

ATMIYA UNIVERSITY

RAJKOT



A

Report On

PROJECT TITLE

Under subject of

MINI PROJECT

B. TECH, Semester – VI

(Computer Engineering)

Submitted by:

1. Fadadu Vasukumar Pravinbhai

180002022

Prof. Nirali P. Borad

(Faculty Guide)

Prof. Tosai M. Bhalodia

(Head of the Department)

Academic Year

(2020-21)

CANDIDATE'S DECLARATION

We hereby declare that the work presented in this project entitled “**Offline Exam management system**” submitted towards completion of project in **6th Semester** of B. Tech. (Computer Engineering) is an authentic record of our original work carried out under the guidance of “**Prof. Nirali P. Borad**”.

We have not submitted the matter embodied in this project for the award of any other degree.

Semester: 6th

Place: Rajkot

Signature:

Vasukumar Fadadu (180002022)

ATMIYA UNIVERSITY

RAJKOT



CERTIFICATE

Date: April 27, 2021

This is to certify that the “**Offline Exam management system**” has been carried out by **Vasukumar Pravinbhai Fadadu** under my guidance in fulfillment of the subject Mini Project in COMPUTER ENGINEERING (6th Semester) of Atmiya University, Rajkot during the academic year 2020-21.

Prof. Nirali P. Borad

(Project Guide)

Prof. Tosai M. Bhalodia

(Head of the Department)

INDEX

| Sr. No. | TITLES | Page No. |
|--------------------|---|---------------------|
| | Acknowledgement | I |
| | Abstract | II |
| | List of Figures | III |
| | List of Tables | IV |
| 1. | Introduction | 5 |
| 1.1 | Purpose | 5 |
| 1.2 | Scope | 5 |
| 1.3 | Technology and tool | 5 |
| 2. | Project Management | 9 |
| 2.1 | Project Planning | 9 |
| 2.1.1 | Project Development Approach and Justification: | 9 |
| 2.1.2 | Milestones and Deliverables: | 10 |
| 2.2 | Project Scheduling | 11 |
| 2.3 | Risk Management | 11 |
| 2.3.1 | Risk Identification | 11 |
| 2.3.2 | Risk Analysis | 12 |
| 3. | System Requirements Study | 14 |
| 3.1 | Hardware and Software Requirements | 14 |
| 3.1.1 | Server-side hardware requirement | 14 |
| 3.1.2 | Software requirement | 14 |
| 3.1.3 | Client-Side requirement | 14 |
| 3.2 | Constraints | 15 |
| 3.2.1 | Hardware Limitation | 15 |
| 3.2.2 | Reliability requirements | 15 |
| 3.2.3 | Interfacing with other systems | 15 |
| 4. | System Analysis | 16 |
| 4.1 | Study of Current System | 16 |
| 4.2 | Problem and Weaknesses of Current System | 16 |

| | | |
|-----------|---|-----------|
| 4.3 | Requirements of New System | 16 |
| 4.3.1 | User Requirements | 16 |
| 4.3.2 | System Requirements | 16 |
| 4.4 | Feasibility Study | 17 |
| 4.4.1 | Does the system contribute to overall objectives of the organization? | 17 |
| 4.4.2 | Can the system be implemented using the current technology and within the given cost and schedule constraints | 17 |
| 4.4.3 | Can the system be integrated with other system which are already in place? | 17 |
| 4.5 | Feature of New System | 17 |
| 4.6 | Selection of Hardware and Software and Justification | 18 |
| 5 | System Design | 19 |
| 5.1 | Input /output interface | 19 |
| 5.2 | Interface Design | 21 |
| 5.2.1 | Class Diagram | 21 |
| 5.2.2 | Use Case Diagram | 22 |
| 5.2.3 | Activity Diagram | 23 |
| 5.2.4 | Data Flow Diagram | 24 |
| 6 | Code Implementation | 26 |
| 6.1 | Implementation Environment | 26 |
| 6.2 | Program/Module Specification | 26 |
| 6.3 | Coding Standards | 26 |
| 7 | Testing | 28 |
| 7.1 | Testing Strategy | 28 |
| 7.2 | Testing Method | 28 |
| 7.2.1 | Unit Testing | 28 |
| 7.2.2 | Integration Testing | 28 |
| 7.2.3 | Validation Testing | 29 |
| 8 | Limitations and Future Enhancement | 30 |
| 8.1 | Limitations | 30 |
| 8.2 | Future Enhancement | 30 |
| 9 | Conclusion | 31 |
| 10 | References | |

ACKNOWLEDGEMENT

We have taken many efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them.

We are highly indebted to **Prof. Nirali P. Borad** for their guidance and constant supervision as well as for providing necessary information regarding the Mini Project titled “**Offline Exam management system**”. We would like to express our gratitude towards staff members of Computer Engineering Department, Atmiya University for their kind co- operation and encouragement which helped us in completion of this project.

We even thank and appreciate to our colleague in developing the project and people who have willingly helped us out with their abilities.

Vasukumar Fadadu (180002022)

ABSTRACT

The purpose of developing offline exam management system is to computerized the traditional way of taking exams. Another purpose for developing this Web application is to generate the detail automatically about the all exam.

Keyword: Student, Admin, program, Subject, exam.

LIST OF FIGURES

| Figure | Table Title | Page |
|----------------|-------------------------|-------------|
| No. | | No. |
| 5.1.1 | Login page | 19 |
| 5.1.2 | Admin page | 20 |
| 5.2.1.1 | Class diagram | 21 |
| 5.2.2.1 | Use case diagram | 22 |
| 5.2.3.1 | Activity diagram | 23 |
| 5.2.3.2 | Activity diagram | 23 |
| 5.2.4.1 | DFD zero level | 24 |
| 5.2.4.2 | DFD first level | 24 |
| 5.2.4.3 | DFD two level | 25 |

LIST OF TABLES

| Table No. | Table Title | Page No. |
|----------------------|---|---------------------|
| 2.1.2.1 | Milestones and Deliverables | 10 |
| 3.1.1.1 | Server-side Hardware Requirement | 14 |
| 3.1.2.1 | Software Requirements | 14 |
| 3.1.3.1 | Client-side Requirements | 14 |
| 4.6.1 | Hardware Requirement | 18 |
| 4.6.2 | Software Requirements | 18 |
| 4.6.3 | Client-side Requirements | 18 |

1. Introduction

1.1 Purpose:

The purpose of developing examination management system is to computerized the traditional way of taking exams. Another purpose for developing this software is to generate the report automatically about the sitting arrangement during exams at the end of the session or in the between of the session.

1.2 Scope:

. The scope of the project is the system on which the software is installed, i.e. the project is developed as a desktop application, and it will work for a particular institute. But later on the project can be modified to operate it online.

1.3 Technology and tool:

1. HTML:

- Hypertext Markup Language (HTML) is the main markup language for creating
- web pages and other information that can be displayed in a web browser.

HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets (like <html>), within the web page content. HTML tags most commonly come in pairs like <h1> and </h1>, although some tags represent empty elements and so are unpaired, for example .

- The first tag in a pair is the start tag, and the second tag is the end tag (they are also called opening tags and closing tags). In between these tags' web designers can add text, further tags, comments and other types of text-based content.
- The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page

2. CSS:

- Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation semantics (the look and formatting) of a document written in a markup language. Its most common application is to style web pages written in HTML and XHTML, but the language can also be applied to any kind of XML document, including plain XML, SVG and XUL. CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colors, and fonts.
- This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content (such as by allowing for table less web design).
- CSS specifies a priority scheme to determine which style rules apply if more than one rule matches against a particular element. In this so-called cascade, priorities or weights are calculated and assigned to rules, so that the results are predictable.
- The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/CSS is registered for use with CSS by RFC 2318 (March 1998), and they also operate a free CSS validation service.

3. JavaScript:

- JavaScript (JS) is an interpreted computer programming language. As part of web browsers, implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed. It has also become common in server-side programming, game development and the creation of desktop applications.
- JavaScript is a prototype-based scripting language with dynamic typing and has first class functions. Its syntax was influenced by C. JavaScript copies many names and naming conventions from Java, but the two languages are otherwise unrelated and have very different semantics. The key design principles within JavaScript are taken from the Self and Scheme programming languages.

- It is a multi- paradigm language, supporting object-oriented, imperative, and functional programming styles.
- JavaScript's use in applications outside of web pages—for example, in PDF documents, site-specific browsers, and desktop widgets—is also significant. Newer and faster JavaScript VMs and frameworks built upon them (notably Node.js) have also increased the popularity of JavaScript for server-side web applications.
- JavaScript was formalized in the ECMA Script language standard and is primarily used as part of a web browser (client-side JavaScript). This enables programmatic access to computational objects within a host environment.

Back End: Back End technologies used in the website are:

1. SQL:

- SQL (Structured Query Language) is a special-purpose programming language designed for managing data held in a relational database management system (RDBMS).
- Originally based upon relational algebra and tuple relational calculus, SQL consists of a data definition language and a data manipulation language. The scope of SQL includes data insert, query, update and delete, schema creation and modification, and data access control. Although SQL is often described as, and to a great extent is, a declarative language (4GL), it also includes procedural elements. SQL was one of the first commercial languages for Edgar F. Codd's relational model, as described in his influential 1970 paper "A Relational Model of Data for Large Shared Data Banks". Despite not entirely adhering to the relational model as described by Codd, it became the most widely used database language.
- SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standards (ISO) in 1987. Since then, the

standard has been enhanced several times with added features. But code is not completely portable among different database systems, which can lead to vendor locking.

2.Net:

.NET is a free, cross-platform, open source developer platform for building many different types of applications.

With .NET, you can use multiple languages, editors, and libraries to build for web, mobile, desktop, games, and IoT.

- Application frameworks and libraries
- .NET Standard
- Runtime Components
- Language Compilers
- Languages – C#, F#, and Visual Basic
- Tools - Visual Studio, Visual Studio for Mac, Visual Studio Code, and Command Line Interface (CLI) are tools used to build, test, and deploy .NET applications.

2. Project Management

2.1 Project Planning:

Project Planning is concerned with identifying and measuring the activities, milestones and deliverables produced by the project. Project planning is undertaken and completed sometimes even before any development activity starts. Project planning consists of following essential activities:

- Offline exam management system also manage the subject details online for class details, semester details, student.
- It tracks all the information of exam, subject, class etc.
- Manage the information of exam.
- Shows the information and description of the student, subject.
- Manage the information of student.
- Editing, adding and updating of records is improved which results in proper resource management of student data.
- Manage the information of course.

2.1.1 Project Development Approach and Justification:

A Software process model is a simplified abstract representation of a software process, which is presented from a particular perspective. A process model for software engineering is chosen based on the nature of the project and application, the methods and tools to be used, and the controls and deliverables that are required. All software development can be characterized as a problem-solving loop which in four distinct stages is encountered:

- Requirement analysis
- Design
- Coding
- Testing
- Deployment

2.1.2 Milestones and Deliverables:

Management needs information. As software is tangible, this information can only be provided as documents that describe the state of the software being developed without this information it is impossible to judge progress at different phases and therefore schedules cannot be determined or updated.

Milestone is an end point of the software process activity. At each milestone there should be formal output such as report that can be represented to the management. Milestones are the completion of the outputs for each activity. Deliverables are the requirements definition and the requirements specification.

Milestone represents the end of the distinct, logical stage in the project. Milestone may be internal project results that are used by the project manager to check progress. Deliverables are usually Milestones but reverse need not be true. We have divided the software process into activities for the following milestone that should be achieved.

| Software Process Activity | Milestone |
|--|--|
| Project Plan | Project schedule |
| Requirement Collection | User requirements, System Requirements |
| Data flow analysis | DFD, System flow |
| Design 1. Database design 2. User Interface design | System Design Document |
| Implementation 1. Code for giving security 2. Code for reports | Access Rights Reports Generation |
| Testing | Setting validations and error messages |

Table 2.1.2.1 Milestones and Deliverables

2.2 PROJECT SCHEDULING:

The scheduling is the peak of a planning activity, a primary component of software project management. When combined with estimation methods and risk analysis, scheduling establishes a roadmap for project management. The characteristics of the project are used to adapt an appropriate task set for doing work.

2.3 Risk Management:

Risk management consists of a series of steps that help a software development team to understand and manage uncertain problems that may arise during the course of software development and can plague a software project.

Risks are the dangerous conditions or potential problems for the system which may damage the system functionalities to very high level which would not be acceptable at any cost. so in order to make our system stable and give its 100% performance we must have identify those risks, analyze their occurrences and effects on our system and must prevent them to occur.

2.3.1 Risk Identification:

Risk identification is a first systematic attempt to specify risks to project plan, scheduling resources, project development. It may be carried out as a team process using brainstorming approach.

Technology risk: Technical risks concern implementation, potential design, interfacing, testing, and maintenance problems.

- Database Corruptness
- Garbage Collection

People Risks: These risks are concerns with the team and its members who are taking part in developing the system.

- Leaking an important data
- Failure of the administration
- Lack of knowledge,
- Lack of clear product vision.
- Technical staff conflict

- Poor communication between people. Tools Risks: These are more concerned with tools used to develop the system
- Tools containing virus.

General Risks: General Risks are the risks, which are concerned with the mentality and resources.

- Rapidly changing requirements.
- Lack of resources can cause great harm to efficiency and timely productivity.
- Changes in requirements can cause a great harm to implementation, designing and schedule of developing the system.
- Insufficient planning and task identification.

2.3.2 Risk Analysis:

“Risk analysis = risk assessment + risk management + risk communication.”

Risk analysis is employed in its broadest sense to include:

Risk assessment:

Involves identifying sources of potential harm, assessing the likelihood that harm will occur and the consequences if harm does occur.

For this project It might be: -

- System Crash

Risk management:

Evaluates which risks identified in the risk assessment process require management and selects and implements the plans or actions that are required to ensure that those risks are controlled.

Precautions taken to make risks minimal are as under:

- Periodical backups are taken to avoid major loss in case of system crash.

Risk communication: Involves an interactive dialogue between stakeholders and risk assessors and risk managers which actively informs the other processes.

Steps taken for risk communication is as under:

- Probability of certain risks is negotiated with client.
- All the possible risks are listed out during communication and project is developed taking care of risks.

3. System Requirements Study

3.1 Hardware and Software Requirements:

3.1.1 Server-side hardware requirement:

| Devices | Description |
|-----------|--------------------------------|
| Processor | Intel Core Duo 2.0 GHz or more |
| RAM | 512 MB or more |
| Hard Disk | 10 GB or more |

Table 3.1.1.1 Server-side Hardware Requirement

3.1.2 Software Requirements :

| For which | Software |
|--------------------|-------------------------|
| Operating System | Windows7/8/10, Linux |
| Front End | Html, CSS |
| Back End | .Net C#, MySQL database |
| Scripting Language | JavaScript |

Table 3.1.2.1 Software Requirements

3.1.3 Client-Side requirement:

| For which | Requirement |
|-----------|-------------------------------|
| Browser | Any compatible browser device |

Table 3.2.3.1 Client-side Requirements

3.2 Constraints:

3.2.1 Hardware Limitations:

The major hardware limitations faced by the system are as follows:

If the appropriate hardware is not there like processor, RAM, hard disks

- The problem in processing requests of client
- If appropriate storage is not there our whole database will crash due to less storage because our main requirement is large storage.

3.2.2 Reliability Requirements:

Since many users can access the server simultaneously, load on the server becomes very high. Hence, the server should be of enough high configurations. There should be high back up storage and management of huge data for overall detail of student, data of result, multiple student profile.

The Reliability requirements are the validations used to protect the system against one or more incorrect activities. Without proper validation of the system, the failure possibilities of it grow higher so it is must to understand the proper validation of the system and must implement them. All the required validator controls spend very good role to keep the system secure from any unauthorized or incorrect information. In all these validation actions if system found one or more entries violating validation rules then user will be warned by proper error messages and the details or the record is not going to be saved until corrections are made to them.

4. System Analysis

4.1 Study of Current System:

Current system in Offline Exam Management System divided into one sections like admin.

4.2 Problem and Weaknesses of Current System:

The current system is undoubtedly well-designed for crowd funding portal expenses but it has some following limitations:

- Lack of awareness of this system.
- Some Security related issues may be created
- Idea stealing problem is there.

4.3 Requirements of New System:

Requirements specification adds further information to the requirements definition.

4.3.1. User Requirements:

Here two types of user can use this project

- Admin: Admin users manage the entire project like login, management of Project
 - student
 - Update
 - Manage exam data

4.3.2 System Requirements:

- Usability:

The interface should use terms and concepts, which are drawn from the experience of the people who will make most of the system. For example, basic social networking concepts are followed.

- Efficiency:

The system should provide easy and fast access.

4.4 FEASIBILITY STUDY:

An important outcome of the preliminary investigation is the determination that the system is feasible or not. The main aim of the feasibility study activity is to determine whether it would be financially and technically feasible to develop a project. The feasibility study activity involves the analysis of the problem and collection of all relevant information relating to the product such as the different data items which would be input to the system, the processing required to be carried out on these data, the output required to be produced by the system as well as the various constraints on the behavior of the system.

4.4.1. Does the system contribute to the overall objectives of the organization?

The main aim of behind development of this system is to provide a Desktop application that can reduce the pre-process printing and reduce work of result. Also provide the best portal who want to learn something and want to know something innovative and for the social improvement.

4.4.2. Can the system be implemented using the current technology and within the given cost and schedule constraints?

o the system can be easily implemented using existing technology. The technology used is PHP which is user friendly and freeware. After seeing the functionality that system provides the cost of developing the application does not matter.

4.4.3. Can the system be integrated with other system which is already in place?

Yes, the system can be integrated with other system which is already in place. If other system wants to use our functionality it can be easily integrate.

4.5 FEATURES OF NEW SYSTEM:

- Simple way to manage offline exam.
- Manage student data.
- Manage course detail.
- Manage all exam detail.

4.6 SELECTION OF HARDWARE AND SOFTWARE:

- **Hardware Selection:**

| Devices | Description |
|-----------|--------------------------------|
| Processor | Intel Core Duo 2.0 GHz or more |
| RAM | 512 MB or more |
| Hard Disk | 10 GB or more |

Table 4.6.1 Hardware Requirement

- **Software Requirements:**

| For which | Software |
|--------------------|-------------------------|
| Operating System | Windows7/8/10, Linux |
| Front End | Html, CSS |
| Back End | .Net C#, MySQL database |
| Scripting Language | JavaScript |

Table 4.6.2 Software Requirements

- **Client-Side requirement:**

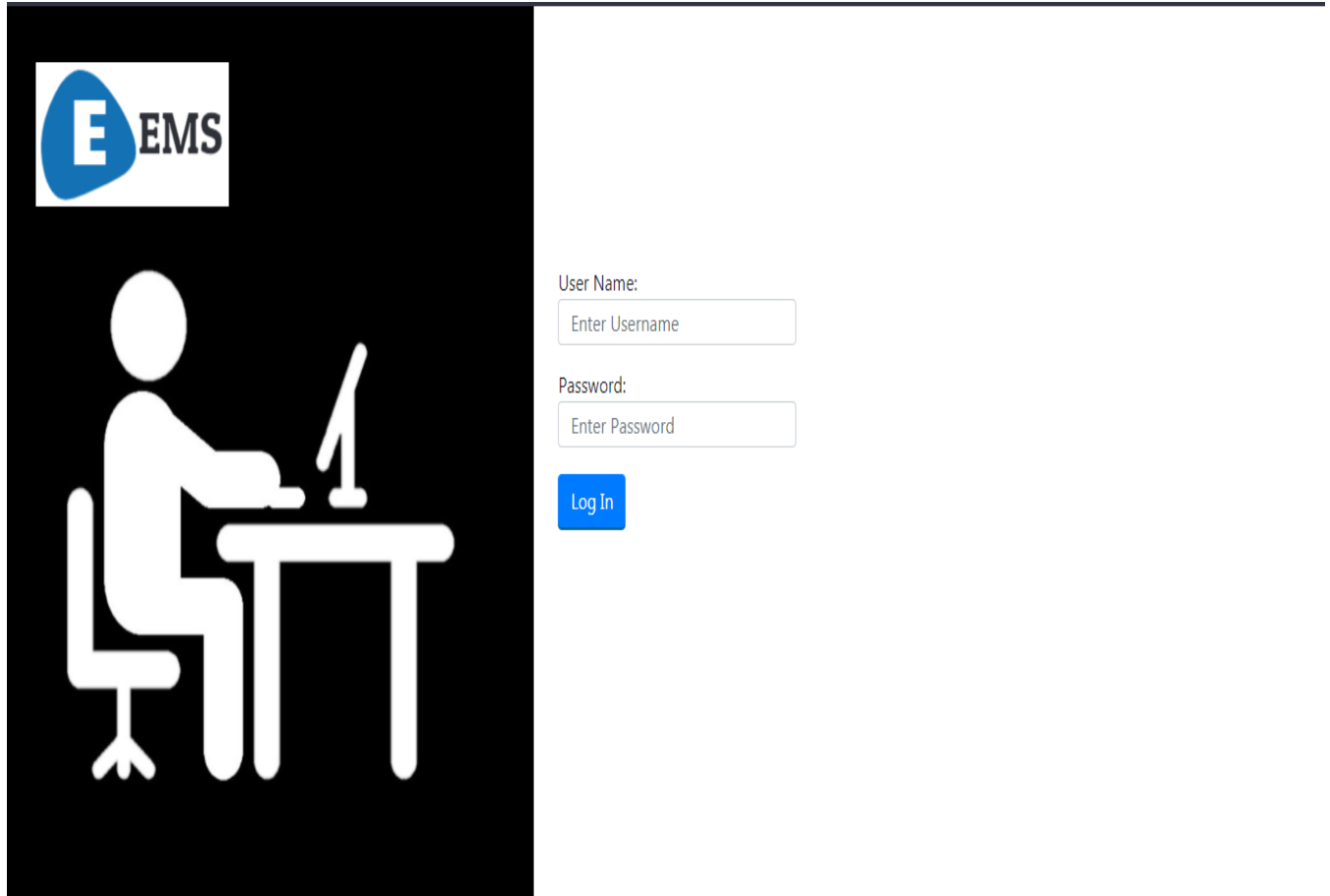
| For which | Requirement |
|-----------|-------------------------------|
| Browser | Any compatible browser device |

Table 4.6.3 Client-side Requirements

5. System Design

5.1 Input/output interface:

Login page:



The login page features a dark sidebar on the left with the EMS logo (a blue circle with a white 'E' and the text 'EMS') and a white icon of a person sitting at a desk with a monitor. The main content area is white and contains the following elements:

- User Name:** A text label above a text input field containing the placeholder text "Enter Username".
- Password:** A text label above a text input field containing the placeholder text "Enter Password".
- Log In:** A blue rectangular button with white text.

Fig.5.1.1 Login page

Admin page:

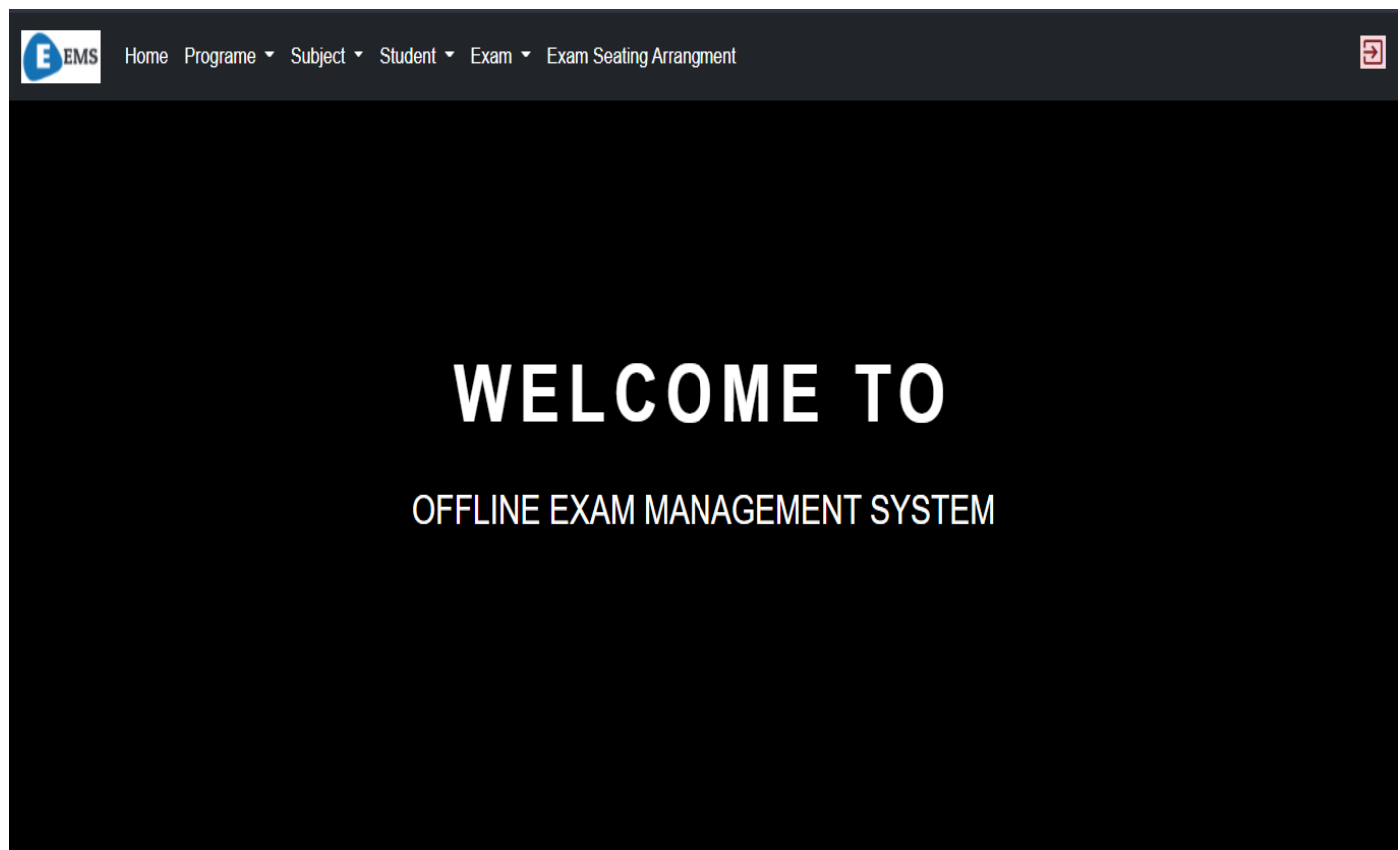


Fig.5.1.2 admin page

5.2 Interface Design:

5.2.1 Class Diagram:

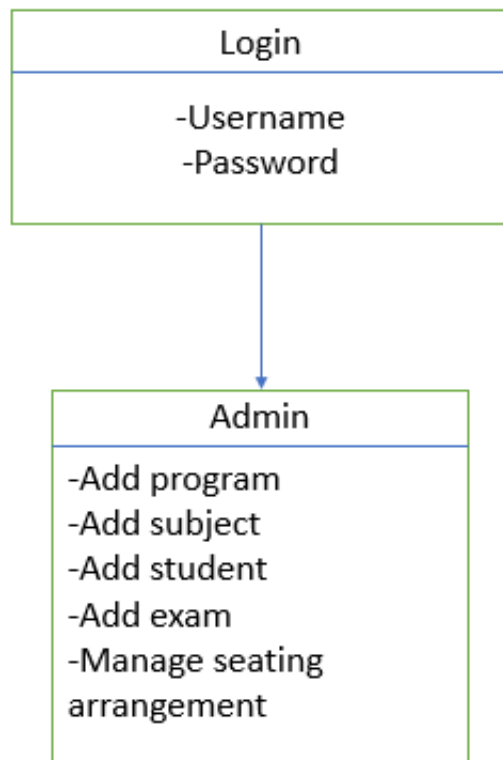


Fig. 5.2.1.1 Class Diagram

5.2.2 Use Case Diagram:

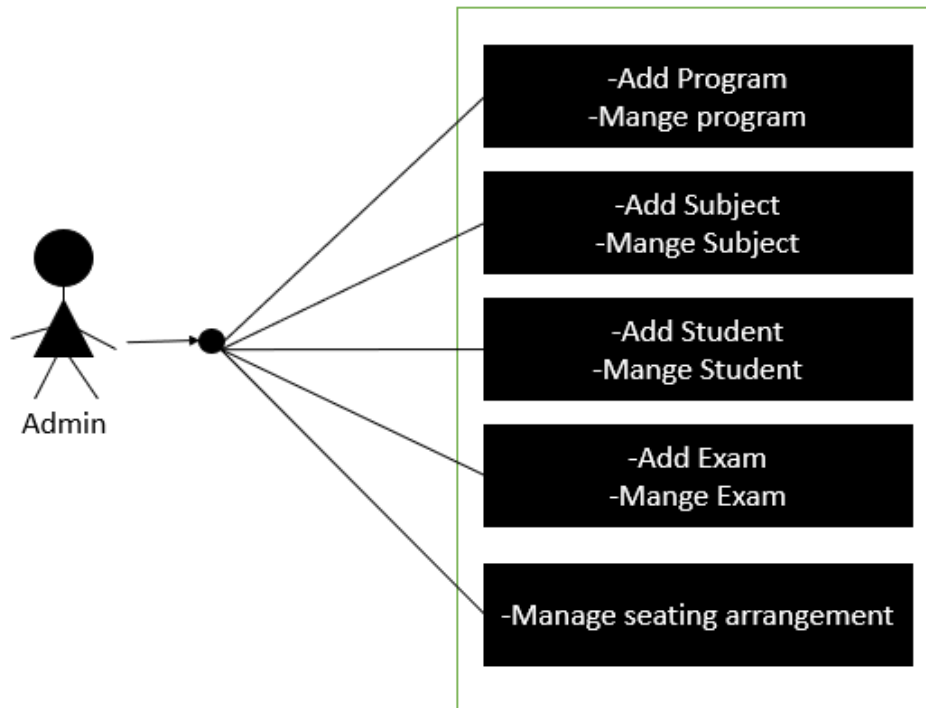


Fig.5.2.2.1 use case diagram

5.2.3 Activity Diagram:

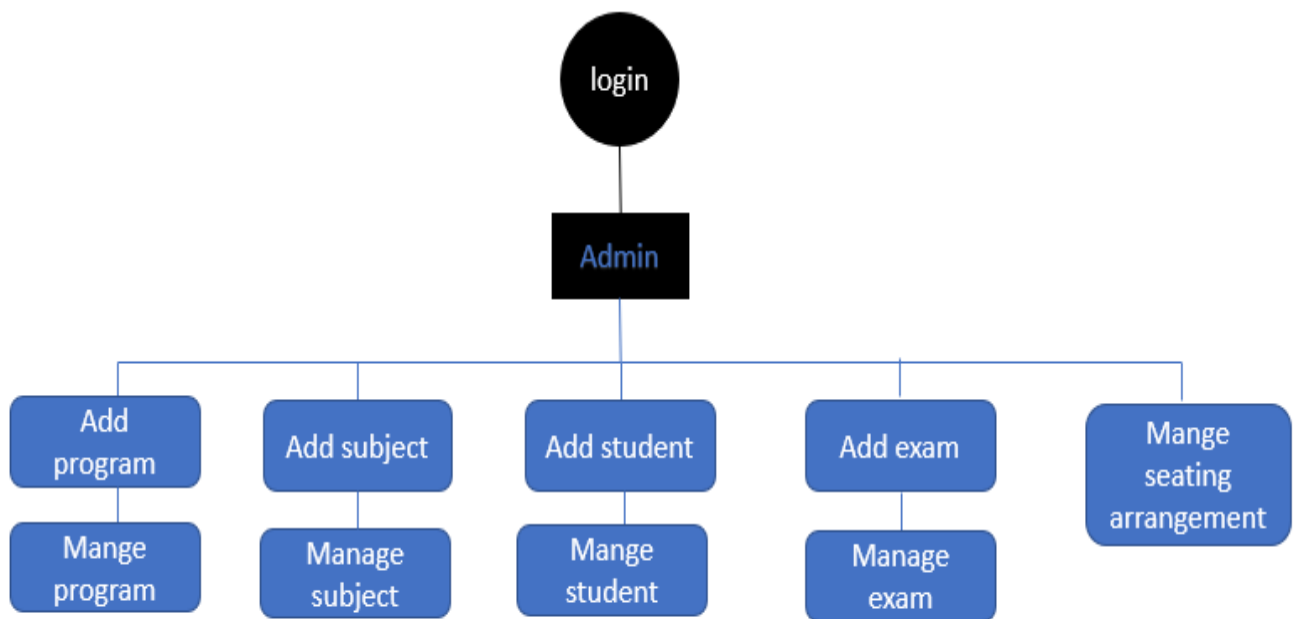


Fig.5.2.3.1 activity diagram

5.2.4 Data Flow Diagram:

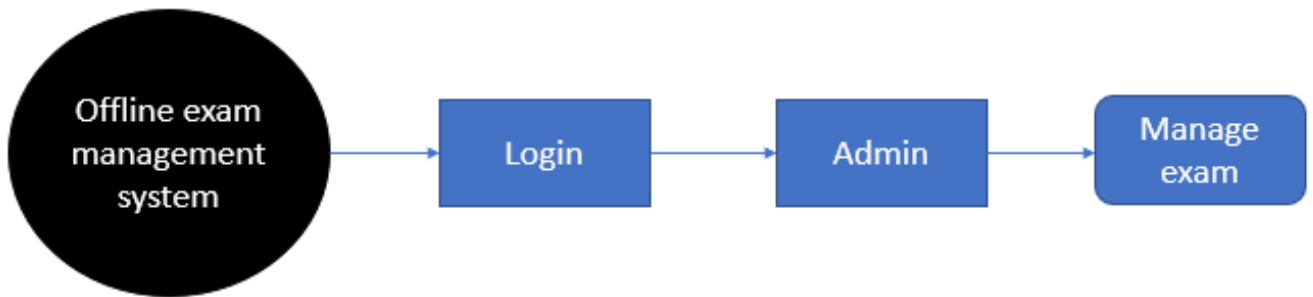


Fig.5.2.4.1 DFD zero level

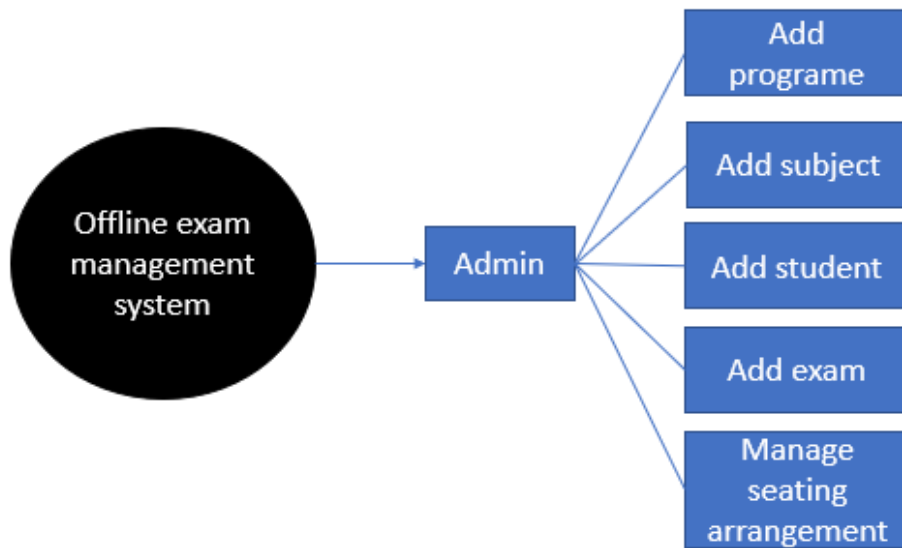


Fig.5.2.4.2 DFD first level

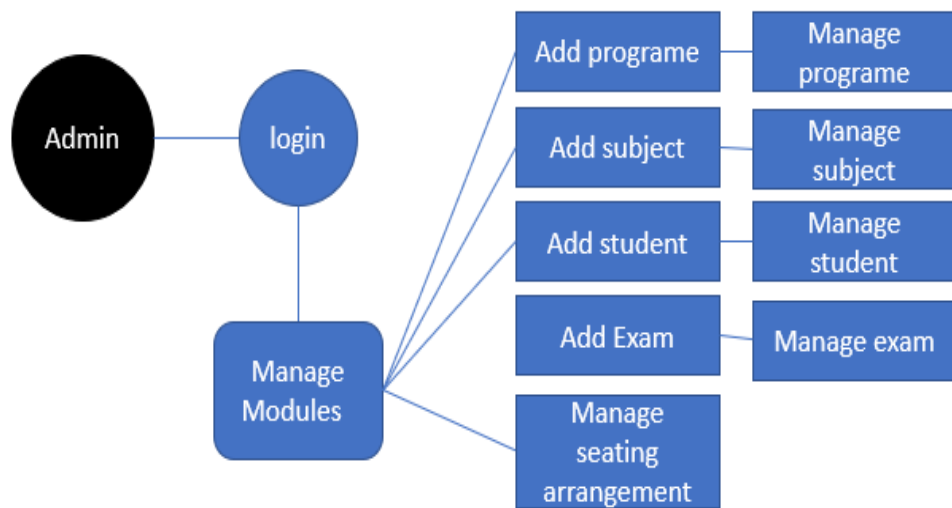


Fig.5.2.4.3 DFD two level

6.Code Implementation

6.1 Implementation Environment:

Challenges identified for successful design and implementation of this project are dominated by: complexity, reliability/availability, transparent data access while respecting security. The project was a result of a group consensus. The team was having two members. The team was guided by project manager. The team structure depends on the management style of the organization, the no. of people in the team, their skill levels and the problem difficulty. After the completion of each module, a module testing was performed on each. When the development was completed, System testing was performed to test the system.

6.2 Program/Module Specification:

- System GUI must be as simple and user friendly as anyone can use it. At front side we implemented registration form for access the system
- Authentication is necessary to enter into the system only if one requires to start his/her own project. This is required to prevent unauthorized access to the system
- If someone steals the password of the administrator or any regular user then he can able to change the database or misuse the system and can enter in restricted area so for this purpose system will provide encrypted password storage format in the database. Option to change the Password.
- A Session is maintained throughout the system when a particular user enters into the system. The Session is regularly checked whenever it is required.
- Proper validation is placed as and when it is required.

6.3 Coding Standards:

Normally, good software development organization requires their programmers to adhere to some well-defined and standard style of coding called coding standard.

6.3.1 Variable Standards:

We have used meaningful variables name.

6.3.2 Comment Standards:

- The comment should describe what is happening, how it is being done, what parameters mean, which global are used and which are modified, and any registration or bugs. The standards I have followed are:
- Every script should begin with a comment block, which describes the scripts purpose; any argument used (if applicable), and return values (if applicable), inputs-outputs, and name of script.
- Comment may also be used in the body of the script to explain individual sections or lines of codes.
- It is also used to describe variable definition or declarations.

7. Testing

7.1 Testing Strategy:

A strategy for software testing integrates software test case design method into a well-planned series of steps that result in the successful construction of the software.

The strategy provides the roadmap that describes the steps to be conducted as a part of testing, then these steps are planned and then undertaken, and how much effort, time and resource will be required.

- We have tested our whole system using bottom up testing strategy.
- Bottom up testing involves integrating and testing the modules to the lower levels in the hierarchy, and then working up hierarchy of modules until the final module is tested.
- Bottom up testing strategy shows how actual testing is to be done with whole system but it does not show any detail about each module testing.
- For each module testing we have decided to test each lower level module with white box testing strategy.
- When all modules are tested successfully then I will move to one step up and continue with white box testing strategy.
- When all modules will be tested successfully then I will integrate those modules and try to test integrated system using black box testing strategy.

7.2 Testing Method:

7.2.1 Unit Testing:

The unit testing is meant for testing smallest unit of software. There are two approaches namely bottom-up and top-down.

In bottom up approach the last module is tested and then moving towards the first module while top down approach reverses the action. In present work we opt for the first one.

The bottom up approach for the current project is carried out as shown in.

7.2.2 Integration Testing:

The integration testing is meant to test all the modules simultaneously because it is possible that all the modules may function correctly when tested individually. But they may not work altogether and may lead to unexpected outcome.

7.2.3 Validation Testing:

After the integration testing software is completely assembled as a package, interfacing error have been uncovered and corrected, and then validation testing may begin.

Validation can be defined in many ways but a simple definition is what a validation succeeds when software functions in a manner that can be reasonably accepted by the user.

8.Limitations and Future Enhancement

8.1 Limitations:

Though we tried our best in developing this system but as limitations are mere parts of any system so are of our system. Some limitations of — offline exam management system. Today portal is as under:

- Storage Capacity low
- User Cannot Effort

8.2 Future Enhancement:

There is always a scope for enhancements in any developed system, especially when our nature of the project is iterative waterfall which allows us to rethink on the method of development to adopt changes in the project. Below mentioned are some of the changes possible in the future to increase the adaptability, and efficiency of the system.

- Improve Storage Capacity
- In Future User also can Effort

9. Conclusion

The offline exam management system is easy to manage record of all exam. The Offline exam management system is developed using .NET, C#, MYSQL, HTML, CSS, JAVASCRIPT and was hosted locally with Apache web server. The ems is simple way to generate all exam detail.

10. References

web site:

<https://www.w3schools.com>

<https://stackoverflow.com/>

<https://phpgurukul.com/>

<https://getbootstrap.com/>