMETABLE | | PART 1/1 |
| TYPES OF OBJECTS: Time Table details to the website | | Indexing: | |

|  |  |
| --- | --- |
| Queries | Timetable\_Id, timetable\_details |
| Commands | Add, remove, display, edit |
| Constraints | Timetable id must be unique  Timetable details must be different |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | DIARY | | PART 1/1 |
| TYPES OF OBJECTS: Diary details to the website | | Indexing: | |

|  |  |
| --- | --- |
| Queries | Diary\_id, diary\_details, student\_id |
| Commands | Add, remove, display, edit |
| Constraints | Diary id must be unique  Diary details must be different for students |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | REPORT | | PART 1/1 |
| TYPES OF OBJECTS: Report details to the website | | Indexing: | |

|  |  |
| --- | --- |
| Queries | Report\_id, report\_details, student\_id, staff\_id, module\_id |
| Commands | Add, remove, display, edit |
| Constraints | Report id must be unique  Each report details given to the students according to module id |
|  |  |

# 3.3 Detailed Dynamic System Design

# 3.3.1 Events Chart

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Events Planning | Week1 | Week2 | Week3 | Week4 | Week5 |
| Introduction Documentation, Mockup architecture and coding environment setup |  |  |  |  |  |
| Implementation of coding for at least 3 module/phase 1 and documentation update |  |  |  |  |  |
| Implementation of coding for at least 4 module/phase 2 and documentation update |  |  |  |  |  |
| Implementation of coding for at least 5 module/phase 3 and documentation update |  |  |  |  |  |
| Final presentation of work |  |  |  |  |  |

These are the events charts of the creation of the system where all the creation is given all these week.

# 3.3.2 Object Creation Chart

Sequence diagram of the system

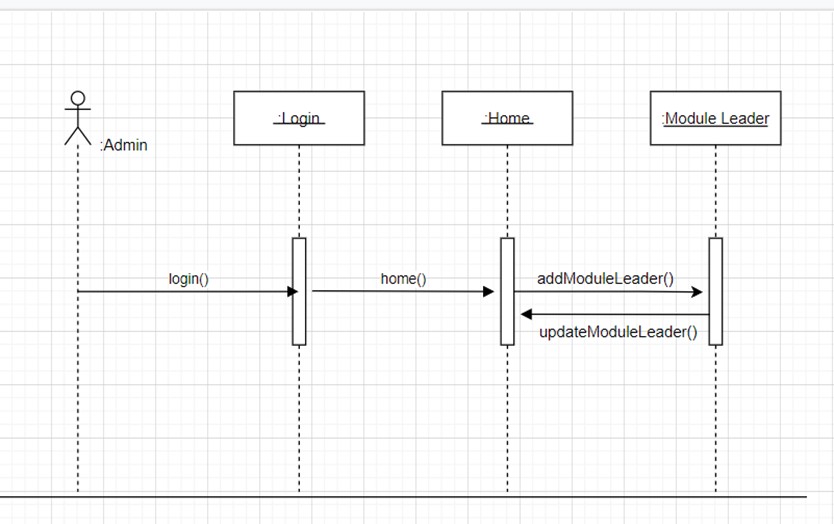


FIG: Sequence of module leader addition

Chart, box and whisker chart

Description automatically generated

FIG: Sequence of Student Addition

Chart, diagram

Description automatically generated

FIG: Sequence of Admin addition

# 3.3.3 System Scenario Chart

Diagram

Description automatically generated

FIG: UML design of Admin

Diagram

Description automatically generated

FIG: UML design of module Leader

Diagram

Description automatically generated

FIG: UML design of Student

# 3.3.4 Dynamic Diagram

# 3.4 System Database Design

# 3.4.1 E-R Model

The real model of our organization is generally shown by the ER (Entity Relationship) Diagram. It demonstrates all the objects in our system, or "entities," as well as their properties and how they relate to one another. It makes use of structured data. It helps make a database that works well with the system.

Diagram

Description automatically generated

# 3.4.2 Attributes listing

|  |  |  |
| --- | --- | --- |
| S.No. | Table Name | Attributes |
| 1 | Students | Student\_id, student\_fullname, address, date\_of\_birth, gender, email, phone\_number |
| 2 | Staff | Staff\_id, fullName, address, role, email, Phone\_number |
| 3 | Course | Course\_id, course\_name, course\_details |
| 4 | Module | Module\_id, module\_name, module\_details, course\_id |
| 5 | Assignment | Assignment\_id,assignment\_name, assignment\_details, assignment\_deadline |
| 6 | Attendance | Attendance\_id,attendance\_details, attendance\_time |
| 7 | Diaries | Diary\_id, diary\_name, diary\_details |
| 8 | TimeTable | Timetable\_id, timetable\_details, course\_id |
| 9 | Report | Report\_id, module\_id, report\_details |
| 10 | Assignment\_module | Assignment\_module\_id, assignment\_id, module\_id |