**2.2.4. Design Constraint**

**Programming language:**

The programming language used is HTML, CSS, and javascript including their libraries and the programming language used was PHP. SQL queries could also be used to obtain data stored in the database.

**Operating system:**

We were asked by clients to make the services available across all platforms, along with IOS and Android.

**Framework:**

The client has not specified, so any framework we think is more appropriate can be used.

**Front-end graphic style:**

The client seeks a bright hue to make it look more enchanting and there should be the use of bold letters.

**Design approach:**

The client advised object-oriented design, which includes both top-down and bottom-up approaches.

**Software Architecture:**

The architecture will be local to a university, but there will be secondary storage sites to safely store data and for backups.

**Hardware requirements:**

The suggested system is supposed to work on any internet-connected device that has an internet browser.

**Development standards:**

The funding is confined, and a functional system must be established within it. Because it interacts with highly personal data, the system must be able to avoid System vulnerabilities.

**2.2.5. Commercial Constraints (Total Project)**

**Schedule:**

The client has set a deadline of mid-July for the design and the study to be completed. The final product designed and created should be delivered to the customer by November 20th.

**Estimated Budget:**

Around £28,960

The expected budget is shown in the table below based on factors like:

1. The amount of time it took to finish the work and the estimated staffing costs.
2. Necessary gain margin
3. The cost and the resources of the project.

|  |  |  |  |
| --- | --- | --- | --- |
| Project details |  |  |  |
| Total project duration(week): | 5 weeks |  |  |
| Earn per hour: | £12 |  |  |
| numbers of employees: | 6 |  |  |
|  |  |  |  |
| Project outline |  |  |  |
| Segments | Weeks | Weekly hours per person | Totals earn |
| Requirement specification | 6 | 4 | £2,231 |
| Design and analysis | 7 | 5 | £2,456 |
| Deconstruction of project | 12 |  |  |
| Console | 28% | 5 | £2,395 |
| Website | 27% | 5 | £2,821 |
| Mobile application | 12% | 5 | £920 |
| Testing | 40% | 5 | £2,398 |
| Evaluation | 5% | 5 | £399 |
|  |  |  |  |
| Added expenses |  |  |  |
| Reason | Cost |  |  |
| Software/hardware | £6,000 |  |  |
| Office expenses (£110 per week) | £2,000 |  |  |
|  |  |  |  |
| Overall project expenditure | £21,250 |  |  |
|  |  |  |  |
| Other charges (10%) | £3,000 |  |  |
| Profitability (20%) | £5,500 |  |  |
| Program right | Acceptable |  |  |
|  |  |  |  |
| Overall project expense | £29,750 |  |  |

3. **System Analysis and Design (Records Management System)**

System analysis and design relate to the procedure of analyzing a company problem to optimize it through better practices and approaches. It can also be termed as the procedure of planning a new system, or one to replace or complement an existing system which basically focuses on continuous improvement and achieving profitable growth targets and it’s primary goal is to optimize better authoritarian schemes.

This section includes the system analysis and design documentation part. The system will be built using the PHP design, which separates user views, access to data, and logical web application features. All the data will be stored in a MySQL database.

**3.1. Preliminary design stages**

The preliminary design stage is a decision-making process whose goal is to explore if the existing system's issue or shortcoming is actual. The designer joins the ultimate detailed planning stage of the project, which is also the focus of this section once the proposal is executed at a higher pace. Certain features of a project's viability could be reinvestigated by the development team in the precise phase. The desire afterward is to continue or terminate the contract which is the eventual consequence.

**3.1.1. Textual analysis**

A textual test is a means of understanding how individuals make logical sense of and convey life events by deciphering the words, signs, and/or visuals included in writings. It can also be used to describe the materials and the kind of messages stored. Graphic, textual, and spoken information often provide indications for the study demonstrated. Sociocultural systems are generally regarded as altering and reflecting insights.

Appropriate classes and behaviors would be featured in the textual analysis contingent on the basis of instruction, materials, and workshops that were supplied.

|  |  |  |
| --- | --- | --- |
| S.N | Applicant category | Applicant features |
| 1 | Student | Create\_student, Amend\_student, Archive\_student, Display\_student, Assign\_student |
| 2 | staff | Create\_staff, Amend\_staff, Archive\_staff, Display\_staff, Assign\_staff |
| 3 | Course | Create\_structure\_course, Amend\_course, Display\_course, Delete\_course, Archive\_course |
| 4 | Module management | Create\_module\_management, Amend\_module management, Delete\_module\_management, Archive\_ module\_management, Display\_module\_management, Assign\_modulemanagement |
| 5 | Assignment management | Create\_assignment\_management,  Amend\_assignment\_management, Delete\_assignment\_management, Archive\_assignment\_management, Display\_assignment\_management, Assign\_assignment\_management, Mark/Grade\_assignment\_management |
| 6 | Attendance records | Create\_attendance, Amend\_attendance, Archive\_attendance, Monitor\_attendance, Display\_attendance, Action\_Poor\_Attendance |

**3.1.2. Significant event analysis**

Significant event analysis outline and investigate some of the possible actions that various kinds of users may do when utilizing the system.

**STUDENT**

|  |  |  |  |
| --- | --- | --- | --- |
| S.N | Events | Performer | Associated fields |
| 1 | View student | Admin | University id, full name, birthdate, gender |
| 2 | Add student | Admin | Full name, email, password, gender, birthdate, contact, address |
| 3 | Edit student | Admin | Full name, email, university id, birthdate |
| 4 | Delete student | Admin | University id, full name, password, gender, birthdate, contact, address |

**STAFF**

|  |  |  |  |
| --- | --- | --- | --- |
| S.N | Events | Performer | Associated fields |
| 1 | View staff | Admin | Id, full name, duty, address |
| 2 | Add staff | Admin | Full name, email, password, duty, address, gender, birthdate |
| 3 | Edit staff | Admin | Full name, email, Id, duty, birthdate |
| 4 | Delete staff | Admin | Id, full name, gender, contact, address |

**COURSE**

|  |  |  |  |
| --- | --- | --- | --- |
| S.N | Events | Performer | Associated fields |
| 1 | View course | Admin, student, module\_leaders | University id, full name, birthdate, gender, password, address, email, address, status, role |
| 2 | Add course | Admin, module\_leaders | University id, full name, gender, address, password, contact, email, role, status |
| 3 | Edit course | Admin | University id, full name, birthdate, password, email |
| 4 | Delete course | Admin | University id, full name,  gender, password, birthdate, email, address, contact, role, status |

**MODULE MANAGEMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| S.N | Events | Performer | Associated fields |
| 1 | View module | Admin  Student  Module\_leaders | University id, full name, gender, password, address, birthdate, email, contact, status role |
| 2 | Add module | Admin | University id, full name, gender, password, address, birthdate, email, contact, status role |
| 3 | Edit module | Admin | University id, full name, gender, password, address, birthdate, email, contact, status role |
| 4 | Delete module | Admin | University id, full name, gender, password, address, birthdate, email, contact, status role |

**ASSIGNMENT MANAGEMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| S.N | Events | Performer | Associated fields |
| 1 | View assignment | Module\_leaders  Students | Student Id, role, status, course id |
| 2 | Add assignment | module\_leaders  students | Student Id, role, status, course id |
| 3 | Edit assignment | Admin | Student id, course id, role, status |
| 4 | Delete assignment | Admin | Course id, status |

**ATTENDANCE RECORDS**

|  |  |  |  |
| --- | --- | --- | --- |
| S.N | Events | Performer | Associated fields |
| 1 | View attendance | Admin  Module\_leaders | University id, full name, password, gender, birthdate, role, contact, email, address, status |
| 2 | Add attendance | module\_leaders | University id, full name, status, |
| 3 | Edit attendance | Admin | Student id, full name, email, status |
| 4 | Delete attendance | Admin | Student id, full name, password, email, address, status |

**3.1.3 Commands queries and constraints**

The guidelines necessary to interact with a database to perform tasks, operations, and analyses of data are classified commands. One can make use of commands to access data as well as perform multiple activities such as designing tables, inserting data into tables, updating data, and removing tables. Queries are an information appeal to the database. Constraints are guidelines for how data in a table should be dealt with.

**STUDENT**

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | *STUDENT* | | Part: 1/1 |
| TYPE OF OBJECT  Append, view, notify, delete and upgrade students | | Author: Bipana  Created: 05/12/2022 | |
| Queries | view\_lecture, view\_assignment, view\_attendance, view\_grade, view\_ announcement, view\_module, append\_files, download\_lectures, upload\_assignment | | |
| Commands | First name, middle name, surname, course, level, address, birth date, email, contact, submit | | |
| Constraints | * Admin adds, updates, and deletes students * students can only view the modules in which they are enrolled * Students can access lectures and materials by downloading them * All students enrolled in that course are notified * Students can submit assignments online * has access to grades | | |

**STAFF**

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | *STAFF* | | Part: 1/1 |
| TYPE OF OBJECT  Append, view, delete, notify, upgrade staff | | Author: Bipana  Created: 05/12/2022 | |
| Queries | Add\_staff, view\_staff, delete\_staff, edit\_staff, update\_staff | | |
| Commands | Staff id, full name, email, gender, address, role, contact, birthdate | | |
| Constraints | * Admin has the ability to append, delete and view staff * Staff generally looks after the courses * The staff gets notified when enrolled * They are divided with their roleplay | | |

**COURSE**

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | *COURSE* | | Part: 1/1 |
| TYPE OF OBJECT  Append, view, notify, delete and upgrade courses | | Author: Bipana  Created: 05/12/2022 | |
| Queries | add\_course, delete\_course, update\_course, edit\_course, assign\_module, add\_lecture, download\_lecture, | | |
| Commands | Course topic, course illustration, course leader, submit | | |
| Constraints | * The module leader has the ability to add, upgrade, and remove courses. * Students could only visualize lectures from the module in which they are enlisted. * Students can download lectures. * A notice is being sent to all students enrolled in that course. | | |

**MODULE MANAGEMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | *MODULE* | | Part: 1/1 |
| TYPE OF OBJECT  Append, view, delete and upgrade module | | Author: Bipana  Created: 05/12/2022 | |
| Queries | University id, Full name, Password, Gender, Birthdate, Address, Contact, Email, Role, Status | | |
| Commands | Append, Alter, login, Logout, Change Password, Archive, Delete | | |
| Constraints | * Administrator is essential to look after the module * Manager should be able to manage all the courses accordingly * Courses should be frequently updated | | |

**ASSIGNMENT MANAGEMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | *ASSIGNMENT* | | Part: 1/1 |
| TYPE OF OBJECT  Append, view, notify and download assignments | | Author: Bipana  Created: 05/12/2022 | |
| Queries | Add\_assignment, upload\_assignment, submit\_assignment, download\_assignment, delete\_assignment, assign\_assignment, notify\_grade | | |
| Commands | Append, View, Download, Archive, Delete | | |
| Constraints | * Assignments could be created, modified, deleted, and notified. * Assignments are available for download. * Assignments include a grade. | | |

**ATTENDANCE RECORDS**

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | *ATTENDANCE RECORDS* | | Part: 1/1 |
| TYPE OF OBJECT  Append, view, notify, delete, and upgrade attendance | | Author: Bipana  Created: 05/12/2022 | |
| Queries | add\_attendence, view\_attendence, update\_attendence, notify\_attendence, delete\_attendence | | |
| Commands | Append, Alter, mail, Delete | | |
| Constraints | * The module leader has the ability to add, upgrade, and remove attendance. * Students can only look at their attendance records. * A notification will be sent to all students enrolled in that module. | | |

**3.4 system database design**

Database design is a series of procedures that aid in the creation, advancement, execution, and upkeep of organization data processing. The project cannot proceed unless a database is designed. A well-designed database is simple to sustain and enhances data reliability whose primary goals are to generate physical and logical concepts of a suggested database system to satisfy the needs of users and perform admirably.

**3.4.1 E-R model**

The ER framework specifies a database's schematic view which is based on factual entities and relationships among them. The ER model is viewed as a great choice for customizing databases. After determining all database tables, ER Diagram was created to ensure that the comprehensive relation between the noticed software tables. The concluded ER Diagram which presents the entire linkage between entities is shown below.

Diagram

Description automatically generated

**3.4.2 Attribute listings**

Attributes are the characteristics that define an entity. After deciding on the connection attribute linking the databases and the complete design of database structure, the database attributes were concluded and are displayed in the below table.

The database entities are as outlined.

* assignment
* assignment\_student
* course
* attendance
* module
* student

The following table details the attributes for every table.

* assignments

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.N.** | **Attribute** | **Datatype** | **Constraint** | **Null** | **Default/Extra** |
| 1 | assignment\_id | int (12) | Primary key (pk) | No | automatic increase |
| 2 | assignment\_description | text |  | No |  |
| 3 | assignment\_title | varchar (250) |  | No |  |
| 4 | assignment\_status | enum (‘Y’, ‘N’) |  | No |  |
| 5 | assignment\_deadline | date |  | No |  |
| 6 | assignment\_module\_id | int (12) | Foreign key (fk)   * module\_id | No | ON DELETE RESTRICT  ON UPDATE RESTRICT |

* assignment\_students

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.N.** | **Attribute** | **Datatype** | **Constraint** | **Null** | **Default/Extra** |
| 1 | assignment\_students\_id | int (12) | Foreign key (fk)   * assignments (assignment\_id) | No | ON UPDATE RESTRICT  ON DELETE RESTRICT |
| 2 | assignment\_students\_files | varchar (260) |  | No |  |
| 3 | assignment\_students\_uid | Int (10) | Foreign key (fk)   * users (uni\_id) | No | ON UPDATE RESTRICT  ON DELETE RESTRICT |

* courses

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.N.** | **Attribute** | **Datatype** | **Constraint** | **Null** | **Default/Extra** |
| 1 | course\_id | int (12) | Primary key (pk) | No | Automatic increase |
| 2 | course\_title | varchar (250) |  | No |  |
| 3 | course\_uid | int (10) |  | No |  |
| 4 | course\_status | enum (‘Y’, ‘N’) |  | No |  |
| 5 | course\_description | text |  | No |  |

* attendances

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.N.** | **Attribute** | **Datatype** | **Constraint** | **Null** | **Default/Extra** |
| 1 | attendance\_id | int (12) | Primary key (pk) | No | Automatic increase |
| 2 | attendance\_uid | Int (10) | Foreign key (fk)   * users (uni\_id) | No | ON UPDATE  RESTRICT  ON DELETE  RESTRICT |
| 3 | attendance\_date | datetime |  | No | CURRENT\_TIMESTAMP |
| 4 | attendance\_status | enum (‘A’, ‘O’,’X’) |  | No |  |

* modules

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.N.** | **Attribute** | **Datatype** | **Constraint** | **Null** | **Default/Extra** |
| 1 | modules\_id | int (13) | Primary key (pk) | No | Automatic increase |
| 2 | modules\_description | text |  | No |  |
| 3 | modules\_name | varchar (220) |  | No |  |
| 4 | modules\_status | enum (‘Y’, ‘N’) |  |  |  |
| 5 | modules\_code | varchar (22) |  |  |  |
| 6 | modules\_course\_id | int (12) | Foreign key (fk)   * courses (course\_id) | No | ON UPDATE RESTRICT  ON DELETE RESTRICT |

* students

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.N.** | **Attribute** | **Datatype** | **Constraint** | **Null** | **Default/Extra** |
| 1 | student\_course\_id | int (13) | Foreign key (fk)   * courses (course\_id) | No | ON UPDATE RESTRICT  ON DELETE RESTRICT |
| 2 | student\_uni\_id | int (8) | Foreign key (fk)   * users (uni\_id) | No | ON UPDATE RESTRICT  ON DELETE RESTRICT |
| 3 | student\_status | enum (‘Live’, ‘Dormant’) |  | No |  |

**8) PROJECT CONCLUSION**

In a conclusion, a fully operational student management website for WoodLand University was created in 7weeks and 5 days which was quite challenging. The development cycle was transitioned to the design phase after an interview with the client to determine their framework. Selective websites were used as references. We assembled as many details as possible. We implemented that gathered information and details to make all those features and functions attractive to the users the website was delivered to clients.

Before moving on to the implementation of the User Interface design, a wireframe and a mock-up were produced. While creating the website, numerous user immersive characteristics such as page navigation ranking, forgiveness applying method, usage of the approachable icon as well as direct link buttons across the website were borne in mind. GitHub, a revision control software, was used to collaborate on writing code with group mates. A private project was established on GitHub and shared between members.   participants were also assigned to the system's pre-analysis. Aside from that, thorough project management analyses were carried out during the activity period. A Gannt chart was developed, which provided a perfect timeline for completing each phase of the work. jira, a project management tool, was used to handle the entire project.

Project development was just not confined to simply completing the features delegated to us by the client. A lot of analysis after the completion by the participation of all the group members. Somewhat, after the project was completed, the entire project was tested. During validation, bugs were discovered. All discovered bugs have been eliminated, ensuring that the project decided to submit meets all customer requirements which might satisfy them. We performed black box testing to ensure that users could integrate with the environment we created.

If the overall operation were to be analyzed, it would be that this project focuses on teamwork instead of designing a personal self. During the project, students can understand the entire project development process, from making plans to writing code to validation and testing. Every and every member performed their best in the project. Everyone was also actively participating in making suggestions and helping other members in every part, which aided in the entire project development, and we wish to develop a similar kind of software with better and improved features.

**9) PROJECT MANAGEMENT**

Project management proved critical to the system's finalization. A total of six people were present in each group.  Each member was responsible for finishing the tasks that had been assigned to them. The document section of the group addressed all concepts on how well the system would look after being finalized.  Total project management details are provided below.

**9.1) Project Gantt Chart/WBDS/Activity List**

For managing projects, a Gantt chart was used. The Gantt chart could be used to plan the entire project opposed to time. The entire project might be summed up fortnightly as the project is scheduled on a weekly basis. The Gantt chart below discusses not only project planning, but also its weekly timeframe, closing date in weeks, and the number of weeks considered necessary to complete the project.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**9.2) Project Meeting Minutes**

MEETING 1

**Date**: 03-05-2022 | **Venue:** Room no. 303, NAMI College

Attendance

Everyone present

Agendas

* Formation of a team
* Division of tasks
* Next steps to take

**Completion of Work**

* There was no progress prior to the start of this meeting

Decisions and Conclusion

* The group planned to conduct research on the different university websites
* Look for the key assets to utilize
* Set up a working pattern
* The introduction section must be accomplished within three days (until project development)
* Member allocations for interview questions were done to plan the interview

**upcoming meeting**

Date: TBD (Individuals will be notified.) | Venue: Lawn, NAMI College

Abbreviation\* TBD – To be Decided

MEETING 2

**Date**: 01-05-2022 | **Venue:** Lawn, NAMI College

Attendance

Everyone present

Agendas

* Prior work evaluation
* Problem identification

**Completion of Work**

* Introduction completed with all aims and objectives and methodology
* Confirmation of first-phase interview questions
* The initial planning documentation has begun.

Decisions and Conclusion

* Interview plans will be finalized by the first week.
* Updates to the specification system are required.
* Work completed until preliminary system analysis and design
* Breaking down documentation and programming sections into smaller sections that can be dispersed to the appropriate members

**upcoming meeting**

Date: 06-05-2022 | Venue: Lawn, NAMI College

MEETING 3

**Date**: 06-05-2022 | **Venue:** Lawn, NAMI College

Attendance

Everyone present

Agendas

* Work evaluation
* Discussion of the issue
* Additional work allocation
* Interview questions finalization
* Discuss system design

**Completion of Work**

* Documentation for the Introduction part was completed
* Requirements study was done
* System analysis was completed till bon chart
* Mockups were ready
* Configuration of the coding environment

Decisions and Conclusion

* Discussion about how and what the system will be and is capable of
* The interview findings must be done within one week
* The residual system analysis and design work must be managed to complete
* Perform and study on various systems reference sites

**upcoming meeting**

Date: 12-05-2022 | Venue: google meet

MEETING 4

**Date**: 12-05-2022 | **Venue:** google meet

Attendance

Everyone present

Agendas

* System modeling discussion
* Conversation about task requirements
* Task distribution
* Entire project discussion

**Completion of Work**

* Interview findings and interview of stakeholders was done
* Completed all basic requirements documentation
* System analysis and design were completed
* Architecture was all ready

Decisions and Conclusion

* Modifications to the system wireframe
* System interface redesign with a new style
* Coding will begin since the wireframes and screen mockups have been completed.

**upcoming meeting**

Date: TBD (Individuals will be notified.) | Venue: classroom, NAMI College

MEETING 5

**Date**: 15-05-2022 | **Venue:** classroom, NAMI College

Attendance

Everyone present

Agendas

* Work allocation in code
* Discussion of the issue

**Completion of Work**

* Execution and coding for at least three modules have been completed
* Documentation till system analysis was fully completed

Decisions and Conclusion

* The use of GitHub for version control will be discussed
* Private repositories can now be created on GitHub
* Almost observable features will be developed until another meeting

**upcoming meeting**

**Date**: 20-05-2022 | **Venue:** Room no. 303, NAMI College

MEETING 6

**Date**: 20-05-2022 | **Venue:** Room no. 303, NAMI College

Attendance

Everyone present

Agendas

* Summary of Work
* Evaluation of Coding Advances
* upcoming projects

**Completion of Work**

* system and technical notes were done
* There hasn't been much progress in the coding department
* Database was modified to meet the needs of the client

Decisions and Conclusion

* Implementation and coding to be completed for the next three modules
* Remaining documentation update on next meeting

**upcoming meeting**

**Date**: 26-05-2022 | **Venue:** Room no. 303, NAMI College

MEETING 7

**Date**: 26-05-2022 | **Venue:** Room no. 303, NAMI College

Attendance

Everyone present

Agendas

* Summary of Work
* Analysis of Coding Progress
* upcoming projects

**Completion of Work**

* three more new modules were added and implemented
* documentation till test strategy was done

Decisions and Conclusion

* Implementation and coding to be completed for the next three modules
* Continuity to documentation

**upcoming meeting**

**Date**: 02-06-2022 | **Venue:** Room no. 303, NAMI College

MEETING 8

**Date**: 02-06-2022 | **Venue:** Room no. 303, NAMI College

Attendance

Everyone present

Agendas

* Final documentation summary
* Project assessment
* Problem analysis

**Completion of Work**

* All documentation has been completed along with system activity diagram
* Next three modules were added and implemented

Decisions and Conclusion

* Additional documentation and testing planning
* Coders were assigned to work on the two sections, and testers were assigned to test the functionalities.
* Changes in color scheme and design

**upcoming meeting**

**Date**: 09-06-2022 | **Venue:** Room no. 303, NAMI College

MEETING 9

**Date**: 09-06-2022 | **Venue:** Room no. 303, NAMI College

Attendance

Everyone present

Agendas

* Finalization of the entire project
* Work allocation for presentations
* Presentation content creation
* Final project testing prior to demo
* Bug fixes, if any exist

**Completion of Work**

* The project is complete (could contain some hidden bugs)

Decisions and Conclusion

* All team members approved the overall work.
* Distribution of presentations among teammates
* PowerPoint slide preparation
* Bug fixes and project test results

**upcoming meeting**

**Date**: 11-06-2022 | **Venue:** Room no. 303, NAMI College

MEETING 10

**Date**: 11-06-2022 | **Venue:** Room no. 303, NAMI College

Attendance

Everyone present

Agendas

* Complete documentation and create a zip file for submitting
* Complete all files before submitting them.

**Completion of Work**

* All the discovered bugs have been fixed, and all documentation has been completed
* Appendices and evidence of tool usage are included in the documentation.

Decisions and Conclusion

* The project has come to an end
* Before submitting, all the files were finalized
* Everything has been prepared for submission (naming, design, and comments)

**upcoming meeting**

*End of project*

**9.3) Project Quality Plan/Strategy**

It is the plan that outlines the tasks, requirements, equipment, and practices that are required to produce a project of quality. The below list consists of features and functionality, also with their level of importance as well as time frame.

Level of importance:

1 = indicates lower preference

2 = indicates a moderate preference

3 = top preference

4 = extremely high preference

5 = Critical

The table below shows the features and functionality planned and their priority for admin authentication and authorization.

|  |  |  |  |
| --- | --- | --- | --- |
| **S.N.** | **Features and Functionality** | **Level of importance** | **Estimated completion date** |
| 1 | There is the admission process for the new students. | 5 | 5 days. |
| 2 | Module, courses can be added to the system. | 5 | 3 days. |
| 3 | Assignments can be added. | 4 | 2 day |
| 4 | Students can be added, edited, and deleted. | 4 | 1 day |
| 5 | New module and courses can be added to the system. | 3 | 2 day |

The table below shows the features and functionality planned and their priority for student’s authentication and authorization.

|  |  |  |  |
| --- | --- | --- | --- |
| **S.N.** | **Features and Functionality** | **Level of importance** | **Estimated completion date** |
| 1 | Students can add the assignment they were provided with | 5 | 3days. |
| 2 | Students can view their Information. | 5 | 2 day |
| 4 | Students can view the courses and modules they are assigned. | 4 | 3 days. |
| 5 | Students have access to diary management where they can add, view and edit diary. | 1 | 2 day |

**9.4) Project Work Log**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **INTRODUCTION** | Member Name | Member Name | Member Name | Member Name | Member Name | Member Name |
| Project Background | Sakshyam  Aryal  (100%) |  |  |  |  |  |
| Project Aims and Objectives | Sakshyam  Aryal  (100%) |  |  |  |  |  |
| Project Development Methodology | Sakshyam  Aryal  (80%) | Prasanna Tandukar  (40%) |  |  |  |  |
| **REQUIREMENTS ENGINEERING** | Member Name | Member Name | Member Name | Member Name | Member Name | Member Name |
| Interview plans | Ankit Dahal  (20%) | Sakshyam  Aryal  (10%) | Kritika Shrestha  (10%) | Bipana Bista  (20%) | Prasanna Tandukar  (10%) | Robin Tondon  (20%) |
| Interviews | Ankit Dahal  (20%) | Bipana Bista (15%) | Kritika Shrestha  (15%) | Prasanna Tandukar  (15%) | Robin Tondon  (15%) | Sakshyam  Aryal  (20%) |
| Interview findings | Sakshyam Aryal  (50%) | Ankit Dahal (50%) |  |  |  |  |
| Comparable Software System Review | Kritika Shrestha (50%) | Prasanna Tandukar (20%) | Sakshyam Aryal (30%) |  |  |  |
| Development Relevant Legislation | Bipana Bista (50%) | Prasanna Tandukar (50%) |  |  |  |  |
| Academic literature Review | Robin Tondon (40%) | Sakshyam Thir (60%) |  |  |  |  |
| User group questionnaire | Prasanna Tandukar (30%) | Robin tondon (30%) | Bipana Bista (20%) | Kritika Shrestha (20%) |  |  |
| **REQUIREMENTS SPECIFICATION** | Member Name | Member Name | Member Name | Member Name | Member Name | Member Name |
| Problem Domain Description | Ankit Dahal (50%) | Bipana Bista  (50%) |  |  |  |  |
| Existing business limitations | Kritika Shrestha (50%) | Robin Tondon  (50%) |  |  |  |  |
| Functional Requirements | Kritika Shrestha (100%) |  |  |  |  |  |
| Performance Requirements | Kritika Shrestha (80%) | Sakshyam Aryal  (20%) |  |  |  |  |
| Design Constraints | Bipana Bista (100%) |  |  |  |  |  |
| Commercial Constraints | Bipana Bista  (100%) |  |  |  |  |  |
| **SYSTEM ANALYSIS AND DESIGN** | Member Name | Member Name | Member Name | Member Name | Member Name | Member Name |
| Preliminary Design Stages |  |  |  |  |  |  |
| Textual Analysis | Bipana Bista (100%) |  |  |  |  |  |
| Significant Event Analysis | Bipana Bista  (50%) | Robin Tondon  (50%) |  |  |  |  |
| Commands Queries and Constraints | Bipana Bista (100%) |  |  |  |  |  |
| Detailed Static System Designs |  |  |  |  |  |  |
| First Draft BON System Architecture Diagram | Prasanna Tandukar (100%) |  |  |  |  |  |
| BON system chart | Prasanna Tandukar (80%) | Ankit Dahal (20%) |  |  |  |  |
| BON cluster Charts | Prasanna Tandukar (100%) |  |  |  |  |  |
| BON Class Charts | Prasanna Tandukar  (100%) |  |  |  |  |  |
| Detailed dynamic system designs |  |  |  |  |  |  |
| Events charts | Prasanna Tandukar  (100%) |  |  |  |  |  |
| Object creation charts | Sakshyam Aryal  (50%) | Ankit Dahal  (50%) |  |  |  |  |
| System scenario charts | Robin Tondon  (50%) | Prasanna Tandukar  (50%) |  |  |  |  |
| Dynamic diagrams | Bipana Bista (50%) | Kritika Shrestha (50%) |  |  |  |  |
| System Database Design |  |  |  |  |  |  |
| E-R Model | Bipana Bista (50%) | Sakshyam Aryal (50%) |  |  |  |  |
| Attribute Listings | Bipana Bista (70%) | Robin Tondon  (30%) |  |  |  |  |
| SYSTEM INTERFACE DESIGNS | Member Name | Member Name | Member Name | Member Name | Member Name | Member Name |
| Draft Interface Designs |  |  |  |  |  |  |
| Wireframes | Prasanna Tandukar  (100%) |  |  |  |  |  |
| System Navigation Diagram | Prasanna Tandukar  (60%) | Kritika Shrestha  (40%) |  |  |  |  |
| System Screen mock-ups | Sakshyam Aryal  (50%) | Robin Tondon  (50%) |  |  |  |  |
| System Activity Event Diagrams | Prasanna Tandukar(60%) | Bipana Bista  (40%) |  |  |  |  |
| Design Revisions | Ankit Dahal (100%) |  |  |  |  |  |
| Heuristic Evaluation | Robin Tondon (100%) |  |  |  |  |  |
| **PROJECT’S CODING** | Member Name | Member Name | Member Name | Member Name | Member Name | Member Name |
| Database Implementation | Prasanna Tandukar  (70%) | Sakshyam Aryal  (30%) |  |  |  |  |
| Front End Coding | Prasanna Tandukar (60%) | Ankit Dahal  (40%) |  |  |  |  |
| Back End Coding  (Making Website Dynamic) | Prasanna Tandukar  (100%) |  |  |  |  |  |
| SYSTEM BUILD AND TECHNICAL NOTES | Member Name | Member Name | Member Name | Member Name | Member Name | Member Name |
| Technically Difficult Code Sections | Robin Tondon  (60%) | Ankit Dahal  (40%) |  |  |  |  |
| On-going Testing Methodology | Kritika Shrestha  (50%) | Sakshyam Aryal  (50%) |  |  |  |  |
| Final System Interface Displays | Prasanna Tandukar  (70%) | Bipana Bista  (20%) | Ankit Dahal  (15%) |  |  |  |
| **TEST STRATEGY** | Member Name | Member Name | Member Name | Member Name | Member Name | Member Name |
| Overview of Test Strategy | Kritika Shrestha  (60%) | Prasanna Tandukar  (30%) | Robin Tondon (20%) |  |  |  |
| Sample Test Results | Kritika Shrestha  (50%) | Bipana Bista  (25%) | Ankit Dahal  (25%) |  |  |  |
| **SYSTEM EVALUATION** | Member Name | Member Name | Member Name | Member Name | Member Name | Member Name |
| Usability Evaluation Strategy | Sakshyam Aryal (50%) | Ankit Dahal (50%) |  |  |  |  |
| Initial System Pilot Trials/Results | Prasanna Tandukar  (100%) |  |  |  |  |  |
| Final System Trial/Results | Prasanna Tandukar (50%) | Robin Tondon (50%) |  |  |  |  |
| **PROJECT MANAGEMENT** | Member Name | Member Name | Member Name | Member Name | Member Name | Member Name |
| Project Gantt Chat/WBDS/Activity List | Bipana Bista  (100%) |  |  |  |  |  |
| Project Meeting Minutes | Bipana Bista  (80%) | Kritika Shrestha  (20%) |  |  |  |  |
| Project Quality Plan/Strategy | Prasanna Tandukar  (60%) | Ankit Dahal  (40%) |  |  |  |  |
| Design Documentation Typewriting and Formatting | Sakshyam Aryal  (50%) | Robin Tondon  (50%) |  |  |  |  |
| Questionnaires | Ankit Dahal  (15%) | Bipana Bista  (15%) | Robin Tondon  (20%) | Kritika Shrestha  (15%) | Prasanna Tanukar  (15%) | Sakshyam Aryal  (20%) |
| Presentation | Ankit Dahal  (15%) | Bipana Bista  (20%) | Robin Tondon  (15%) | Kritika Shrestha  (20%) | Prasanna Tanukar  (15%) | Sakshyam Aryal  (15%) |
| Presentation Delivery | Ankit Dahal  (20%) | Bipana Bista  (15%) | Robin Tondon  (15%) | Kritika Shrestha  (15%) | Prasanna Tanukar  (20%) | Sakshyam Aryal  (15%) |