

INTRODUCTION

Student Automation from Admission to Alumni (SAA) is designed to help Educational institutions to maintain the information of a student in automated form. Extensive information is available at your fingertips through this System. Viewing student data, managing admission and reshuffling, semester, faculty, category and for examination, block allocation, subject management, scheduling exam, result and related issues are made simple and easy. There are custom search capabilities to aid in finding student information and working on student records. This can make the system easier to navigate and to use maximizing the effectiveness of time and other resources. SAA allows the keeping of personnel data in a form that can be easily accessed and analyzed in a consistent way.

The SAA module is a component covering many other student aspects from admission to alumni. The system records basic personal information, admission information, education information regarding student. Leading edge systems provide the ability to "read" applications and enter relevant data to applicable database fields, notify student and provide result. Student management function involves

- Manage new admission and enrolment
- Fetch the predefined data of student
- Handling academic details
- Providing statistical reports based on performance
- student Basic Information
- Fee category of the student
- Manage semester and year
- admission seat management
- Exams scheduling
- Result management
- Subject management
- Block management

In SAA, every user has a Login ID and Password. Also all the users have different permission rights to access the applications. These rights are Dynamic and can be changed.

There are three main roles in the system. Admin, Faculty and Student. Admin has complete access to the whole system, while accountant is only concerned with payment of fees for the admission of the student. Operator is the role that is responsible for the use of the system.

The Admin role can be as follows:

- Set fees structures
- Manage faculties
- Manage subjects
- Management of semester
- Set examination
- Allocating subjects
- Generating timetables

The Faculty role is:

- To set the Examinations
- Viewing the student information
- Maintaining the marks of the Individual
- Viewing and updating the Timetable

The student can:

- View his/her basic information
- Timetable
- Examination Schedule
- Faculty allotment

Now when the user with the particular role Logs on he can see only those pages which are allowed to them.

1.1 PURPOSE

The project is about to handle all the information of the student regarding admission and examination. Also it manages resources which were managed and handled by manpower previously. The main purpose of the project is to integrate distinct sections of the organization into consistent manners so that complex functions can be handled smoothly by any technical or non-technical persons.

The project aims at the following matters:

- Automation of admission and enrollment.
- Assistance in decision-making.
- To manage information of student, faculty and courses.
- Consistently update information of all the students.
- Reports- To gather all the related information about any student.

All the above-mentioned matters are to be incorporated in the application along with some additional requirements.

The main purpose of the Admin Module is to introduce new things and configure important aspects. For e.g. only admin is authorized to introduce subject allotment, student category and only admin is allowed to configure exam and set fees structure. So the master screens for all these are visible to only admin role. This is done by the Admin Module. It also creates the users and Physical and Logical Locations. Thus the main purpose of the Admin Module is to managing the dynamic working of the system

1.2. EXISTING SYSTEM

The Present Existing System was using the registers and if the management wants to search the student details they should open many files to get the information. They should maintain separate file for each student. It is very confusion able and time cost. They have to update every student files. It is very difficult to maintain each student detail for each lecturer.

1.3.DRAWBACKSOFEXISTINGSYSTEM

- It is very difficult to maintain each record for every student in big universities.
- The full details of the student like marks, address, phone number, attendance was very difficult to maintain in separate records.
- The present Existing System is very expensive.
- The records of the passed out students was very difficult to maintain.

1.4. PROPOSED SYSTEM

The system is maintained for collection of student data and ensuring the accuracy of such data to serve as a database for the following purposes:

- Enrolling the student registration number
- Easy to access student details
- Compilation of student related statistics
- Monitoring the student education grants

1.5.SCOPE

The scope of the project includes the following

- Any college can use this system as it is not client centric.
- All admission and examination related work for the student can be done using this system.
- Application Support & Maintenance after deployment to production
- The Admin Module can be reused for projects as well which have many users with different rights. Hence it is reusable.

SYSTEMANALYSIS

2.SYSTEMANALYSIS

2.1.INTRODUCTION

During analysis, the focus is on what needs to be done intendment of how it is done. During design, decisions are made about how the problem will be solved, first at a high level, then at increasingly detailed levels.

System design is the first stage in which the basic approach to solving the problem is selected. During system designing the overall structure and style are decided. The system architecture is the overall organization of the system into components called system. System design deals with transforming the customer requirements, as described in the SRS document, into a form that is implementable using the programming language. Certain items such as modules, relationships among identified modules, data structures, relationships between the data structures, and algorithms for implementation should be designed during this phase.

As a system designer we are tried to take following design decisions:

- Organize the system into modules
- Organize sub-modules for each module
- Allocate tasks to processors
- Choose an approach to manage data store
- Handle access to global resources
- Choose implementation logic

2.2 Feasibility Study

Feasibility study is the test of a system proposal according to its workability, impact on the organization, ability to meet user needs, and effective use of resources. It focuses on the evaluation of existing system and procedures analysis of alternative candidate system cost estimates. Feasibility analysis was done to determine whether the system would be feasible.

The development of a computer based system or a product is more likely plagued by resources and delivery dates. Feasibility study helps the analyst to decide whether or not to proceed,

amend, postpone or cancel the project, particularly important when the project is large, complex and costly.

Once the analysis of the user requirement is complement, the system has to check for the compatibility and feasibility of the software package that is aimed at. An important outcome of the preliminary investigation is the determination that the system requested is feasible.

2.2.1 Technical Feasibility:

The technology used can be developed with the current equipments and has the technical capacity to hold the data required by the new system.

- This technology supports the modern trends of technology.
- Easily accessible, more secure technologies.

Technical feasibility on the existing system and to what extend it can support the proposed addition. We can add new modules easily without affecting the Core Program. Most of parts are running in the server using the concept of stored procedures.

2.2.2 Operational Feasibility:

This proposed system can easily implemented, as this is based on PHP coding (Hyper Text Pre Processor) & HTML .The database created is with Myself server which is more secure and easy to handle. The resources that are required to implement/install these are available. The personal of the organization already has enough exposure to computers. So the project is operationally feasible

2.2.3 Economical Feasibility:

Economic analysis is the most frequently used method for evaluating the effectiveness of a new system. More commonly known cost/benefit analysis, the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs. If benefits outweigh costs, then the decision is made to design and implement the system. An entrepreneur must accurately weigh the cost versus benefits before taking an action. This system is more economically feasible which assess the brain capacity with quick & online test. So it is economically a good project.

System design is the first stage in which the basic approach to solving the problem is selected. During system designing the overall structure and style are decided. The system architecture is the overall organization of the system into components called system. System design deals with transforming the customer requirements, as described in the SRS document, into a form that is implementable using the programming

2.3 System Requirement Specification

- Processor : Pentium IV with 2.4 GHZ & above
- Ram : 2 GB & above
- Disk Space : 100GB & above
- Web Browser : Internet Explorer 6.0 & above, Mozilla Firefox and Chrome
- **JavaScript:** - JavaScript can generate HTML dynamically on the client. This is a useful capability, but only handles situations where the dynamic information is based on the client's environment. With the exception of cookies, HTTP and form submission data is not available to JavaScript. And, since it runs on the client, JavaScript can't access server-side resources like databases, catalogs, pricing information, and the like.

Design

3.DESIGN

3.1 Introduction

The website is designed in such a way that a professional look and feel is appeared. The designing is done using HTML and CSS for the look. The project is mainly designed based on many diagrams. All the diagrams that are developed give us a way to design the functionalities of the project and the flow that is to be maintained till the end of the project. The diagrams used in the development of the project are UML diagrams, Class diagrams, Sequence diagrams, Activity diagrams, ER diagrams.

3.2. Design Objectives

Design is the first step in moving from problem domain to the solution domain. Design is essentially the bridge between requirements specification and the final solution. The goal of design process is to produce a model or representation of a system. It is the most creative and challenging phase of the life cycle. The term design describes a final system and the process by which it is developed. It refers to the technical specifications that will be applied in implementations the candidate system. The design may be defined as “the process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient details to permit its physical realization”.

The designer's goal is how the output is to be produced and in what format samples of the output and input are also presented, second input data and database files have to be designed to meet the requirements of the proposed output. . Finally, details related to justification of the system and an estimate of the impact of the candidate system on the user and the organization are documented and evaluated by management as a step toward implementation.

The importance of software design can be started in a single word “quality”. Design provides us with representations of software that can be assessed for quality .Design is the only way that we can accurately translate a customer's requirements into a finished software product or system without design we risk building an unstable system, that might fail if small changes are made or may be difficult to test, or one whose quality can't be tested. So it is an essential phase in the development of a software product.

3.3.UnifiedModelingLanguage

AnOverviewOfUML

UML is a method for describing the system architecture in detail using the blueprint. UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is very important parts of developing object oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects. Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software.

Definition

The UML is a general purpose visual modeling language for

- Visualizing
- Specifying
- Constructing
- Documentation

These are the artifacts of a software-intensive system.

UML is a language

It will provide vocabulary and rules for communications and function on conceptual and physical representation. So it is modeling language.

UML Specifying

Specifying means building models that are precise, unambiguous and complete. In particular, the UML address the specification of all the important analysis, design and implementation decisions that must be made in developing and displaying a software intensive system.

UML Visualization

The UML includes both graphical and textual representation. It makes easy to visualize the system and for better understanding.

UML Constructing

UML models can be directly connected to a variety of programming languages and it is sufficiently expressive and free from any ambiguity to permit the direct execution of models.

UML Documenting

UML provides variety of documents in addition raw executable codes.

A Conceptual Model of UML

The three major elements of UML are

- The UML's basic building blocks.
- The rules that dictate how those building blocks may be put together.
- Some common mechanisms that apply throughout the UML.

Goal of UML

The primary goals in the design of the UML are

- Provide users with a ready-to-use, expressive visual modelling language so they can develop and exchange meaningful models.
- Provide extensibility and specialization mechanisms to extend the core concepts.
- Be independent of particular programming languages and development processes.
- Provide a formal basis for understanding the modeling language.
- Encourage the growth of the OO tools market.
- Support higher-level development concepts such as collaborations, frameworks, patterns and components.
- Integrate best practices.

Rules of UML

The UML has semantic rules for

- NAMES: It will call things, relationships and diagrams.

- **SCOPE:** The content that gives specific meaning to a name.
- **VISIBILITY:** How those names can be seen and used by others.
- **INTEGRITY:** How things properly and consistently relate to another.
- **EXECUTION:** What it means is to run or simulate a dynamic model.

Basic Building blocks of UML

The vocabulary of the UML encompasses 3 kinds of building blocks

- Things
- Relationships
- Diagrams

Unified Modeling Language Diagrams

The unified modelling language allows the software engineer to express an analysis model using the modelling notation that is governed by a set of syntactic semantic and pragmatic rules. A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.

User Model View

This view represents the system from the user's perspective. The analysis representation describes a usage scenario from the end-users perspective.

Structural Model View

In this model the data and functionality are arrived from inside the system. This model view models the static structures.

Behavioral Model View

It represents the dynamic of behavioral as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

Implementation Model View

In this the structural and behavioral as parts of the system are represented as they are to be built.

Environmental Model View

In this the structural and behavioral aspects of the environment in which the system is to be implemented are represented.

UML is specifically constructed through two different domains they are:

- UML Analysis modelling, this focuses on the user model and structural model views of the system.
- UML design modelling, which focuses on the behavioral modelling, implementation modelling and environmental model views.
- Use case Diagrams represent the functionality of the system from a user's point of view. Use cases are used during requirements elicitation and analysis to represent the functionality of the system. Use cases focus on the behavior of the system from external point of view.

Actors are external entities that interact with the system. Examples of actors include users like administrator, bank customer etc., or another system like central database. Diagrams group the interesting collection of things.

Things in UML

There are four kinds of things in the UML

- Structural things
- Behavioral things
- Grouping things
- Annotational things

These things are the basic object oriented building blocks of the UML. They are used to write well-formed models.

Structural Things

Structural things are the nouns of the UML models. These are mostly static parts of the model, representing elements that are either conceptual or physical. In all, there are seven kinds of structural things:

Class

A class is a description of a set of objects that share the same attributes, operations, relationships, and semantics. A class implements one or more interfaces.

Graphically a class is rendered as a rectangle, usually including its name, attributes and operations, as shown below.



Figure 3.3.1 class

Interface

An interface is a collection of operation that specifies a service of a class or component. An interface describes the externally visible behavior of that element.

Graphically, the interface is rendered as a circle together with its name.

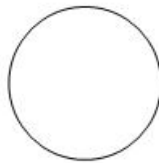


Figure 3.3.2. An Interface

Usecase

Use case is a description of a set of sequence of actions that a system performs that yields an observable result of value to a particular things in a model.

Graphically, use case is rendered as an ellipse with dashed lines, usually including only its name as shown below.



Figure 3.3.3 A use case

Component

Component is a physical and replaceable part of a system that conforms to and provides the visualization of a set of interfaces.

Graphically, a component is rendered as a rectangle with tabs, usually including only its name, as shown below.

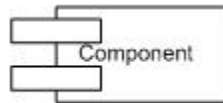


Figure 3.3.4 component

Node

A node is a physical element that exists at run time and represents a computational resource, generally having at least some memory and often, processing capability.

Graphically, a node is rendered as a cube, usually including only its name, as shown below.

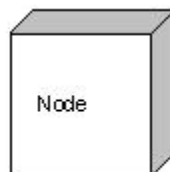


Figure 3.3.5 a node

Behavioral Things

Behavioral things are the dynamic parts of UML models. These are the verbs of a model, representing behavior over time and space.

Interaction

An interaction is a behavior that comprises a set of messages exchanged among set of objects within a particular context to accomplish a specific purpose.

Graphically, a message is rendered as a direct line, almost always including the name of its operation, as shown below.



Figure 3.3.6 an Interface

State Machine

A state machine is a behavior that specifies the sequence of states of object or an interaction goes through during its lifetime on response to events, together with its responses to those events. Graphically, a state is rendered as a rounded rectangle usually including its name and its sub-states, if any, as shown below.



Figure 3.3.7 state

Grouping Things

Grouping things are the organizational parts of the UML models. These are the boxes into which a model can be decomposed.

Package

A package is a general-purpose mechanism for organizing elements into groups.



Figure 3.3.8 package

Annotational Things

Annotational things are the explanatory parts of UML models. It doesn't correspond to any object but serves as a component

Note

A note is simply a symbol for rendering constraints and comments attached to an element or a collection of elements.

Graphically, a note is rendered with a dog-eared corner, with a textual or graphical comment.

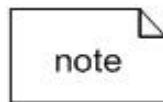


Figure 3.3.9 Note

3.4. Relationships In The Uml

There are three kinds of relationships in the UML:

1. Dependency
2. Association

3.Generalization

4.Realization

3.4.1 Class Diagram for SAA

The class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing and documenting different aspects of a system but also for constructing executable code of the software application.

The class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object oriented systems because they are the only UML diagrams which can be mapped directly with object oriented languages.

The class diagram shows a collection of classes, interfaces, associations, collaborations and constraints. It is also known as a structural diagram.

Purpose

The purpose of the class diagram is to model the static view of an application. The class diagrams are the only diagrams which can be directly mapped with object oriented languages and thus widely used at the time of construction.

The UML diagrams like activity diagram, sequence diagram can only give the sequence flow of the application but class diagram is a bit different. So it is the most popular UML diagram in the coder community.

So the purpose of the class diagram can be summarized as:

- Analysis and design of the static view of an application.
- Describes responsibilities of a system.
- Base for component and deployment diagrams.

- Forwardandreverseengineering

Contents

Classdiagramcommonlycontainthefollowingthings:

- Classes
- Interfaces
- Collaboration
- Dependency,generalizations,andassociationrelationships

The following diagram represents the classes used in SAA and their relationships. Each entity is identified as a class .the classes are

1. Student
2. Faculty
3. Admin
4. Exam
5. Timetable
6. Admission

The relationship represents the flow of control and the association among the different classes. The dependency shows that the exam, timetable, student details are under the control of admin. The Association shows that the admission, student, faculty are associated to admin i.e. they are controlled by admin. He has all the access permissions on these modules. The dependency relationship is among

1. Admin and login
2. Admin and exam
3. Admin and timetable
4. Student and login

5. Faculty and login
6. Faculty and exam

The Association is among

1. Admin and faculty
2. Admin and student
3. Faculty and student
4. Admin and admission

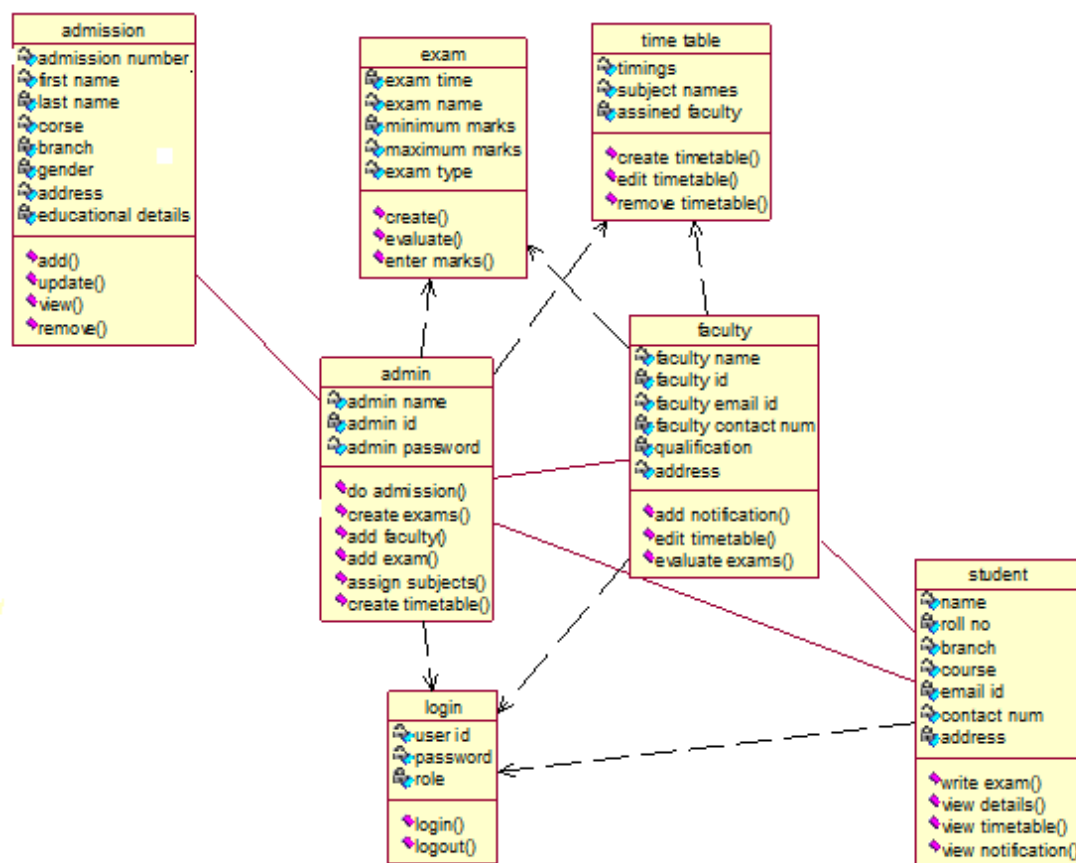


Figure 3.4.1 Class Diagram for SAA

3.4.2 Use Case Diagram for SAA

To model a system the most important aspect is to capture the dynamic behavior. To clarify a bit in details, dynamic behavior means the behavior of the system when it is running /operating. So only static behavior is not sufficient to model a system rather dynamic behavior is more important than static behavior. In UML there are five diagrams available to model dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature there should be some internal or external factors for making the interaction.

These internal and external agents are known as actors. So use case diagrams are consists of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system.

So to model the entire system numbers of use case diagrams are used.

Purpose

The purpose of use case diagram is to capture the dynamic aspect of a system. But this definition is too generic to describe the purpose.

Because other four diagrams (activity, sequence, collaboration and State chart) are also having the same purpose. So we will look into some specific purpose which will distinguish it from other four diagrams.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So when a system is analyzed to gather its functionalities use cases are prepared and actors are identified.

Now when the initial task is complete use case diagrams are modeled to present the outside view.

So in brief, the purposes of use case diagrams can be as follows:

- Used to gather requirements of a system.
- Used to get an outside view of a system.
- Identify external and internal factors influencing the system.
- Show the interacting among the requirements are actors.

Contents

Use case diagrams commonly contain

- Use case
- Actors
- Dependency
- Generalization

In this the actors are

- Admin
- Faculty
- Student

The use cases for admin

- Login
- Enter student details
- View student details
- Add faculty
- Assign subjects
- Add faculty details
- Add announcements
- Create exams
- Create timetable

The Use cases for faculty

- View student details
- Create exams
- Enter marks
- Generate reports
- Edit timetable
- Add notifications

The use cases for student

- View his/her details
- View notifications
- View marks
- Search faculty

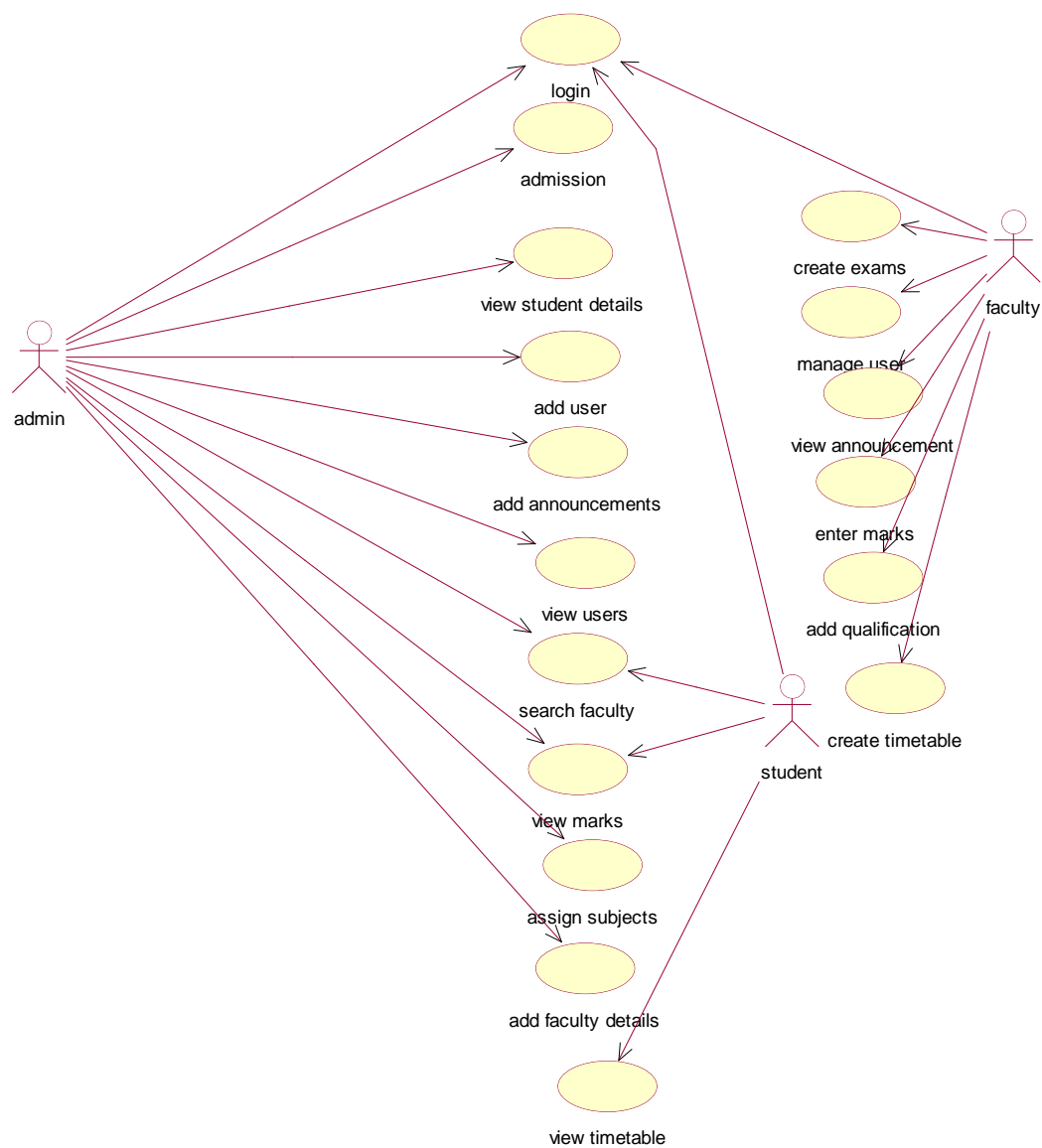


Figure: 3.4.2. Usecase diagram for SAA

3.4.3. Sequence Diagram for SAA

UML sequence diagrams are used to show how objects interact in a given situation. An important characteristic of a sequence diagram is that time passes from top to bottom: the interaction starts near the top of the diagram and ends at the bottom (i.e. Lower equals later).

A popular use for them is to document the dynamics in an object-oriented system. For each key collaboration, diagrams are created that show how objects interact in various representative scenarios for that collaboration.

The following sequence diagram shows the time ordering of messages between admin, faculty and student

The admin orders the functions of faculty and student. The functions are

- To add faculty
- Entering details
- Assigning subjects
- Add students
- Update student details
- conduct exams
- update faculty details

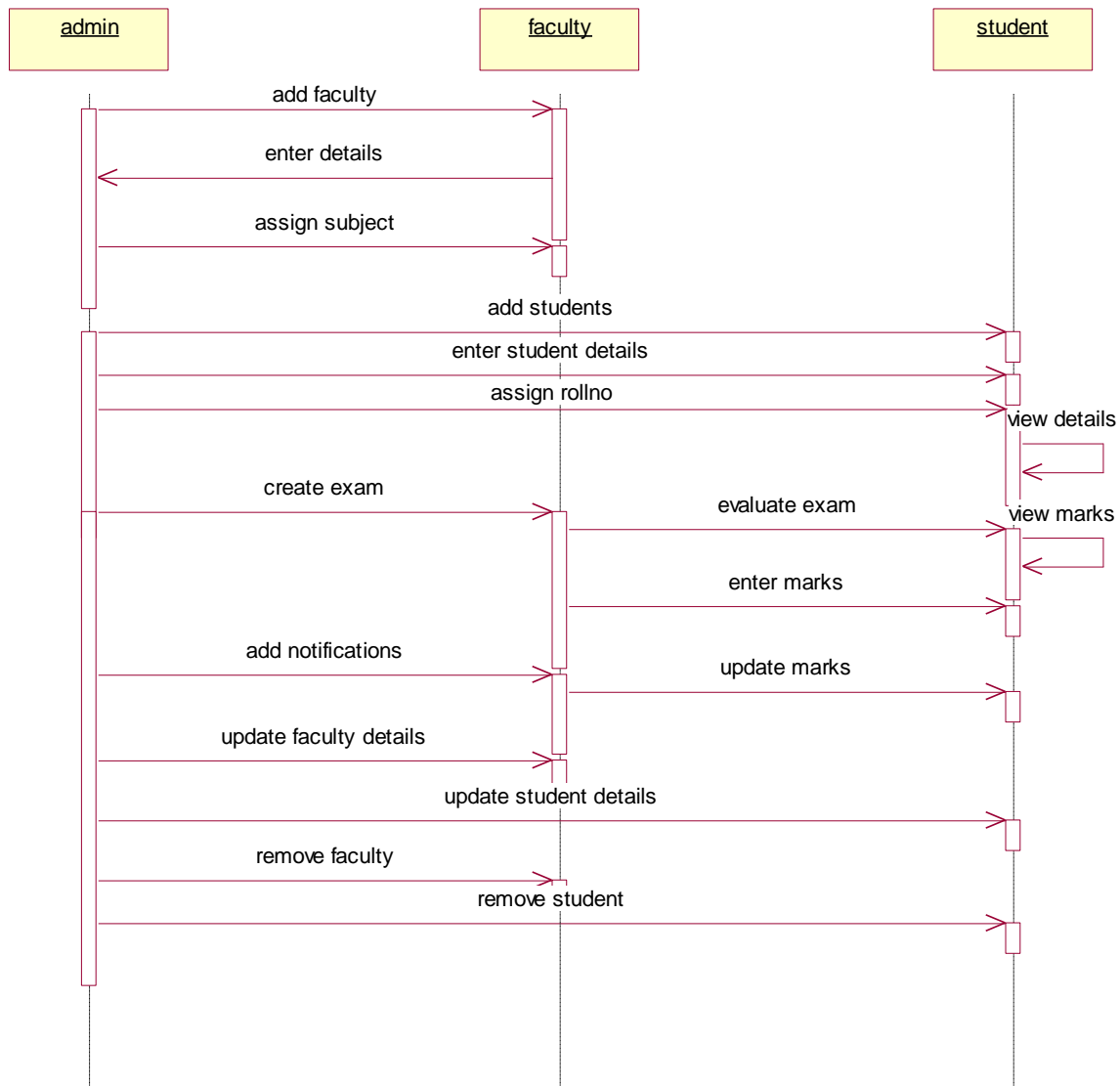


Figure 4.4.3 Sequence Diagram for roles of admin, faculty, student in SAA

3.4.4. Activity Diagram

Activity diagram is another important diagram in UML to describe dynamic aspects of the system.

Activity diagram is basically a flow chart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent. Activity diagrams deal with all type of flow control by using different elements like fork, join etc.

Purpose

The basic purposes of activity diagrams are similar to other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing dynamic nature of a system but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in activity diagram is the message part.

It does not show any message flow from one activity to another. Activity diagram is some time considered as the flow chart. Although the diagram looks like a flow chart but it is not. It shows different flow like parallel, branched, concurrent and single.

So the purposes can be described as:

- Draw the activity flow of a system.
- Describe the sequence from one activity to another.
- Describe the parallel, branched and concurrent flow of the system.

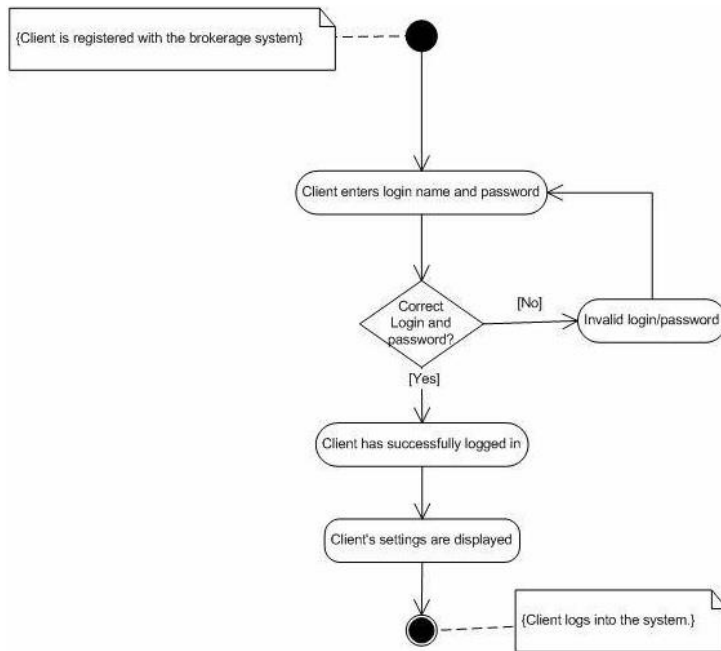


Figure 3.4.4 Activity Diagram for SAA

3.4.5. Component Diagram

Component diagrams are different in terms of nature and behavior. Component diagrams are used to model physical aspects of a system.

Now the question is what are these physical aspects? Physical aspects are the elements like executable, libraries, files, documents etc. which resides in a node.

So component diagrams are used to visualize the organization and relationships among components in a system. These diagrams are also used to make executable systems.

Purpose

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities.

So from that point component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files etc.

Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment.

A single component diagram cannot represent the entire system but a collection of diagrams are used to represent the whole.

So the purpose of the component diagram can be summarized as:

- Visualize the components of a system.
- Construct executable by using forward and reverse engineering.
- Describe the organization and relationships of the component

3.5 Module design and organization:

In SAA there are various modules as mentioned below

- Admission
- Student Details
- Manage users
- Manage news
- Timetable
- Attendance
- Human Resource

Admission module:

In this we manage the registration of a new student. The details of the student like personal info, previous education details etc., are collected and stored into a database. These details can be updated and altered. These all details are managed by the admin. The student can view his/her details. The students have only “READ” permission.

Student Details:

This module gives the entire information of a particular student. The admin can manage these details. He can modify and update the information. He has READ and WRITE permissions.

The faculty can also view these information. This module consists of the details like name, gender, address, department, roll number etc.,

Manage users:

This module manages the users of the system. There are three different types of users. They are

- Admin
- Faculty
- Student

Admin:

The admin acts as a “HEAD”. He manages the entire system. Modifications to the system can be done by him. He has all the access permissions

Faculty:

The role of the faculty is to maintain the attendance, generate exams, entering the marks, adding any news, giving announcements. Faculty has read and write permissions based on the designation.

Student:

The student has only read permission. He can view only his details. He has no permission to change the information. He can read the announcements, view the timetable.

Manage News:

In this module we can just maintain the notifications regarding exams, any announcements. The Admin and Faculty can manage the news but student can only read the notifications.

Academics Module

Attendance:

In this the attendance of the student is maintained. This module is managed by the faculty. The student can just view the attendance of his own. He has no permission to modify the details.

Timetable:

This module is used to generate timetables, create new timetables for a class, edit timetables, view the generated timetables.

Human Resources:

In these the information regarding faculty is maintained. There are many submodules like

- Employment search
- Employment management
- HR settings

3.6 CONCLUSION

Based on all the diagrams we are able to design the required functionalities and the flow of data that is to be maintained between each of them. By doing all this we are able to maintain the mobile application without any bugs and errors. All the diagrams that are developed show us the functionalities of the mobile application.

Implementation

4.1 Introduction

The user enters into the web site and navigates through many pages. First the user login through the login page and then enters into the pages in the app. If the user is not an authenticated person he enters into registration page and then registers and login.

4.2 Explanation of Key functions

The key functions of the project are to analyze the data of the project in various types. Here the employee will update his information and that will be updated in the databases related accordingly.

- **What is Ruby?**

Ruby is a computer programming language. You might be familiar with Basic, Java, C++, or some other programming language. In certain ways, all these languages are the same. They all provide ways for you to give instructions to a computer. “Move this value from that memory location to that other location on your hard drive.” A computer language is a way of expressing instructions in a precise, unambiguous manner. What makes Ruby different from so many other computer programming languages? In what way does Ruby support agile development? Here’s the answer: Ruby is a dynamically typed, interpreted, reflective, object-oriented language. That’s a great answer, but what does it mean? Ruby is dynamically typed in many languages; you have to declare each variable’s type. You write `int date; date = 25092006;` The first line tells the computer that the date must store an integer — a whole number — a number without a decimal point — a number like 25092006. Later in the same program, you might write `date = “September 25, 2006”;` But the computer refuses to accept this new line of code. The computer flags this line with an error message. The value “September 25, 2006” isn’t an integer. (In fact, “September 25, 2006” isn’t a number.) And because of the `int date;` line, the non-Ruby program expects date to store an integer. The word `int` stands for a type of value. In a statically typed language, a variable’s type doesn’t change. In contrast, Ruby is dynamically typed. The following lines form a computer

- Ruby is interpreted
- Many commonly used programming languages are compiled. When the computer compiles a program, the computer translates the program into a very detailed set of instructions (a set more detailed than the set that the programmer originally writes). So, picture yourself developing code in a compiled language. First you write the code. Then you compile the code. Then you run the code. The code doesn’t run exactly the way you want it to run, so you modify the code. Then

you compile again. And then you run the code again. This cycle takes place hundreds of times a day. “Modify, compile, run.” You get tired of saying it inside your head. In contrast, to the compiled languages, Ruby is interpreted. An interpreted language bypasses the compilation step. You write the code, and then you run the code. Of course you don’t like the results. (That’s a given.) So you modify and rerun the code. The whole cycle is much shorter. A piece of software (the Ruby interpreter) examines your code and executes that code without delay.

Testing and Validation

5.1 TESTING PLAN

Software Testing has a dual function; it is used to identify the defects in program and it is used to help judge whether or not program is usable in practice. Thus software testing is used for validation and verification, which ensure that software conforms to its specification and meets need of the software customer.

Developer resorted Alpha testing, which usually comes in after the basic design of the program has been completed. The project scientist will look over the program and give suggestions and ideas to improve or correct the design. They also report and give ideas to get rid of around any major problems. There is bound to be a number of bugs after a program have been created.

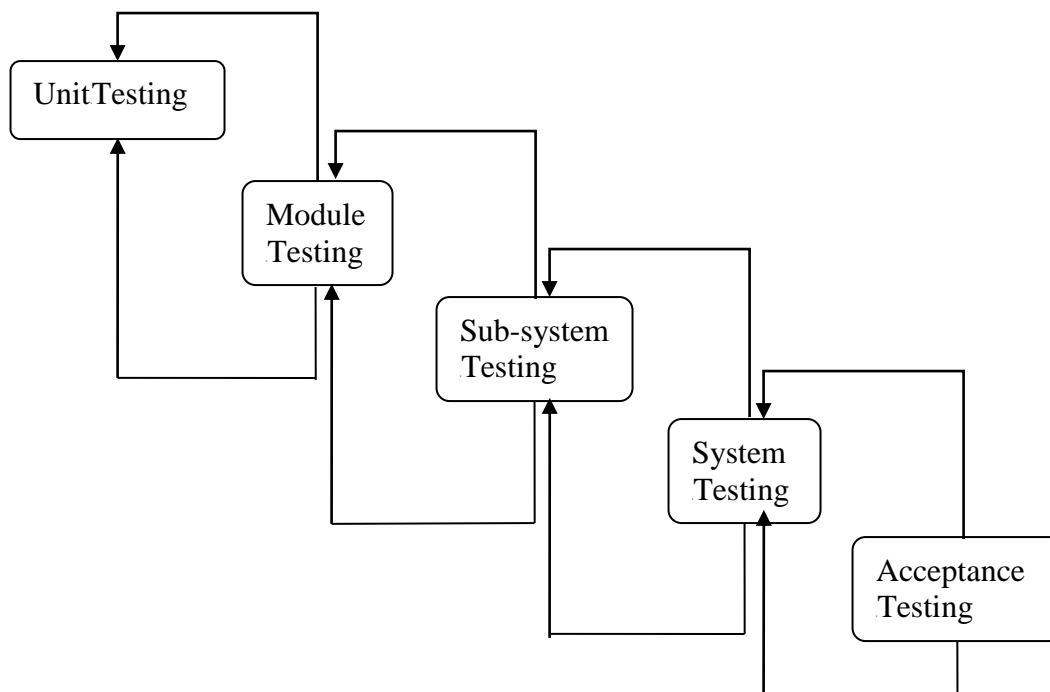


Fig5.1 Testing Step

Branch coverage

It is strategy in which test cases are designed to make each branch condition assume true & false values.

Conditional coverage

In this testing test cases are designed to make each component of composite conditional expression both true & false.

The Testing Process

We test the software process activities such as Design, Implementation, and Requirement Engineering. Because, design errors are very costly to repair once system has been started to operate, it is quite obvious to repair them at early stage of the system. So analysis is the most important process of any project.

Requirement Traceability

As most interested portion is whether the system is meeting its requirements or not, for that testing should be planned so that all requirements are individually tested. We checked the output of certain combination of inputs so that we can know whether it gives desirable results or not. Strictly sticking to your requirements specifications, give you the path to get desirable results from the system.

Testing Schedule

We have tested each procedure back-to-back so that errors and omissions can be found as early as possible. Once the system has been developed fully we tested it on other machines, which differs in configuration.

5.2 TESTING STRATEGY

There are types of testing that we implement. They are as follows:

- While deciding on the focus of testing activities, study project priorities. For example, for an on-line system, pay more attention to response time. Spend more time on the features used frequently.
- Decide on the effort required for testing based on the usage of the system. If the system is to be used by a large number of users, evaluate the impact on users due to a system failure before deciding on the effort.
- A necessary part of the test case is a definition of the expected result.
- Write test cases for invalid and unexpected as well as valid and expected input conditions.
- Thoroughly inspect the results of each test.

We have performed both Unit Testing and System Testing on WIMS to detect and fix errors. A brief description of both is given below.

▪ Unit Testing

Objective

The objective of Unit Testing is to test a unit of code (program or set of programs) using the Unit Test Specifications, after coding is completed. Since the testing will depend on the completeness and correctness of test specifications, it is important to subject these to quality and verification reviews.

Input

- Unit Test Specifications
- Code to be tested

Testing Process

- Checking for availability of Code Walk-through reports which have documented the existence of and conformance to coding standards.
- Verify the Unit Test Specifications conform to the program specifications.

- Verify that all boundary and null data conditions are included.

- **Features to be tested**

Table 5.2 Feature tested table

Test Specification	Description
GUI and general Tests	Screen consistency with respect to project specific standards and checklist.
Menu commands are executed at least once	Functionality with respect to description in menu or image in toolbar.
Functionality Test	All possible scenarios to test the functionality of the component are listed here. This list is made very exhaustive to cover all the expected functionality described in the Software Requirement Specifications and Design document completely.
Boundary Value Analysis for EOF/BOF and variables	Checks for EOF/BOF, closing of result sets/connections
'Null Data' cases are covered	Handling of Null values.
'Valid Data' as well as 'Invalid Data' cases are covered.	Suitable Error/Warning Messages
Access Control	Access controls as specified in the security module
Print Testing	If on preprinted stationary or on some specific paper size.

▪ Unit Test Specifications

A sample Unit Test Specification is as follows.

Form Template Functionality

Table 5.3 Form template table

Event	Action	Expected Result	Observed result	Verified
1.	On pressing Add Button	Should clear the screen and allow entry for new record. State of other Buttons in Toolbar Modify=Disable Save=Enable Query=Disable Fetch=Disable Delete=Disable Cancel=Enable	As Expected	YES
2.	On pressing Query Button	Set the fields to accept the query condition. State of other Buttons in Toolbar Modify=Disable Save=Disable Add=Disable Fetch=Enable Delete=Disable Cancel=Enable	As Expected	YES
3.	On pressing Fetch button	Get all the records, which satisfy the query condition. State of other Buttons in Toolbar Modify=Enable	As Expected	YES

		Save=Disable Query=Enable Add=Enable Delete=Enable Cancel=Disable		
4.	On pressing Savebutton	Record should be inserted in database StateofotherButtonsinToolbar Modify=Enable Add=Enable Query=Enable Fetch=Disable Delete=Enable Cancel=Disable	AsExpected	YES
5.	On pressing Modify button	Recordshouldbeupdated. StateofotherButtonsinToolbar Add=Disable Save=Enable Query=Disable Fetch=Disable Delete=Disable Cancel=Enable	AsExpected	YES
6.	On pressing Cancel button	Screen should go into default mode of screen StateofotherButtonsinToolbar Modify=Disable Save=Disable Query=Enable Fetch=Disable Delete=Disable Add=Enable	AsExpected	YES

7.	Deletebutton.	NotApplicable	AsExpected	YES
8.	Formloadevent	The form should open in Cancel Mode	AsExpected	YES
9.	Statusbarstatus	User ID, Screen ID and Screen Mode is set. Also on the focus event of any input control the tool tip is displayed in the status bar	AsExpected	YES
10.	Security Features	According to the Access rights defined the corresponding buttons should be Enabled/Disabled	AsExpected	YES

▪ **Integration Testing**

After our individual modules were tested out we proceed to the integration testing to create a complete system. This integration process involves building the system and testing the resultant system for problems that arise from component interactions.

We have applied top-down strategy to validate high-level components of a system before design and implementations have been completed. Our development process started with high-level components and we worked down the component hierarchy.

▪ **System Testing**

System testing is actually a series of tests whose purpose is to fully exercise the computer-based system. It verifies that system elements have been properly integrated and perform allocated functions. It checks whether the system as a whole works as per requirement. We have used Performance testing. Performance testing - designed to test the run-time performance of software, especially real-time software.

▪ **Performance Testing**

This is designed to test the run-time performance of software within the context of an integrated system. Performance testing occurs throughout all steps in the testing process. Our system is checked for high load as well as low load

▪ Statistical Testing

Statistical Testing is used to test the program's performance and reliability and to check how it works under operational conditions. Tests are designed to reflect the actual user inputs and their frequency.

The stages involved in the static analysis for this system are follows.

Control flow analysis

- Unreachable code
- Unconditional branches into loops

Data use analysis

- Variable used before initialization
- Variables declared but never used
- Variables assigned twice but never used between assignments
- Possible array bound violations
- Declared variables

Interface analysis

- Parameter type mismatches
- Parameter number mismatches
- Non-usage of the results of functions
- Uncalled functions and procedures

Storage management faults

- Images not Stored in Resources
- Out of Bound -> Program's non-volatile memory.

5.3 TEST CASES

A test case is a set of conditions or variables and inputs that are developed for a particular goal or objective to be achieved on a certain application to judge its capabilities or features.

It might take more than one test case to determine the true functionality of the application being tested. Every requirement or objective to be achieved needs at least one test case. Some software development methodologies like Rational Unified Process (RUP) recommend creating at least two test cases for each requirement or objective; one for performing testing through positive perspective and the other through negative perspective.

- **Test Case Structure**

A formal written test case comprises of three parts -

Information

Information consists of general information about the test case. Information incorporates Identifier, test case creator, test case version, name of the test case, purpose or brief description and test case dependencies.

Activity

Activity consists of the actual test case activities. Activity contains information about the test case environment, activities to be done at test case initialization, activities to be done after test case is performed, and step by step actions to be done while testing and the input data that is to be supplied for testing.

Results

Results are outcomes of a performed test case. Results data consist of information about expected results and the actual results.

Designing Test Cases

Test cases should be designed and written by someone who understands the function or technology being tested. A test case should include the following information -

- Purpose of the test
- Software requirements and Hardware requirements (if any)
- Specific setup or configuration requirements
- Description on how to perform the test(s)
- Expected results or success criteria for the test

Designing test cases can be time consuming in a testing schedule, but they are worth giving time because they can really avoid unnecessary retesting or debugging or at least lower it. Organizations can take the test cases approach in their own context and according to their own perspectives. Some follow a general step way approach while others may opt for a more detailed and complex approach. It is very important for you to decide between the two extremes and judge on what would work the best for you. Designing proper test cases is very vital for your software testing plans as a lot of bugs, ambiguities, inconsistencies and slipups can be recovered in time as also it helps in saving your time on continuous debugging and re-testing test cases.

Table 5.3.1 Test Cases for Admission

Sr.No.	Test Condition	Expected Output	Actual Output	Pass/Fail
1	Click on admission mode	The page with details form should be opened	Page with detail form is opened	Pass
2	Click on new	The new form is opened	New form is opened	Pass
3	Name field	Only characters should be accepted	Only characters are opened	Pass
4	Click on roll number	Characters are not accepted	It rejects the strings	Pass
5	Mandatory fields	All the mandatory fields should be filled	It does not enable until all fields are entered	Pass
6	Update and submit	The details added should be submitted	All the details are submitted	Pass

Table 5.3.2. Test cases for entering details

Sr.No.	Test Condition	Expected Output	Actual Output	Pass/Fail
7	Initial mode of opening the page.	Initial mode should be cancel.	Initial mode is cancel.	Pass
8	Clicking on add button	System should only enable save and cancel mode and automatically generated student ID should be shown.	Only save and cancel mode are enabled, student ID is shown in textbox as read only mode	Pass
9	In add mode filling "father's income" text box	System should only allow digits	System inserts allow digits	Pass
10	In add mode filling phone number and mobile number	System only allow digits and with limited length	System only allows digits and after certain length user can not enter more digit	Pass
11	Clicking 'save' button without filling mandatory fields	System should show alert for required fields	System shows alert box for required fields.	Pass
12	Clicking on 'cancel' in any mode	System should clear all fields and go to cancel mode	System resets all control and cancel mode is shown	Pass
13	Type of course should be selected first in 'add' mode for admission details	System should not allow to select other details before selecting type of course for admission	System shows alert box to select type of course	Pass
14	After selecting type of course in 'add' mode	System should alert if there is no available seat.	System alerts for seat is not there.	Pass
15	Clicking on 'save' button in add mode	System should insert the record in the Database and default mode should be shown with all fields reset.	System enters the record in the Database, cancel mode is shown and all controls are reset.	Pass
16	Clicking on 'query' button	'Fetch' and 'cancel' button should be enabled and student ID is allowed to be selected.	Only 'fetch' and 'cancel' are available and LOV and text box for student ID are	Pass

			enabled.	
17	In query mode clicking on 'fetch' button without entering student ID	System should show alert for Student ID is not entered.	System displays alert for student ID is not selected.	Pass
18	On clicking 'fetch' button in query mode if student ID is provided.	System should show all details of student as read only mode and 'modify' and 'cancel' button should be enabled only.	Details of student is shown as read only manner, and modify and cancel mode are only allowed.	Pass
19	Clicking on modify button	It should allow to edit details and 'cancel' and 'save' button are only enabled.	Fields are editable and 'cancel' and 'save' buttons are only enabled.	Pass

Table 5.3.3 Test Cases for Admission

Sr.No.	Test Condition	Expected Output	Actual Output	Pass/Fail
1	Initial mode of opening the page.	Initial mode should be cancel.	Initial mode is cancel.	Pass
2	Clicking on add button	System should only enable save and cancel mode and student ID fields should not be disabled or read only.	Only save and cancel mode are enabled, student ID textbox and LOV controls are enabled	Pass
3	Providing student ID in add mode	System should all status of student as well as result status	If student is eligible for enrollment as per result status then fields are editable for the enrollment.	Pass
4	Clicking 'save' button without filling mandatory fields	System should show alert for required fields	System shows alert box for required fields.	Pass
5	Clicking on 'cancel' in any mode	System should clear all fields and go to cancel mode	System resets all control and cancel mode is shown	Pass
6	Clicking on 'save' button in add mode	System should insert the record in the Database and default mode should	System enters the record in the Database, cancel mode is shown and	Pass

		be shown with all fields reset.	all controls are reset.	
7	Clicking on 'query' button	'fetch' and 'cancel' button should be enabled and student ID is allowed to be selected.	Only 'fetch' and 'cancel' are available and LOV and text box for student ID are enabled.	Pass
8	In query mode clicking on 'fetch' button without entering student ID	System should show alert for Student ID is not entered.	System displays alert for student ID is not selected.	Pass
9	On clicking 'fetch' button in query mode if student ID is provided.	System should show all details of student as read only mode and 'modify' and 'cancel' button should be enabled only.	Details of student is shown as read only manner, and modify and cancel mode are only allowed.	Pass
10	Clicking on modify button	It should allow to edit details and 'cancel' and 'save' button are only enabled.	Fields are editable and 'cancel' and 'save' buttons are only enabled.	Pass
11	Modify button	'Modify' button should be never enabled	'modify' button is always disabled	Pass

Table 5.3.4 Test Cases for modifying student details

Sr.No.	Test Condition	Expected Output	Actual Output	Pass/Fail
1	Initial mode of opening the page.	Initial mode should be cancel.	Initial mode is cancel.	Pass
2	'Add' button	'Add' button should be disabled	'Add' button is always disabled	Pass
3	Clicking on 'query' button	System should enable all controls and 'fetch' and 'cancel' button should be enabled	System allows all the controls and 'cancel' and 'fetch' mode are available.	Pass
4	In query mode clicking on 'fetch' button without entering student ID	System should show alert for Student ID is not entered.	System displays alert for student ID is not selected.	Pass

5	On clicking 'fetch' button in query mode if student ID is provided.	System should show all details of student as read only mode and 'modify' and 'cancel' button should be enabled only.	Details of student is shown as read only manner, and modify and cancel mode are only allowed.	Pass
6	Clicking on 'cancel' in any mode	System should clear all fields and go to cancel mode	System resets all control and cancel mode is shown	Pass
7	Clicking on modify button	It should allow to edit details and 'cancel' and 'save' button are only enabled.	Fields are editable and 'cancel' and 'save' buttons are only enabled.	pass
8	Clicking 'save' button without filling mandatory fields	System should show alert for required fields	System shows alert box for required fields.	Pass
10	Clicking on 'query' button	'fetch' and 'cancel' button should be enabled and student ID is allowed to be selected.	Only 'fetch' and 'cancel' are available and LOV and text box for student ID are enabled.	Pass

Table 5.3.5 Test Cases for Searching Student

Sr.No.	Test Condition	Expected Output	Actual Output	Pass/Fail
1	Initial mode of opening the page	Initial mode should be cancel.	Initial mode is cancel.	Pass
2	'Add' button	'Add' button should be disabled	'Add' button is always disabled	Pass
3	Clicking on 'query' button	System should enable all controls and 'fetch' and 'cancel' button should be enabled	System allows all the controls and 'cancel' and 'fetch' mode are available.	Pass
4	In query mode clicking on 'fetch' button without selecting search criteria or search details	System should show alert for criteria or details is not provided.	System displays alert for mandatory fields	Pass
5	On clicking 'fetch'	System should show	Alert box is shown if	Pass

	button in query mode with improper search details for particular searchcriteria.	alert for improper details.	details are not valid as per search criteria.	
6	On clicking 'fetch' button with proper details.	System should display student details as per records in database.	System shows student details.	Pass
7	Clicking on modify button	It should allow to edit details and 'cancel' and 'save' button are only enabled.	Fields are editable and 'cancel' and 'save' buttons are only enabled.	Pass
8	Clicking 'save' button without filling mandatory fields	System should show alert for required fields	System shows alert box for required fields.	Pass

Table 5.3.6 Test Cases for Setting Fees structure

Sr.No.	Test Condition	Expected Output	Actual Output	Pass/Fail
1	Initial mode of opening the page	Initial mode should be cancel.	Initial mode is cancel	Pass
2	'Add' button	'add' button should be always disabled	'add' button is always disabled.	Pass
3	Clicking 'modify' button	Fields should be editable and 'save' and 'cancel' button should be enabled	Fields are editable and 'save' and 'cancel' button are enabled	pass
4	Clicking 'save' without filling mandatory details	System should show alert for the mandatory fields	Alert is shown by the system for the mandatory fields.	pass
5	Textboxes for fees	Only digits should be allowed	User is not able to enter other than digits	Pass
6	Textbox for year	should not be editable and show current year only	Year text box is not editable and shows current year	
7	Clicking 'save' with filling complete details	System should update data to database and 'query' button should be enabled.	System inserts data to database and 'query' button is enabled.	Pass
8	Clicking on 'query'	System should	System allows all	Pass

	button	enable all controls and 'fetch' and 'cancel' button should be enabled	the controls and 'cancel' and 'fetch' mode are available.	
9	In query mode clicking on 'fetch' button without entering required details.	System should show alert.	System displays alert.	Pass
10	On clicking 'fetch' button in query mode if required details are provided.	System should show all details in read only mode and 'modify' and 'cancel' button should be enabled only.	Details are shown as read only manner, and modify and cancel mode are only allowed.	Pass
11	Clicking on 'cancel' in any mode	System should clear all fields and go to cancel mode	System resets all control and cancel mode is shown	Pass
12	Clicking on modify button	It should allow to edit details and 'cancel' and 'save' button are only enabled.	Fields are editable and 'cancel' and 'save' buttons are only enabled.	pass
13	Dropdown for 'semester' and 'year'	Should only be selectable after selecting course	'semester' and 'year' are not selectable till the 'course' is selected.	Pass

Table 5.3.7 Test Cases for scheduling exam

Sr.No.	Test Condition	Expected Output	Actual Output	Pass/Fail
1	Initial mode of opening the page	Initial mode should be cancel.	Initial mode is cancel.	Pass
2	Clicking 'Add' button	'save' and 'cancel' button should be enabled and fields are editable.	'Save' and 'cancel' button are enabled and fields are editable.	Pass
3	Clicking 'save' without filling mandatory details	System should show alert for the mandatory fields	Alert is shown by the system for the mandatory fields.	pass
6	Clicking on 'query' button	System should enable all controls and 'fetch' and 'cancel' button should be enabled	System allows the entire controls and 'cancel' and 'fetch' mode are available.	Pass

7	In query mode clicking on 'fetch' button without entering mandatory fields.	System should show alert for fields are not entered.	System displays alert for fields not selected.	Pass
8	On clicking 'fetch' button in query mode if fields are provided.	System should show all details of scheduled exam for selected details and 'modify' and 'cancel' button should be enabled	System should show all details of scheduled exam for selected details and 'modify' and 'cancel' button are enabled.	Pass
9	Clicking on 'cancel' in any mode	System should clear all fields and go to cancel mode	System resets all control and cancel mode is shown	Pass
10	Clicking on modify button	It should allow editing details and 'cancel' and 'save' button are only enabled.	Fields are editable and 'cancel' and 'save' buttons are only enabled.	pass
11	In add mode	Other fields are only allowed to be entered if course is selected	other fields are not allowed to enter till course is selected	pass

Table 5.3.8 Test Cases for declaring result

Sr.No.	Test Condition	Expected Output	Actual Output	Pass/Fail
1	Initial mode of opening the page	Initial mode should be cancel.	Initial mode is cancel.	Pass
2	Clicking 'Add' button	'save' and 'cancel' button should be enabled and fields are editable.	'save' and 'cancel' button are enabled and fields are editable.	Pass
3	Clicking 'save' without filling mandatory details	System should show alert for the mandatory fields	Alert is shown by the system for the mandatory fields.	Pass
6	Clicking on 'query' button	System should enable all controls and 'fetch' and 'cancel' button should be enabled	System allows all the controls and 'cancel' and 'fetch' mode are available.	Pass
7	In query mode clicking on 'fetch'	System should show alert for fields are	System displays alert for fields not	Pass

	button without entering mandatory fields.	not entered.	selected.	
8	On clicking 'fetch' button in query mode if fields are provided.	System should show result status for selected year, course semester, exam-type, subject.	System shows status of result for selected year, course, semester, exam-type, subject	Pass
9	Clicking on 'cancel' in any mode	System should clear all fields and go to cancel mode	System resets all control and cancel mode is shown	Pass
10	Clicking on modify button	It should allow to edit details and 'cancel' and 'save' buttons are only enabled.	Fields are editable and 'cancel' and 'save' buttons are only enabled.	Pass
11	In add mode, text box of marks	Status of pass/fail should be automatic as per entered mark.	Status is set automatically as per entered marks.	Pass
12	In add mode, text box of marks	Only digits are allowed to be entered.	Only digits are entered.	Pass
13	In add mode	Other fields are only allowed to be entered if course is selected	other fields are not allowed to enter till course is selected	Pass
14	Clicking on 'fetch student' button	If mandatory fields are not entered then it should show alert.	Alert is shown if mandatory fields are not entered.	Pass
15	Clicking on 'fetch student' button	If mandatory fields are entered then it should show list of all students	List of all student t is shown if mandatory fields are entered.	Pass

5.4 Output Screens

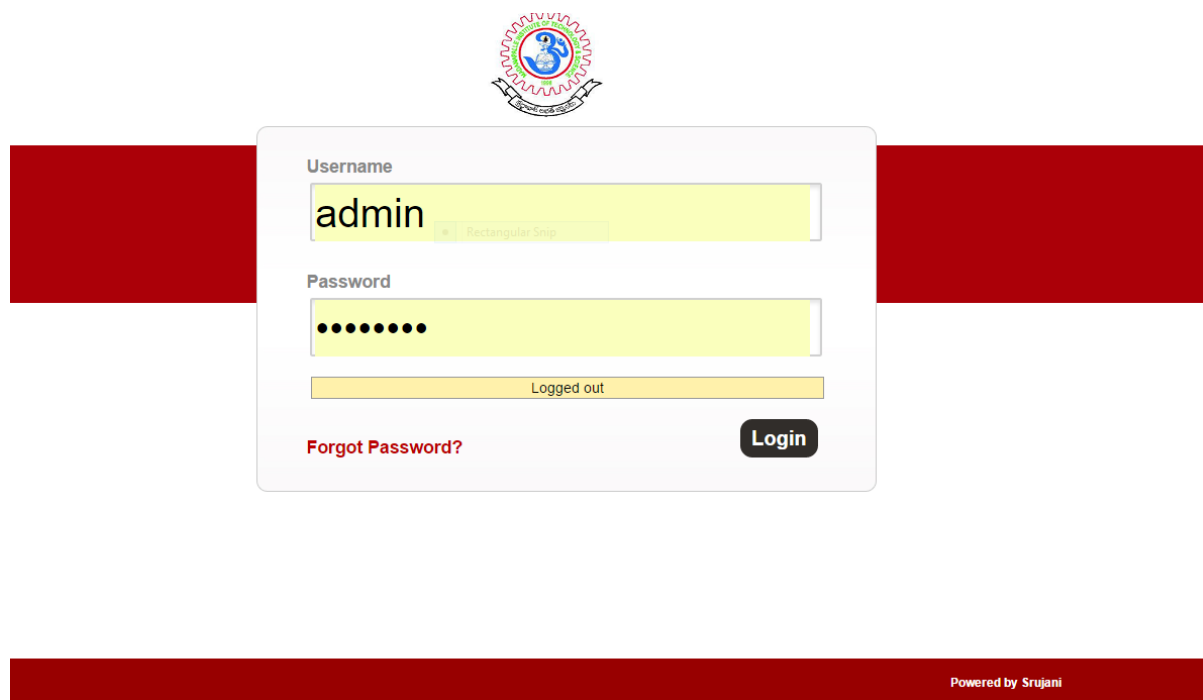


Fig:5.4.1 Login page for different users

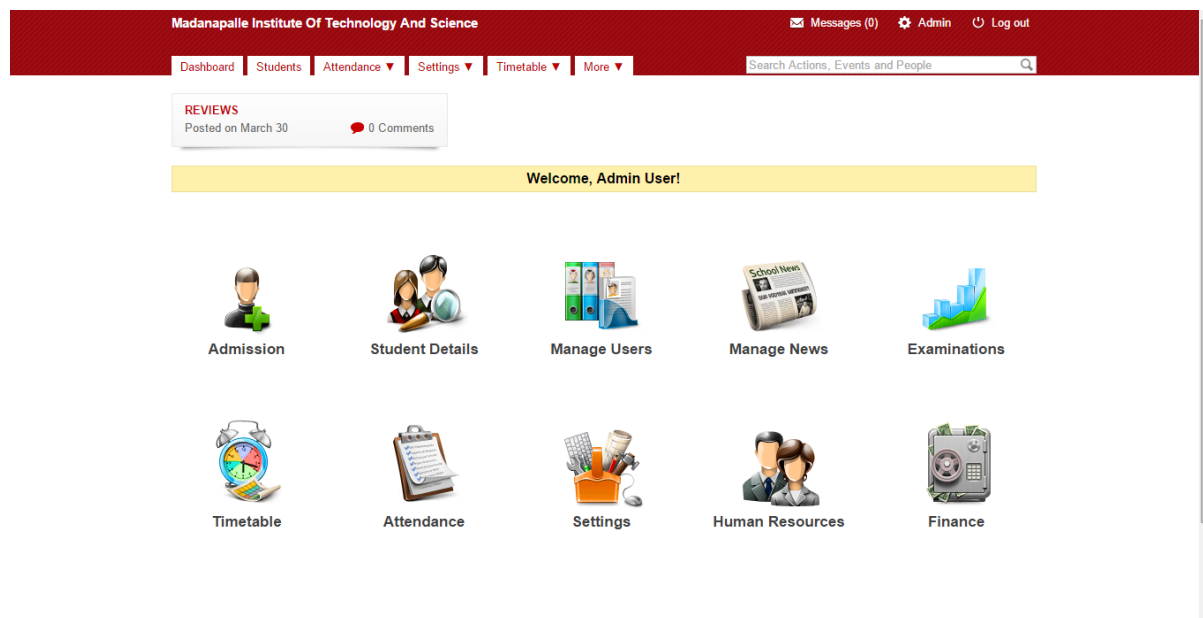


Fig:5.4.2. Dashboard consists of different modules

Madanapalle Institute Of Technology And Science

Messages (0) Admin Log out

Dashboard Students Attendance Settings Timetable More

Search Actions, Events and People

Admission
Step 1 - Student details

Back

Fields marked with * must be filled.

Admission number * 11691A05a3

Admission Date * 2015-04-08

Personal Details

First Name *

Middle Name

Last Name

Course & Batch 111 - Btech 11-15

Date of Birth * 2010-04-08

Fig:5.4.3Formforadmission

Madanapalle Institute Of Technology And Science

Messages (0) Admin Log out

Dashboard Students Attendance Settings Timetable More

Search Actions, Events and People

Create User
Make new admin user

Back

Fields marked with * must be filled. (Only Admin User can be created)

Username *

First Name *

Last Name faculty

Password *

Email *

Create

Fig:5.4.4.pagefornewadminuser

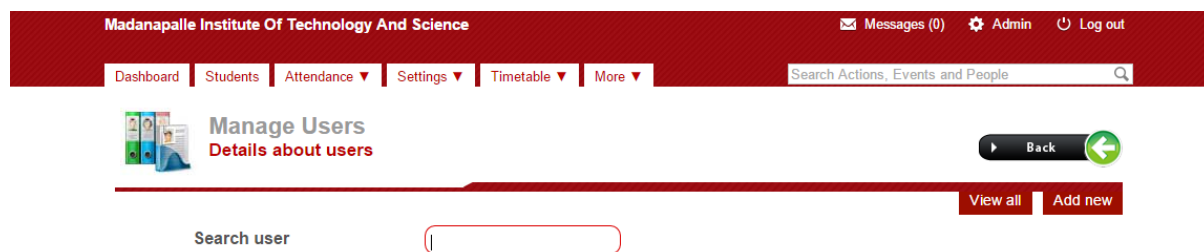


Fig:5.4.5.seachingforuser

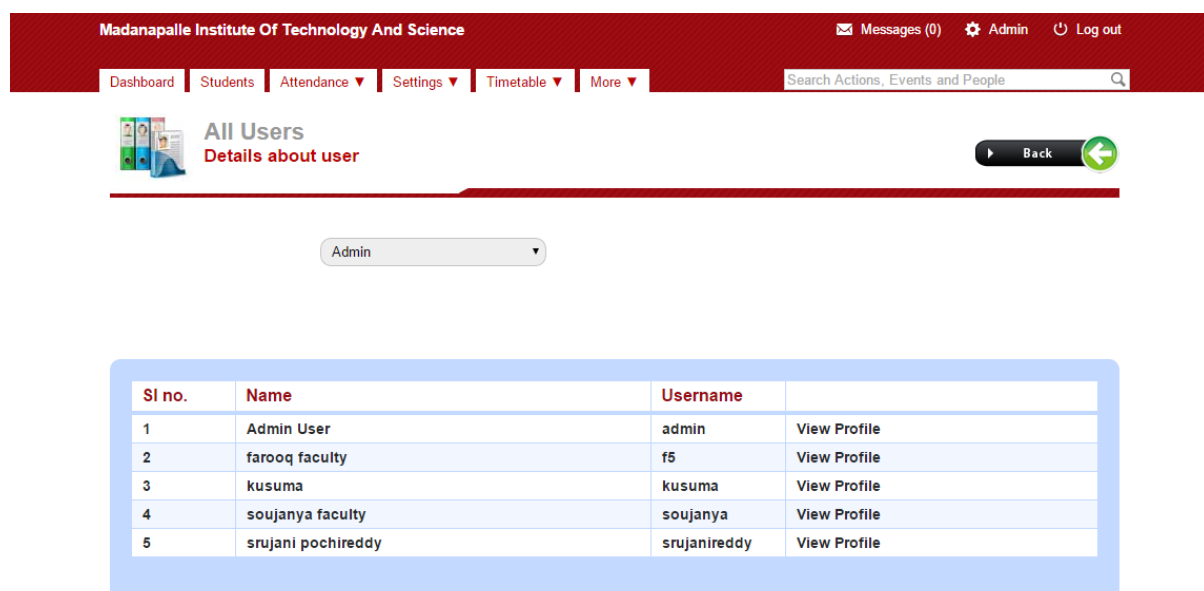


Fig:5.4.6viewingtheexistingusers

Madasapalle Institute Of Technology And Science
Dashboard | Students | Attendance | Settings | Timetable | Store | Log out

Employee admission
Step - 1

Employee number of last employee is 100

Fields marked with * must be filled.

General Details

Employee Number*

Joining date

First Name*

Middle Name

Last Name

Email

Gender ☒ Male ☐ Female

Date of Birth*

Department*

Category*

Position*

Grade

Job Title

Qualification

Experience Info

Total Experience

Status ☒ Active ☐ Inactive

Personal Details

Marital Status

Father Name

Mother Name

Blood Group

Nationality

Upload User Photo


Fig:5.4.7.Employee admission form

Madanapalle Institute Of Technology And Science

Messages (0)AdminLog out


DashboardStudentsAttendance▼Settings▼Timetable▼More▼

Search Actions, Events and People

Employee Profile
Details

Back

Profile▼Salary▼Leaves▼TimetableDeleteMore▼



Farooq

Employee ID : F5

Joining date	March 11, 2008
Department	Computer Science And Engineering
Category	Faculty
Position	Assistant Professor
Grade	
Job Title	
Manager	Change
Gender	Male
Email	
Status	Active
Qualification	
Total Experience	7 Years 0 Months
Experience Info	

Edit

PDF Report

Fig:5.4.8viewingemployeedetails

The screenshot shows the Madanapalle Institute Of Technology And Science dashboard. The top navigation bar includes links for Dashboard, Students, Attendance, Settings, Timetable, and More. A search bar is available for actions, events, and people. The main content area displays a 'View all' button for employees. Below this, a dropdown menu allows selecting a department, currently set to 'Computer Science And Engi'. A table lists 15 employees with their names, employee numbers, and departments.

Name	Employee Number	Department
Bala Krishna	G0	Computer Science And Engineering
Farooq	F5	Computer Science And Engineering
Jaganatha Reddy	E6	Computer Science And Engineering
Kusuma	F2	Computer Science And Engineering
Nageshwara	F3	Computer Science And Engineering
Narasimha	G1	Computer Science And Engineering
Naveen	G2	Computer Science And Engineering
Padhmanabha Reddy	F4	Computer Science And Engineering
Prabhakar	F8	Computer Science And Engineering
Rajarajeswari	G3	Computer Science And Engineering
Shabeer	F6	Computer Science And Engineering
Sri Devi	G7	Computer Science And Engineering
Sudhakar Yadav	G6	Computer Science And Engineering
Surya Bahadur	F9	Computer Science And Engineering
Vikram Phanindra	G8	Computer Science And Engineering

Fig:5.4.9view all the employee

The screenshot shows the 'Attendance register' interface. The top navigation bar is the same as the previous screenshot. The main content area has a title 'Attendance Attendance register' and a 'Back' button. Below this, there are dropdown menus for 'Select a batch' (set to '111 - Batch 11-15') and 'Select a subject' (set to 'Electronic Devices And Circuits'). A date selector shows 'April 2015'. A table lists names for attendance recording.

Name
shiva
Sudhakar
anusha
sritha
srinika
srinani
srinani
srinani
srinani
srinani

Fig:5.4.10.Maintaining the attendance

Conclusion

6. Conclusion

A complete automation of a school is done with ease and effectiveness with the software we've designed. Statistical data can be managed easily and viewed using the timely reports. Organization of student data includes his/her pre-entry details and post entry details (alumni) along with the present information in the school. Student information includes his/her academic data as well as management related data such as fees, attendance e.t.c., Our software not only provides the categorized information regarding the student, it also facilitates the graphical view of the corresponding information in various presentation tools like graphs and charts for a better viewing experience.

7. REFERENCES

1. www.ruby-lang.org/en/documentation/
2. www.w3schools.com/css/
3. www.dev.mysql.com/doc/
4. www.api.rubyonrails.org/
5. www.w3schools.com/jsref/