# **TABLE OF CONTENTS**

TITLE PAGE

[**TABLE OF CONTENTS**](#_vkiqk1kc1afs)

[**LIST OF FIGURES**](#_c0a4n5h1twyh)

[**CHAPTER 1: INTRODUCTION**](#_39vj0i19n7m8) **1**

[1.1 Background](#_s70t99xz7vkg) 1

[1.2 Problem Statement](#_go9vejqekz5a) 2

[1.3 Objectives](#_jzwynp1n3564) 2

[1.4 Project Features](#_4i15gmx7ob01) 3

[1.5 Feasibility Analysis](#_n8wztdnthqxj) 4

[1.5.1 Technical Feasibility](#_j5ofmlcacka9) 4

[1.5.2 Operational Feasibility](#_toizku1bdflg) 4

[1.5.3 Economical Feasibility](#_h2vi4h3frqeu) 5

[1.6 System Requirement](#_h8qnakyptpmn) 5

[1.6.1 Software Requirement](#_fwpfix9odyr9) 5

[1.6.2 Hardware Requirement](#_co17i7hhb93t) 5

[1.6.3 Functional Requirement](#_49bj95sfrnxc) 6

[1.6.4 Non-functional Requirement](#_4fbx0m5m0z2b) 6

[**CHAPTER 2: LITERATURE REVIEW**](#_2w28zt45l10r) **7**

[**CHAPTER 3: METHODOLOGY**](#_twdikgi7yakv) **9**

[3.1 System Development Life Cycle](#_uf98d7wxv5f4) 9

[3.2 Technology Overview](#_e9flpb921520) 10

[3.3 Approach](#_a96uxs8gu232) 11

[3.4 Flowchart](#_nko7sdtosse8) 12

[3.5 Use Case Diagram](#_q0b83vn8f0zs) 13

[3.6 Data Flow Diagram](#_yez9eyupb04m) 14

[3.6.1 DFD 0](#_blxtqj5ku8lc) 14

[3.6.2 DFD 1](#_hn8mrgfspb8a) 15

[3.7 Software Testing](#_rnlkb1m6tkjl) 16

[3.7.1 Unit Testing](#_me7q4kyp3e4) 16

[3.7.2 Integration Testing](#_4m5ugvoelsiz) 17

[3.7.3. System Testing](#_2suvhpvpdb3x) 17

[3.7.4 Acceptance Testing](#_wcy38x1mupr7) 17

[**CHAPTER 4: EPILOGUE**](#_k8727rm1e8wn) **18**

[4.1 Expected Output](#_hyip8bmha8ha) 18

[4.2 Budget Analysis](#_h5s6lgvwe2kz) 18

[4.3 Work Schedule](#_jg1r5c56q1d8) 19

[4.3.1 Requirement Gathering](#_b8belmeiu3we) 19

[4.3.2 Quick Design](#_hvsgpyezpeun) 19

[4.3.3 Building Prototype](#_dif2pxv6h9w7) 19

[4.3.4 Evaluation](#_qq1gy2xww99u) 19

[4.3.5 Refining Prototype](#_voedf5lse3k8) 19

[4.3.6 Implement and Final Product](#_afe1xadq3dj4) 20

[**REFERENCES**](#_5jm6ki29wppk) **21**

# **LIST OF FIGURES**

FIGURE PAGE

Prototype Model: ------------------------------------------------------------------------------------9

General Workflow: ---------------------------------------------------------------------------------12

Use Case Diagram: -------------------------------------------------------------------------------- 13

DFD 0: -----------------------------------------------------------------------------------------------14

DFD 1: -----------------------------------------------------------------------------------------------15

Software Testing Level: ---------------------------------------------------------------------------16

Gantt Chart: -----------------------------------------------------------------------------------------19

# **CHAPTER 1: INTRODUCTION**

## **1.1 Background**

Digitization has evolved in each sector with the help of advanced technology. That sector involved the education sector as well. The education sector has been rapidly adopting these technologies. And surprisingly, more advanced technologies have been solely developed for this sector. Among them is the Class Management System that every institute seems to be talking about.

The management system is a kind of system that provides services to organize, evaluate, and efficiently manage information within an organization or a company in a very digital and chronological way. In the modern era, the management systems are built by professionals and published as a particular website or software needed according to the client. An organization’s information management allows for quick, accurate, and reliable access to the records.

Records are crucial for day to day operations of the academic institutions. They aid in the decision-making process and evaluate the progress of the institution. Some kind of management is required to maintain the records. Controlling what is created, developing effective and efficient systems to preserve the records, and the processes for their preservation are all part of the management.

The main goal in developing the Class Management System is to make it simple to automate the academic functions. The system will function as a bridge between students and teachers. The system will manage tasks relating to students as well as save time spent searching for individual records. Its integrated structure, modules, and core databases help to reduce data redundancy and duplication of work, leading to efficient use of time, resources and cost.

## **1.2 Problem Statement**

The traditional method of keeping records is the most common. The procedure of collecting data is done using manual record-keeping like paper records and information has to be filled in again and again manually. These records are not easily searchable which may take a long time. It is quite difficult to keep track of a student's performance in this process. Some of the lecturers do not keep track of files, reports, assignments submitted by a student, and so on. The class is required to have at least one attendance sheet or carbon copy paper printed by the academic institution. This is not the issue of paper cost only but it's the price of printer ink and paper. If a teacher or student wishes to know about the exam details and results, they must wait a large amount of time using the traditional paper-based procedure. As a result, it's time-consuming and lots of paper gets wasted. Sometimes storing paper records may consume a lot of physical space. Many paper records lead to much time and effort needed to access any stored information. The old paper system may reduce the productivity of the staff. The staff gets tired handling lots of paper records and as a result, the staff’s productivity suffers.

## **1.3 Objectives**

The main aim is to design and perform the implementation of CMS for the institute that computerized all the details that are maintained manually.

Others objectives of this project are as:

1. To provide a user-friendly and effective system to clients
2. To retrieve students' and teachers' information
3. To show detailed reports of exam and attendance
4. To reduce unwanted waste of time and money
5. To improve and reduce the workload of staffs

## **1.4 Project Features**

Admin, teachers, and students are the three primary categories in which the project workflow is separated. Initially, there will be a limit on the number of administrators, but administrators can create teachers as sub-administrators, and students as users. Sub-admin will have some of the features of admin with additional extra features.

The features provided to the administrator are: -

1. Teacher and Student Registration
2. Manage and Control Account
3. Make announcements and notices
4. Manage teachers and students information with ease
5. Publish student's results and have a record of them
6. Schedule class routine

The features provided to teachers are:-

1. Take Students attendance
2. Give Assignment to students
3. Give a digital copy of lecture notes to students
4. View schedules class routine
5. Provide assessments marks to students
6. Announce notices

The features provided to students are:-

1. View own attendance
2. Submit Assignment to teachers
3. View and download a digital copy of lecture notes to students
4. View schedules class routine
5. View Exam Results

## 

## **1.5 Feasibility Analysis**

After studying and analyzing the required functionalities of the systems, the next task is to do a feasibility study for the project. It is said that “All Projects are feasible given unlimited resources and times”. However, both resources and times are limited in reality. The project should adhere to the time and make efficient use of the available resources. The proposed solution should satisfy all the user requirements and should be flexible enough to allow for future changes based on new requirements.

The following areas are covered by the feasibility study: -

1. Technical Feasibility
2. Operational Feasibility
3. Economical Feasibility

### **1.5.1 Technical Feasibility**

To establish whether the proposed system is technically feasible, we must consider the technical issues that affect it. Class Management System will use web technologies, which are widely used these days all over the world. With the help of available technologies, the proposed system can be built effectively. Today's world would be unimaginable without web technologies and the internet. That proves that the system under consideration is technically feasible.

### **1.5.2 Operational Feasibility**

While determining the operational feasibility, we must consider the awareness level of users. No doubt, the system will have a user-friendly user interface. This system is operationally practicable but training should be provided to ensure that everyone understands the system's purpose.

### 

### **1.5.3 Economical Feasibility**

This is a very important aspect to be considered. To decide whether a project is economically feasible, we have to consider various factors such as -

1. Cost-Benefit Analysis
2. Long Term returns
3. Maintenance costs

The proposed system is web-based so it requires access to the internet and average computing capabilities. This is easily affordable to the college so it does not have any additional financial costs, making the system economically feasible.

## **1.6 System Requirement**

System Requirements are essential for the system to work efficiently.

### **1.6.1 Software Requirement**

Software Requirements of the projects are as follows:

1. Operating System: Windows 7 or higher, Linux, or Mobile Phones
2. Browser: Any web browser like chrome, firefox, etc,

### **1.6.2 Hardware Requirement**

Hardware Requirements of the projects are: -

1. Processor: Intel Pentium IV processor or equivalent or higher
2. Memory: Minimum 1GB of RAM to handle browser
3. Internet Access

### **1.6.3 Functional Requirement**

The initial functional requirements for our projects are

1. Manage User Account
2. Announcement and Notice Management
3. Teacher and Student Registration
4. Attendance Management
5. Reports Management
6. Provide Lecture Notes
7. Assignment Management

### **1.6.4 Non-functional Requirement**

Security requirements are critical factors in this system as data will be stored in the database. The authentication of the user should be done to validate user logins. In this system, user authentication will be done by login with email and password where passwords are encrypted and stored in the databases. Users will get access to the system as permissions according to their roles specified.

The system has a consistent user interface and easy tracking of all the records so that the system is easy to use and in the interface of our system, buttons and forms are used to enter data related to a specific module.

# **CHAPTER 2: LITERATURE REVIEW**

The use of technology to update the academic facilities has a significant impact on every responsible stakeholders’ achievements of a particular institution. Therefore, the most important aspect of developing and implementing efficient information systems for any academic institution is to start with the genuine needs of administration, teachers, and students (Breiter and Light, 2006).

A study of (Durnali, 2013) when comparing the data collection, processing, storage, accuracy and analysis, and dissemination of student data before and after an e-class management was implemented in their academic institutions, it has been shown through his studies that there are improvements in terms of data collection, processing, storage, accuracy, and analysis and dissemination of student data. If technology is employed for class management and development, it has an impact on how society reflects the socioeconomic, cultural, and technical change (Yıkıcı et al., 2019).

Research on creative information management in Taiwan illustrated the value of the class management method. Innovative information management makes students more inspired, enhances their learning effectively, and increases the sense of classes and academic institutions being defined (Chen et al., 2014).

A study in Turkey shows how teachers and the principal believe that the e-class management system is adequate in terms of administrative relations, student affairs, and student report card work time (Polat & Arabaci, 2013). A developed web-based information system for class management provides leverage for academic institutions that need the application to facilitate learning, teaching, and administration quality and effectiveness. It is always necessary to have a modern class management and information system(Pavlović, Ranðić & Paunović, 2014).

It is necessary to keep academic records and manage them properly because it is an important aspect of the institution’s leverage in terms of keeping things in their rightful place to ensure quality processing and record-keeping. This will also help institution managers in their decision-making process and to also enhance the implementation of usable records in academic institutions that will lead to cost savings, transparency, easy accessibility, accountability, and retrieval of required information from their storage (Akinloye, Adu, & Ojo, 2017). The key areas that need to be present in an e-class information and management system are open standards, interoperability, transition, accessibility, cost efficiency, statutory-based innovations, and usability (Strickley, 2011).

Information systems are created to provide solutions and feedback to encourage the effectiveness of learning, teaching, and administrative purposes. There are various information conveying systems and class management systems that are used by different educational institutions like PowerSchool SIS, Skyward Student Management Suite, and so on. Each system has its features, advantages, and disadvantages. These systems are considered for providing the necessary information, announcements of the particular institutions to their respective stakeholders but are failing to ensure the reach and engagement of the conveyed information and other managerial tasks for the institution.

Our Class Management System will be a better solution for providing effective communication and engagement among administration, teachers, and students, track academic progress, efficient staff and student management, organized learning materials, and proper record-keeping process.

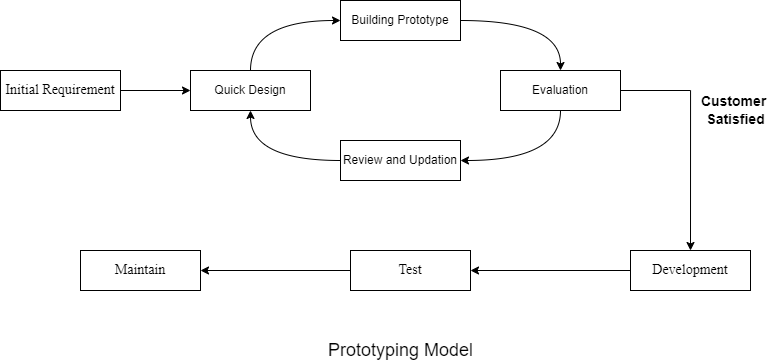
# **CHAPTER 3: METHODOLOGY**

## **3.1 System Development Life Cycle**

The project does not have fixed user requirements, so features of the project can be changed during development. So, we are planning to use the Prototype model for project development since the requirements are not clear. One of the main advantages of prototype models is errors can be detected in the initial stage of the software development process.

The reasons for selecting the prototype model are:-

1. Users are actively involved in development.
2. Missing functionalities can be identified, which helps to reduce the risk of failure.
3. Errors can be detected much earlier, saving lots of time and effort.
4. New Requirements can be easily achieved as there is a scope for improvement.



*Fig 3.1 Prototype Model*

## **3.2 Technology Overview**

The project will use the following technologies and languages to make the project successful.

1. **React:** React is a free and open-sourced front-end Javascript library created by Meta for building user interfaces. Building user interfaces is based on components so it makes building complex designs easier. More importantly, React enables us to develop reusable user interface components.
2. **CSS:** CSS is the language we use to style the pages. It makes web pages more attractive. CSS describes how elements and components are to be displayed on the screen. It can control the layout of multiple web pages all at once
3. **Git:** Git is a popular version control system that is easy and free to use. It helps to keep track of changes and improvements. Using Git will free us from the trouble of losing files. It is designed to handle everything from small to very large projects with speed and efficiency.
4. **Django:** Backend programming will be done with Django. It's an open-source, free-to-use Python framework. It is chosen because of its widespread popularity and ease of scalability. Various significant companies use it effectively in their projects. It follows the model-views-template (MVT) architectural pattern.

## 

## **3.3 Approach**

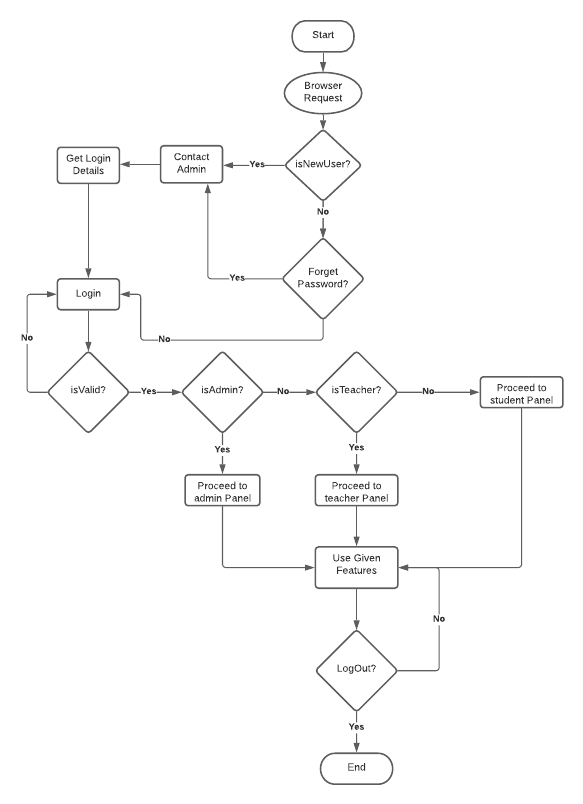
We discuss how to improve the current methods of class management. We plan to split the roles of users so that they can make use of the features that have been assigned to them. As a result, users are separated into three groups: administrators, teachers, and students.

By default, there will be a limit on the total number of admins, but the admin can adjust it if necessary. Teachers and students will not be restricted in any way. To access their dashboards, all users must first log in to the website. Admins will be able to easily manage the accounts of other users, as well as the details of teachers and students. When necessary, the administrator can make announcements or notices, which teachers and students will be able to see. The class routine can be scheduled by the administrator. The final exam results can be published via the dashboard by the administrator. Admin has access to the information of each student's performance as well.

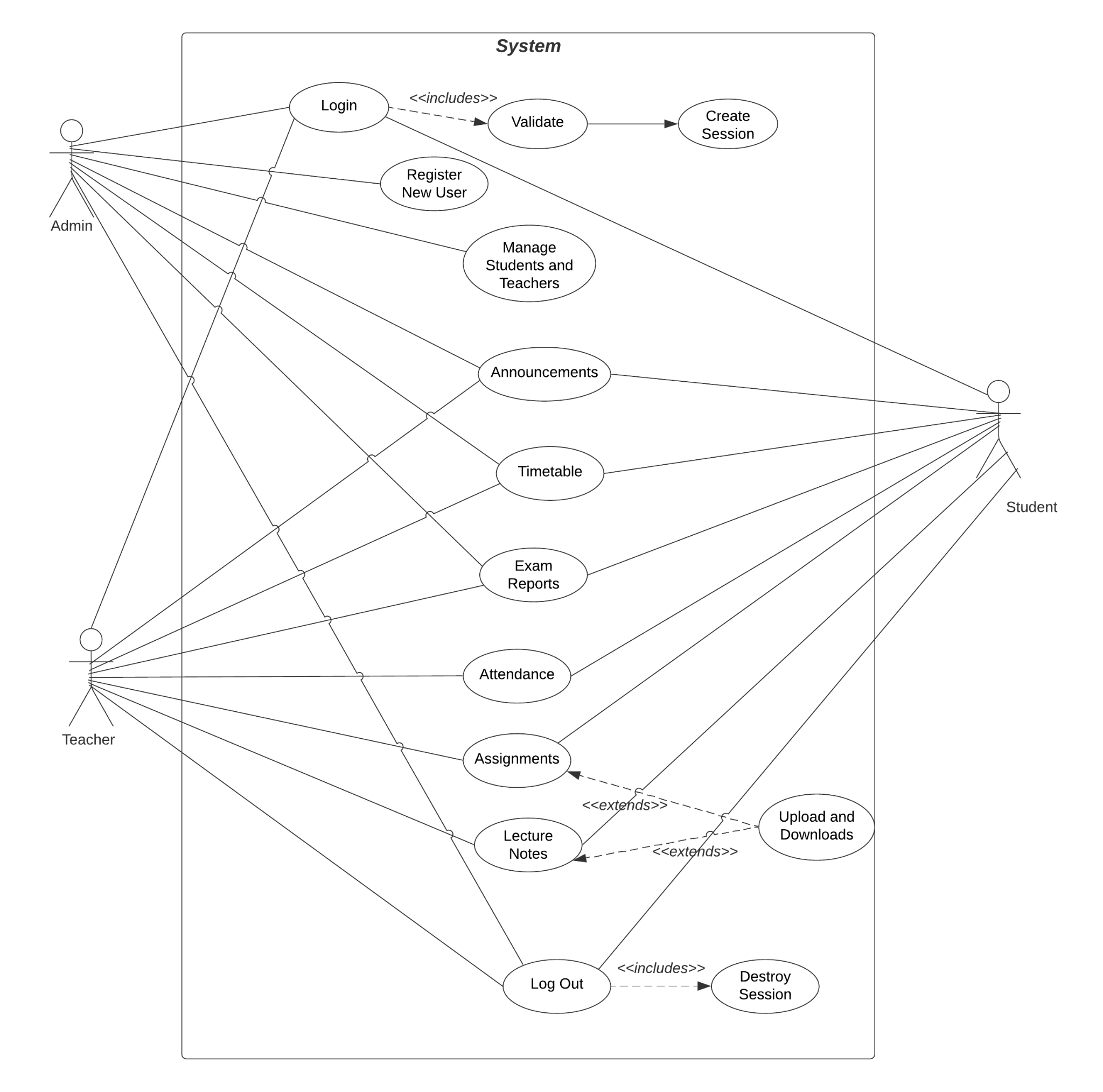
Teachers can read the admin's announcements or notices, and they can also announce their notices to the students. Teachers can assign assignments to students and announce them. Students can access a digital copy of the teacher's lecture notes. They can take the attendance of students digitally.

Students can use their dashboard to view details about their performance, such as attendance reports, exam reports, and so on. They can read the announcements made by the administration and teachers. They can see and download the lecture notes that their faculty teachers have provided. They can also see the planned lesson schedule.

## **3.4 Flowchart**

  
*Fig 3.2: General Workflow of the System*

## **3.5 Use Case Diagram**



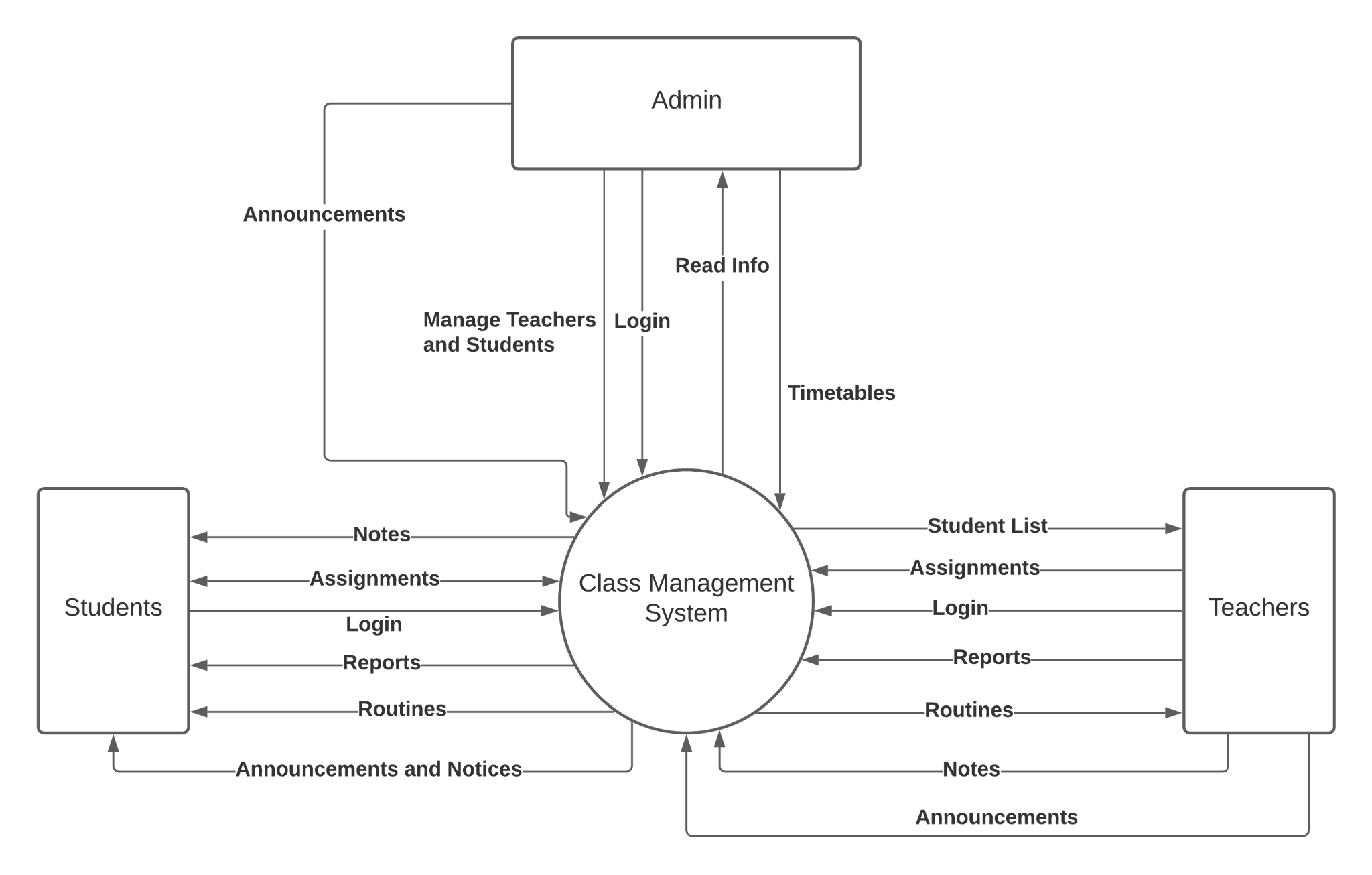
*Fig 3.3: Use Case Diagram of the System*

## 

## 

## **3.6 Data Flow Diagram**

### **3.6.1 DFD 0**



*Fig 3.4: DFD 0*

### **3.6.2 DFD 1**

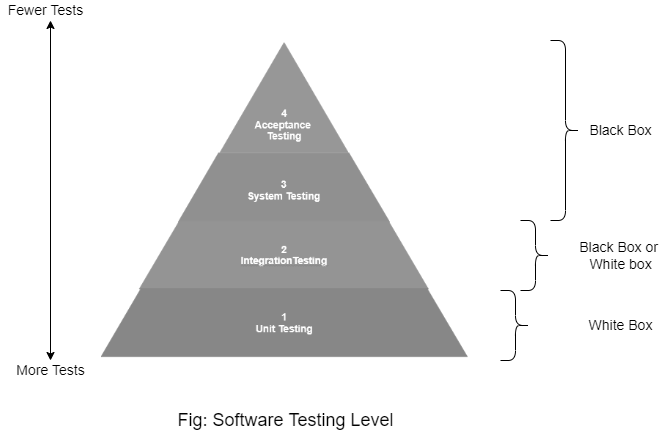
### 

### 

*Fig 3.5: DFD 1*

## **3.7 Software Testing**

The goal of testing is to find mistakes. Testing is the practice of attempting to find all possible flaws or weaknesses in a work product. It allows us to test the functionality of individual components, modules, and/or a whole product. It is the process of testing software to ensure that it meets its requirements and meets user expectations and that it does not fail in an unacceptable way. There are many different types of tests. Each test type addresses a specific testing requirement. Various testing would be performed to find mistakes and flaws.



*Fig 3.6: Software Testing Level**.*

### **3.7.1 Unit Testing**

Different components will be examined independently to see if they are fully functional. We would be the ones to conduct the testing. Unit testing will be performed when we include new functionalities.

### **3.7.2 Integration Testing**

After unit testing is completed, then we will combine the functionalities to perform integration testing. This will ensure that different system units work like a charm or not when they are integrated and tested as a group. Some members would be the ones to conduct the testing as testers.

### **3.7.3. System Testing**

Following the completion of integration testing, we will conduct system testing to ensure that the system complies with the requirements. It examines the overall interaction of the various components. Performance, reliability, functionality, and security testing are all part of this process. This testing will assist us in determining both functional and non-functional requirements. With the concept of white-box testing and black-box testing, system testing can be accomplished.

### **3.7.4 Acceptance Testing**

After all testing is accomplished, acceptance testing will be performed to find if the requirements of a specification are met as per user requirements. This will be done by the user (client), however, other stockholders can be involved in this process.

# **CHAPTER 4: EPILOGUE**

## **4.1 Expected Output**

When the project is completed, it will be useful to academic institutes that still use the old traditional method of class management. Students can view the details of their performance in order to gain insight into how they can improve. In comparison to prior methods, teachers can easily manage all of the information. It will replace the previous manual procedures, saving clients a significant amount of time. All the mentioned features will be live in the final product with additional features if needed.

## **4.2 Budget Analysis**

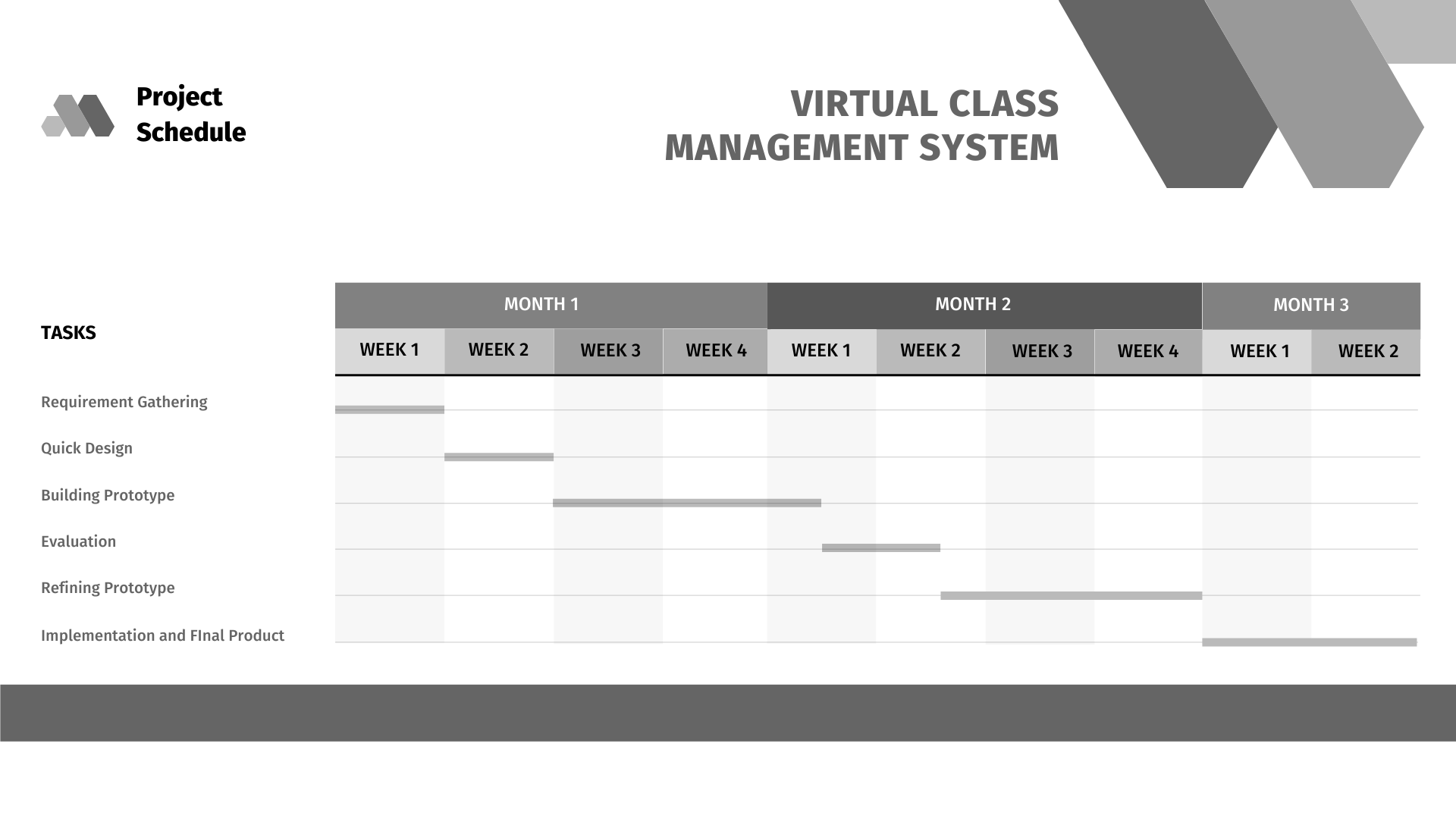
We have discussed the budget analysis as follows:

| S.N. | Name | Cost (Rs.) |
| --- | --- | --- |
| 1. | Internet and Electricity | 1500 per month |
| 2. | Domain registration and Hosting | 3000 per month |
| 3 | Stationary and Miscellaneous | 2000 per month |

## 

## **4.3 Work Schedule**

The full project is expected to take 12 weeks to complete. The schedule is broken down into six sections: requirement gathering, quick design, prototype building, evaluation, prototype refining, and final product.

  
*Fig 4.1: Gantt Chart for Project Scheduling*

### **4.3.1 Requirement Gathering**

In this phase, the requirements of the system are defined in detail. During the process, the expectations from the system are researched and analyzed.

### **4.3.2 Quick Design**

A simple system design is created at this stage. It is, however, not a complete design. This will give a certain idea of the system being built. It could take up to one to two weeks.

### **4.3.3 Building Prototype**

An actual prototype will be designed based on the information gathered from quick design. It will be a small working model of the required system.

### **4.3.4 Evaluation**

The proposed system will be presented to friends or clients for an initial evaluation. It helps to find out the strength and weaknesses of the working model. Comments and suggestions will be collected from the friends.

### **4.3.5 Refining Prototype**

The prototype will be refined according to the feedback and suggestions gathered.

### **4.3.6 Implement and Final Product**

The final system will be developed based on the final prototype with thorough testing and deployment.

# **REFERENCES**

Akinloye, G. M., Adu, E. O., & Ojo, O. A. (2017). Record-keeping management practices and legal issues in the school system. The Anthropologist, 28(3), 197-207. doi:

10.1080/09720073.2017.1335832

Breiter, A., & Light, D. (2006). Data for school improvement: Factors for designing

effective information systems to support decision-making in schools. Journal of

Educational Technology & Society, 9(3), 206-217.

Chen, P. C., Lan, T. S., Lan, Y. H., & Hsu, H. Y. (2014). Dynamic effect of knowledge

management system on school management. Journal of Theoretical and Applied

Information Technology, 61(2), 249-253.

Durnalı, M. (2013). The contributions of e-school, a student information management

system, to the data processes, environment, education, and economy of Turkey.

Proceedings of the Asian Conference on Technology in the Classroom, (pp. 170-184).

Retrieved from <http://papers.iafor.org/wp-content/uploads/papers>

/actc2013/ACTC2013\_0233.pdf

Pavlović, M. I. J., Ranðić, S., & Paunović, L. (2014). Information technologies in the

contemporary school management system. Emerald Emerging Markets Case

Studies, 4(6). https://doi.org/10.1108/EEMCS-11-2012-0195

Polat, M., & Arabaci, I. B. (2013). Evaluation of E-school applications as a management

information system. Elementary Education Online, 12(2), 320–333.

<https://doi.org/10.17051/io.49201>

Strickley, A. (2011). A baseline for a school management information system. In A. Tatnall, O. C. Kereteletswe, & A. Visscher (Eds.), Information Technology and Managing

Quality Education. Paper presented at the IFIP Advances in Information and Communication Technology (ITEM 2010), vol. 348, Botswana (pp. 62-74). Berlin: Springer. DOI: 10.1007/978-3-642-19715-4\_7

Yıkıcı, B., Bastas, M., Altınay, F., Dagli, G., & Altınay, Z. (2019). The role of technology for school management and development. Revista Inclusiones, 6, 100-115.

# 