



# SRM VALLIAMMAI ENGINEERING COLLEGE

(An Autonomous Institution)



SRM Nagar, Kattankulathur-603203.

## DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

Regulation 2019  
Lab Manual

**ACY:2024-2025**

**1922507 - SOFTWARE ENGINEERING & MANAGEMENT LAB**

(V semester)

Prepared By

**Mr.R. Danu, Asst.Prof(O.G)**

**1922507 SOFTWARE ENGINEERING AND MANAGEMENT LABORATORY L T P C**

0 0 4 2

**OBJECTIVES:**

- To understand the software engineering methodologies for project development.
- To gain knowledge about open source tools for implementing software engineering methods.
- To develop an efficient software using case tools.
- To exercise developing product-startups implementing software engineering methods.
- To have hands on experience in developing a software project by using various software engineering principles and methods in each of the phases of software development.

**LIST OF EXPERIMENTS:**

Prepare the following documents for each experiment and develop the software using software engineering methodology.

1. **Problem Analysis and Project Planning** - Thorough study of the problem – Identify Project scope, Objectives and Infrastructure.
2. **Software Requirement Analysis** - Describe the individual Phases/modules of the project and Identify deliverables.
3. **Data Modelling** - Use work products – data dictionary, use case diagrams and activity diagrams, build and test class diagrams, sequence diagrams and add interface to class diagrams.
4. **Software Development and Debugging** – implement the design by coding
5. **Software Testing** - Prepare test plan, perform validation testing, coverage analysis, memory leaks, develop test case hierarchy, Site check and site monitor.
6. **Project Management** – Project Scheduling, Planning and RISK Management.

**SAMPLE EXPERIMENTS:****Academic domain**

1. Course Registration System
2. Student marks analysing system

**Railway domain**

3. Online ticket reservation system
4. Platform assignment system for the trains in a railway station

**Medicine domain**

5. Expert system to prescribe the medicines for the given symptoms
6. Remote computer monitoring

### **Finance domain**

7. ATM system
8. Stock maintenance

### **Human Resource management**

9. Quiz System
10. E-mail Client system

### **OUTCOMES:**

At the end of the course, the student should be able to:

- Use open source case tools to develop software.
- Analyze and design software requirements in efficient manner.
- Ability to translate end-user requirements into system and software requirements
- Ability to generate a high-level design of the system from the software requirements
- Get experience and/or awareness of testing problems and will be able to develop a simple testing report.

### **LIST OF EQUIPMENTS FOR A BATCH OF 25 STUDENTS:**

#### **SOFTWARE:**

Argo UML / StarUML / UMLGraph / Topcased or Equivalent.

#### **HARDWARE:**

Standalone desktops - 25 Nos. (or) Server supporting 25 terminals or more.

### **PROGRAM EDUCATIONAL OBJECTIVES (PEOs)**

1. To afford the necessary background in the field of Artificial Intelligence and data Science to deal with engineering problems to excel as engineering professionals in industries.
2. To improve the qualities like creativity, leadership, teamwork and skill thus contributing towards the growth and development of society.
3. To develop ability among students towards innovation and entrepreneurship that caters to the needs of Industry and society.
4. To inculcate and attitude for life-long learning process through the use of Artificial Intelligence and Data Science sources.
5. To prepare them to be innovative and ethical leaders, both in their chosen profession and in other activities.

### **PROGRAMME OUTCOMES (POs):**

After going through the four years of study, Bachelor of Technology in Artificial Intelligence and Data Science Graduates will exhibit ability to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

## **PROGRAM SPECIFIC OUTCOMES (PSOs)**

After the completion of Bachelor of Technology in Artificial Intelligence and Data Science programme the student will have following Program specific outcomes.

1. Design and develop secured database applications with data analytical approaches of data preprocessing, optimization, visualization techniques and maintenance using state of the art methodologies based on ethical values.
2. Design and develop intelligent systems using computational principles, methods and systems for extracting knowledge from data to solve real time problems using advanced technologies and tools.
3. Design, plan and setting up the network that is helpful for contemporary business environments using latest software and hardware.
4. Planning and defining test activities by preparing test cases that can predict and correct errors ensuring a socially transformed product catering all technological needs.

**COURSE OUTCOMES:**

**Course Name: 1922507 – Software Engineering and Management Laboratory**

**Year of study: 2023 –2024**

1922507.1	Perform OO analysis and design for a given problem specification.
1922507.2	Identify and map basic software requirements in UML mapping.
1922507.3	Use the UML analysis and design diagrams.
1922507.4	Improve the software quality using design patterns and to explain the rationale behind applying specific design patterns.
1922507.5	Create code from design.

**CO-PO Matrix:**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
1922507.1	3	-	-	-	-	-	-	-	-	-	-	-
1922507.2	-	-	3	-	3	-	-	-	-	-	-	-
1922507.3	-	3	-	-	-	-	-	-	-	-	-	-
1922507.4	-	-	-	2	3	-	-	-	-	-	-	-
1922507.5	3	-	3	2	3	-	-	-	-	-	-	-

**Justification:**

Course Outcome	Program Outcome	Value	Justification
1922507.1	PO1	3	Applying the basic engineering knowledge to perform the analysis of the problem domain to get solution.
1922507.2	PO3	3	Describing the Ideas and key features for UML Mapping. Identifying the approach for mapping requirements to design solutions for complex problems.
	PO5	3	Selection and application of appropriate techniques and tools for mapping the requirements
1922507.3	PO2	3	Identify the problem domain and perform analysis using UML and appropriate design diagram to find solution.

1922507.4	PO4	2	Deriving the appropriate pattern for designing the system
	PO5	3	Analyzing the design pattern and implementing the solution using modern tools.
1922507.5	PO1	3	Apply the knowledge of mathematics, engineering fundamentals, to the solution of complex engineering problems using the UML analysis and design diagrams
	PO3	3	Design solutions for complex engineering problems and design system components using the UML design diagrams.
	PO4	2	Identify, formulate, review research literature, and analyze complex engineering problems to provide valid solutions.
	PO5	3	Select and apply appropriate IT tools for predicting, analyzing and designing solutions for complex engineering activities.

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
1922507	3	3	3	2	3	-	-	-	-	-	-	-

CO-PSO Matrix:

CO	PSO1	PSO2	PSO3	PSO4
1922507.1	-	2	-	-
1922507.2	-	2	-	-
1922507.3	-	1	-	-
1922507.4	-	-	-	3
1922507.5	-	1	-	-

**Justification:**

Course Outcome	Program Specific Outcome	Value	Justification
1922507.1	PSO 2	2	Solutions of the existing system are used for problem solving and designing.
1922507.2	PSO 2	2	Requirements are identified and mapped using UML diagrams.
1922507.3	PSO 2	1	Developing the solutions for modern business environment using design diagrams and tools
1922507.4	PSO 4	3	Implementing the solution using appropriate design pattern and defining the test activities to ensure the product meets the requirements and needs.
1922507.5	PSO 2	1	Developing code from design diagram for implementation.

CO	PSO 1	PSO 2	PSO 3	PSO 4
1922507	-	2	-	3

**ASSESSMENT METHOD**

<b>MARK SPLIT UP</b>	
<b>AIM &amp; PRE LAB VIVA-QUESTIONS</b>	<b>20</b>
<b>OBSERVATION</b>	<b>30</b>
<b>CONDUCTION &amp; EXECUTION</b>	<b>30</b>
<b>OUTPUT &amp; RESULT</b>	<b>10</b>
<b>POST LAB VIVA QUESTIONS</b>	<b>10</b>
<b>TOTAL</b>	<b>100</b>



CHAPTER NO	TITLE
1.	SOFTWARE ENGINEERING PHASES
	A. SOFTWARE DEVELOPMENT ACTIVITIES
	I. ANALYSIS PHASE II. DESIGN PHASE III. IMPLEMENTATION/CODING PHASE IV. TESTING V. DEPLOYMENT VI. MAINTENANCE
2.	INTRODUCTION TO UML (UNIFIED MODELING LANGUAGE)
	A. NOTATION ELEMENTS B. USE CASE DEFINITION C. CLASS DIAGRAM D. ACTIVITY DIAGRAM E. SEQUENCE DIAGRAM F. COLLABORATION DIAGRAM
3.	EXPERIMENTS
	A. COURSE REGISTRATION SYSTEM B. STUDENT MARKS ANALYSING SYSTEM C. ONLINE TICKET RESERVATION SYSTEMS D. PLATFORM ASSIGNMENT SYSTEM FOR THE TRAINS IN A RAILWAY STATION E. EXPERT SYSTEM TO PRESCRIBE THE MEDICINES FOR THE GIVEN SYMPTOMS F. REMOTE COMPUTER MONITORING G. ATM SYSTEM H. STOCK MAINTENANCE I. QUIZ SYSTEM J. E-MAIL CLIENT SYSTEM
4.	TOPIC BEYOND THE SYLLABUS
	A. JDBC CONNECTIVITY B. CREATING JDBC APPLICATION C. JDBC PROGRAM

**TABLE OF CONTENTS**

<b>EXP NO.</b>	<b>NAME OF THE EXPERIMENT</b>
1.	Course Registration System
2.	Student marks analyzing System
3.	Online ticket reservation System
4.	Platform assignment system for the trains in a railway station
5.	Expert system to prescribe medicines for the given symptoms
6.	Remote computer monitoring
7.	ATM system
8.	Stock maintenance
9.	Quiz system
10.	E-mail client system
	Additional Viva Voce Questions

## CHAPTER 1: SOFTWARE ENGINEERING PHASES

**Scope of Software Engineering:** Software engineering is a discipline whose aim is the production of fault-free software that is delivered on time, within budget, and satisfies the user's needs. Software engineering is about teams. The problems to solve are so complex or large, that a single developer cannot solve them anymore. Software engineering is also about communication. Teams do not consist only of developers, but also of testers, architects, system engineers, customer, project managers, etc.

Software projects can be so large that we have to do careful planning. Implementation is no longer just writing code, but it is also following guidelines, writing documentation and also writing unit tests. But unit tests alone are not enough. The different pieces have to fit together. And we have to be able to spot problematic areas using metrics. They tell us if our code follows certain standards. Once we are finished coding, that does not mean that we are finished with the project: for large projects maintaining software can keep many people busy for a long time. Since there are so many factors influencing the success or failure of a project.

We also need to learn a little about project management and its pitfalls, but especially what makes projects successful. There are four fundamental phases in most, if not all, software engineering methodologies. These phases are analysis, design, implementation, and testing. These phases address what is to be built, how it will be built, building it, and making it high quality.

### 1.1. SOFTWARE DEVELOPMENT ACTIVITIES

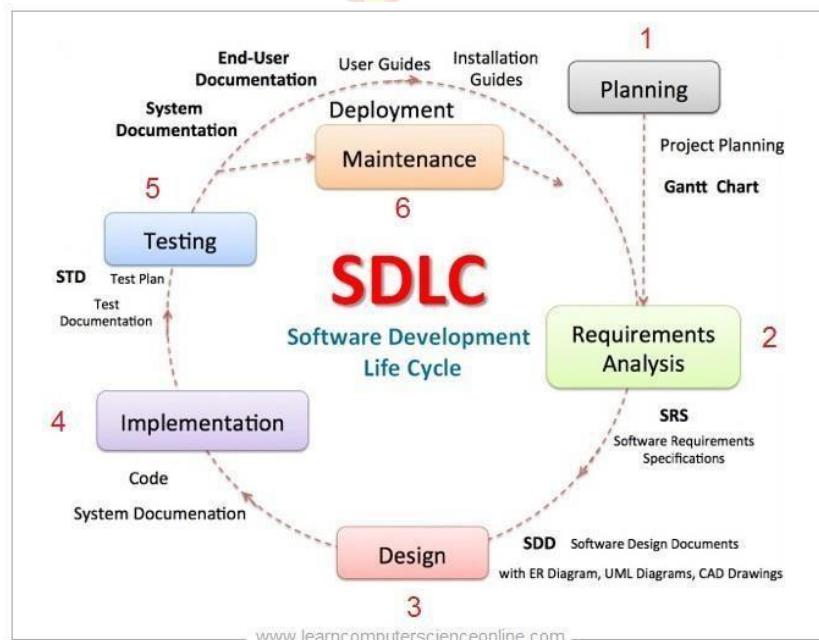


Fig 1.1. Software Development Life Cycle

Gathering Requirements, Software Design, Coding, Testing, Documentation & Software Maintenance.

## I. ANALYSIS PHASE

The analysis phase defines the requirements of the system, independent of how these requirements will be accomplished. This phase defines the problem that the customer is trying to solve. The deliverable result at the end of this phase is a requirement document. Ideally, this document states in a clear and precise fashion what is to be built. This analysis represents the “what” phase. The requirement document tries to capture the requirements from the customer’s perspective by defining goals and interactions at a level removed from the implementation details.

The requirement document may be expressed in a formal language based on mathematical logic. Traditionally, the requirement document is written in English or another written language. The requirement document does not specify the architectural or implementation details, but specifies information at the higher level of description. The problem statement, the customer’s expectations, and the criteria for success are examples of high-level descriptions. There is a fuzzy line between high-level descriptions and low-level details.

Sometimes, if an exact engineering detail needs to be specified, this detail will also appear in the requirement document. This is the exception and should not be the rule. These exceptions occur for many reasons including maintaining the consistency with other established systems, availability of particular options, customer’s demands, and to establish, at the requirement level, a particular architecture vision. An example of a low-level detail that might appear in the requirement document is the usage of a particular vendor’s product line, or the usage of some accepted computer industry standard, or a constraint on the image size of the application behaviors.

The deliverable design document is the architecture. The design document describes a plan to implement the requirements. This phase represents the “how” phase. Details on computer programming languages and environments, machines, packages, application architecture, distributed architecture layering, memory size, platform, algorithms, data structures, global type definitions, interfaces, and many other engineering details are established.

## II. DESIGN PHASE

The result of the software requirements analysis (SRA) usually is a specification. The design helps us turning this specification into a working system. As we have seen there are different kinds of software designs.

- Architectural Design: the process of defining a collection of hardware and software components and their interfaces to establish the framework for the development of a computer system.

- Detailed Design: the process of refining and expanding the preliminary design of a system or component to the extent that the design is sufficiently complete to begin implementation.
- Functional Design: the process of defining the working relationships among the components of a system.
- Preliminary Design: the process of analyzing design alternatives and defining the architecture, components, interfaces, and timing/sizing estimates for a system or components.

There are many aspects to consider in the design of a piece of software. The importance of each should reflect the goals the software is trying to achieve. Some of these aspects are:

- Compatibility - The software is able to operate with other products that are designed for interoperability with another product. For example, a piece of software may be backward-compatible with an older version of itself.
- Extensibility - New capabilities can be added to the software without major changes to the underlying architecture.
- Fault-tolerance - The software is resistant to and able to recover from component failure.
- Maintainability - The software can be restored to a specified condition within a specified period of time. For example, antivirus software may include the ability to periodically receive virus definition updates in order to maintain the software's effectiveness.
- Modularity - the resulting software comprises well defined, independent components. That leads to better maintainability. The components could be then implemented and tested in isolation before being integrated to form a desired software system. This allows division of work in a software development project.
- Packaging - Printed material such as the box and manuals should match the style designated for the target market and should enhance usability. All compatibility information should be visible on the outside of the package. All components required for use should be included in the package or specified as a requirement on the outside of the package.
- Reliability - The software is able to perform a required function under stated conditions for a specified period of time.
- Reusability - the software is able to add further features and modification with slight or no modification.
- Robustness - The software is able to operate under stress or tolerate unpredictable or invalid input. For example, it can be designed with resilience to low memory conditions.
- Security - The software is able to withstand hostile acts and influences.
- Usability - The software user interface must be usable for its target user/audience. Default values for the parameters must be chosen so that they are a good choice for the majority of the users.

### III. IMPLEMENTATION / CODING PHASE

On receiving system design documents, the work is divided in modules/units and actual coding is started. Since, in this phase the code is produced so it is the main focus for the developer. This is the longest phase of the software development life cycle. Implementation is the part of the process where software engineers actually program the code for the project. The implementation phase deals with issues of quality, performance, baselines, libraries, and debugging. The end deliverable is the product itself.

### IV. TESTING

After the code is developed it is tested against the requirements to make sure that the product is actually solving the needs addressed and gathered during the requirements phase. During this phase unit testing, integration testing, system testing, acceptance testing are done.

### V. DEPLOYMENT

After successful testing the product is delivered / deployed to the customer for their use.

### VI. MAINTENANCE

Once when the customers starts using the developed system then the actual problems comes up and needs to be solved from time to time. This process where the care is taken for the developed product is known as maintenance.

## CHAPTER 2. INTRODUCTION TO UML (UNIFIED MODELING LANGUAGE)

The UML is a language for specifying, constructing, visualizing, and documenting the software system and its components. The UML is a graphical language with sets of rules and semantics. The rules and semantics of a model are expressed in English in a form known as OCL (Object Constraint Language). OCL uses simple logic for specifying the properties of a system. The UML is not intended to be a visual programming language. However it has a much closer mapping to object-oriented programming languages, so that the best of both can be obtained. The UML is much simpler than other methods preceding it. UML is appropriate for modeling systems, ranging from enterprise information system to distributed web based application and even to real time embedded system. It is a very expressive language addressing all views needed to develop and then to display system even though understand to use. Learning to apply UML effectively starts forming a conceptual mode of languages which requires learning.

Three major language elements:

- o UML basic building blocks
- o Rules that dictate how this building blocks put together
- o Some common mechanism that apply throughout the language. The primary goals in the design of UML are:
  1. Provides users ready to use, expressive visual modeling language as well so they can develop and exchange meaningful models.
  2. Provide extensibility and specialization mechanisms to extend the core concepts.
  3. Be independent of particular programming languages and development processes.
  4. Provide formal basis for understanding the modeling language.
  5. Encourage the growth of the OO tools market.
  6. Support higher-level development concepts.
  7. Integrate best practices and methodologies.

Every complex system is best approached through a small set of nearly independent views of a model. Every model can be expressed at different levels of fidelity. The best models are connected to reality.

The UML defines nine graphical diagrams:

1. Class diagram
2. Use-case diagram
3. Behavior diagram
  - 3.1. Interaction diagram
    - 3.1.1. Sequence diagram
    - 3.1.2. Collaboration diagram
  - 3.2. State chart diagram
  - 3.3. Activity diagram

4. Implementation diagram
  - 4.1 component diagram
  - 4.2 deployment diagram

### 1. UML class diagram:

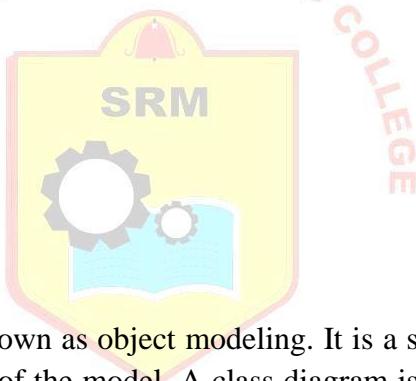
The UML class diagram is also known as object modeling. It is a static analysis diagram. These diagrams show the static structure of the model. A class diagram is a connection of static model elements, such as classes and their relationships, connected as a graph to each other and to their contents.

### 2. Use-case diagram:

The functionality of a system can be described in a number of different use cases, each of which represents a specific flow of events in a system. It is a graph of actors, a set of use-cases enclosed in a boundary, communication, associations between the actors and the use-cases, and generalization among the use-cases.

### 3. Behavior diagram:

It is a dynamic model unlike all the others mentioned before. The objects of an object oriented system are not static and are not easily understood by static diagrams. The behavior of the class's instance (an object) is represented in this diagram. Every use-case of the system has an associated behavior diagram that indicates the behavior of the object. In conjunction with the use-case diagram we may provide a script or interaction diagram to show a time line of events. It consists of sequence and collaboration diagrams



#### 4. Interaction diagram

It is the combination of sequence and collaboration diagram. It is used to depict the flow of events in the system over a timeline. The interaction diagram is a dynamic model which shows how the system behaves during dynamic execution.

#### 5. State chart diagram:

It consists of state, events and activities. State diagrams are a familiar technique to describe the behavior of a system. They describe all of the possible states that a particular object can get into and how the object's state changes as a result of events that reach the object. In most OO techniques, state diagrams are drawn for a single class to show the lifetime behavior of a single object.

#### 6. Activity diagram:

It shows organization and their dependence among the set of components. These diagrams are particularly useful in connection with workflow and in describing behavior that has a lot of parallel processing. An activity is a state of doing something: either a real-world process, or the execution of a software routine.

#### 7. Implementation diagram:

It shows the implementation phase of the systems development, such as the source code structure and the run-time implementation structure. These are relatively simple high level diagrams compared to the others seen so far. They are of two sub diagrams, the component diagram and the deployment diagram.

#### 8. Component diagram:

These are organizational parts of a UML model. These are boxes to which a model can be decomposed. They show the structure of the code itself. They model the physical components such as source code, user interface in a design. It is similar to the concept of packages.

#### 9. Deployment diagram:

The deployment diagram shows the structure of the runtime system. It shows the configuration of runtime processing elements and the software components that live in them. They are usually used in conjunction with deployment diagrams to show how physical modules of code are distributed on the system.

## A. NOTATION ELEMENTS:

These are explanatory parts of UML model. They are boxes which may apply to describe and remark about any element in the model. They provide the information for understanding the necessary details of the diagrams.

Relations in the UML:

These are four kinds of relationships used in an UML diagram, they are:

Dependency

Association

Generalization

Realization

Dependency:

It is a semantic relationship between two things in which a change one thing affects the semantics of other things. Graphically a dependency is represented by a non-continuous line.

Association:

It is a structural relationship that describes asset of links. A link is being connected among objects. Graphically association is represented as a solid line possibly including label.

Generalization:

It is a specialized relationship in which the specialized elements are substitutable for object of the generalized element. Graphically it is a solid line with hollow arrow head parent.

Realization:

It is a semantic relation between classifiers. Graphically it is represented as a cross between generalization and dependency relationship.

Where UML can be used:

UML is not limited to modeling software. In fact it is expressive to model non-software such as to show in structure and behavior of health case system and to design the hardware of the system.

Conceptual model be UML:

UML you need to form the conceptual model of UML. This requires three major elements:

- UML basic building blocks.
- Rules that dictate how this building blocks are put together.

- Some common mechanism that apply throughout the language.

Once you have grasped these ideas, you may be able to read UML create some basic ones. As you gain more experience in applying conceptual model using more advanced features of this language.

**Building blocks of the UML:**

The vocabulary of UML encompasses these kinds of building blocks.

#### B. USE CASE DEFINITION:

**Description:**

A use case is a set of scenarios tied together by a common user goal. A use case is a behavioral diagram that shows a set of use case actions and their relationships.

**Purpose:**

The purpose of use case is login and exchange messages between sender and receiver (Email client).

**Main flow:**

First, the sender gives his id and enters his login. Now, he enters the message to the receiver id.

**Alternate flow:**

If the username and id by the sender or receiver is not valid, the administrator will not allow entering and “Invalid password” message is displayed.

**Pre-condition:**

A person has to register himself to obtain a login ID.

**Post-condition:**

The user is not allowed to enter if the password or user name is not valid.

#### C. CLASS DIAGRAM:

**Description:**

A class diagram describes the type of objects in system and various kinds of relationships that exists among them. Class diagrams and collaboration diagrams are alternate representations of object models. During analysis, we use class diagram to show roles and responsibilities of entities that provide email client system behaviors design. We use to capture the structure of classes that form the email client system architecture.

The classes used in system are:

1. user
2. login

A class diagram is represented as:

```
<<Class name>>
<<Attribute 1>>
<<Attribute n>>
<<Operation ()>>
```

Relationship used:

A change in one element affects the other

Generalization:

It is a kind of relationship

State chart:

Description:

The state chart diagram made the dynamic behavior of individual classes. State chart shows the sequences of states that an object goes through events and state transitions. A state chart contains one state „start“ and multiple „end“ states.

The important objectives are:

Decision:

It represents a specific location state chart diagram where the work flow may branch based upon guard conditions.

Synchronization:

It gives a simultaneous workflow in a state chart diagram. They visually define forks and joins representing parallel workflow.

Forks and joins:

A fork construct is used to model a single flow of control.

Every work must be followed by a corresponding join.

Joints have two or more flow that unit into a single flow.

State:

A state is a condition or situation during a life of an object in which it satisfies condition or waits for some events.

Transition:

It is a relationship between two activities and between states and activities.

Start state:

A start state shows the beginning of a workflow or beginning of a state machine on a state chart diagram.

End state:

It is a final or terminal state.

#### D. ACTIVITY DIAGRAM

Description:

Activity diagram provides a way to model the workflow of a development process. We can also model this code specific information such as class operation using activity diagram.

Activity diagrams can model different types of diagrams. There are various tools involved in the activity diagram.

Activity:

An activity represents the performance of a task on duty. It may also represent the execution of a statement in a procedure.

Decision:

A decision represents a condition on situation during the life of an object, which it satisfies some condition or waits for an event.

Start state:

It represents the condition explicitly the beginning of a workflow on an activity.

Object flow:

An object on an activity diagram represents the relationship between activity and object that creates or uses it.

Synchronization:

It enables us to see a simultaneous workflow in an activity.

End state:

An end state represents a final or terminal state on an activity diagram or state chart diagram.

#### E. SEQUENCE DIAGRAM:

Description:

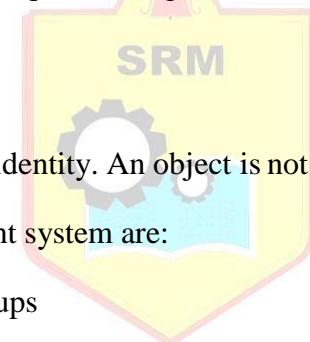
A sequence diagram is a graphical view of scenario that shows object interaction in a time based sequence what happens first what happens next. Sequence diagrams are closely related to collaboration diagram. The main difference between sequence and collaboration diagram is that sequence diagram show time based interaction while collaboration diagram shows objects associated with each other. The sequence diagram for the e-mail client system consists of the following objectives:

Object:

An object has state, behavior and identity. An object is not based is referred to as an instance.

The various objects in e-mail client system are:

User, Website, Login, Groups



Message icon:

A message icon represents the communication between objects indicating that an action will follow. The message icon is the horizontal solid arrow connecting lifelines together.

#### F. COLLABORATION DIAGRAM:

Description:

Collaboration diagram and sequence diagrams are alternate representations of an interaction. A collaboration diagram is an interaction diagram that shows the order of messages that implement an operation or a transaction. Collaboration diagram is an interaction diagram that shows the order of messages that implement an operation or a transaction. Collaboration diagram shows object s, their links and their messages. They can also contain simple class instances and class utility instances. During, analysis indicates the semantics of the primary and secondary interactions. Design, shows the semantics of mechanisms in the logical design of system.

**Ex. No: 1a****PASSPORT AUTOMATION SYSTEM****AIM:**

To write the problem statement for the Passport automation system.

**PROBLEM STATEMENT:**

Passport Automation System is used to process the passport application form and dispatch passport to the applicants. This system adopts a comprehensive approach to minimize the manual work and schedule resources, time in an effective manner. The core of the system is to get the online application form (with details such as personal information, address details, etc) filled by the applicant whose testimonials are verified for its genuineness by the system with respect to already existing information in the database. This forms the first and foremost step in the processing of passport application.

After the first round of verification done by the system, the information is in turn forwarded to the regional administrator's office. The system forwards the necessary details to the police for its separate verification, whose report is then presented to the administrator. After all the necessary criteria have been met, the original information is added to the database and the passport is sent to the applicant. The administrator will be provided with the option to display the current status of the application to the applicant, which they can view in their online interface. The system also provides the user with the facility to apply for the renewal of expired passports and to make changes to current details in the passport.

**RESULT:**

Thus the problem statement for the Passport automation system has been done successfully.

**Ex. No: 1b****SOFTWARE REQUIREMENT SPECIFICATION****AIM:**

To write the software requirement specification for Passport Automation System.

**1. Introduction:**

Passport Automation System is an interface between the applicant and the authority responsible for the issue of passport. It aims at improving the efficiency in the issue of passport and reduce the complexities involved in it to the maximum extent possible.

**1.1 Purpose:**

If the entire process of ‘issue of passport’ is done in a manual manner then it would take several months for the passport to reach the applicant. Considering the fact that the number of applicants for passport is increasing every year, an automated system becomes essential to meet the demand. So this system uses several programming and database techniques to elucidate the work involved in this process. As it is a matter of national security, the system has to carefully verify to satisfy it.

**1.2 Scope:**

This system provides an online interface for the applicant to submit the application form filled with details. It also provides payment facilities for the user to pay online. The details given by the applicant is cross-checked with already existing information in other databases. The system also provides the user with options to check the status of his application, for renewal of expired passports and make changes to existing information. The passport officer can update the status of the application, view the verification report submitted by the police and access the information submitted by the applicant for verification. The police can submit the verification report using this system.

**1.3 Definitions, Acronyms and Abbreviations**

- Administrator/Passport Officer - refers to the authority who is vested with the privilege to manage the entire system. It can be a higher official in the Regional Passport Office of the Ministry of External Affairs
- Applicant – one who applies for the passport
- PAS – Passport Automation System
- SRS – Software Requirements Specification
- LAN - Local Area Network
- GUI – Graphical User Interface
- 

**1.4 References**

IEEE Software Requirements Specification format from <http://www.ieee.org>

## **1.5 Overview:**

SRS includes three chapters.

The 1<sup>st</sup> chapter is the Introduction specifying the purpose and scope of the PAS system.

The 2<sup>nd</sup> chapter is the Overall Description which includes the product perspective, product functions, the user characteristics and the constraints.

The 3<sup>rd</sup> chapter is the Specific Requirements which includes the software product features, the functional requirements, the product requirements for safety and performance, and the system attributes.

## **2. Overall Description**

### **1.1 Product Perspective**

The PAS acts as an interface between the applicant and the passport officer. It tries to make the interface as simple as possible at the same time not risking the security of data stored. The processing of the application is automated and this reduces the time duration in which the user receives the passport.

### **1.2 System Interfaces**

The PAS will use a relational database as backend for data storage. The client system should be able to share the data available in the database through the network connection.

#### **1.2.1 User Interfaces**

The PAS will provide an easy to use GUI as part of the working environment for the applicant and the passport officer to interact with the system.

#### **1.2.2 Hardware Interfaces**

A scanner is used to scan and upload the documents needed for verification.

#### **1.2.3 Software Interfaces**

The System uses ODBC drive version 6.0 to connect and control the database.

#### **1.2.4 Communication Interfaces**

LAN is used for communication among the client and server system.

#### **1.2.5 Memory Constraints**

A minimum of 64MB RAM is needed for the client system.

#### **1.2.6 Operations**

The applicant is required to fill and submit the application form and also scan and upload documents needed for the verification process.

The police must verify the details forwarded by the system and submit the verification report to the passport officer.

The passport officer must verify and approve the dispatch of passport for the applicants based on the verification report submitted by the police. He must also update the status of the application in each stage of the process.

### **1.3 Product Functions**

- Secure registration of information by the applicants
- Display of passport application status

- SMS and e-mail updates to the applicants from the passport office
- Passport officer can generate reports from the information and add eligible application to the database

#### **1.4 User Characteristics**

Applicant – person who applies for the passport and submits information. He need not be a person with special skills and training. He is the average common man.

Passport Officer – person with the privilege to update the application status, approve the issue of passport and access the applicant information. He should have the skills and necessary training to manage the database and the system.

Police – person who verifies the details on receiving intimation from the passport officer. He should have basic computer knowledge to communicate via the PAS to submit the verification report.

#### **1.5 Constraints**

- The applicant requires a computer with a scanner to upload documents for verification
- Security is of much importance as there is always a chance of intrusion in the web world
- Should support many users simultaneously
- Must display appropriate error messages while filling the application form

#### **1.6 Assumptions and Dependencies**

If the applicant is under 18 years of age, the details of the parents passport information can be used to obtain the passport.

### **2. Specific Requirements**

#### **2.1 Functional Requirements**

##### **2.1.1 Apply**

**Input:** The Applicant fills in the application form available in the online interface giving the personal details and family information, and submits the application form.

**Process:** The details furnished by the applicant are stored in the database

**Output:** The database is updated with the new applicant details and appropriate message is sent to the applicant

##### **2.1.2 Check Status**

**Input:** The applicant enters the application number and login password

**Process:** The applicant password is verified and if correct, the database table containing the status information is searched with the application number provided by the user.

**Output:** The status message from the table is displayed for the applicant on the screen

##### **2.1.3 Renewal**

**Input:** The passport holder gives the passport number, passport holder's name and the expiry date for the validity.

**Process:** The given details are verified in reference with the existing information in the database. If found to be valid, the passport is renewed and the database updated.

**Output:** The renewed passport is dispatched to the user.

#### 2.1.4 Update Status

**Input:** The passport officer gives the application number and the status for that application

**Process:** The database is updated with the new status message

**Output:** The updated database and the message to the applicant

#### 2.1.5 Verification

**Input:** The details furnished by the applicant in the application form

**Process:** The details are cross checked with information in already existing databases

**Output:** The verification report

#### 2.1.6 Online Payment

**Input:** The account details of the applicant

**Process:** The service of an external module provided by the bank is used to process the payment.

**Output:** The updated status and message to the applicant

### 2.2 Performance Requirements

The system should have a very short response time while displaying forms and messages.

As the system handles personal information and payment details, it should ensure privacy to avoid malpractices.

The system should recover from crashes without any serious loss of data and should always maintain a backup of important information.

### 2.3 Design Constraints

The user interface should be an easy to use GUI that contains forms with hints to fill in the form in correct format and display error messages when not in required format.

The database should be accessible only to the passport officer.

The system should easily adapt to future changes and hence must be modular.

### 2.4 Software System Attributes

#### 2.4.1 Reliability

To make the PAS system reliable, we must ensure that duplication is avoided

#### 2.4.2 Availability

The system should be available to the applicant to check his status at any stage of the process, once it has been submitted. It must support more than one user at a time.

#### 2.4.3 Security

- To avoid unauthorized access, the applicant, passport officer and police must be given unique login and password protected
- The police should be able to access the information only when notified by the passport officer through the system
- All the documents uploaded by the applicant should be accessible only to the system administrator
- A log of all user logins to the system must be maintained

#### **2.4.4 Maintainability**

- The PAS system should be frequently updated with any changes that may have occurred in the rules or application interface.
- For easy updation and maintenance, the system should be modular and the user interface simple.

#### **2.4.5 Portability**

The PAS system should be supported by any operating system and should occupy very less memory space.

#### **2.5 Logical Database Requirements**

Several tables are used in the database to store various information.

They are:

Register – used to store the details of users registered with the system. It contains the fields: Name, Date of Birth, Username, and Password.

Personal – used to store the details filled in by the applicant in the application form. It contains the fields: Name, Date of Birth, Sex, Address, City, State, Qualification, Profession, Mother's name, Father's Name, Marital status, and Spouse's Name.

Report - contains the verification reports submitted by the police for the reference of the passport officer. It contains the field: Application number, Recommendation.

Renewal Requests – contains the information regarding the renewal requests submitted by passport holders. It contains: Passport number, Passport holder's name and Expiry date.

Status - contains the application status. It contains the fields: Application number, Status and Passport number (if status is approved).

#### **Pre lab Questions:**

1. What Is A Software Requirements Specification?
2. What Is Srs In Project?
3. What Is Requirement Specifications Of The System?
4. What Is Requirement Gathering?
5. How Many Types Of Software Requirements Are There? Specify Them.

#### **Post Lab Questions:**

1. What Are User Interface Requirements?
2. What Are The Contents Of An Effective SRS Document?
3. What Are The Software Requirement Validations?
4. What are Functional and Non Functional Requirements?
5. What Is SRS In Software Engineering?

#### **RESULT:**

Thus the software requirement specification for Passport Automation System has been done successfully

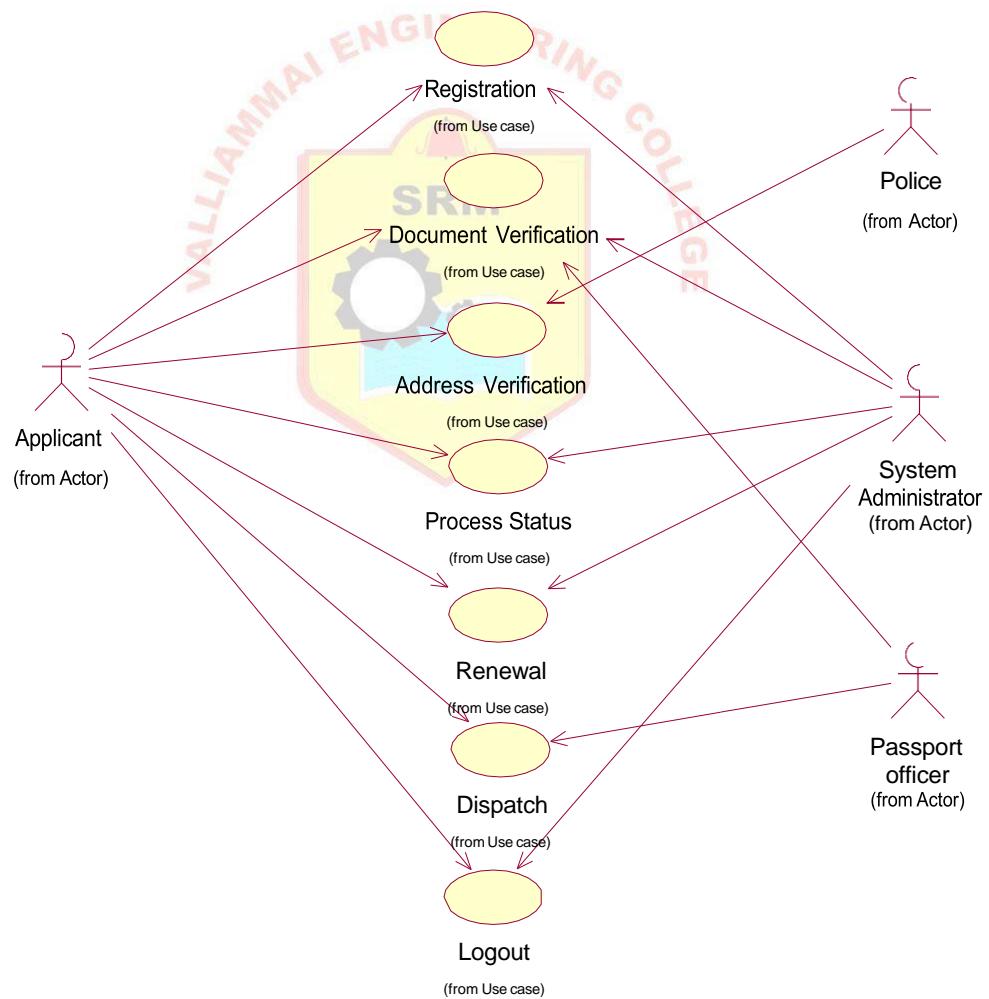
**Ex. No: 1c****USECASE DIAGRAM****AIM:**

To draw use case diagram for Passport Automation System.

**Introduction:**

In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system.

- Scenarios in which your system or application interacts with people, organizations, or external systems
- Goals that your system or application helps those entities (known as actors) achieve
- The scope of your system



**Pre Lab Questions:**

1. Define UML?
2. What is use case diagram?
3. What is use case?
4. List out the notations of use case diagram.
5. Write the differences of Use case diagram and use case.

**Post Lab Questions:**

1. What is actor in usecase diagrams?
2. Can you explain primary and secondary actors?
3. How does a simple usecase look like?
4. Can you explain ‘Extend’ and ‘Include’ in usecases?
5. Define modeling in UML and it advantages.

**RESULT:**

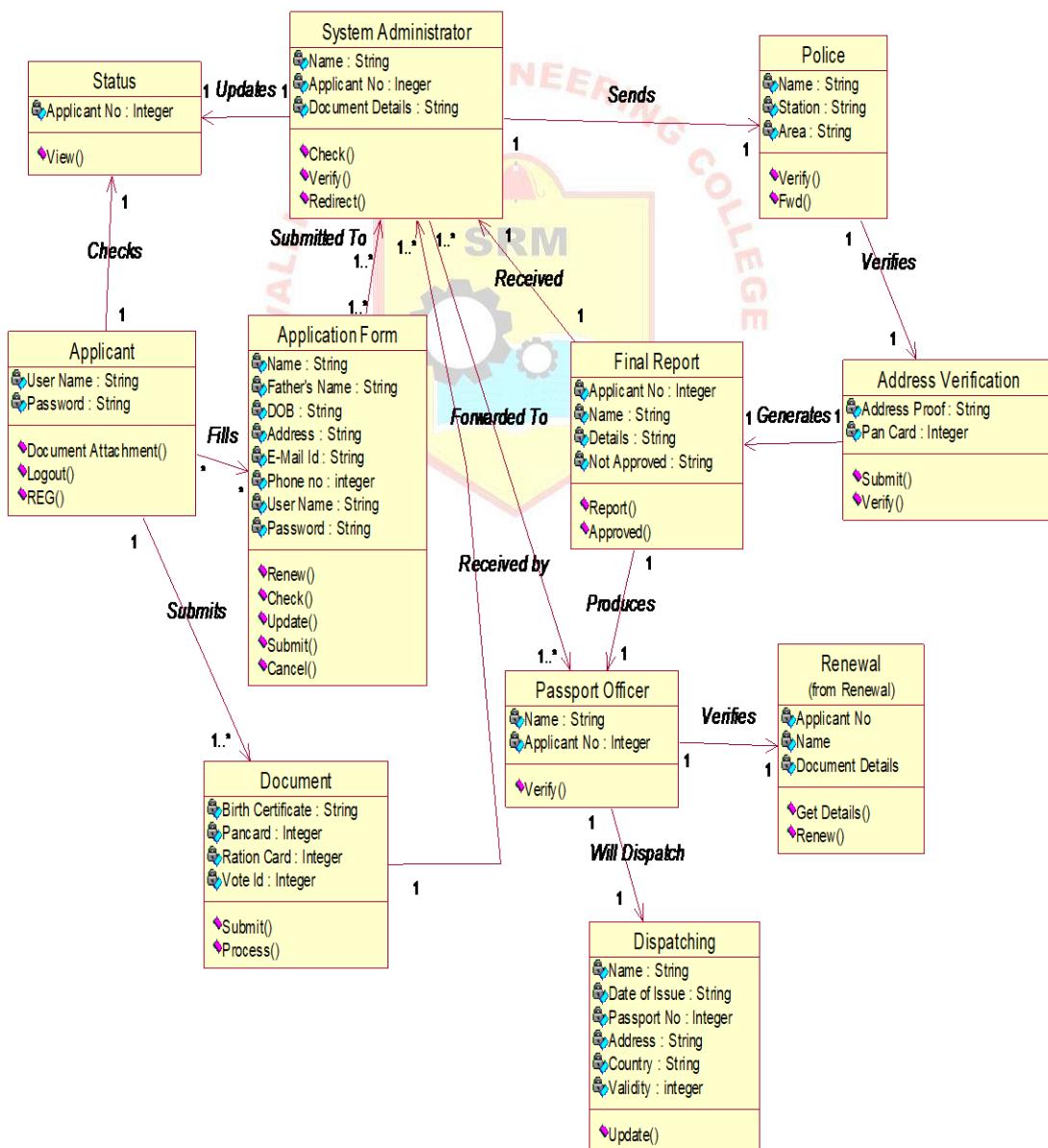
Thus the use case diagram for Passport Automation System has been drawn successfully.

**Ex. No: 1d****CLASS DIAGRAM****AIM:**

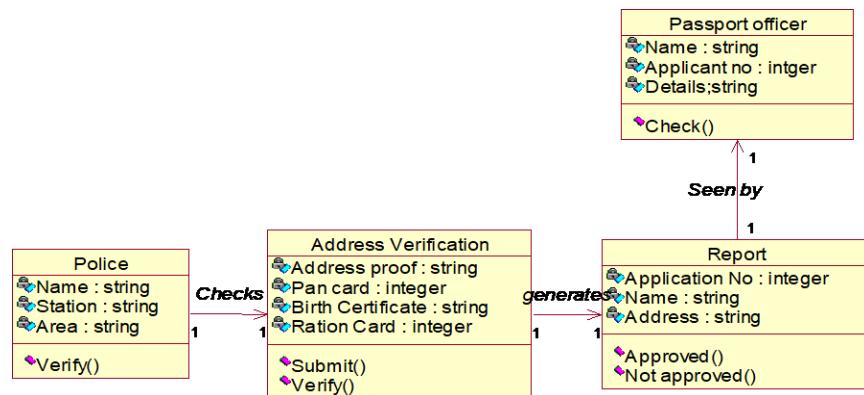
To draw class diagram for Passport Automation System.

**Introduction:**

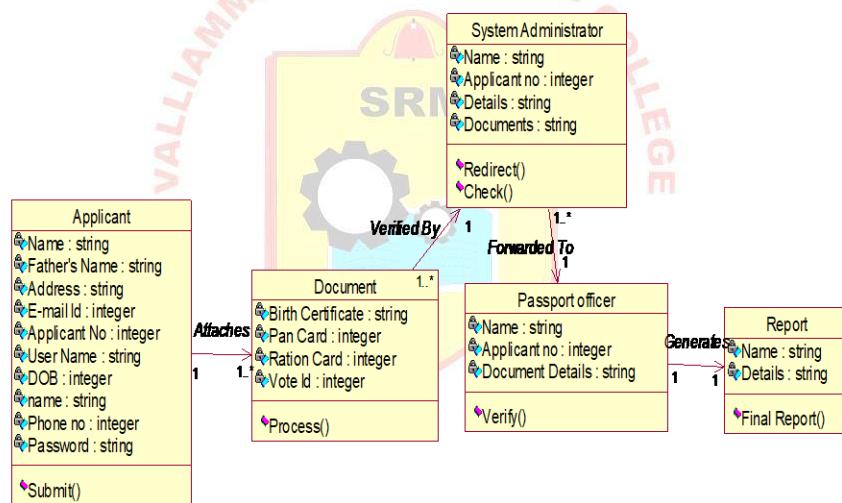
A class diagram models the static structure of a system. It shows relationships between classes, objects, attributes, and operations.

**Overall Class Diagram**

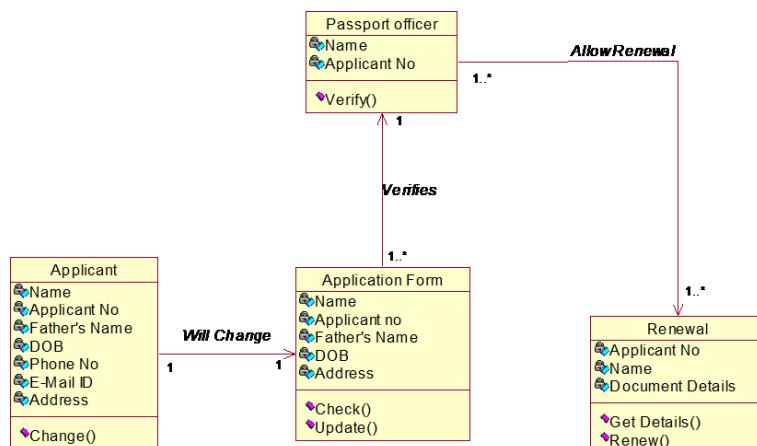
## Address Verification



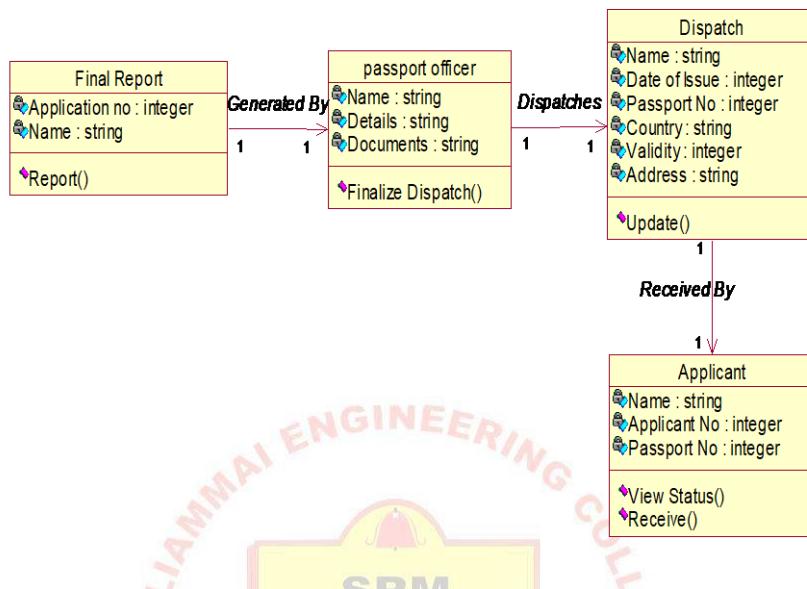
## Document Verification



## Renewal



## Dispatch



### Pre lab Questions:

1. What do class diagrams illustrate?
2. What are the notations and partitions of class diagram?
3. What three things does a UML class define?
4. What are the relationships that can be shown in the class diagram?
5. How a UML Use Case is Diagram different from a traditional flow chart?

### Post Lab Questions:

1. How do we represent private, public and protected in class diagrams?
2. What does associations, aggregation and composition in a class diagram mean?
3. What are the four principal relationships classes participate in?
4. Which illustrates an “is a” relationship between instances of a class.
5. When is it necessary to use a static or dynamic view class diagram?

### RESULT:

Thus the class diagram for Passport Automation System has been drawn successfully.

**Ex. No: 1e****ACTIVITY DIAGRAM****AIM:**

To draw activity diagram for Passport Automation System

**Introduction:**

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

Before drawing an activity diagram, we should identify the following elements –

- Activities
- Association
- Conditions
- Constraints

Activity diagram can be used for –

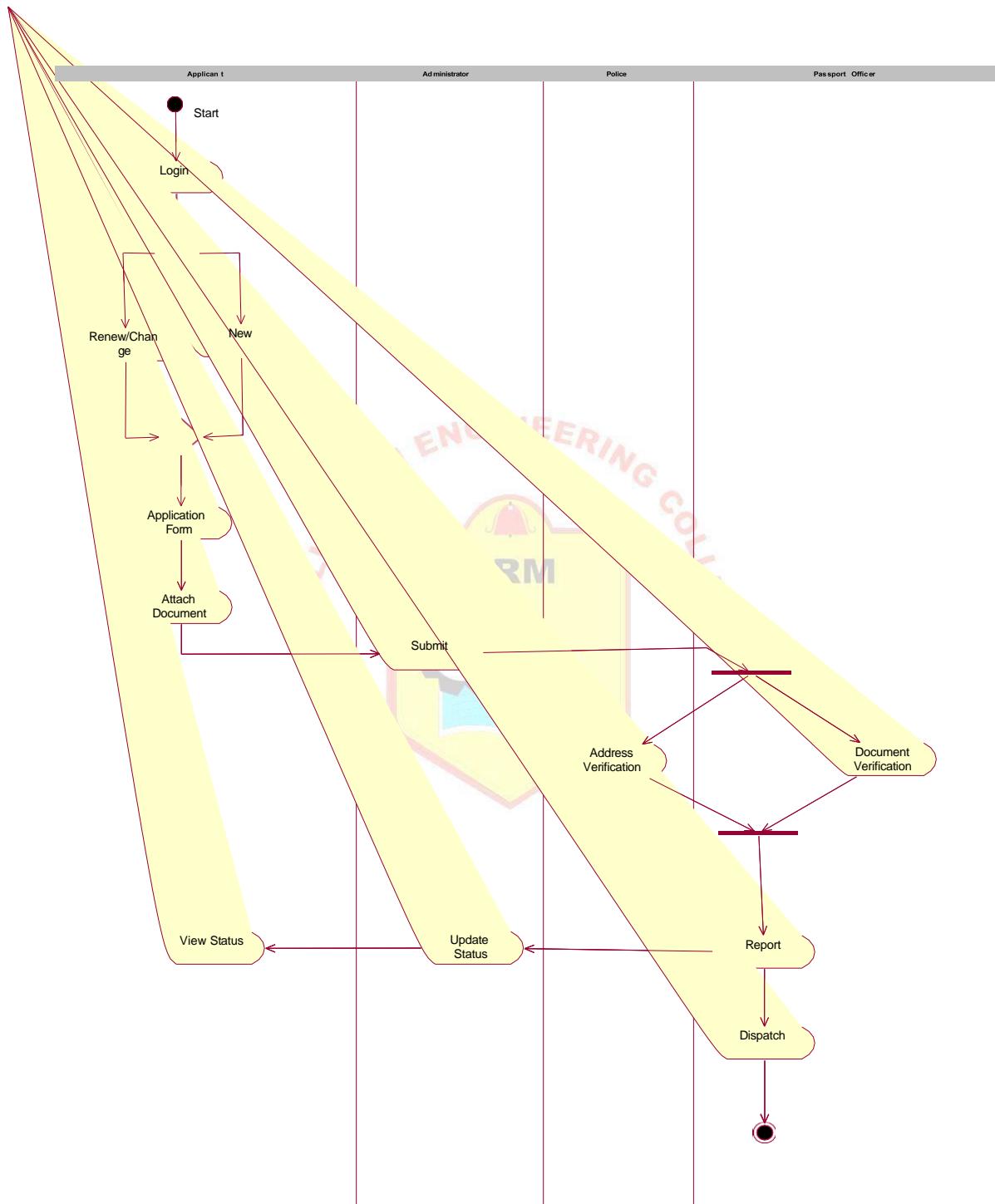
- Modeling work flow by using activities.
- Modeling business requirements.
- High level understanding of the system's functionalities.
- Investigating business requirements at a later stage.

**Pre Lab Questions:**

1. What the purpose of activity diagram?
2. Draw the activity flow of a system.
3. How the activity diagram is used in a system?
4. What are the benefits of activity diagram?
5. What is Conditional logic?

**Post Lab Questions:**

1. What are the components of activity diagram?
2. What is decision node?
3. State the differences of fork and join.
4. When the activity diagram cannot be used?
5. What are the pros and cons of activity diagram?



## RESULT:

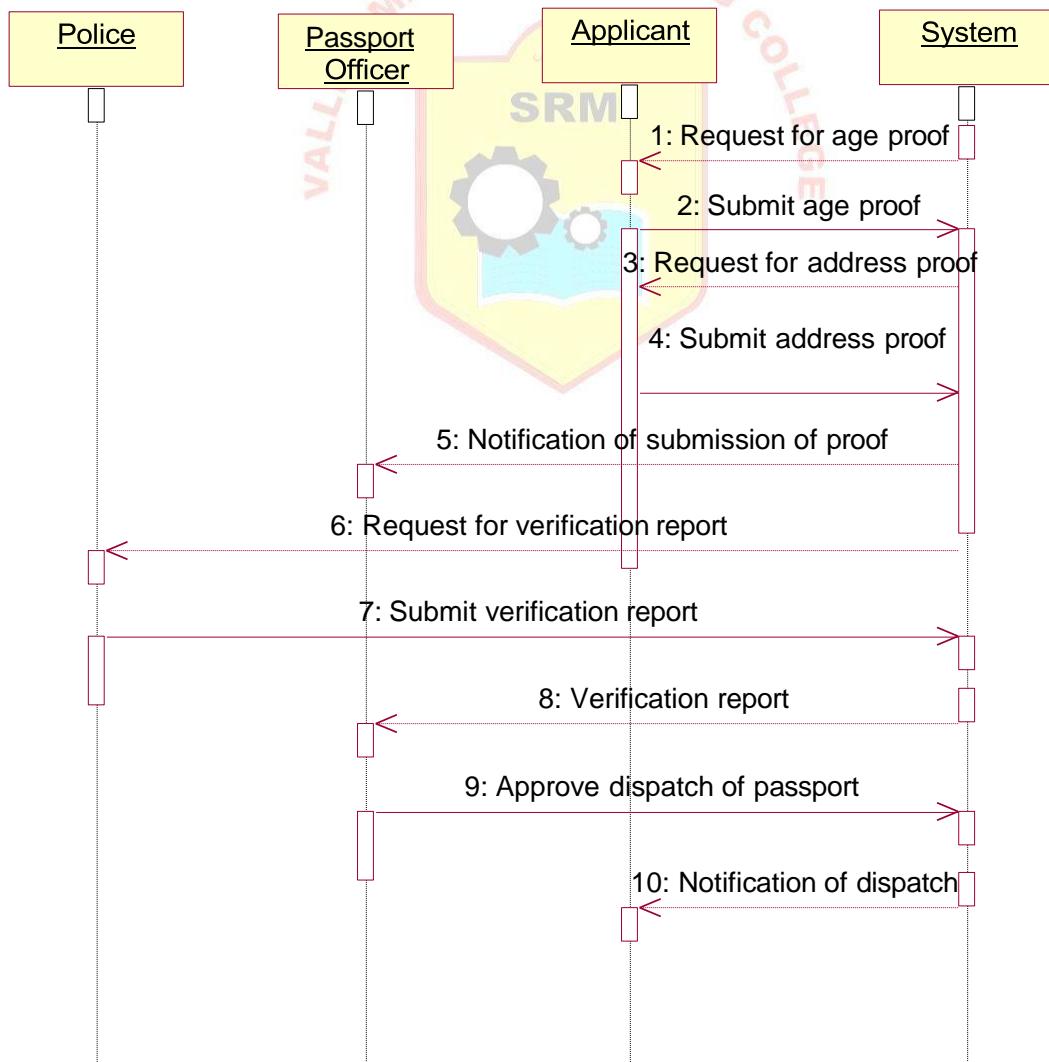
Thus the activity diagram for Passport Automation System has been drawn successfully.

**Ex. No: 1f****INTERACTION DIAGRAM****AIM:**

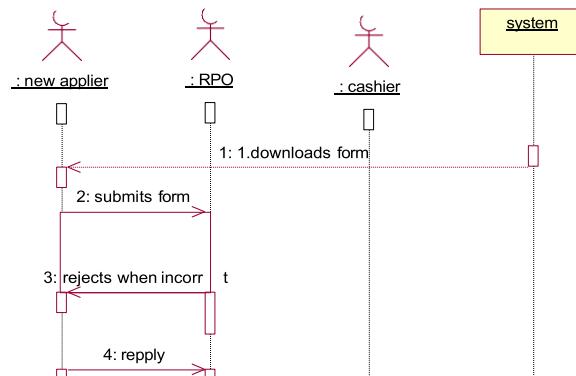
To draw interaction diagram for Passport Automation System

**SEQUENCE DIAGRAM****Introduction:**

Sequence diagrams are sometimes called event diagrams or event scenarios. A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur.

**SEQUENCE DIAGRAM FOR VERIFICATION AND DISPATCH:**

## SEQUENCE DIAGRAM FOR FRESH APPLIER:



### Pre lab questions:

1. What is Interaction diagram?
2. What are the different interaction diagram notations does UML have?
3. What is a sequence diagram?
4. What is a lifeline?
5. What does a message mean?

### Post Lab Questions:

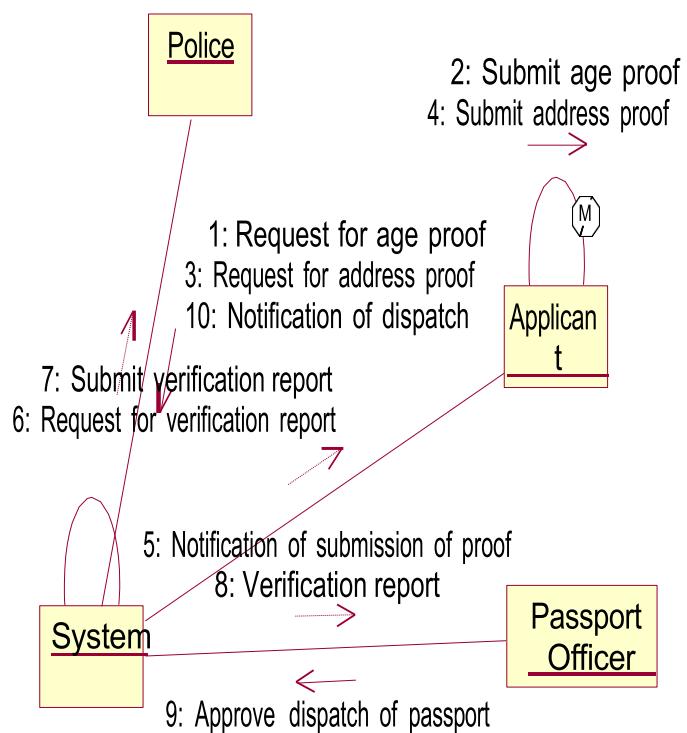
1. What is an interaction fragment?
2. When is the operation executing, suspended and active?
3. What is break fragment?
4. What are the three different types of message arrows?
5. What are the heuristics which sequencing diagram follows?

## COLLABORATION DIAGRAM

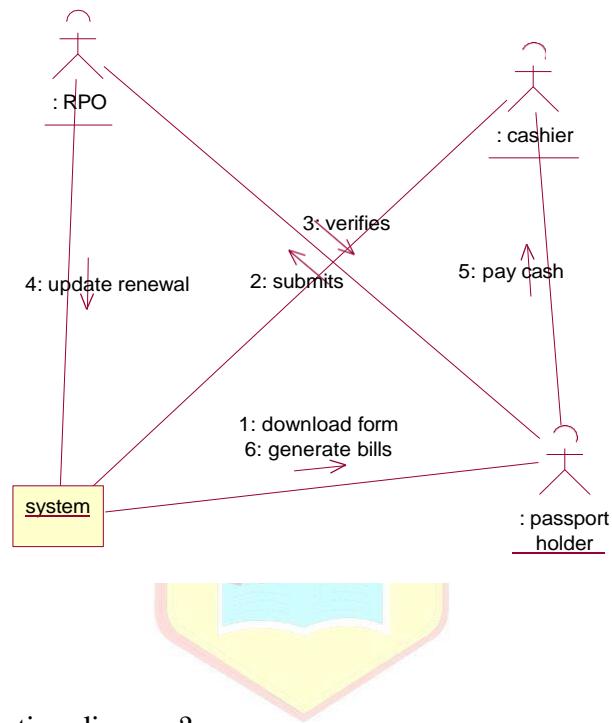
### **Introduction:**

A collaboration diagram, also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). A collaboration diagram often comes in the form of a visual chart that resembles a flow chart.

### **COLLABORATION DIAGRAM FOR VERIFICATION AND DISPATCH:**



### Collaboration diagram for renewal:



### Pre Lab Questions:

1. What is collaboration diagram?
2. What are the symbols used in collaboration diagram?
3. What are the notations used in collaboration diagram?
4. How the messages interact between the objects.
5. How the conditional statements are denoted by?

### Post Lab Questions:

1. What is sequence number?
2. Is it possible to group the sequence number?
3. How the messages are represented by?
4. What is looping?
5. What are the advantages of collaboration diagram?

### RESULT:

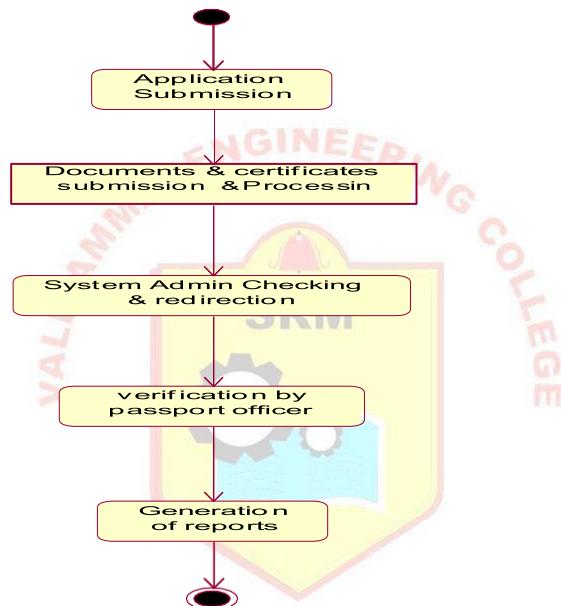
Thus the interaction diagram for Passport Automation System has been drawn successfully.

**Ex. No: 1g****STATE CHART DIAGRAM****AIM:**

To draw state chart diagram for Passport Automation System

**Introduction:**

State diagram describes the behavior of a single object in response to a series of events in a system. This UML diagram models the dynamic flow of control from state to state of a particular object within a system.

**Pre Lab Questions:**

1. How to draw a state chart Diagram?
2. Where to use state chart diagram?
3. What is a state?
4. What is an event?
5. What is composite state?

**Post Lab Questions:**

1. What is self-transition?
2. Give the notation for initial state and final state?
3. What is the difference between state chart diagram and sequence diagram?
4. What are the basic components of state chart diagram?
5. Can we use fork and join in state chart diagram? If yes, how to use?

**RESULT:**

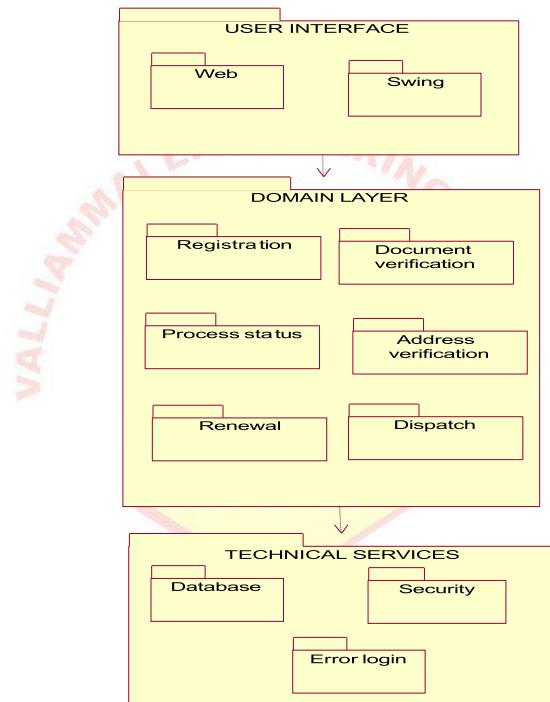
Thus the state chart diagram for Passport Automation System has been drawn successfully.

**Ex. No: 1h****PACKAGE DIAGRAM****AIM:**

To draw package diagram for Passport Automation System.

**Introduction:**

Package diagram is used to simplify complex class diagrams; we can group classes into packages. A package is a collection of logically related UML elements. Packages appear as rectangles with small tabs at the top.

**Pre lab Questions:**

1. What are the three layers of package diagram?
2. How the public elements of a package are accessible?
3. How the packages can merge?
4. What is visibility?
5. Name the elements of package diagram.

**Post lab Questions:**

1. What is the difference between template and package template?
2. How the packages can merge?
3. How the packages can import?
4. What does '+' symbol and '-' symbol represents in package diagram.
5. What is the difference between public and private visibility?

**RESULT:**

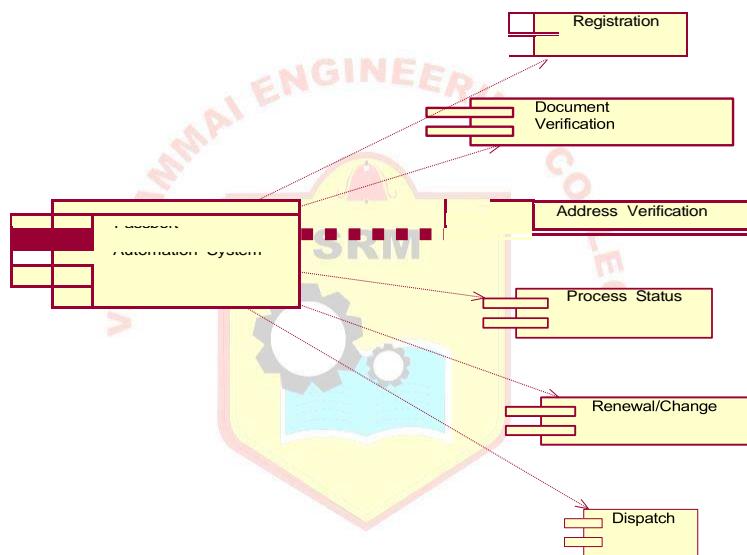
Thus the package diagram for Passport Automation System has been drawn successfully.

**Ex.No:1i****COMPONENT DIAGRAM****AIM:**

To draw component diagram for Passport Automation System.

**Introduction:**

The purpose of a component diagram is to show the relationship between different components in a system. The term "component" refers to a module of classes that represent independent systems or subsystems with the ability to interface with the rest of the system.

**Pre Lab Questions:**

1. What is the purpose of component diagram?
2. How to draw a component diagram?
3. Where to use component diagrams?
4. What is meant by artifacts?
5. What are the benefits of component diagram?

**Post Lab Questions:**

1. Recollect the symbols and shapes used in component diagram.
2. What is an interface?
3. What is dependency?
4. How the relationship among the artifact can be represented?
5. What is the difference between libraries and folders?

**RESULT:**

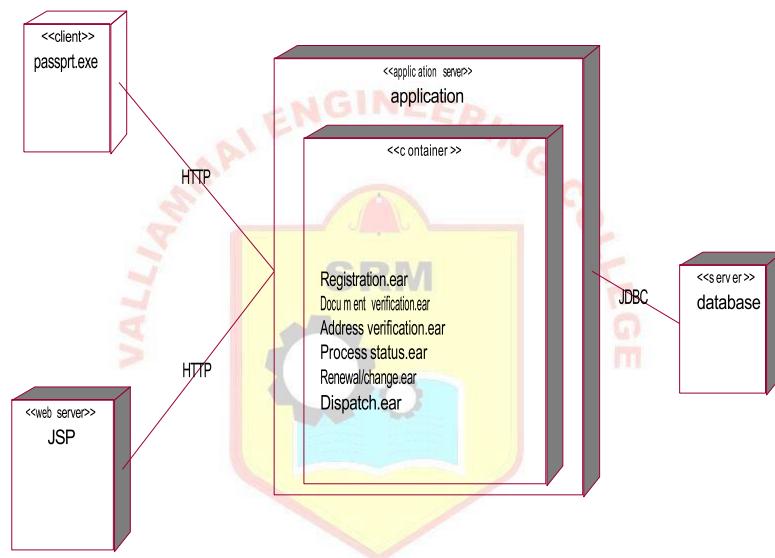
Thus the component diagram for Passport Automation System has been drawn successfully.

**Ex. No: 1j****DEPLOYMENT DIAGRAM****AIM:**

To draw deployment diagram for Passport Automation System.

**Introduction:**

Deployment diagram is a structure diagram which shows architecture of the system as deployment (distribution) of software artifacts to deployment targets. Artifacts represent concrete elements in the physical world that are the result of a development process.

**Pre Lab Questions:**

1. What is deployment diagram?
2. What are the limitations of your physical hardware?
3. Which distribution architecture are you using?
4. Do you have all the nodes you need? Do you know how they are all connected?
5. Do you know which components are going to be on which nodes?

**Post Lab Questions:**

1. What are the elements of deployment diagram?
2. What is the difference between software component and software component?
3. Name some of the deployment diagram applications.
4. How the database can be connected?
5. What are runtime processing nodes?

**RESULT:**

Thus the deployment diagram for Passport Automation System has been drawn successfully.

## **Ex. No : 1k IMPLEMENTATION OF PASSPORT AUTOMATION SYSTEM**

### **AIM:**

To write a code for the passport automation system.

### **FORM1:**

#### **Private Sub Command1\_Click()**

```
If (Text1.Text = "vignesh" And Text2.Text = "1234") Then
    MsgBox "valid"
    Form1.Hide
    Form2.Show
Else
    MsgBox "invalid"
End If
```

**End Sub**

### **FORM2:**

#### **Private Sub Command1\_Click()**

```
Form2.Hide
Form3.Show
```

**End Sub**

#### **Private Sub Command2\_Click()**

```
Form2.Hide
Form4.Show
```

**End Sub**

#### **Private Sub Command3\_Click()**

```
Form2.Hide
Form5.Show
```

**End Sub**

#### **Private Sub Command4\_Click()**

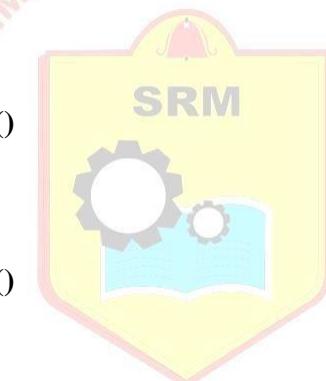
```
Form2.Hide
Form1.Show
```

**End Sub**

### **FORM3:**

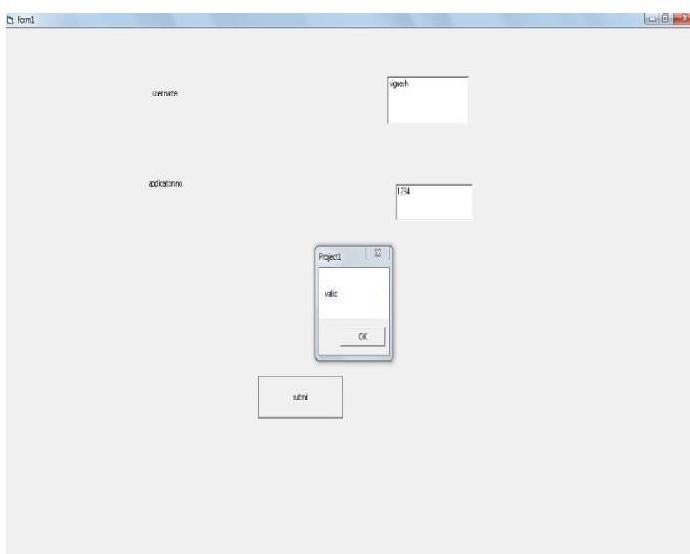
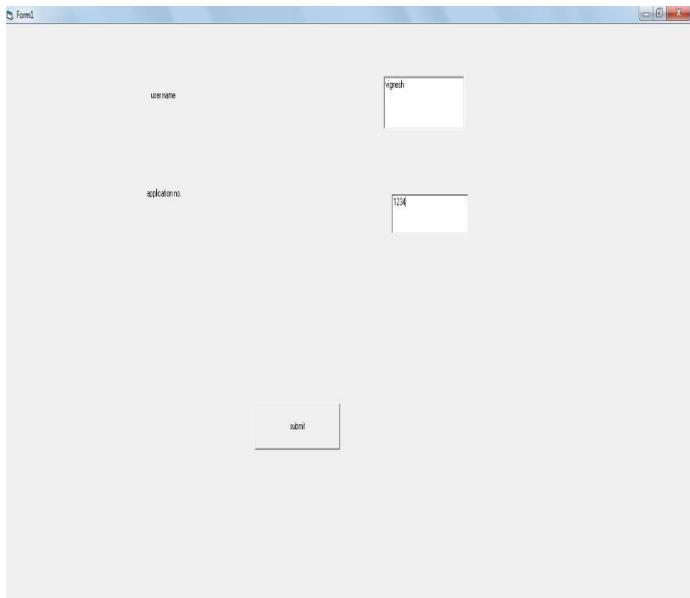
#### **Private Sub Command1\_Click()**

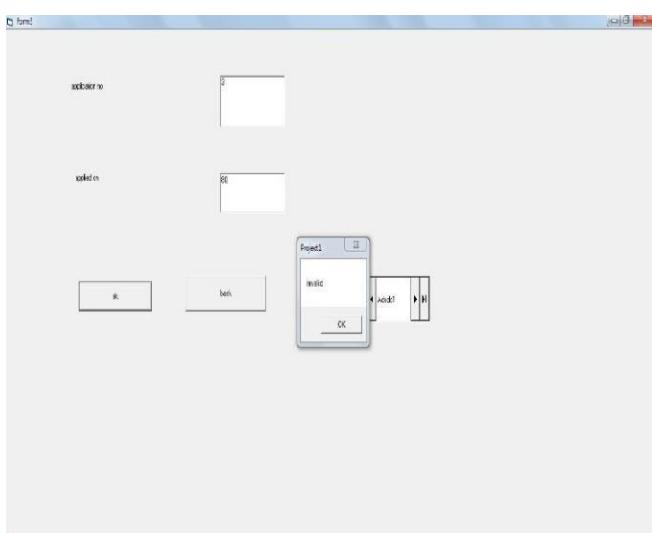
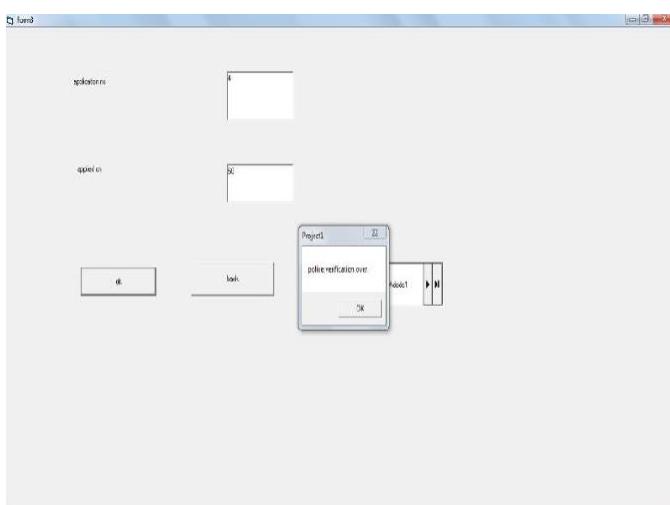
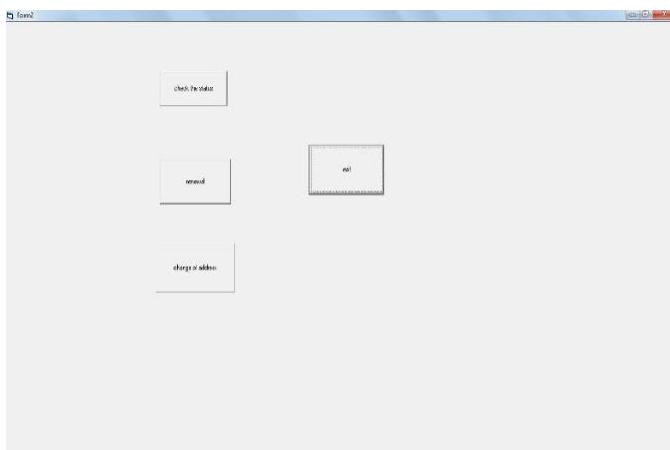
```
If (Text2.Text < 30) Then
    MsgBox "stillprocessing"
Else
    If (Text2.Text > 30 And Text2.Text < 50) Then
        MsgBox "police verification goin on"
    Else
        If (Text2.Text < 80) Then
            MsgBox "police verification over"
```

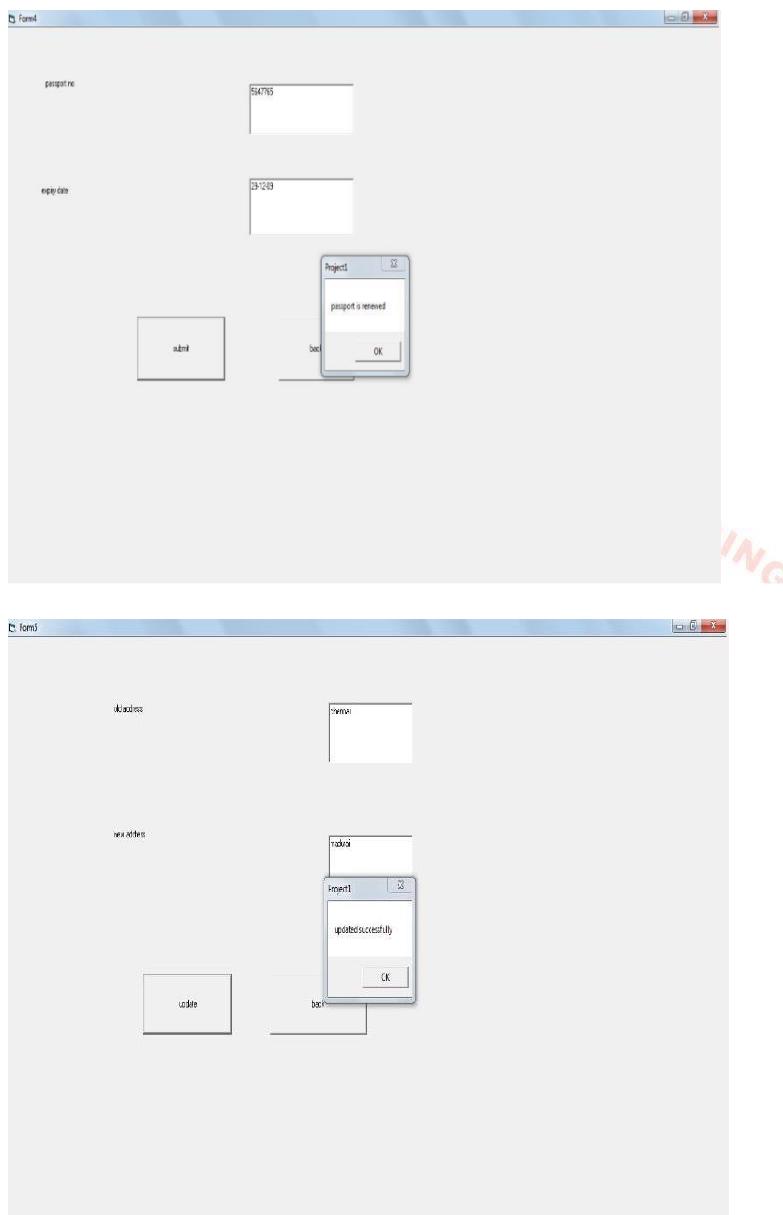


```
Else
MsgBox "invalid"
End If
End If
End If
End Sub
Private Sub Command2_Click()
Form3.Hide
Form2.Show
End Sub
FORM4:
Private Sub Command1_Click()
MsgBox "passport is renewed"
End Sub
Private Sub Command2_Click()
Form4.Hide
Form2.Show
End Sub
FORM5:
Private Sub Command1_Click()
MsgBox "updated successfully"
End Sub
Private Sub Command2_Click()
Form5.Hide
Form2.Show
End Sub
```



**GUI DESIGN:**



**RESULT:**

Thus the project to develop a Passport Automation System is done successfully.

## CHAPTER 3. EXPERIMENTS

### A. COURSE REGISTRATION SYSTEM

#### **1. PROBLEM STATEMENT:**

The software which displays the list of courses available for the mark that the student get and the student can able to allocate best course from the choice available. It displays and allocates courses based on student ranking. The student ranking is based on marks, caste and community. Based on caste and community the ranking may varied. This software allows the student to choose the best college for the available courses.

#### **2. OVERALL DESCRIPTIONS:**

##### **2.1 MODULES:**

- 1.Login
- 2.student\_info
- 3.course\_details
- 4.course\_registration
- 5.confirmation

##### **2.2 MODULE DELIVARABLES:**

###### **Login:**

Basicflow:To authenticate the user,the student has to enter username and password

Alternateflow:If the password is wrong,it will ask the student to answer security question and retrieve the password

Precondition:The system asks the student to enter the password

Postcondition:On success,the student displays the student information

###### **Student\_info:**

Basicflow:The system displays the basic information about the student.It should be verified by the user.

Alternateflow:If there is any error in displayed information ,the student can do changes.

Precondition:The student checks the basic details displayed.

Postcondition:After verifying all the details,it should be updated and moves on to the next state.

###### **Course\_details:**

Basicflow:The courses that are available for the student are displayed and the student should select one.

Alternateflow:If the student finds that the courses available for him are not displayed,he can report it.

Precondition:The details of the corses available for his mark should be known by the user.

Postcondition:After selecting the course,it will moves on to the next stage.

###### **Course\_registration:**

Basicflow:All the colleges that are available for the courses are displayed.

Alternateflow:If the student finds that the colleges are displayed in error,he can report it.

Precondition:The student should have to select the colleges available.

Postcondition:After registering the course,it will move on to the next stage.

###### **Confirmation:**

Basicflow:The college and the course that the student choose is displayed.

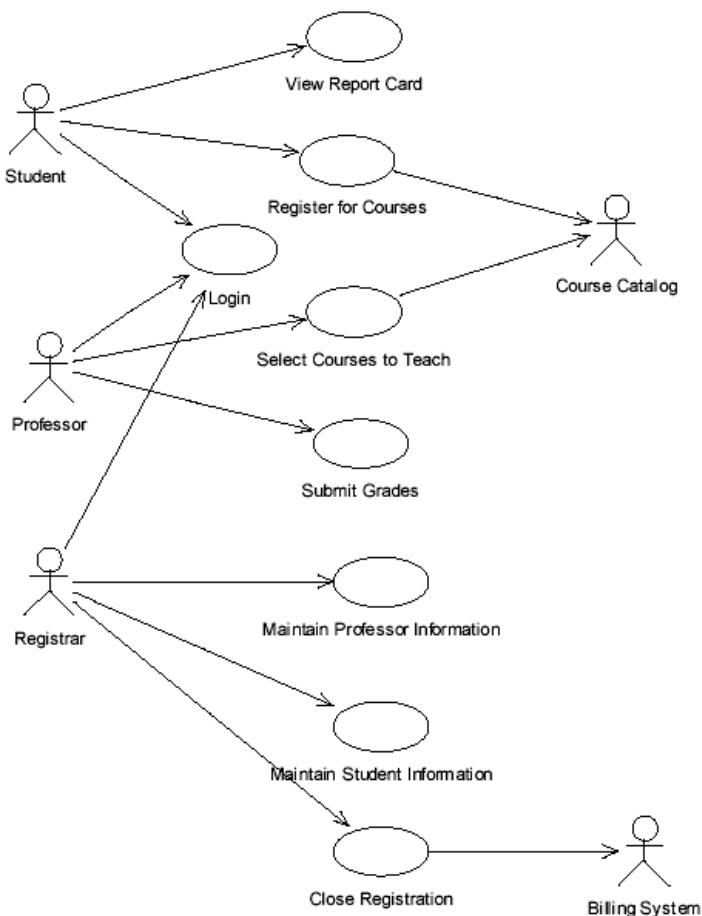
Alternateflow:If the student finds any fault regarding his course/college,he can report it.

Precondition:The student should have to confirm the college and the course.

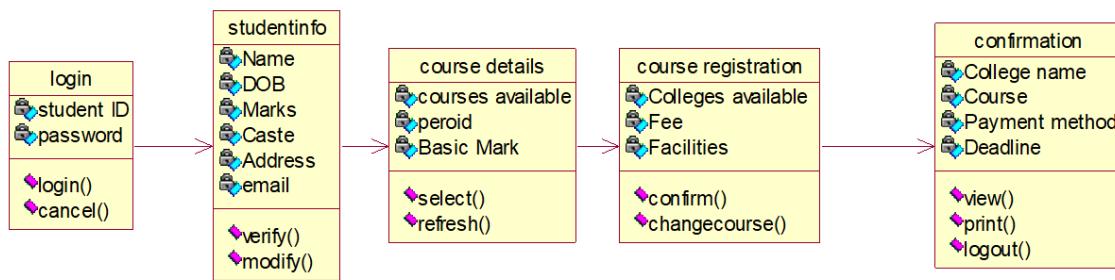
Postcondition: The course and the college is confirmed and the student confirms it by choosing confirm button.

### 3. UML DIAGRAMS:

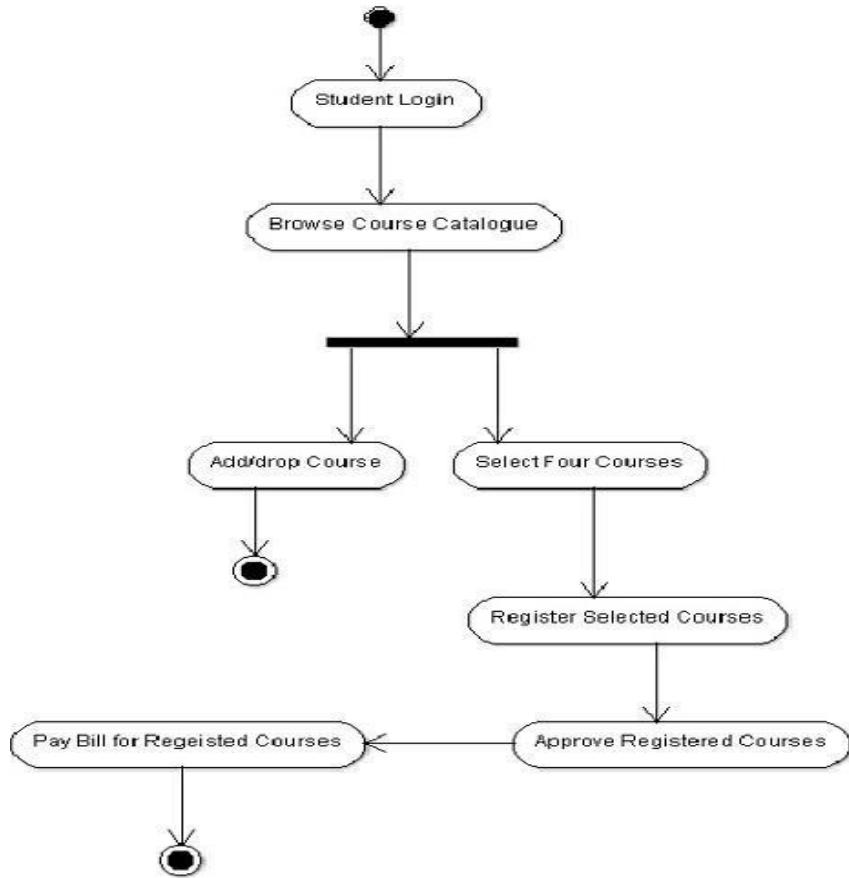
#### 3.1. UseCase Diagram:



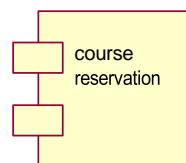
#### 3.2. Class Diagram:



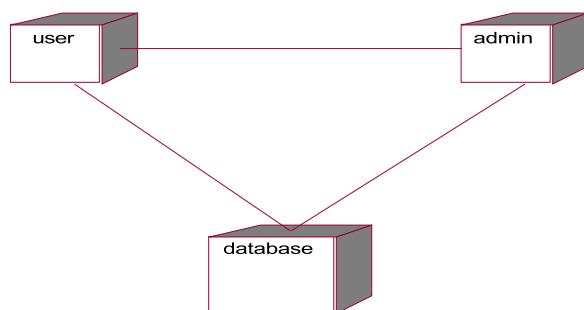
### 3.3. ActivityDiagram:



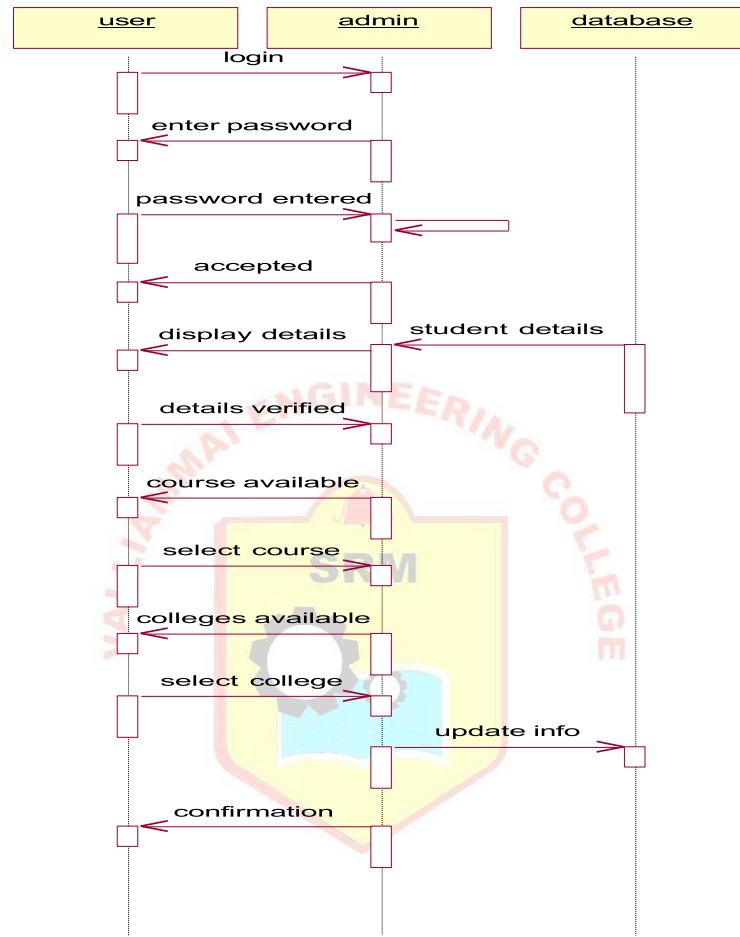
### 3.4. Component Diagram:



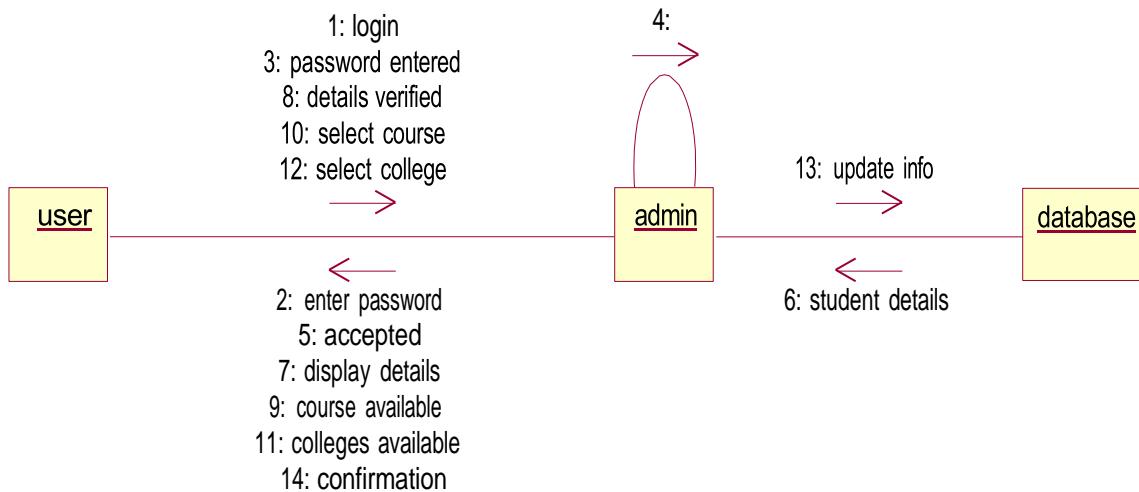
### 3.5. Deployment Diagram:



### 3.6. Sequence Diagram:



### 3.7. Collaboration Diagram:



#### 4. DATABASE DESIGN

Database name: student

Table name : studentdetails

Fields	Data type
NAME	text
dob	text
MARK	integer
caste	text
ADDRESS	text
email	text
gender	text

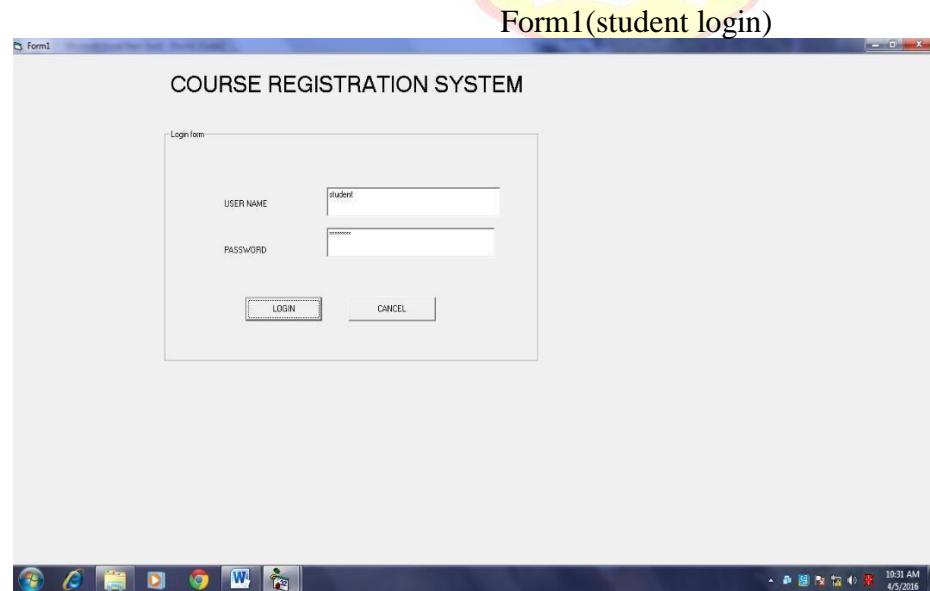
Table name : coursedetails

Fields	Data type
courseavailable	text
timeperiod	text
basicmark	integer

Table name : conformation

Fields	Data type
availablecolleges	text
paymentmethod	text

#### 5. IMPLEMENTATION:

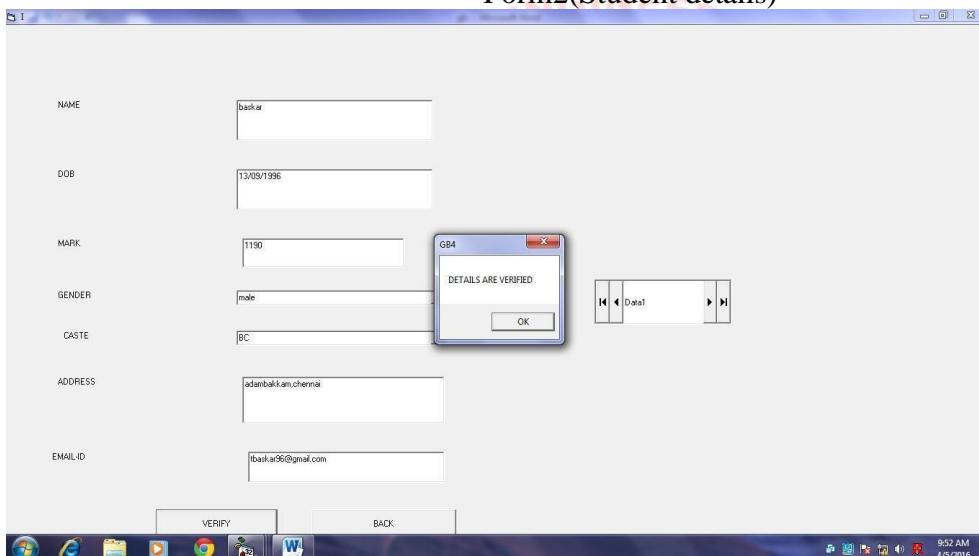


Coding:

```
Private Sub Command1_Click()
If Text1.Text = "student" and Text2.Text = "password" Then
form2.Show
else
msgBox ("incorrect username or password")
End If
End Sub
```

```
Private Sub Command2_Click()
End
End Sub
```

Form2(Student details)

Coding:

```
Private Sub Command1_Click()
form2.Show
Data1.Recordset.AddNew
Data1.Recordset.Fields("NAME") = Text5.Text
Data1.Recordset.Fields("dob") = Text4.Text
Data1.Recordset.Fields("MARK") = Text3.Text
Data1.Recordset.Fields("caste") = Combo1.Text
Data1.Recordset.Fields("ADDRESS") = Text2.Text
Data1.Recordset.Fields("email") = Text1.Text
Data1.Recordset.Fields("gender") = Combo2.Text
MsgBox ("DETAILS ARE VERIFIED")
Form3.Show
Data1.Recordset.Update
End Sub

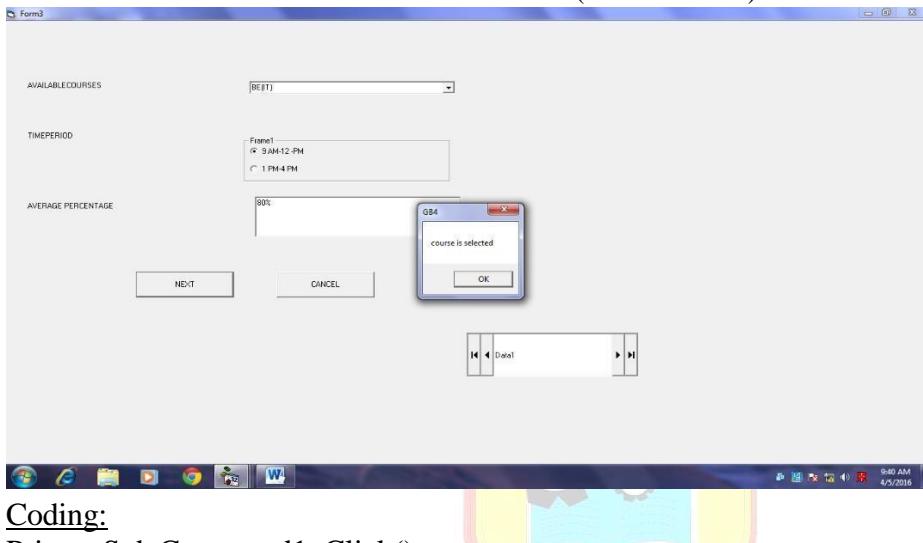
Private Sub Command2_Click()
Form1.Show
End Sub
```

```

End Sub
Private Sub Form_Load()
Combo1.AddItem ("BC")
Combo1.AddItem ("SC")
Combo2.AddItem ("male")
Combo2.AddItem ("female")
Combo2.AddItem ("other")
End Sub

```

Form3(Course details)

Coding:

```

Private Sub Command1_Click()
Form3.Show
Data1.Recordset.AddNew
Data1.Recordset.Fields("courseavailable") = Combo1.Text
If Option1.Value = True Then
Data1.Recordset.Fields("timeperiod") = "9am-12pm"
Else
Data1.Recordset.Fields("timeperiod") = "1am-4pm"
End If
Data1.Recordset.Fields("basicmark") = Val(Text1.Text)
MsgBox ("course is selected")
Form4.Show
Data1.Recordset.Update
End Sub

```

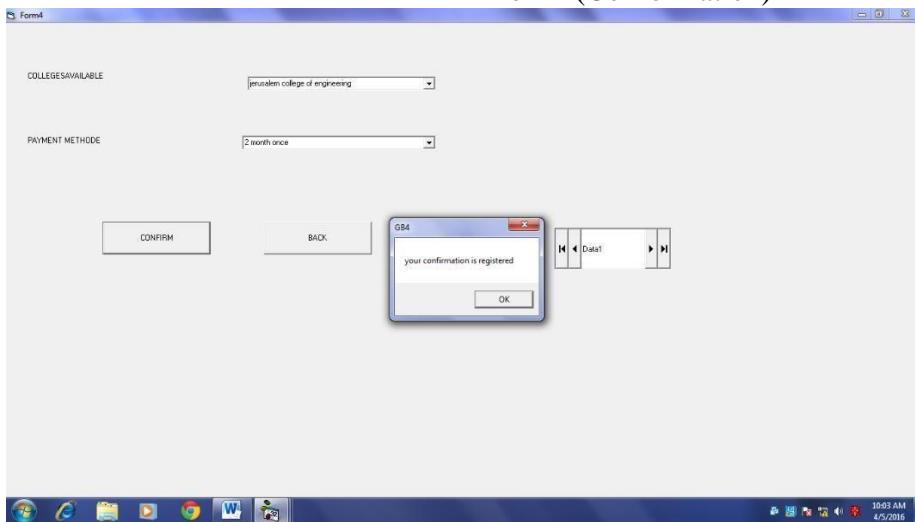
```

Private Sub Command2_Click()
Form2.Show
End Sub
Private Sub Form_Load()
Combo1.AddItem ("BE(ECE)")
Combo1.AddItem ("BE(IT)")
Combo1.AddItem ("BE(CIVIL)")
Combo1.AddItem ("BE(E&I)")

```

End Sub

### Form4(Conformation)



#### Coding:

```

Private Sub Command1_Click()
Form4.Show
Data1.Recordset.AddNew
Data1.Recordset.Fields("availablecolleges") = Combo1.Text
Data1.Recordset.Fields("paymentmethod") = Combo2.Text
MsgBox (" your confirmation is registered")
Form1.Show
End Sub

Private Sub Command2_Click()
Form3.Show
End Sub

Private Sub Form_Load()
Combo1.AddItem ("jerusalem college of engineering")
Combo1.AddItem ("prathushya engineering college")
Combo1.AddItem ("tagore engineering college")
Combo2.AddItem ("single payment")
Combo2.AddItem ("2 month once")
Combo2.AddItem ("3 month once")
End Sub

```

## 6.TESTING:

<b>Test case ID: Test_01</b>					
Test priority(Low/Medium/High):Medium					
Module name: login					
Test title :verify login with valid username and password					
S.N O	TEST STEPS	EXPECTED RESULTS	ACTUAL RESULTS	STATUS	NOTES
1	Provide valid User name	User should Be able to login	The user is able to move to next Entry	Success	-
2	Provide valid password	User should be Able to Login	The user is able To login Successfully	Success	Incase of wrong Password was given an error Message box was displayed
3	Click login	User should be able to navigate to next page after validation	User name and password is validated and next page is displayed	Success	Incase user gives wrong entry the sign in page remains active
4	Click signup	User should be able to navigate to next page where user enters his credentials	User navigates to the signup page where his user name and password is validated	success	-

## RESULT:

The Course registration system was designed and implemented successfully.

## B. STUDENT MARKS ANALYZING SYSTEM

### 1. PROBLEM STATEMENT:

Student marks analyzing system has to be developed for analyzing obtained by the students who scored in Semester Examination The System should provide following functionalities

1. The System obtains following information's from the faculty generates report Roll No, Name, Department, Semester, Marks obtained in each subject.
2. The total for each student should be calculated and ranked based on total and pass in all the subject appeared.
3. The Final report should display rank, percentage, Class, Pass/Fail Status for each student.

### 2. OVERALL DESCRIPTIONS:

#### 2.1 MODULES:

- 1.Login
- 2.student\_mark
3. View report

#### 2.2 MODULE DELIVARABLES:

##### 1. Login

**Basic Flow:** This use case starts when the Faculty wishes to Login to the Student Marks Analyzing System.

1. The System requests that the Faculty enter his/her name and password
2. The Faculty enters his/her name and password
3. The System validates the entered name and password and logs the Faculty into the System

**Alternative Flows:** Invalid Name/Password

If, in the Basic flow, the Faculty enters an invalid name and/or password, the system displays an error message. The Faculty chooses to either return to the beginning of the Basic flow or cancel the login, at which point the use case ends.

**Pre-Conditions:** None

**Post-Conditions:** If the use case was successful, the Faculty is now logged into the system. If not, the system State is unchanged.

##### 2. Student\_Mark

**Basic flow:** The Faculty uses this usecase to enter marks for each student. The faculty enters the following details namely Roll No, Student Name, Department, Marks for each student.

**Alternative Flows:** If faculty not entered any details or invalid marks then gives error Message

**Pre-Conditions:** The Faculty must logged into the system

**Post-Conditions:** If this Use case was successful, Student Mark Analysis Report will be generated for the Student.

##### 3. View Report

**Basic flow:** The Actor uses this usecase view the Report .The report contains the following details Namely Roll No, Student Name, Marks in each subject, total, class, Pass/Fail Status, No of subjects failed, Rank.

Apart from this there is a separate report Overall Pass percentage of class, No of students cleared in First class, Overall Top 3 persons of the class.

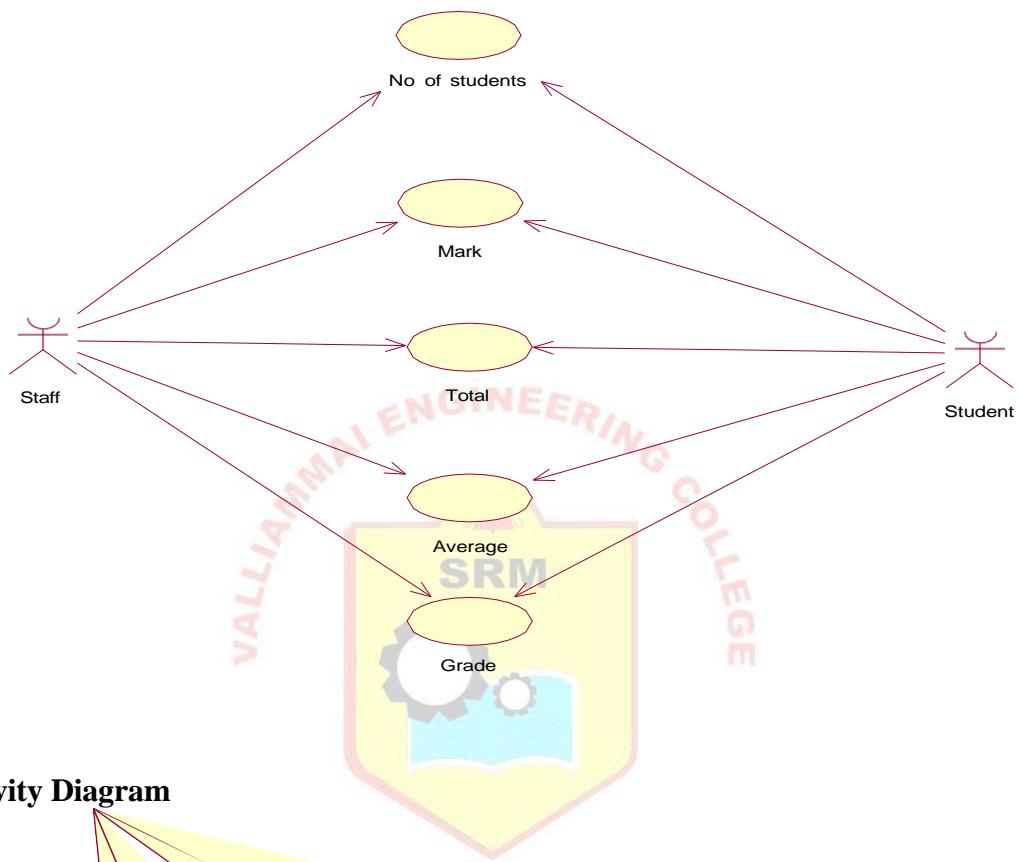
**Alternative Flow:** If the Marks is not entered for all the students the use case will ask the faculty to the enter the marks.

**Pre-Conditions:** The Faculty must entered marks for all the students in a class.

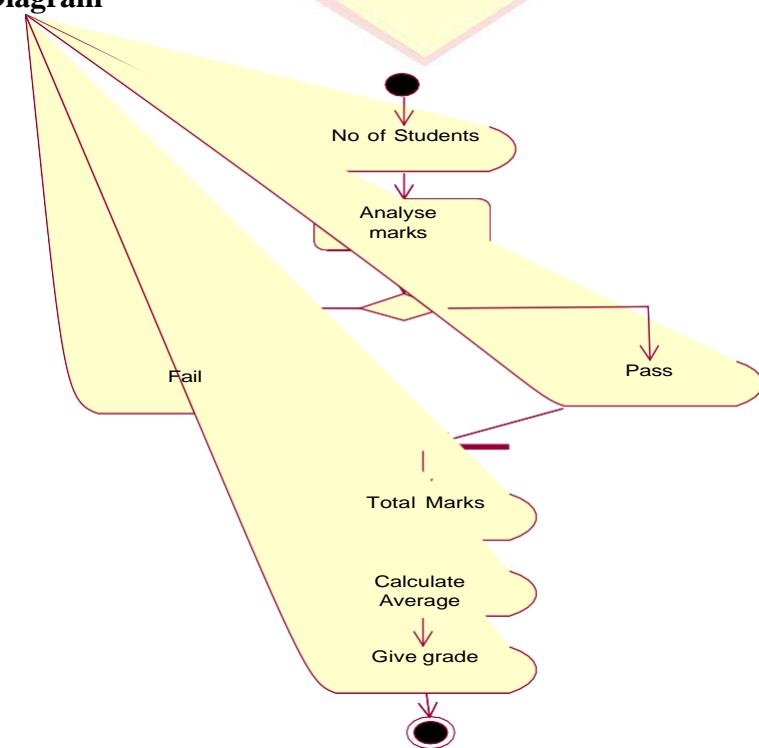
**Post-Conditions:** None

### 3. UML DIAGRAMS:

#### 3.1. UseCase Diagram:



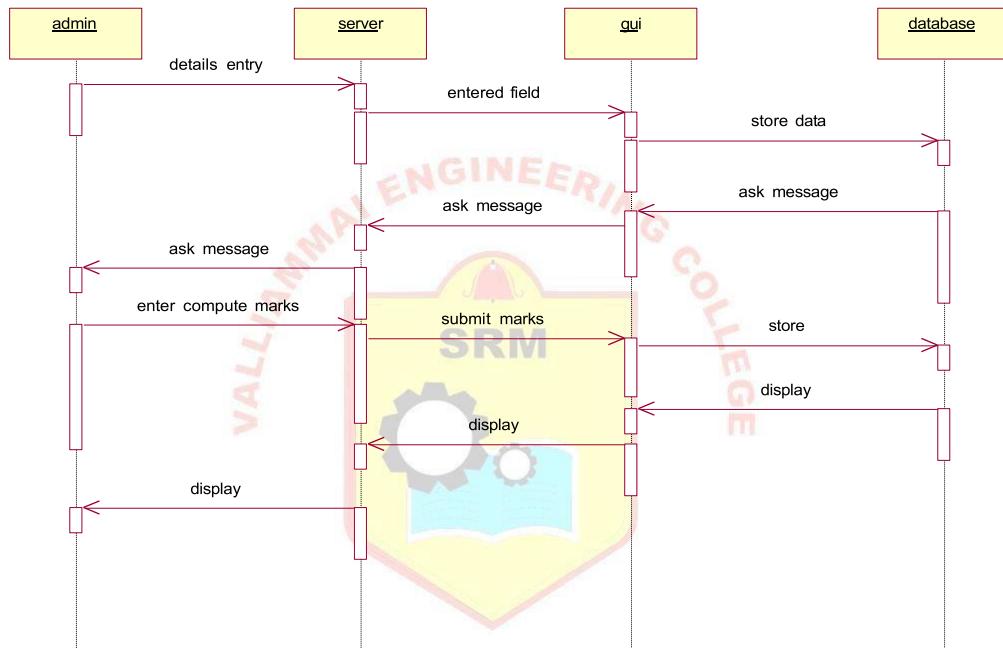
#### 3.2 Activity Diagram



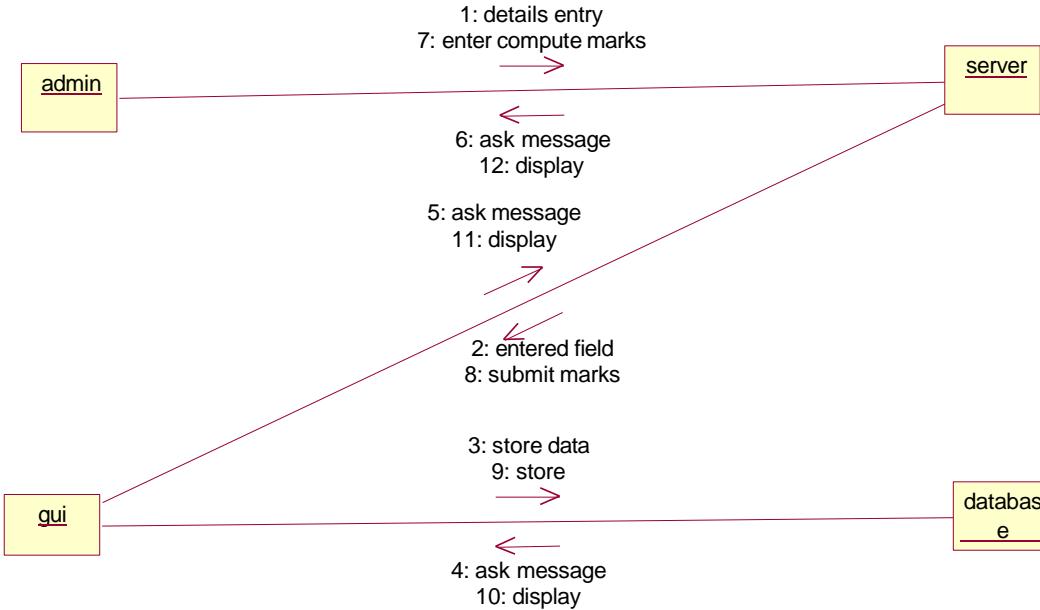
### 3.3 Class Diagram



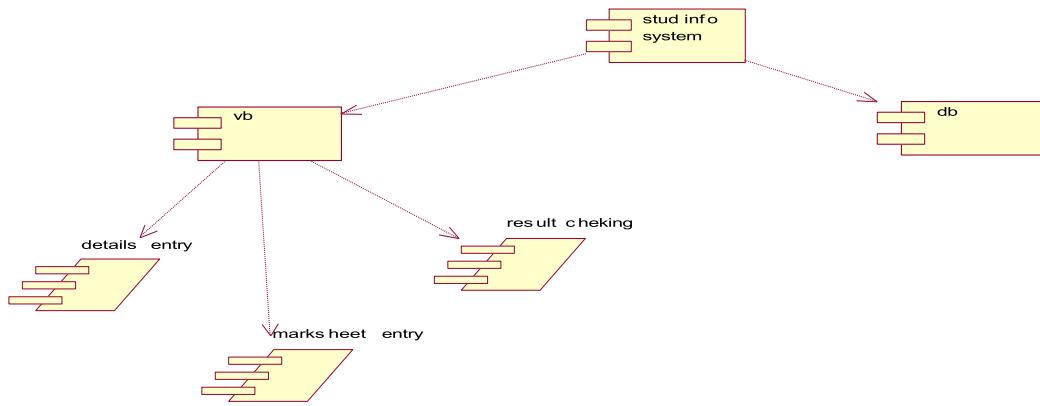
### 3.4 Sequential Diagram



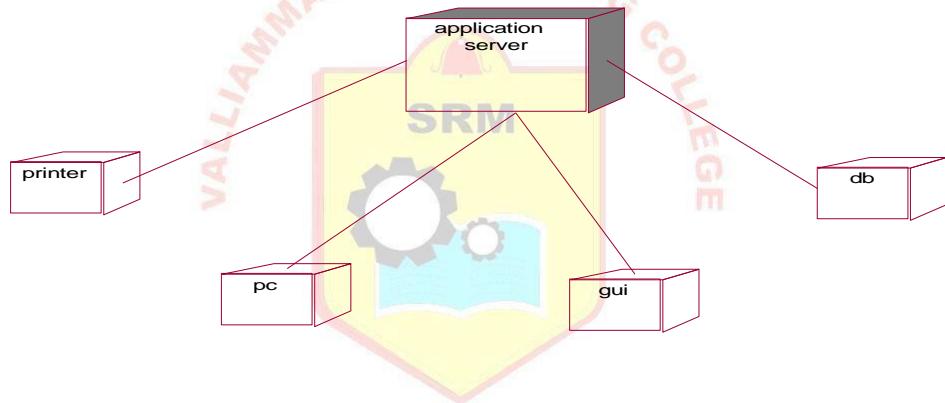
### 3.5 Collaboration diagram



### 3.6 Component Diagram of student info System



### 3.7 Deployment Diagram of Student info System



## 4. DATABASE DESIGN

Database name: student

Table name:marks

Fields	Data type
student name	text
register number	integer
Pqt	integer
Daa	integer
Mup	integer
Se	integer
Os	integer
Total	integer
Percentage	double
Result	text

## 5. IMPLEMENTATION:

FORM1 (Login form)

Coding:

```
Private Sub Command1_Click()
if text1.text="student" and text2.text="itdept" then
form2.Show
Else
msgbox("Please enter correct user name and password")
end if
End Sub
```

```
Private Sub Command2_Click()
End
End Sub
```

FORM2 (Entry form)



Coding:

```
Private Sub Command1_Click()
Dim total, per, result
Data1.Recordset.AddNew
Data1.Recordset.Fields("sname") = Text1.Text
Data1.Recordset.Fields("regno") = Text2.Text
Data1.Recordset.Fields("m1") = Text3.Text
Data1.Recordset.Fields("m2") = Text4.Text
Data1.Recordset.Fields("m3") = Text5.Text
Data1.Recordset.Fields("m4") = Text6.Text
Data1.Recordset.Fields("m5") = Text7.Text
total = Text3.Text + Text4.Text + Text5.Text + Text6.Text + Text7.Text
Data1.Recordset.Fields("total") = total
per = (total / 5) * 100
Data1.Recordset.Fields("per") = per
If (Text3.Text > 49 And Text4.Text > 49 And Text5.Text > 49 And Text6.Text > 49 And Text7.Text > 49) Then result = "pass"
Data1.Recordset.Fields("result") = result
Data1.Recordset.Update
MsgBox ("record saved")
Form3.Show
End Sub

Private Sub Command2_Click()
End
End Sub
```

### FORM3 (Display Students)

Form3

sname	regno	m1	m2	m3	m4	m5	total	per	result
brij	11	56	67	78	89	90	5667788990	113352779800	pass
bhavi	13	66	77	88	99	77	6677889977	13557799540	pass
ara	14	56	65	76	87	98	5665768798	113315279560	pass
rachel	12	66	77	65	88	98	6677658898	13553177960	pass
chandler	15	99	77	56	87	85	9977568765	19951375700	pass
ross	16	78	98	88	66	98	7898866598	157977733960	pass
joey	17	65	75	86	85	97	6575986597	131517371940	pass
monica	19	56	87	95	45	77	5687954577	113759091540	
phoebe	20	88	76	55	78	87	8876557887	177531157740	pass

cancel

Coding:

```
Private Sub Command1_Click()
End
End Sub
```



## 6. TESTING:

<b>Test case ID: Test_01</b>					
Test priority (Low/Medium/High):Medium					
Module name: login					
Test title :verify login with valid username and password					
S.NO	TEST STEPS	EXPECTED RESULTS	ACTUAL RESULTS	STATUS	NOTES
1	Provide valid User name	User should Be able to login	The user is able to move to next Entry	Success	-
2	Provide valid password	User should be Able to Login	The user is able To login Successfully	Success	Incase of wrong Password was given an error Message box was displayed
3	Click login	User should be able to navigate to next page after validation	User name and password is validated and next page is displayed	Success	Incase user gives wrong entry the sign in page remains active
4	Click signup	User should be able to navigate to next page where user enters his credentials	User navigates to the signup page where his user name and password is validated	success	-

## RESULT:

The Student marks analyzing system was designed and implemented successfully.

## C. ONLINE TICKET RESERVATION SYSTEM

### 1. PROBLEM STATEMENT:

This project is about online ticket reservation and consists of two modules. The reservation and the cancellation module. The reservation module allows the user to reserve tickets for a particular train on a particular date. If there is a ticket available, the users can know the vacancy details through the enquiry module. The cancellation module allows user to cancel the tickets for a particular date through reservation officer (system). This module performs status reveal before tickets are being reserved and after they get booked. All these modules together prove to be a flexible online reservation system and it provides complete flexibility to end users and it assumes the desired performance.

### 2. OVERALL DESCRIPTION:

#### 2.1 MODULES:

- Login
- Display train list
- Search for train
- Reservation
- Cancellation
- Train Status

#### 2.2 MODULE DELIVERABLES:

##### 1. LOGIN

Basic Flow

This use case starts when the passenger wishes to Login to the Online Ticket Reservation system

1. The System requests that the passenger enter his/her name and password
2. The passenger enters his/her name and password
3. The System validates the entered name and password and logs the passenger into the System

Alternative Flows: Invalid Name/Password

If, in the Basic flow, the passenger enters an invalid name and/or password, the system displays an error message. The passenger chooses to either return to the beginning of the Basic flow or cancel the login, at which point the use case ends.

Pre-Conditions: None

Post-Conditions: If the use case was successful, the passenger is now logged into the system. If not, the system State is unchanged.

##### 2. Display Train List

Basic Flow: This use case gives passenger information about each train namely train no, train name, Stations passes, Arrival Time, Departure Time etc

Alternative Flows: None

Pre-Conditions: None

Post-Conditions: If the use case was successful, the passenger information about each train namely train no, train name, Stations passes, Arrival Time, Departure Time etc

##### 3. Search for Train

Basic flow

The passenger can obtain train information either by entering train no or Source and Destination Station

1. If the passenger train no gives the information about train

2. If the passenger enter Source and Destination Station from list gives information about list of trains passing through station. From the list link will be provided to each train, which contains the information

Alternate flow: If the passenger enters an invalid train no then it gives error message invalid train no and asks the passenger to enter a valid train no.

Pre-Conditions: None

Post-Conditions: If the use case was successful, the passenger can able to view the list of trains.

#### **4. Reservation**

Basic flow

1. The user reserves the ticket by giving following

a) Passenger name, Sex, Age, Address

b) Credit Card No, Bank Name

c) Class through passenger is going to travel i.e First class or Second class or AC

d) Train no and Train name, Date of Journey and number of tickets to be booked.

2. If the ticket is available in a train then the ticket will be issued with PNR No.else the ticket will be issued with a waiting list number.

Alternative flow: If the passenger gives an invalid credit card no or specified a bank where does have any account. Error message will be displayed.

Pre-Conditions: The passenger has to decide about the train he is going to travel.

Post-Conditions: If the use case was successful, the passenger will get the ticket.

#### **5. Cancellation**

Basic flow

This use case used by passenger to cancel the ticket, which he/she booked earlier by Entering PNR No. The cancellation has been done reallocating the tickets allotted to the Passenger.

Alternate flow: If the Passenger had entered invalid PNR No then has been asked to enter valid PNR No.

Pre-Conditions: The Passenger had reserved tickets in a train.

Post-Conditions: If the use case was successful, the passenger can cancel the ticket.

#### **6. Ticket Status**

Basic flow

1. The passenger should give PNR No to know the status of ticket, which he/she booked earlier.

2. If the PNR No is valid, the status of the ticket will be displayed.

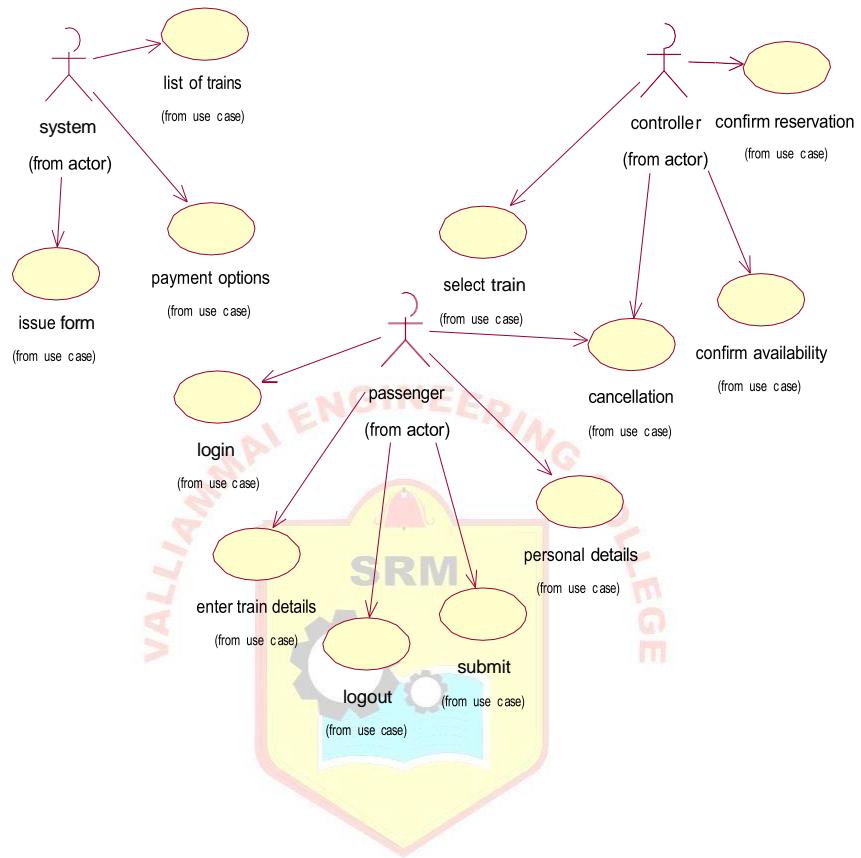
Alternate flow: If passenger had entered an invalid no or PNR NO, which does not exists then error Message will be displayed.

Pre-Conditions: The Passenger had reserved tickets in a train.

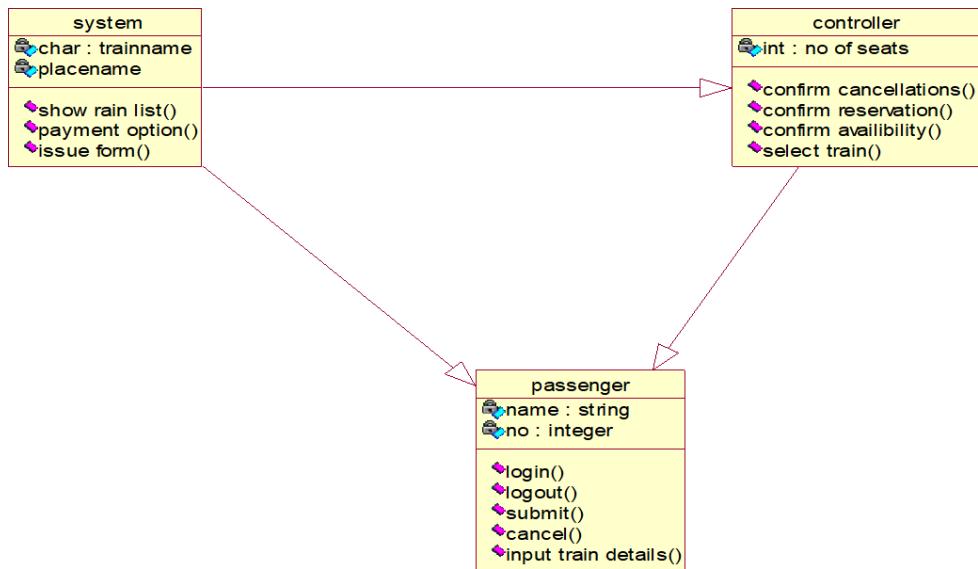
Post-Conditions: If the use case was successful, the passenger can view status of the ticket.

### 3. UML DIAGRAMS:

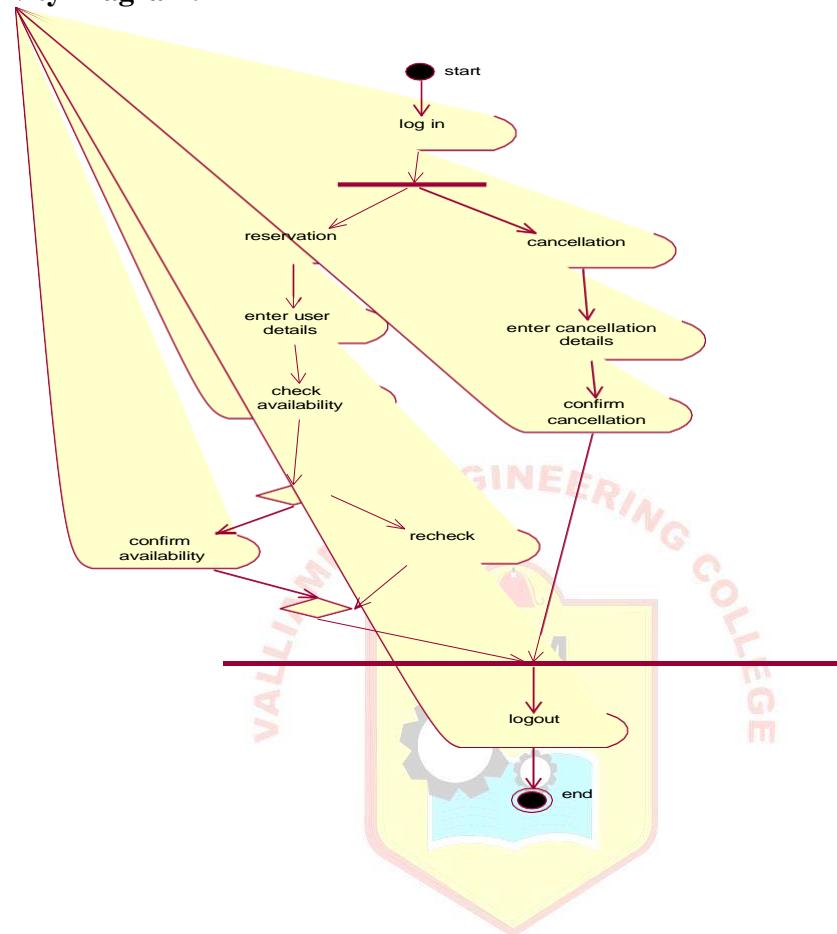
#### 3.1. Usecase Diagram:



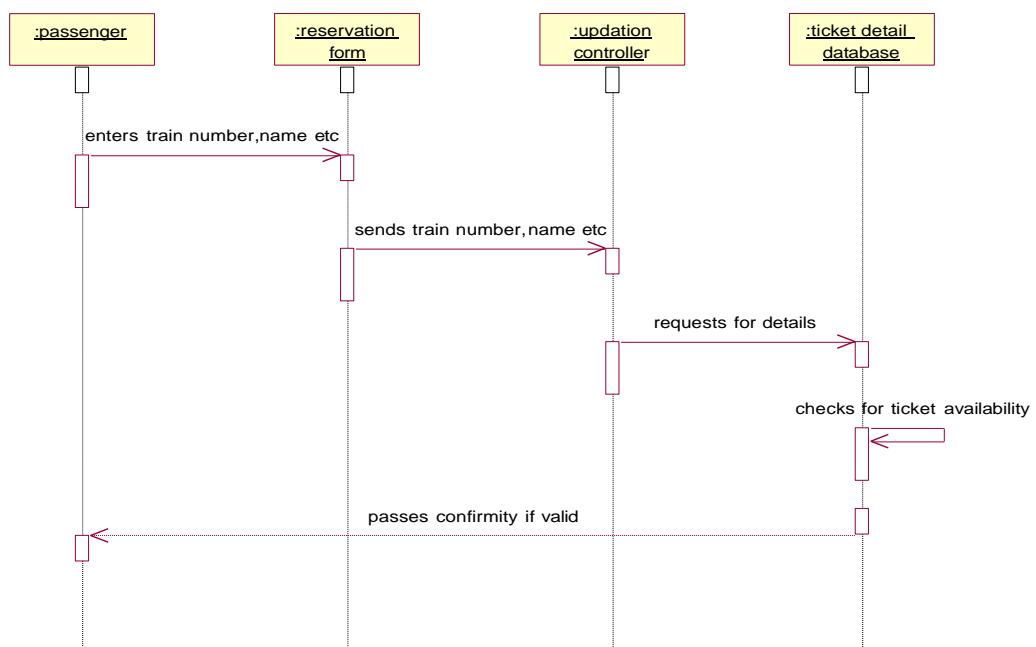
#### 3.2 Class Diagram:



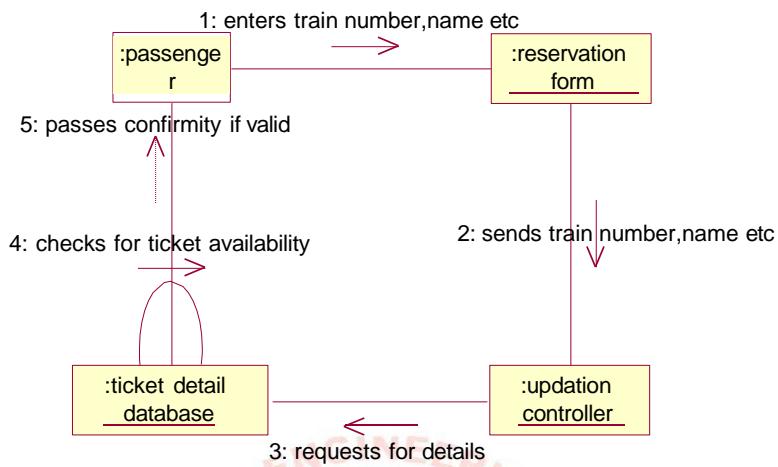
### 3.3 Activity Diagram:



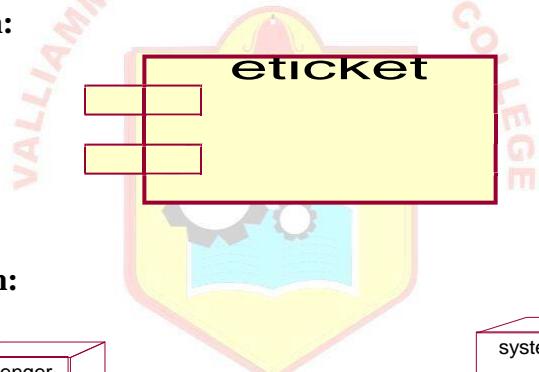
### 3.4 Sequence Diagram:



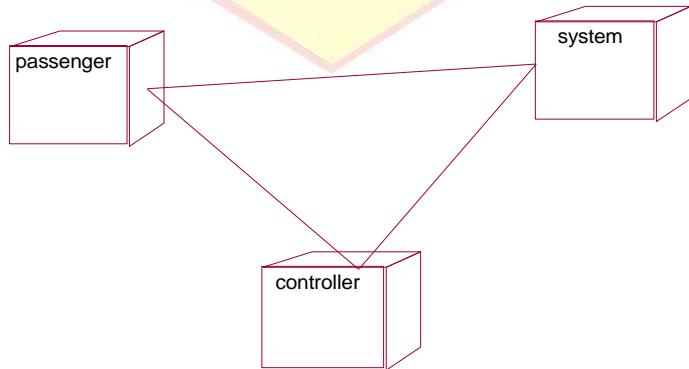
### 3.5 Collaboration Diagram



### 3.6 Component Diagram:



### 3.7 Deployment Diagram:



## 4. DATABASE DESIGN

Database name: Rail

Table name :student name

Field name	Datatype
Name	Text(50)
Place	Integer
Mobile number	integer

Table name: reservation

Field name	Datatype

Name	Text
Age	Integer
Place	Text
Train and number	Text
Seat Selection	Text
Sex	Text
Date	Text
Time	Text
From	Text
To	Text
Food	Text
Class	Text

Table name : reservation 2

Field name	Data type
Adhar no	Text
Pan card number	Text
e-mail id	Text
Mobile number	Text

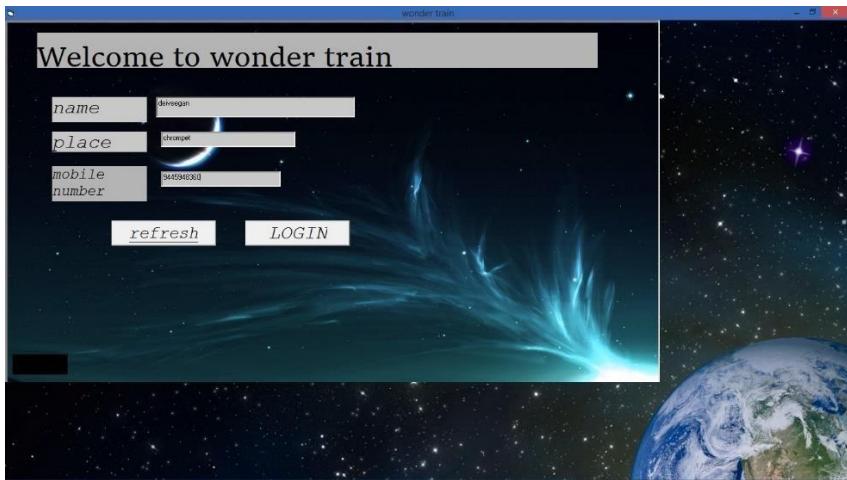
Table name: cancellation

Field name	Data type
Train name & number	Text
Name	Text
Pnr number	Text
e-mail ID	Text
Phone number	Text

Table name: ticket status

Field name	Data type
Train name & number	Text
Name	Text
Pnr number	Text
e-mail ID	Text
Phone number	Text

## **5. IMPLEMENTATION**



### **Coding:**

```
Private Sub Command1_Click()
Form1.Show
Data1.Recordset.AddNew
Data1.Recordset.Fields("name") = Text1.Text
Data1.Recordset.Fields("place") = Text2.Text
Data1.Recordset.Fields("mobile number") = Text3.Text
Data1.Recordset.Update
End Sub
```



```
Private Sub Command2_Click()
Dim ctr As Control
For Each ctr In Me.Controls
    If TypeOf ctr Is TextBox Then
        ctr.Text = Empty
    End If
Next ctr
End Sub
```



**coding**

```
Private Sub Command1_Click()
```

```
Form3.Show
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
Form7.Show
```

```
End Sub
```

```
Private Sub Command3_Click()
```

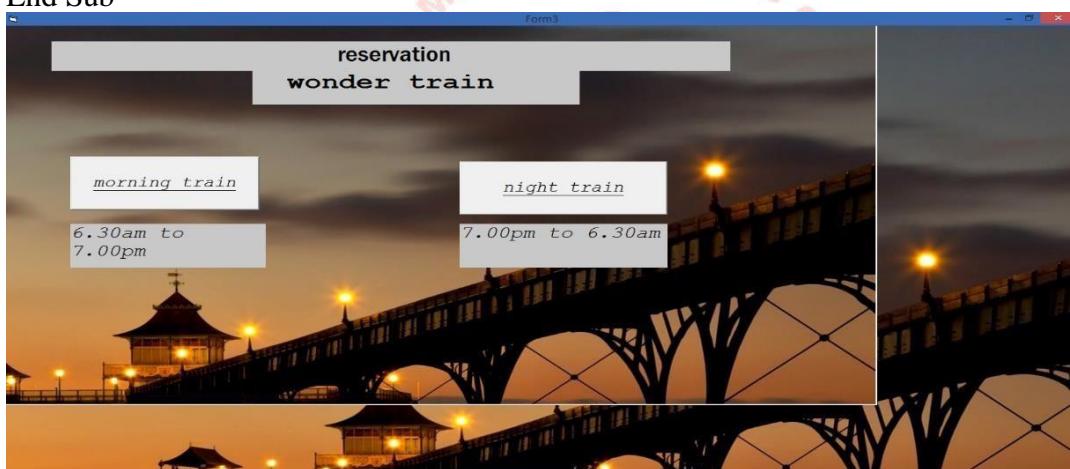
```
Form9.Show
```

```
End Sub
```

```
Private Sub Command4_Click()
```

```
Form12.Show
```

```
End Sub
```

**coding**

```
Private Sub Command1_Click()
```

```
Form2.Show
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
Form6.Show
```

```
End Sub
```

**coding**

```
Private Sub Command1_Click()
```

```
Form4.Show
```

```
Data1.Recordset.AddNew
```

```
Data1.Recordset.Fields("name") = Text1.Text
```

```
Data1.Recordset.Fields("age") = Text2.Text
```

```
Data1.Recordset.Fields("place") = Text3.Text
```

```
Data1.Recordset.Fields("train name and number") = Text8.Text
```

```

If Option1.Value = True Then
Data1.Recordset.Fields("seat selection") = "upper"
Else
Data1.Recordset.Fields("seat selection") = "lower"
End If
If Option6.Value = True Then
Data1.Recordset.Fields("sex") = "male"
Else
Data1.Recordset.Fields("sex") = "femal"
End If
If Option10.Value = True Then
Data1.Recordset.Fields("food") = "veg"
Else
Data1.Recordset.Fields("food") = "non-veg"
End If
If Option3.Value = True Then
Data1.Recordset.Fields("class") = "1 class"
Else
Data1.Recordset.Fields("seat selection") = "2 class"
End If
Data1.Recordset.Fields("date") = Text4.Text
Data1.Recordset.Fields("time") = Text5.Text
Data1.Recordset.Fields("from") = Text6.Text
Data1.Recordset.Fields("to") = Text7.Text
Data1.Recordset.Update
End Sub

```

```

Private Sub Command4_Click()
Dim ctr As Control
For Each ctr In Me.Controls
If TypeOf ctr Is TextBox Then
    ctr.Text = Empty
End If
Next
End Sub

```

### **coding**

```

Private Sub Command2_Click()
Form5.Show
Data1.Recordset.AddNew
Data1.Recordset.Fields("adhar no") = Text1.Text
Data1.Recordset.Fields("pan card number") = Text2.Text
Data1.Recordset.Fields("e-mail ID") = Text3.Text
Data1.Recordset.Fields("mobile number") = Text4.Text
Data1.Recordset.Update
End Sub

```

```

Private Sub Command4_Click()
Dim ctr As Control
For Each ctr In Me.Controls
    If TypeOf ctr Is TextBox Then
        ctr.Text = Empty
    End If
    Next
End Sub

```

**coding**

```

Private Sub Command1_Click()
train.Show
End Sub

```

**coding**

```

Private Sub Command1_Click()
Form4.Show
Data1.Recordset.AddNew
Data1.Recordset.Fields("name") = Text1.Text
Data1.Recordset.Fields("age") = Text2.Text
Data1.Recordset.Fields("place") = Text3.Text
Data1.Recordset.Fields("train name and number") = Text8.Text
If Option1.Value = True Then
    Data1.Recordset.Fields("seat selection") = "upper"
Else
    Data1.Recordset.Fields("seat selection") = "lower"
End If
If Option6.Value = True Then
    Data1.Recordset.Fields("sex") = "male"
Else
    Data1.Recordset.Fields("sex") = "femal"
End If
If Option10.Value = True Then
    Data1.Recordset.Fields("food") = "veg"
Else
    Data1.Recordset.Fields("food") = "non-veg"
End If
If Option3.Value = True Then
    Data1.Recordset.Fields("class") = "1 class"
Else
    Data1.Recordset.Fields("seat selection") = "2 class"
End If
Data1.Recordset.Fields("date") = Text4.Text
Data1.Recordset.Fields("time") = Text5.Text
Data1.Recordset.Fields("from") = Text6.Text

```

```
Data1.Recordset.Fields("to") = Text7.Text
Data1.Recordset.Update
End Sub
```

```
Private Sub Command4_Click()
Dim ctr As Control
For Each ctr In Me.Controls
If TypeOf ctr Is TextBox Then
    ctr.Text = Empty
End If
Next
End Sub
```

**coding**

```
Private Sub Command2_Click()
Form5.Show
Data1.Recordset.AddNew
Data1.Recordset.Fields("adhar no") = Text1.Text
Data1.Recordset.Fields("pan card number") = Text2.Text
Data1.Recordset.Fields("e-mail ID") = Text3.Text
Data1.Recordset.Fields("mobile number") = Text4.Text
Data1.Recordset.Update
End Sub
```

```
Private Sub Command4_Click()
Dim ctr As Control
For Each ctr In Me.Controls
If TypeOf ctr Is TextBox Then
    ctr.Text = Empty
End If
Next
End Sub
```

**coding**

```
Private Sub Command1_Click()
train.Show
End Sub
```

**coding**

```
Private Sub Command1_Click()
Form8.Show
End Sub
```

```
Private Sub Command2_Click()
Form8.Show
End Sub
```



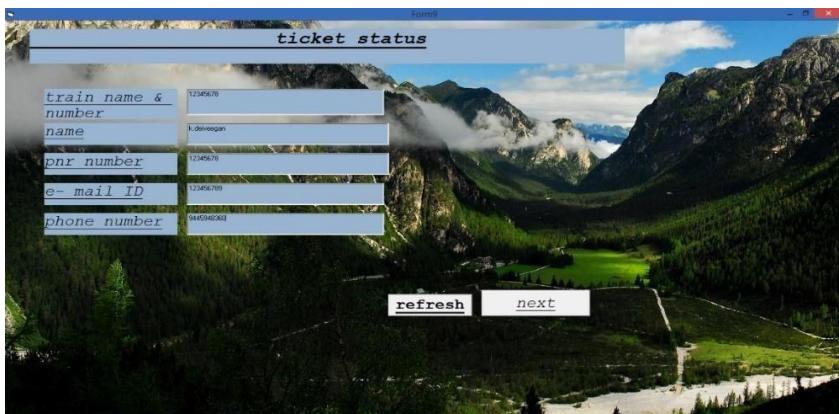
### coding

```

Private Sub Command2_Click()
Form10.Show
Data1.Recordset.AddNew
Data1.Recordset.Fields("train name & number") = Text1.Text
Data1.Recordset.Fields("name") = Text2.Text
Data1.Recordset.Fields("pnr number") = Text3.Text
Data1.Recordset.Fields("e-mail ID") = Text4.Text
Data1.Recordset.Fields("phone number") = Text5.Text
Data1.Recordset.Update
End Sub

Private Sub Command4_Click()
Dim ctr As Control
For Each ctr In Me.Controls
If TypeOf ctr Is TextBox Then
    ctr.Text = Empty
End If
Next
End Sub

```



**coding**

```
Private Sub Command1_Click()
train.Show
End Sub
```

**coding**

```
Private Sub Command2_Click()
Form11.Show
Data1.Recordset.AddNew
Data1.Recordset.Fields("train name & number") = Text1.Text
Data1.Recordset.Fields("name") = Text2.Text
Data1.Recordset.Fields("pnr number") = Text3.Text
Data1.Recordset.Fields("e-mail ID") = Text4.Text
Data1.Recordset.Fields("phone number") = Text5.Text
Data1.Recordset.Update
End Sub
```

```
Private Sub Command4_Click()
Dim ctr As Control
For Each ctr In Me.Controls
    If TypeOf ctr Is TextBox Then
        ctr.Text = Empty
    End If
Next ctr
End Sub
```

**coding**

```
Private Sub Command1_Click()
train.Show
End Sub
```

**coding**

```
Private Sub Command1_Click()
    train.Show
End Sub
```

## 6. TESTING:

Test case ID: Test_01					
Test priority (Low/Medium/High):Medium					
Module name: login					
Test title :verify login with valid username and password					
Precondition: user has invalid username and password					
S. N O	TEST STEPS	EXPECTED RESULTS	ACTUAL RESULTS	STATU S	NOTES
1	Provide valid User name	User should Be able to login	The user is able to move to next Entry	Success	-
2	Provide valid password	User should be Able to Login	The user is able To login Successfully	Success	Incase of wrong Password was given an error Message box was displayed
3	Click login	User should be able to navigate to next page after validation	User name and password is validated and next page is displayed	Success	Incase user gives wrong entry the sign in page remains active
4	Click signup	User should be able to navigate to next page where user enters his credentials	User navigates to the signup page where his user name and password is validated	success	-

## RESULT:

The online ticket reservation system was designed and implemented successfully.

## D. PLATFORM ASSIGNMENT SYSTEM FOR THE TRAINS IN A RAILWAY STATION

### **1. PROBLEM STATEMENT:**

This project is about platform assignment system for the train in a railway station. It contains 3 modules. The first module gets the current time, scheduled time of the train, expected time to enter the station, departure time and platform no to calculate the time delay of the train. If the time delay is more than the specified time also if arrival time and platform no of next train gets clashed then next train cannot be allocated in the same platform.

### **2. OVERALL DESCRIPTION:**

#### **2.1 MODULES:**

- Login
- Train details
- Delay calculation
- Platform assignment

#### **2.2 MODULE DELIVERABLES:**

##### **Login:**

Basicflow: To authenticate the user, the admin has to enter username and password

Alternateflow: If the password is wrong, it will ask the admin to answer security question and retrieve the password

Precondition: The system asks the admin to enter the password

Postcondition: On success, the admin displays the admin information

##### **Train Details:**

Basicflow: The train details like train no, scheduled time, expected time to arrive, scheduled time for departure, expected time to start, platform number and current number is given as input to calculate the time delay.

Alternateflow: If the train details is not correct admin can report it.

Precondition: The train details should be known.

Postcondition: After verifying all the details, admin is moved to the next state.

##### **Delay calculation:**

Basicflow: The entered details are verified with the database and time delay is calculated and also checks whether there is any clash with the next train timing

Alternateflow: If the details are incorrect, error message is generated.

Precondition: The details of the database should be up to date.

Postcondition: After verification the platform assignment is done.

##### **Platform Assignment:**

Basicflow: Based on the delay and arrival of the next train the platform assignment is done.

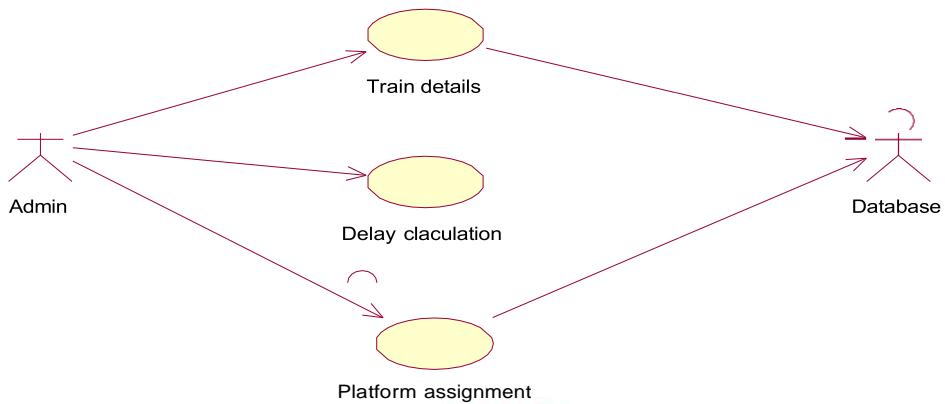
Alternateflow: If no platform is free then the train is made to wait.

Precondition: The availability of the platform should be known.

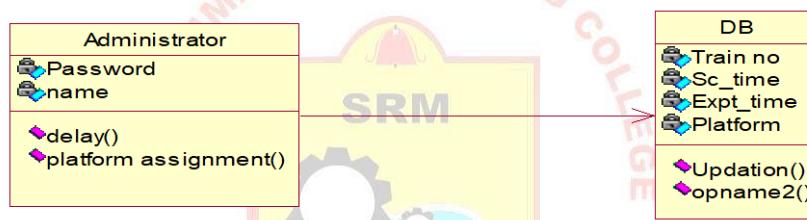
Postcondition: The platform is confirmed and the signal is given for the train to stop.

### 3. UML DIAGRAM:

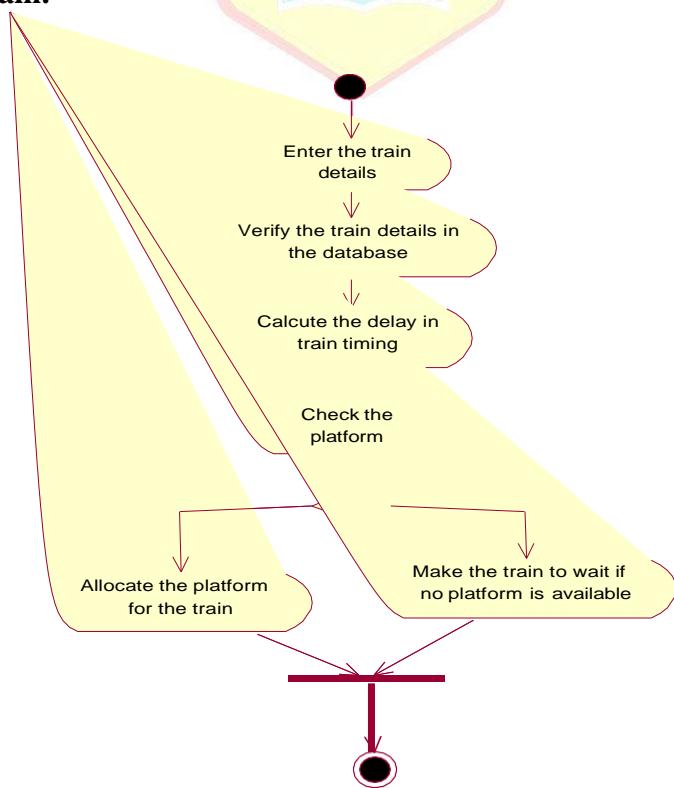
#### 3.1 Usecase Diagram:



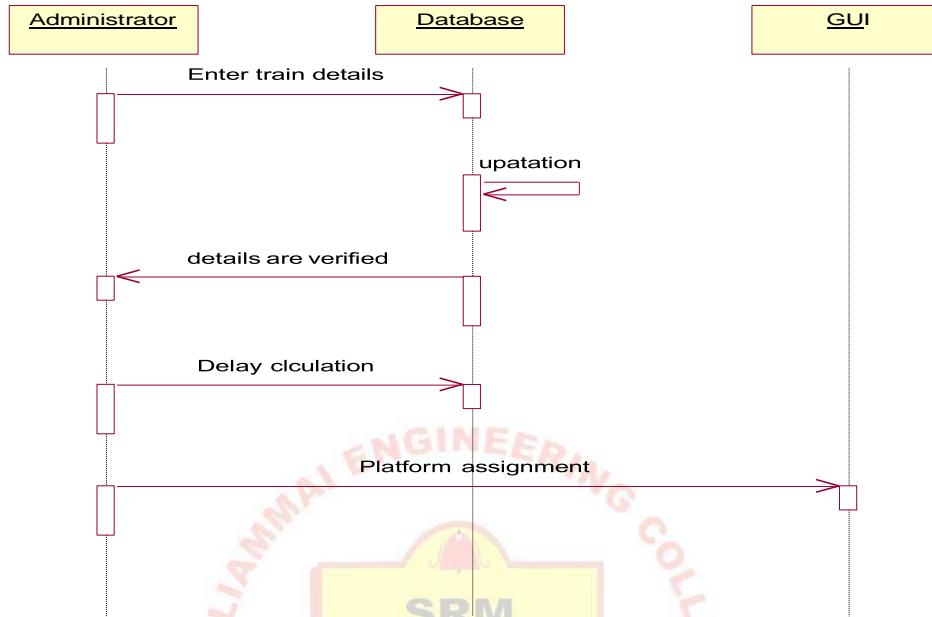
#### 3.2 Class Diagram:



#### 3.3 Activity Diagram:



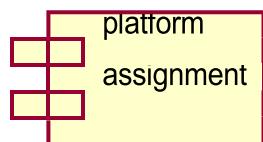
### 3.4 Sequence Diagram:



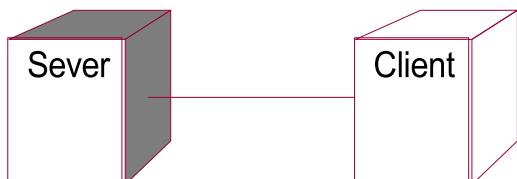
### 3.5 Collaboration Diagram:



### 3.6 Component Diagram:



### 3.7 Deployment Diagram



#### 4. DATABASE DESIGN:

Database name: platform  
 Table name : rail

Fields	Data type
train number	integer
train name	text
platform	integer
timing	text

#### 5. IMPLEMENTATION:

Form1 (Login form)

Coding:

```
Private sub command1_click()
if text1.text="HARIHARAN" and text2.text="1234" then
Form2.show
End if
End sub
Private sub command2_click()
exit
End sub
```

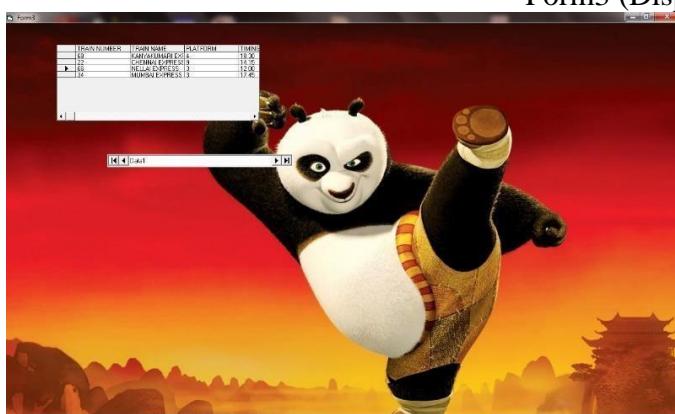
Coding:

```
Private Sub Command1_Click()
Data1.Recordset.addnew
Data1.Recordset.fields("train number") = Text1.Text
Data1.Recordset.fields("train name") = Text2.Text
Data1.Recordset.fields("platform") = Text3.Text
Data1.Recordset.fields("timing") = Text4.Text
Data1.Recordset.Update
End Sub
Private Sub Command2_Click()
Form3.Show
End Sub
```



Form2 (Train details)

Form3 (Display)



## 6. TESTING:

<b>Test case ID: Test_01</b>					
Test priority					
(Low/Medium/High):Medium					
Module name: login					
Test title :verify login with valid username and password					
Precondition: user has invalid username and password					
S.N O	TEST STEPS	EXPECTED RESULTS	ACTUAL RESULTS	STATUS	NOTES
1	Provide valid User name	User should Be able to login	The user is able to move to next Entry	Success	-
2	Provide valid password	User should be Able to Login	The user is able To login Successfully	Success	Incase of wrong Password was given an error Message box was displayed
3	Click login	User should be able to navigate to next page after validation	User name and password is validated and next page is displayed	Success	Incase user gives wrong entry the sign in page remains active
4	Click signup	User should be able to navigate to next page where user enters his credentials	User navigates to the signup page where his user name and password is validated	success	-

## RESULT:

The Platform assignment system was designed and implemented successfully.

## E. EXPERT SYSTEM FOR PRESCRIBING MEDICINE FOR GIVEN SYMPTOMS

### 1. PROBLEM STATEMENT:

The project is mainly focused on medical field in expert system where the person or patient login to the system and must select what are the symptoms for him that field information will be given to expert system. Expert system will diagnose what type of disease using medical database. The medical database will get all the information about medicine from pharmacy module and generate the prescription for the right symptoms and it gives to expert system. The medical database gives types of disease information to expert system.

### 2. OVERALL DESCRIPTION:

#### 2.1 MODULES:

- Symptoms
- Expert System
- Medical DB
- Prescription
- Pharmacy

#### 2.2 MODULE DELIVERABLES:

##### SYMPTOMS:

Basic Flow:

Patient must enter the symptoms that occurs for him. The symptoms must be exact so that it can be verified.

Alternate Flow:

If there are 4 symptoms then the patient is having big problem disease so, exit. If there are more than 4 then it is a serious problem

Precondition:

All the symptoms related to the disease are entered. Patient must enter the symptoms that occurs correctly.

Postcondition:

If the symptoms are corresponding to particular disease they are entered in expert system.

##### PRESCRIPTION:

Basic Flow:

The medicine is prescribed based on the symptoms. Medical prescription is given by expert system.

Alternate Flow:

If medicine is not available then exit. If there is no medicine then it comes under stock unavailable condition so update information in medical DB.

Precondition:

If medicine is only available in pharmacy it will generate prescription. If medicine is not available then store the details in expert system.

Postcondition:

If medicine are prescribed it will give to patient. The patient then gets the prescribed medicine in the shop.

##### PHARMACY:

**Basic Flow:**

Based on disease it will give medicine. Pharmacy gets the disease of prescribed medicine from the expert system.

**Alternate Flow:**

If medicine are not available it will pass the information as no medicine available

Precondition:

If prescribed medicine is available in pharmacy expert system must generate the required code.

Postcondition:

If prescribed medicine is available in pharmacy then expert system must load the data that is available.

### **EXPERT SYSTEM:**

Basic Flow:

It will get symptoms and information from medical database to diagonise disease

Alternate Flow:

If any of the condition fails then system will exit.

Precondition:

Medical database and symptoms of patient should be present

Postcondition:

Prescribe the medicine to the patient from expert system to the patient.

### **MODEL DATABASE:**

Basic Flow:

Medical database contains the backup and additional details which is not there in expert system.

Alternate Flow:

If medical database from expert system is not validated in medical database then first we have to update them

Precondition:

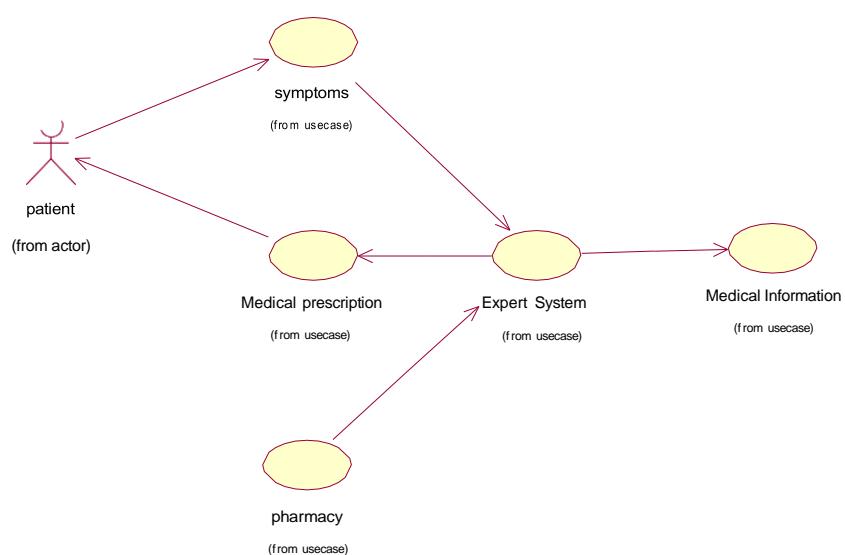
Medical database must contain some basic medicine stored in the memory of the system or computer

Postcondition:

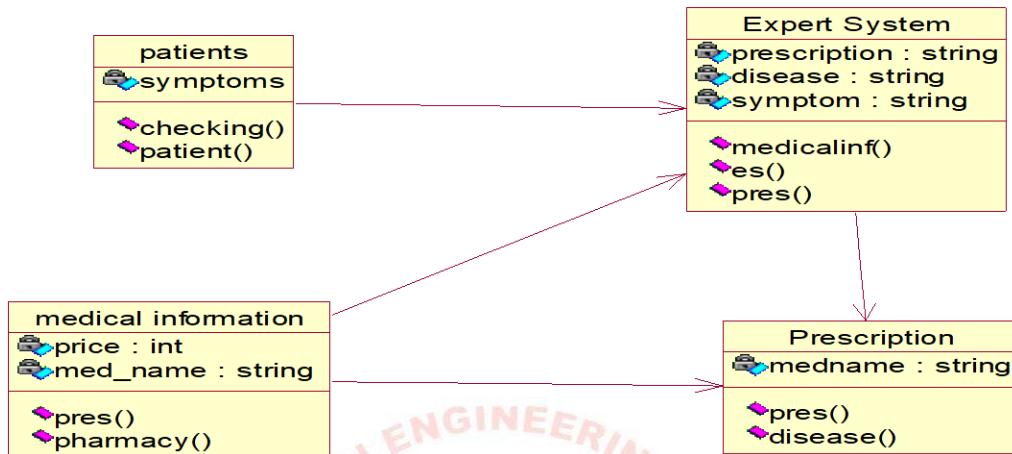
If the details which is not present in expert system contained in database then we can extract information.

### **3. UML DIAGRAM:**

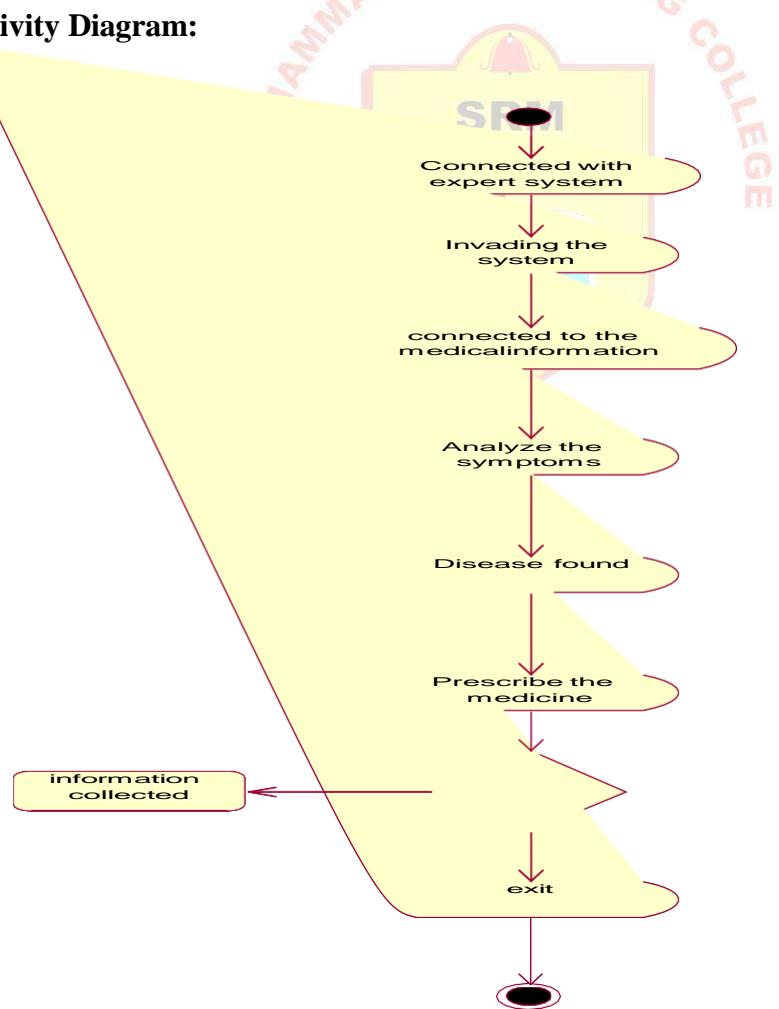
#### **3.1 Usecasediagram:**



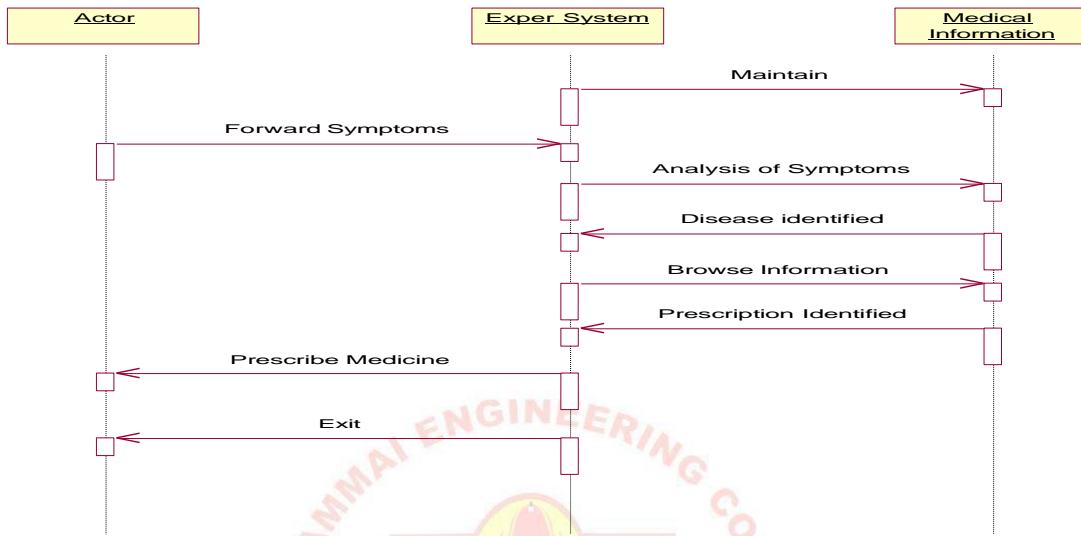
### 3.2 Class Diagram:



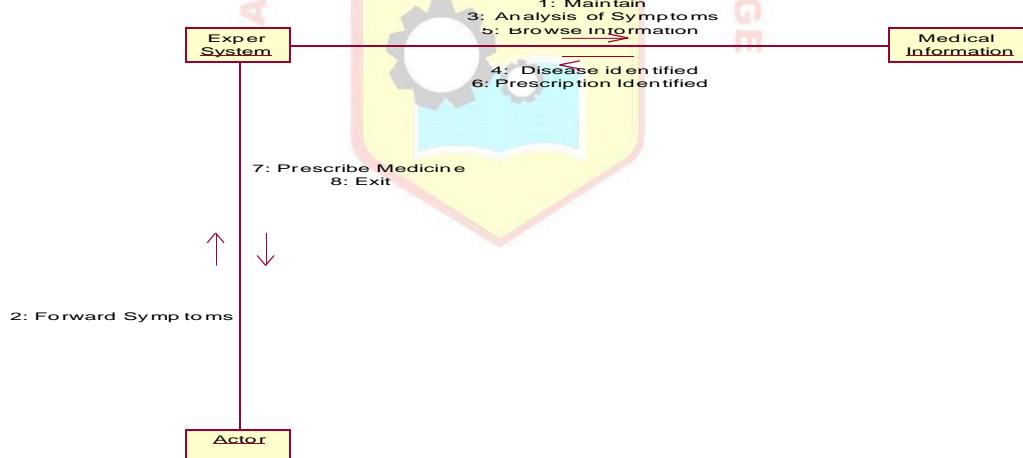
### 3.3 Activity Diagram:



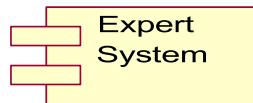
### 3.4 Sequence Diagram:



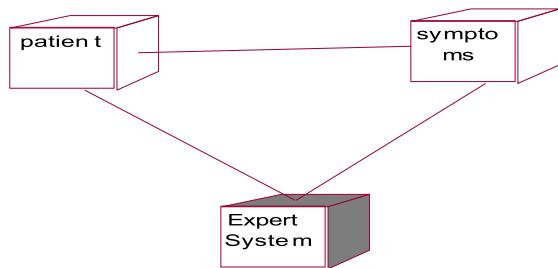
### 3.5 Collaboration Diagram:



### 3.6 Component Diagram:



### 3.7 Deployment Diagram



#### 4. DATABASE DESIGN

Database Name: Medical  
Table Name : Patient

Fields	Data type
pid	Integer
pname	Text
age	Integer
gender	Text
symptoms	Text
drug	Text

#### 5. IMPLEMENTATION:

Form1 (Login form)

Coding:

```
Private Sub Command1_Click()
Form2.Show
End Sub
```

```
Private Sub Command2_Click()
End
End Sub
```



Form 2 (Main menu)

Coding:

```
Private Sub Command1_Click()
Form3.Show
End Sub
```

```
Private Sub Command2_Click()
Form4.Show
End Sub
```

```
Private Sub Command3_Click()
Form1.Show
End Sub
```

FORM 3 (Patient data entry)

Coding:

```
Private Sub Command1_Click()
Data1.Recordset.AddNew
Data1.Recordset.Fields("pID") = Text1.Text
Data1.Recordset.Fields("pName") = Text2.Text
Data1.Recordset.Fields("age") = Text3.Text
If Option1.Value = True Then
```

```

Data1.Recordset.Fields("Gender") = "Male"
Else
Data1.Recordset.Fields("Gender") = "Female"
End If
Data1.Recordset.Fields("Symptoms") = Combo2.Text
Data1.Recordset.Update
MsgBox ("Data Added")
End Sub

```

```

Private Sub Command2_Click()
Form2.Show
End Sub

```

```

Private Sub Form_Load()
Combo2.AddItem ("Fever")
Combo2.AddItem ("Cough")
Combo2.AddItem ("Cold")
End Sub

```

Coding:

```

Private Sub Command1_Click()
Data1.Recordset.Edit
Data1.Recordset.Fields("Drug") = drug.Text
Data1.Recordset.Update
MsgBox ("Data Added")
End Sub

```

```

Private Sub Command2_Click()
Form2.Show
End Sub

```

```

Private Sub Command3_Click()
Data1.Recordset.MoveFirst
Do While Not Data1.Recordset.EOF
If Data1.Recordset.Fields("pid") = pid.Text Then
pid.Text = Data1.Recordset.Fields("pid")
pname.Text = Data1.Recordset.Fields("pname")
age.Text = Data1.Recordset.Fields("age")
gender.Text = Data1.Recordset.Fields("Gender")
symptoms.Text = Data1.Recordset.Fields("Symptoms")
GoTo out
End If
Data1.Recordset.MoveNext
Loop
MsgBox ("not found")
out:
End Sub

```



## 6. TESTING:

<b>Test case ID: Test_01</b>					
Test priority					
(Low/Medium/High):Medium					
Module name: login					
Test title :verify login with valid username and password					
Precondition: user has invalid username and password					
S.N O	TEST STEPS	EXPECTED RESULTS	ACTUAL RESULTS	STATUS	NOTES
1	Provide valid User name	User should Be able to login	The user is able to move to next Entry	Success	-
2	Provide valid password	User should be Able to Login	The user is able To login Successfully	Success	Incase of wrong Password was given an error Message box was displayed
3	Click login	User should be able to navigate to next page after validation	User name and password is validated and next page is displayed	Success	Incase user gives wrong entry the sign in page remains active
4	Click signup	User should be able to navigate to next page where user enters his credentials	User navigates to the signup page where his user name and password is validated	success	-

## RESULT:

The Expert system to prescribe the medicines for the given symptoms was designed and implemented successfully.

F.

## REMOTE COMPUTER MONITORING

### 1. PROBLEM STATEMENT

For Organizations with huge data centers having a lot of servers hosting numerous applications, it is always a major problem to monitor if each of the servers is up and functional all the time. The problem is more acute during late night shifts when the usual number of network/systems engineers working is less. Usually, when organizations host the applications on their servers on behalf of their clients, they sign-up a service level agreement (SLA), specifying the allowed down time for each of the applications. Any lack of commitment on the part of the organizations in meeting the SLA could result in loss of business or legal action or both.

### 2. OVERALL DESCRIPTION:

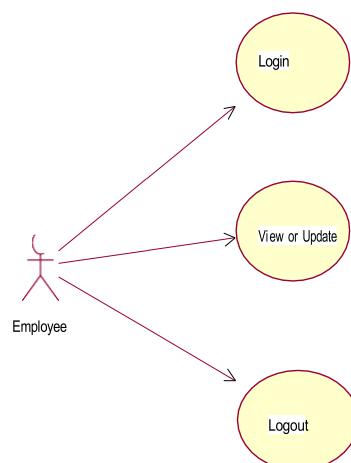
#### 2.1 MODULES:

Organization Information management Module: Allows Admin users to capture and update Organization information related to Users, Specialists, Servers, IP addresses etc

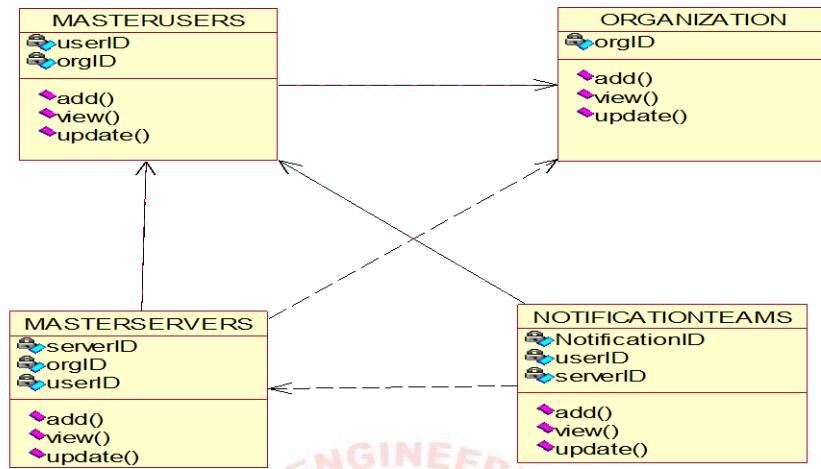
- **Access management Module:** Allows Admin users to give admin privileges to other users as well as managing the userid/passwords of all the Network/System engineers
- **Automated Server Monitoring Module:** Runs continuously to check if each of the servers is up and running and logs failures into a database.
- **SMS Failure Notification Module:** Sends out SMS to the specified list of mobile numbers along with the failed server information.
- **Corrective Action Module:** Allows Network/Systems engineers to put in the corrective action they have taken to rectify the failure.
- **Management Reporting:** Allows Admin users to run reports based on Organization, Servers, Specialists and Corrective action taken.
- **Archiving and cleanup Module:** Allows Admin users to Archive/Cleanup old data on the system

### 3. DESIGN:

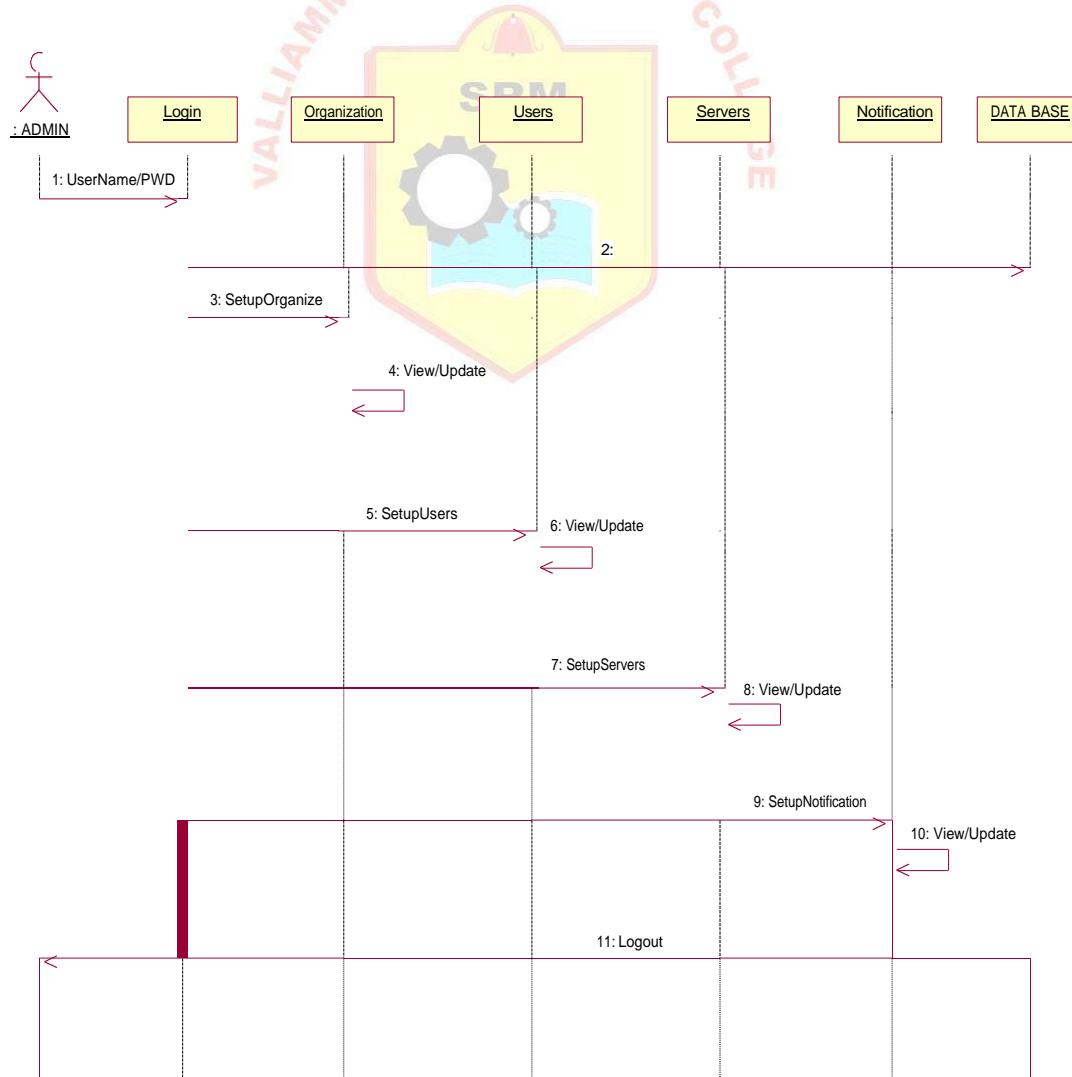
#### 3.1 USECASE DIAGRAM:



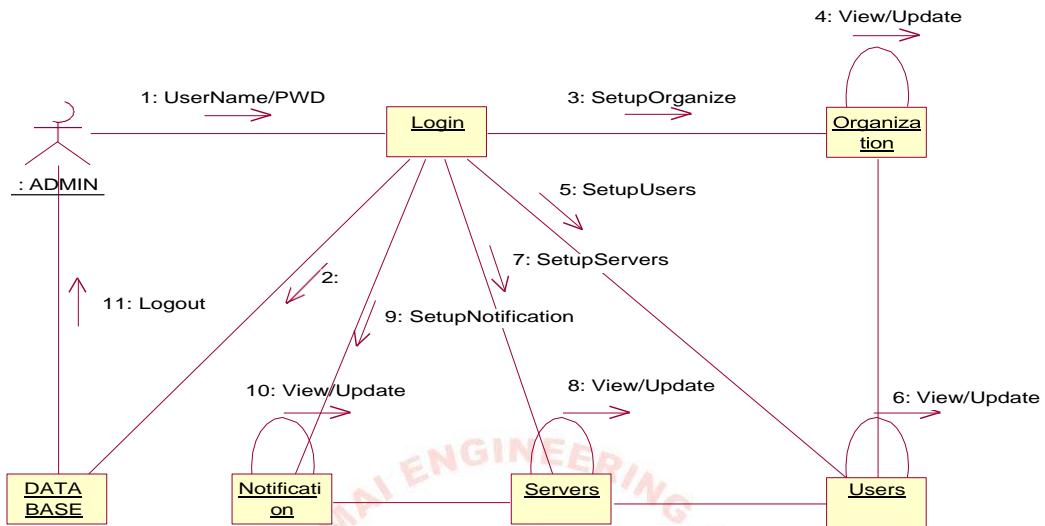
### 3.2 CLASS DIAGRAM:



### 3.3 SEQUENCE DIAGRAM:



### 3.4 COLLABORATION DIAGRAM:



### 4. DATABASE DESIGN

Database Name: Empl  
Table Name : Employee

Fields	Data type
Id	Integer
Name	Text
Address	Text
Project	Text
Domain	Text

### 5. IMPLEMENTATION:

Form1 (Login form)



Coding:

```
Private Sub BACK_Click()
```

```
Tsys.Show
```

```
End Sub
```

```
Private Sub del_Click()
```

```
Data1.Recordset.MoveFirst
```

```
Do While Not Data1.Recordset.EOF
```

```
If Data1.Recordset.Fields("id") = empid.Text Then
```

```
Data1.Recordset.delete
```

```
MsgBox ("Record deleted")
```

```
GoTo out
```

```
End If
```

```
Data1.Recordset.MoveNext
```

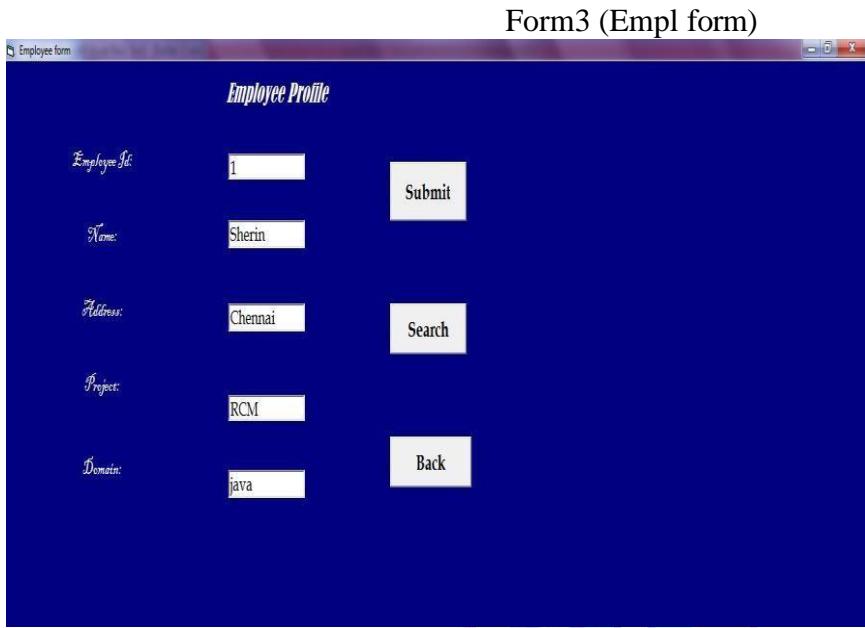
```
Loop
```

```
MsgBox ("Not found")
```

```
out:
```

```
End Sub
```

```
Private Sub delete_Click()
Tsys.Hide
Deldet.Show
End Sub
```



Coding:

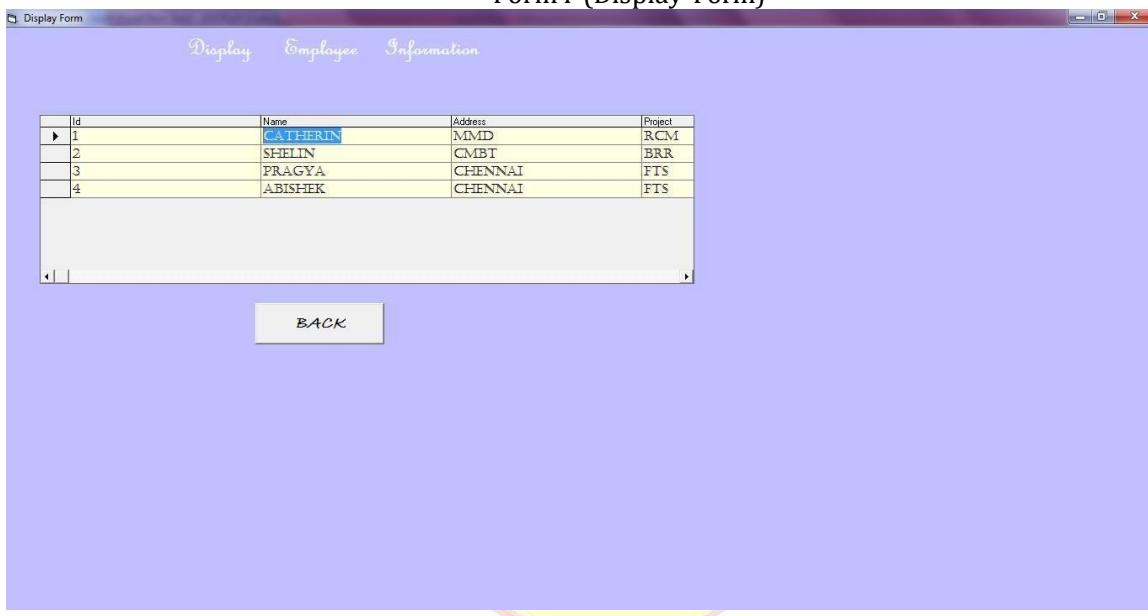
```
Private Sub BACK_Click()
Tsys.Show
End Sub
```

```
Private Sub submit_Click()
Data1.Recordset.AddNew
Data1.Recordset.Fields("Id") = id.Text
Data1.Recordset.Fields("Name") = name1.Text
Data1.Recordset.Fields("Address") = address.Text
Data1.Recordset.Fields("Project") = project.Text
Data1.Recordset.Fields("Domain") = domain.Text
Data1.Recordset.Update
MsgBox "Inserted successfully"
End Sub

Private Sub search_Click()
Data1.Recordset.MoveFirst
Do While Not Data1.Recordset.EOF
If Data1.Recordset.Fields("id") = id.Text Then
id.Text = Data1.Recordset.Fields("Id")
name1.Text = Data1.Recordset.Fields("Name")
address.Text = Data1.Recordset.Fields("Address")
project.Text = Data1.Recordset.Fields("Project")
domain.Text = Data1.Recordset.Fields("Domain")
End If
Loop
End Sub
```

```
GoTo out
End If
Data1.Recordset.MoveNext
Loop
MsgBox ("Not Found")
out:
End Sub
```

Form4 (Display Form)



## 6. TESTING:

<b>Test case ID: Test_01</b>					
Test priority					
(Low/Medium/High):Medium					
Module name: login					
Test title :verify login with valid username and password					
Precondition: user has invalid username and password					
S.N O	TEST STEPS	EXPECTED RESULTS	ACTUAL RESULTS	STATUS	NOTES
1	Provide valid User name	User should Be able to login	The user is able to move to next Entry	Success	-
2	Provide valid password	User should be Able to Login	The user is able To login Successfully	Success	Incase of wrong Password was given an error Message box was displayed
3	Click login	User should be able to navigate to next page after validation	User name and password is validated and next page is displayed	Success	Incase user gives wrong entry the sign in page remains active
4	Click signup	User should be able to navigate to next page where user enters his credentials	User navigates to the signup page where his user name and password is validated	success	-

## RESULT:

The Remote computer monitoring system was designed and implemented successfully.

## G. ATM SYSTEM

### **1. PROBLEM STATEMENT:**

This project on atm is based on the processing of debit cards in atm. There are two main section and the administrator section. In withdraw module the requested amount is deducted from the user's account if sufficient balance is available. If the balance in the user's account is insufficient the service is denied. In deposit module the deposited amount is added to the user's balance. In balance enquiry module the amount that is present in the user's account is displayed. The user must enter his pin and account number to perform any of the transactions.

### **2. OVERALL DESCRIPTION:**

#### **2.1 MODULES:**

- Login
- Deposit
- Withdraw
- Changing pin number
- Balance enquiry

#### **2.1.1 MODULE DELIVERABLES:**

##### **2.1.1.1 LOGIN:**

###### **Basic Flow :**

User enters correct pin and logs into the machine

###### **Alternative Flow :**

User enters wrong pin and the card gets rejected

###### **Pre-condition :**

User must know the correct pin number.

##### **2.1.1.2 DEPOSIT AMOUNT:**

###### **Basic Flow:**

User enters the amount and feds cash to machine.

###### **Alternative Flow:**

User cancels transaction.

###### **Pre condition:**

User should know the amount to be deposited.

###### **Post condition:**

User balance gets updated in the account.

##### **2.1.1.3 WITHDRAWAL:**

###### **Basic Flow:**

User enters amount and gets cash from machine.

###### **Alternative Flow:**

Not enough cash in account and so the card gets rejected.

###### **Pre condition:**

User knows the amount to withdraw.

###### **Post condition:**

Balance is updated in account.

##### **2.1.1.4 BALANCE ENQUIRY:**

###### **Basic Flow:**

User enquires about balance and the balance is displayed.

#### **Alternative Flow:**

User cancels transaction.

#### **2.1.5 DELETE ACCOUNT:**

#### **Basic Flow:**

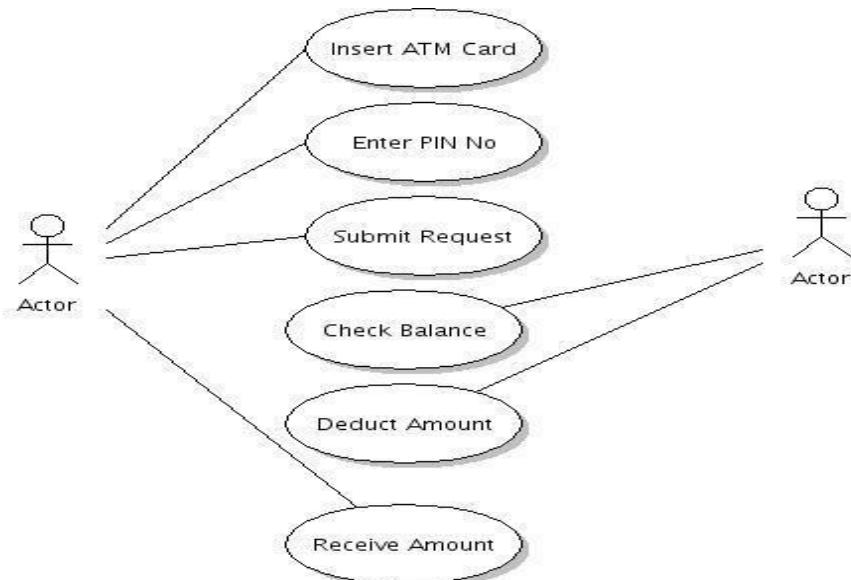
User confirms deletion and account gets deleted.

#### **Alternative Flow:**

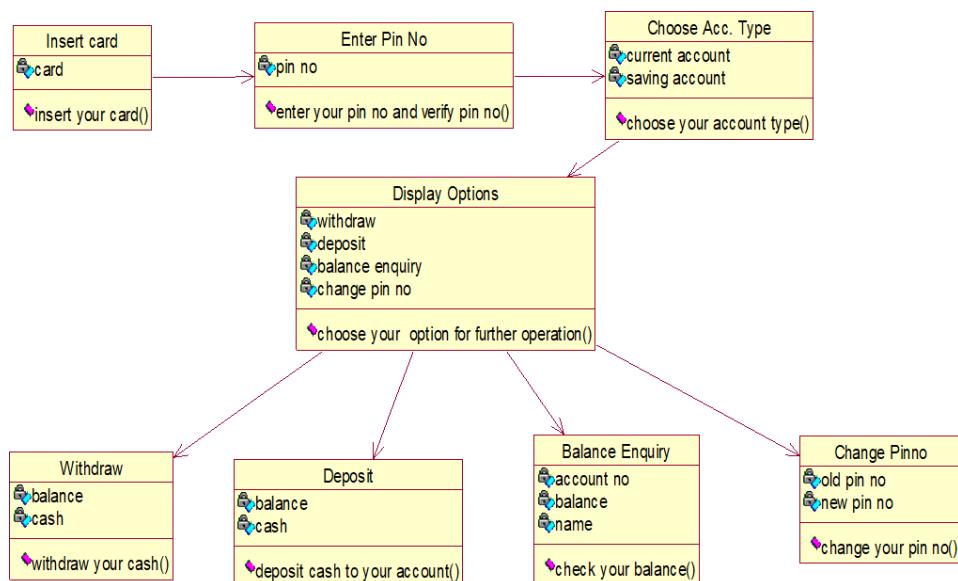
User doesn't confirm and resumes transaction.

### **3. DESIGN**

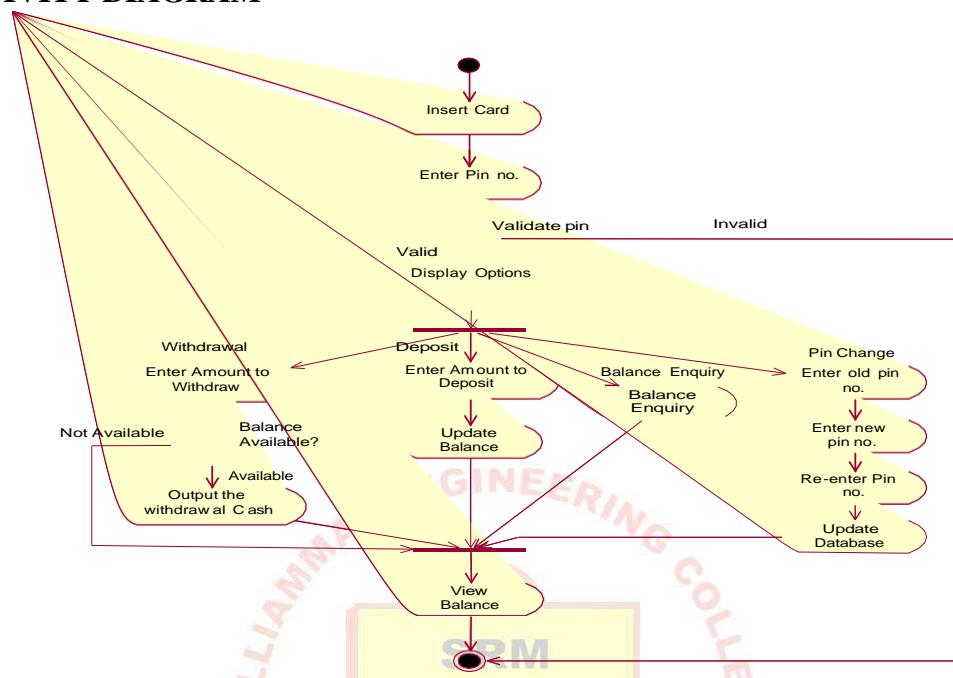
#### **3.1 USECASE DIAGRAM**



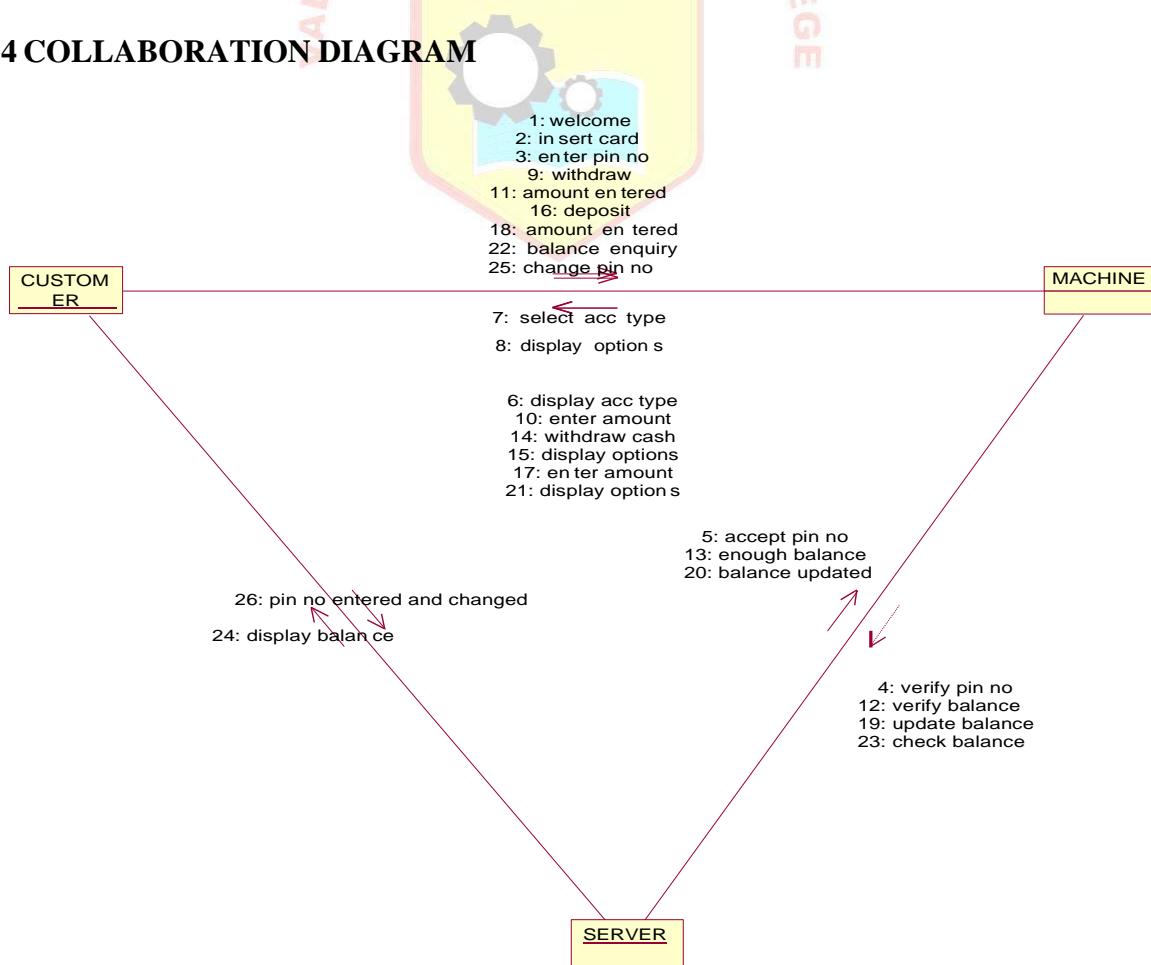
#### **3.2 CLASS DIAGRAM**



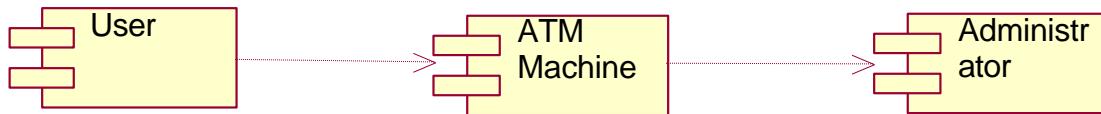
### 3.3 ACTIVITY DIAGRAM



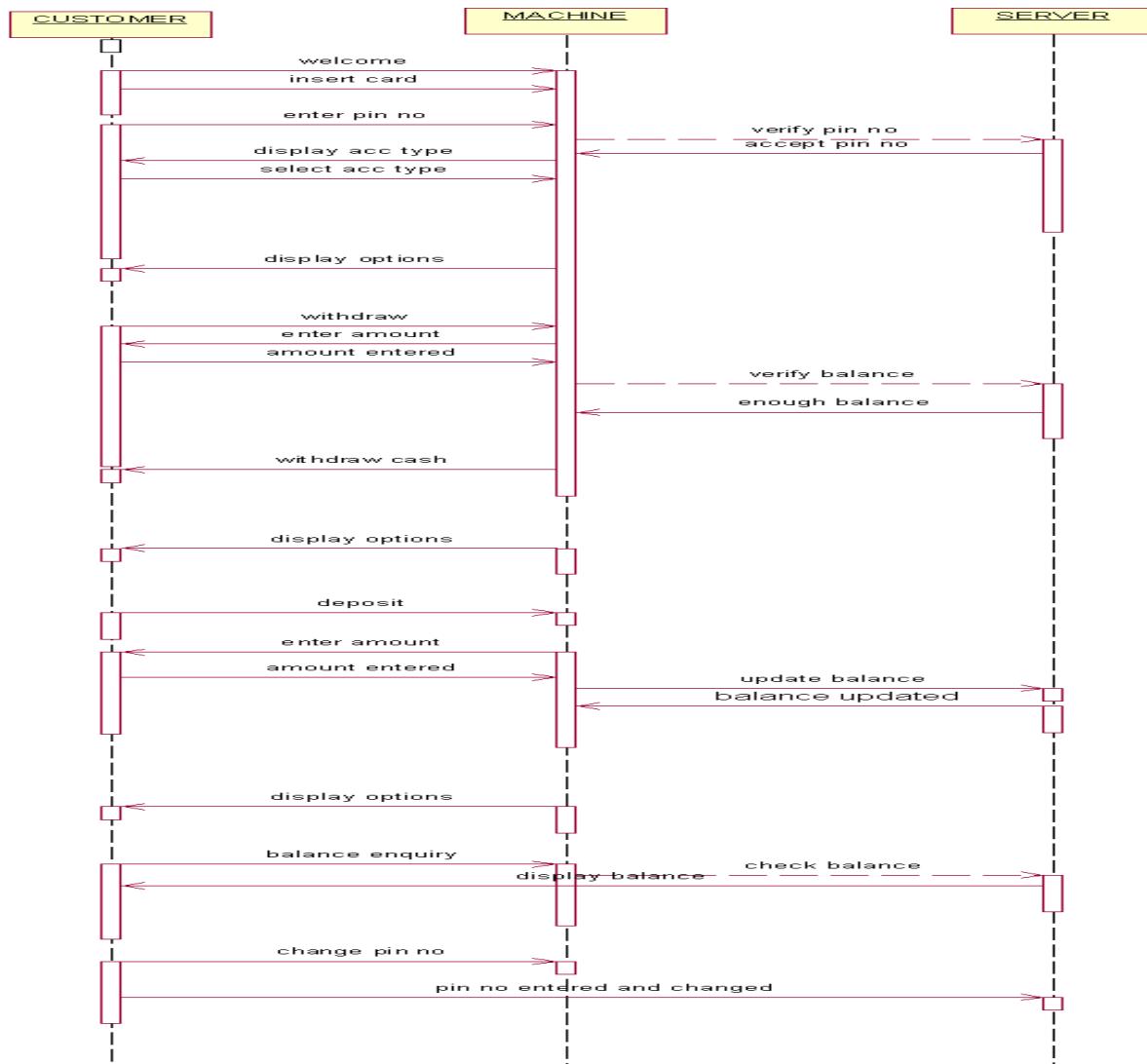
### 3.1.4 COLLABORATION DIAGRAM



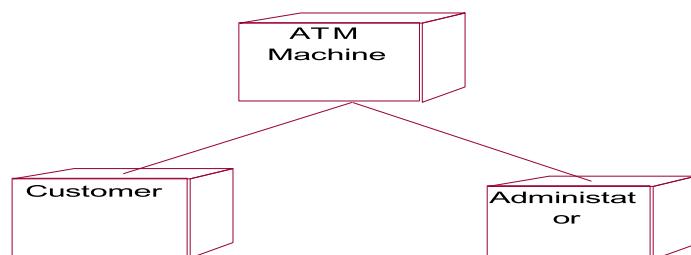
### 3.1.7 COMPONENT DIAGRAM



### 3.1.5 SEQUENCE DIAGRAM



### 3.1.5 DEPLOYMENT DIAGRAM



#### 4. DATABASE DESIGN

Database Name: ATM  
 Table Name : Infor atm

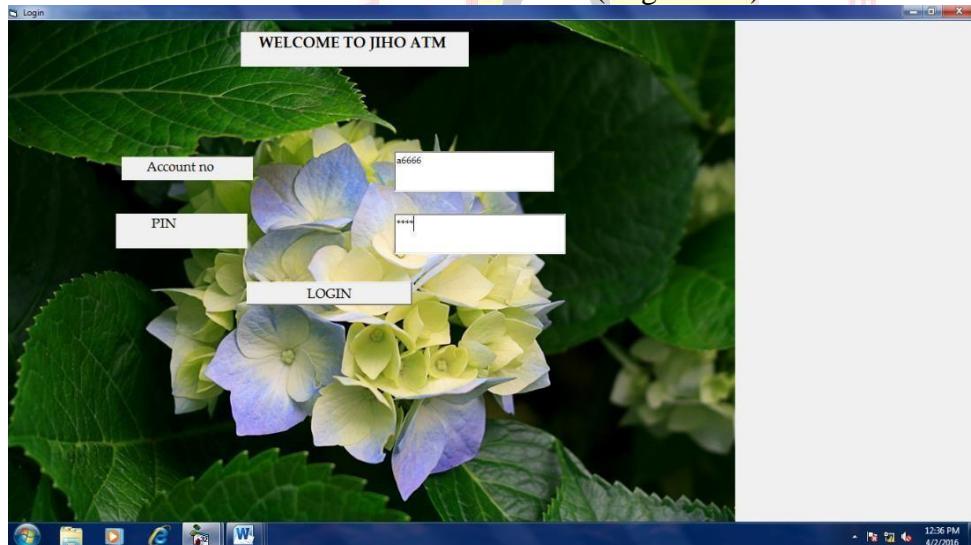
Fields	Data type
ACC NO	Text
USER NAME	Text
PIN NO	Integer
ACCOUNT BALANCE	Double(8)

Table data:

ACC NO	PIN NO	USER NAME	ACCOUNT BALANCE
a6666	9999	Magdalene	Rs.19000
b7777	7777	hhhh	Rs.35000

#### 5. IMPLEMENTATION:

SRM  
 Form1 (Login form)



#### Coding:

```

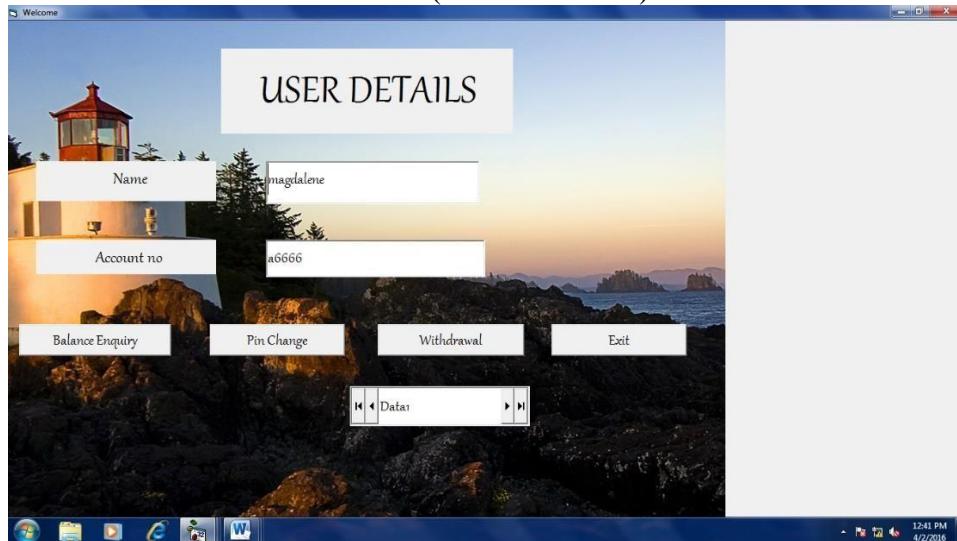
Private Sub Command1_Click()
Data1.Recordset.MoveFirst
While Not Data1.Recordset.EOF
If ((Text1.Text = Data1.Recordset.Fields("AccNO")) And (Text2.Text =
Data1.Recordset.Fields("PIN"))) Then
Welcome.Text1.Text = Data1.Recordset.Fields("UserName")
Welcome.Text2.Text = Data1.Recordset.Fields("AccNO")
Welcome.Show
GoTo out
End If
End Sub
  
```

```

Else
Data1.Recordset.MoveNext
End If
MsgBox ("Enter correct AccNO and PIN")
Wend
out:
End Sub

```

Form2 (Welcome Form)

Coding:

```

Private Sub Command1_Click()
Data1.Recordset.MoveFirst
While Not Data1.Recordset.EOF
If ((Text1.Text = Data1.Recordset.Fields("UserName")) And (Text2.Text =
Data1.Recordset.Fields("accno"))) Then
MsgBox (Data1.Recordset.Fields("Account Balance"))
Balanceenquiry.Text1.Text = Data1.Recordset.Fields("Account Balance")
Balanceenquiry.Show
GoTo out
Else
Data1.Recordset.MoveNext
End If
MsgBox ("check user name and account number")
Wend
out:
End Sub

```

```
Private Sub Command2_Click()
```

```
Pinchange.Show
```

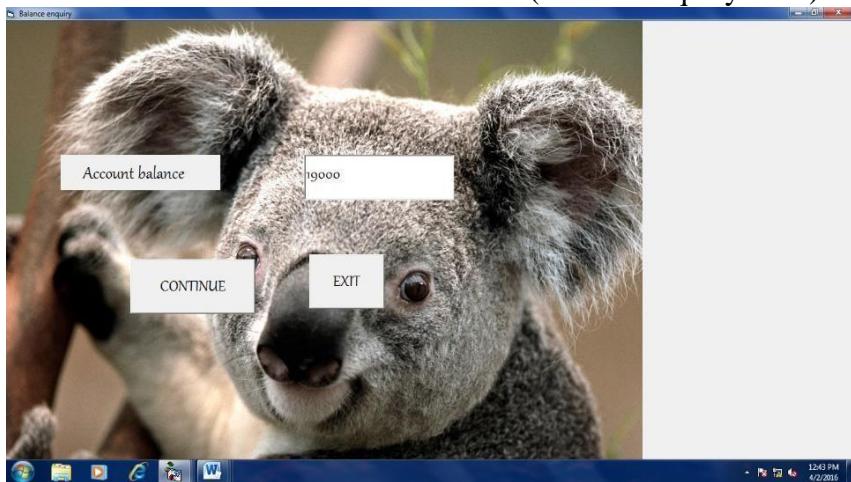
```
End Sub
```

```
Private Sub Command3_Click()
```

```
Withdrawal.Show
```

```
End Sub
```

Form3(Balanceenquiry form)

Coding:

```
Private Sub Command1_Click()
```

```
Welcome.Show
```

```
End Sub
```

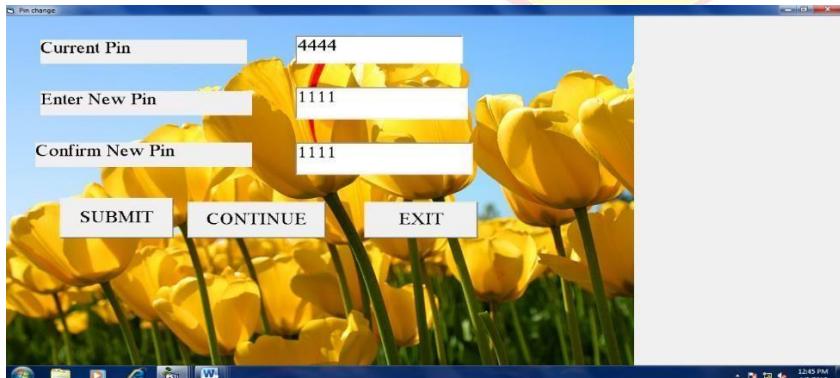
```
Private Sub Command2_Click()
```

```
End
```

```
End Sub
```



Form4 (Pinchange form)

Coding:

```
Private Sub Command1_Click()
```

```
Data1.Recordset.MoveFirst
```

```
Data1.Recordset.Edit
```

```
While Not Data1.Recordset.EOF
```

```
If (Text1.Text = Data1.Recordset.Fields("PIN")) Then
```

```
Data1.Recordset.Fields("PIN") = Text3.Text
```

```
MsgBox ("Pin Change Success")
```

```
Data1.Recordset.Update
```

```
GoTo out
```

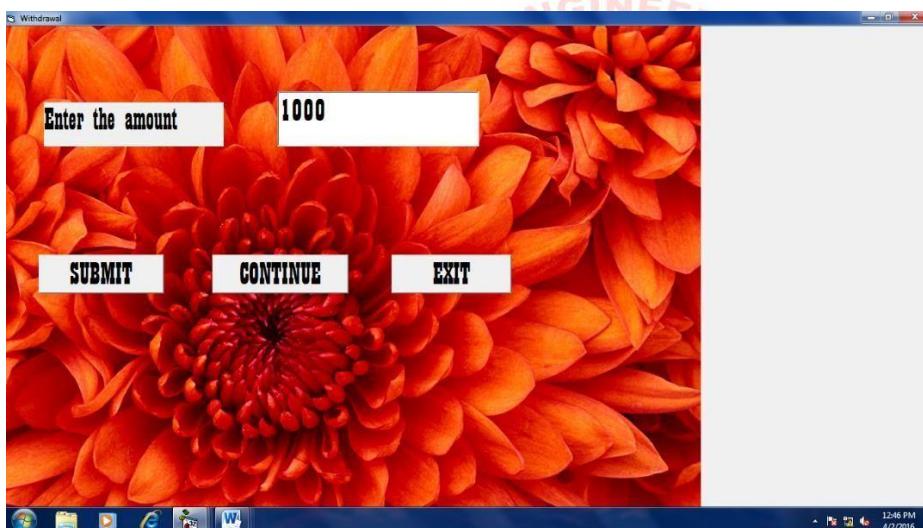
```
Else
```

```

Data1.Recordset.MoveNext
End If
MsgBox ("Pin change not succesful")
Wend
out:
End Sub
Private Sub Command2_Click()
Welcome.Show
End Sub
Private Sub Command3_Click()
End
End Sub

```

Form5 (Withdrawal form)

Coding:

```

Private Sub Command1_Click()
Dim s
Welcome.Data1.Recordset.Edit
If Welcome.Data1.Recordset.Fields("Account Balance") > Text1.text then
s=val(Welcome.Text2.text) – val(Text1.text)
Welcome.Data1.Recordset.Fields("Account Balance") = s
Welcome.Data1.Recordset.Update
Else
Msgbox("Insufficient amount")
End if
End Sub
Private Sub Command2_Click()
Welcome.Show
End Sub
Private Sub Command3_Click()
End
End Sub

```

## 6.TESTING:

<b>Test case ID: Test_01</b>					
Test priority					
(Low/Medium/High):Medium					
Module name: login					
Test title :verify login with valid username and password					
Precondition: user has invalid username and password					
S.N O	TEST STEPS	EXPECTED RESULTS	ACTUAL RESULTS	STATUS	NOTES
1	Provide valid User name	User should Be able to login	The user is able to move to next Entry	Success	-
2	Provide valid password	User should be Able to Login	The user is able To login Successfully	Success	Incase of wrong Password was given an error Message box was displayed
3	Click login	User should be able to navigate to next page after validation	User name and password is validated and next page is displayed	Success	Incase user gives wrong entry the sign in page remains active
4	Click signup	User should be able to navigate to next page where user enters his credentials	User navigates to the signup page where his user name and password is validated	success	-

## RESULT:

The ATM system was designed and implemented successfully.

## H. STOCK MAINTENANCE SYSTEM

### 1. PROBLEM STATEMENT

Inventory control is used to keep track of stocks in an industry. Its main purpose is to update stock when it reaches an optimum level. This System will provide a search functionality to facilitate the search of resources. This search will be based on various categories viz. results to search various categories simultaneously. Further the Student can add/update/remove the resources and the resource users from the system.

### 2. OVERALL DESCRIPTION:

#### 2.1 Modules:

The features that are available to the telephone directory are:

- ✓ Selection of items from the list.
- ✓ Addition of item and deletion of item.
- ✓ Updating of stock
- ✓ Cost estimation for selected items.
- ✓ View the available stock.

#### 2.2 User Classes and Characteristics

There are various kinds of users for the product. Usually web products are visited by various users for different reasons. The users include: Students who will be using the above features by accessing

#### 2.3 Operating Environment

The product will be operating in windows environment. Also it will be compatible with the IE 6.0. Most of the features will be compatible with the Mozilla firefox & Opera 7.0 or higher version. The only requirement to use this online product would be the internet connection.

#### 2.4 User Documentation

The product will include user manual. The user manual will include product overview, complete configuration of the used software (such as SQL server), technical details, backup procedure and contact information which will include email address. There will be no online help for the product at this moment. The product will be compatible with the Internet Explorer 6.0 or higher. The databases will be created in the Microsoft access

### 3. System Features

#### 3.1. Database – Storage

#### 3.2. Functional Requirements

This section gives the list of Functional and non functional requirements which are applicable to the Student Marks Analyzing System

##### 3.2.1 Interface Requirements

This section describes how the software interfaces with other. This section describes how the software interfaces with other software products or users for input or output.

###### 3.2.1.1 UserInterfaces

Describes how this product interfaces with the user GUI .Describes the graphical user interface if present. This section should include a set of screen dumps or mockups to illustrate user interface features.

#### 1. Description

The user interface must be customizable by the administrator

## **2. Criticality**

This issue is essential to the overall system. All the modules provided with the software must fit into this graphical user interface and accomplish to the standard defined.

## **3. Technical issues**

In order to satisfy this requirement the design should be simple and all the different interfaces should follow a standard template. There will be the possibility of changing colors and images, plus switching between interfaces with the minimum impact for the users.

## **4. Risks**

To reduce the circumstances under which this requirement might not able to be satisfied, all the designers must have been developed web sites previously and they must be aware of html restriction and cross browsers implementations before starting the designing. In order to reduce the probability of this occurrence the entire design team will be trained in basic html development and macromedia fireworks, this tool will be used instead of Photoshop.

## **5. Dependencies with other requirements**

All user interfaces should be able to interact with the user management module and a part of the interface must be dedicated to the login/logout module

### **Non Functional Requirements**

4.1. User Interfaces

4.2. Hardware Interfaces

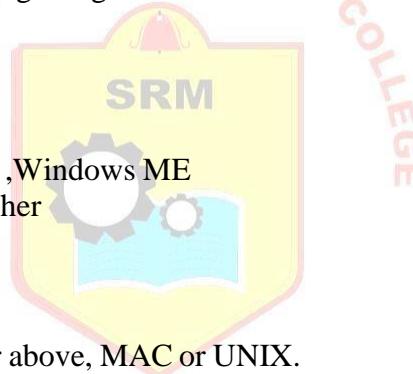
#### **Server Side:**

Operating System: Windows 9x/xp ,Windows ME

Processor: Pentium 3.0 GHz or higher

RAM: 256 Mb or more

Hard Drive: 10 GB or more



#### **Client side:**

Operating System: Windows 9x or above, MAC or UNIX.

Processor: Pentium III or 2.0 GHz or higher.

RAM: 256 Mb or more

### **4.3. Software Interfaces**

Database: Ms Access

## **5. Other Nonfunctional Requirements**

### **5.1 Performance Requirements**

The proposed system that we are going to develop will be used as the Chief performance system within the different documentation purpose. Therefore, it is expected that the database would perform functionally all the requirements that are specified by the university.

### **5.2 Safety Requirements**

The database may get crashed at any certain time due to virus or operating system failure. Therefore, it is required to take the database backup

### **5.3 Security Requirements**

We are going to develop a secured database for the university .There are different categories of users namely teaching staff, administrator, library staff ,students etc.,Depending upon the category of user the access rights are decided .It means if the user is an administrator then he can be able to modify the data, delete, append etc., All other users other than library staff only have the rights to retrieve the information about database.

### **Software Quality Attributes**

The Quality of the database is maintained in such a way so that it can be very user friendly to all the users of the database

#### **5.4 Hardware Constraints**

The system requires a database in order to store persistent data. The database should have backup capabilities.

#### **5.5 Software Constraints**

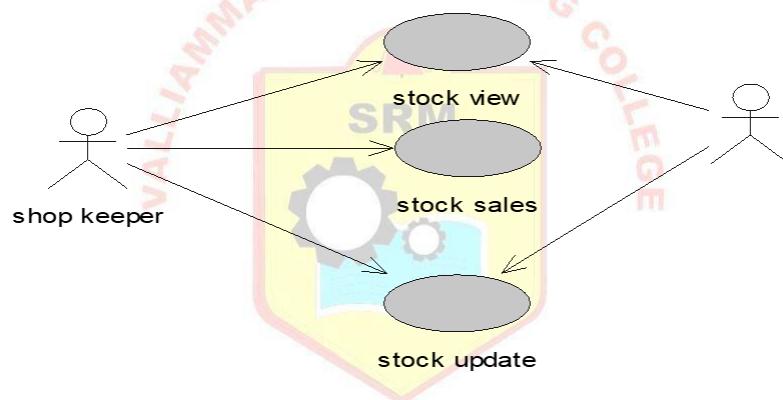
The development of the system will be constrained by the availability of required software such as web servers, database and development tools.

#### **5.6 Design Constraints**

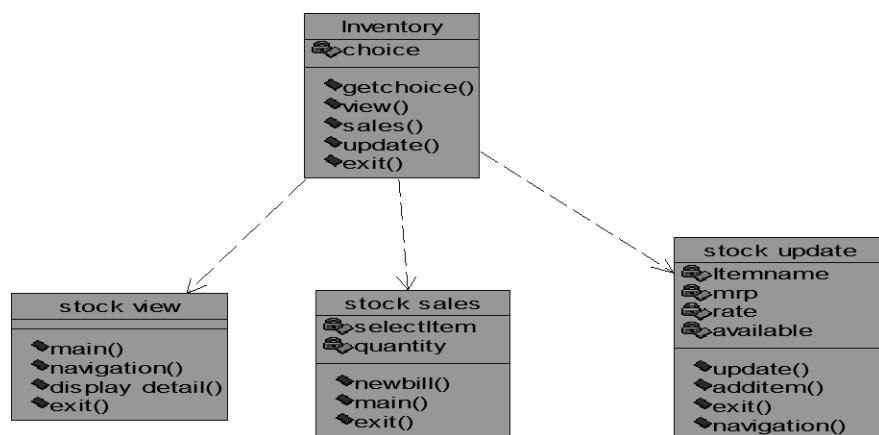
The system must be designed to allow web usability. That is, the system must be designed in such a way that will be easy to use and visible on most of the browsers.

## **6. SOFTWARE DESIGN**

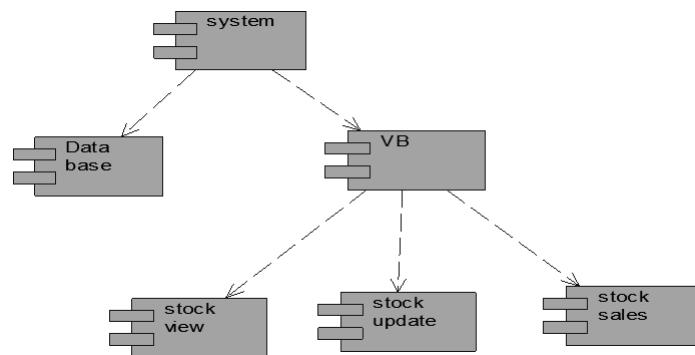
### **6.1 USECASE DIAGRAM:**



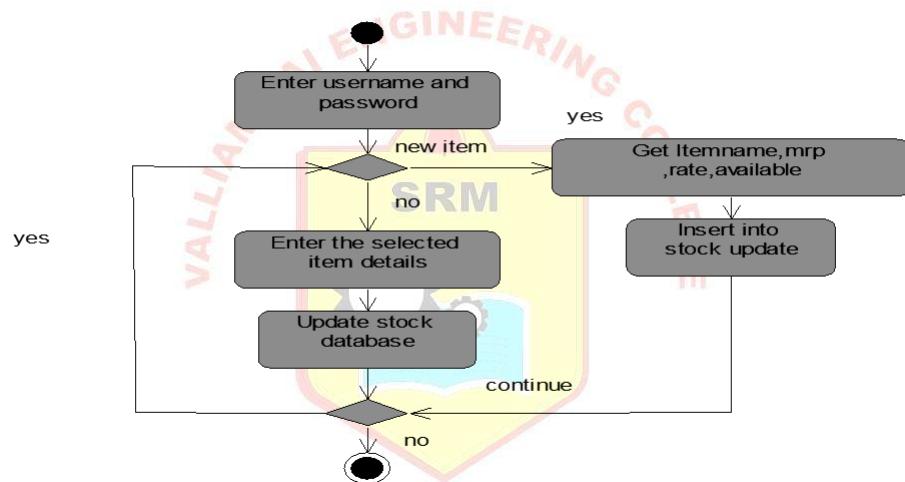
### **6.2 CLASS DIAGRAM:**



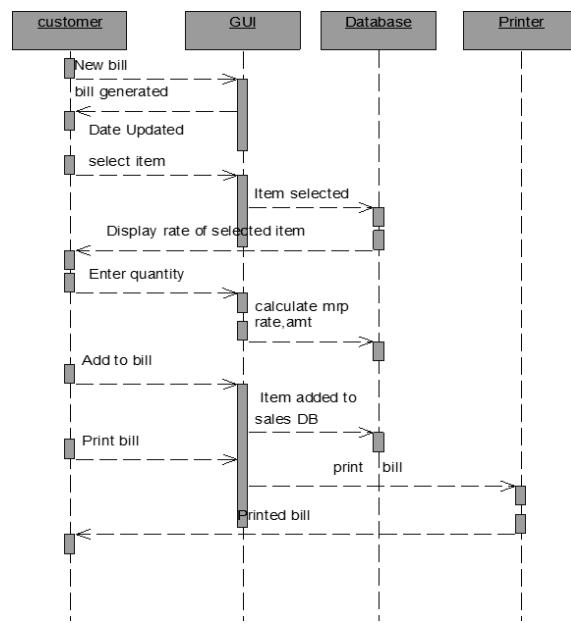
### 6.3 COMPONENT DIAGRAM:



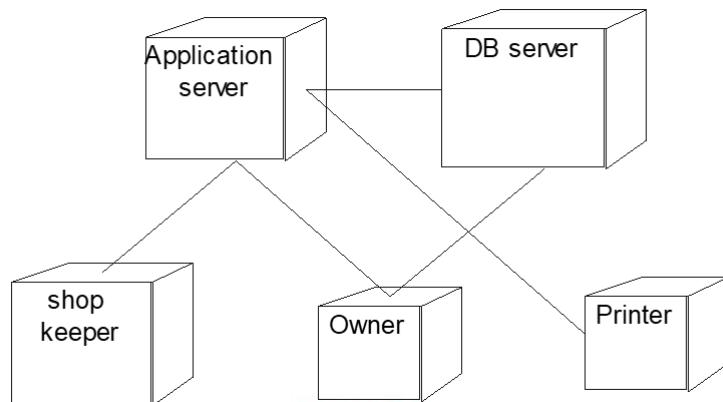
### 6.4 ACTIVITY DIAGRAM:



### 6.5 SEQUENCE DIAGRAM:



## 6.6 DEPLOYMENT DIAGRAM:



## 7. DATABASE DESIGN

Database name: STOCK

Table Name : CUSTOMER DETAILS

FIELD NAME	FIELD TYPE
Name	Text
Id	Integer
Address	Text
Mobile no	Double

Table Name : PURCHASE ITEM (MOBILE)

FIELD NAME	FIELD TYPE
Total number of stocks	Integer
Needed	Integer
Price	Double

Table Name : PURCHASE ITEM (LAPTOP)

FIELD NAME	FIELD TYPE
Total number of stocks	Integer
Needed	Integer
Price	Double

## 8. IMPLEMENTATION:

Form1 ( Login form)



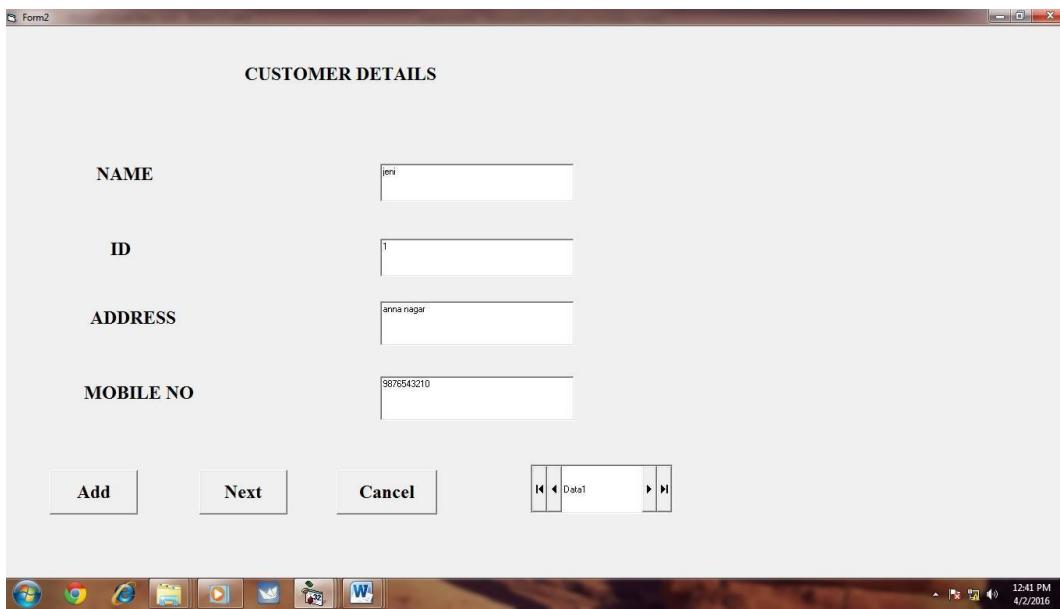
### Coding:

```
Private Sub Command1_Click()
If text1.text="stock" and text2.text="1234" then
Form2.show
Else
Msgbox("Invalid user name and password")
End if
End Sub
```

```
Private Sub Command2_Click()
End
End Sub
```



Form2(Customer details)



### Coding:

```

Private Sub Command1_Click()
Data1.Recordset.AddNew
Data1.Recordset.Fields("LOGIN") = Text1.Text
Data1.Recordset.Fields("PASSWORD") = Val(Text2.Text)
MsgBox ("data added")
Data1.Recordset.Update
End Sub

```

```

Private Sub Command2_Click()
Form3.Show
End Sub

```

```

Private Sub Command3_Click()
End
End Sub

```

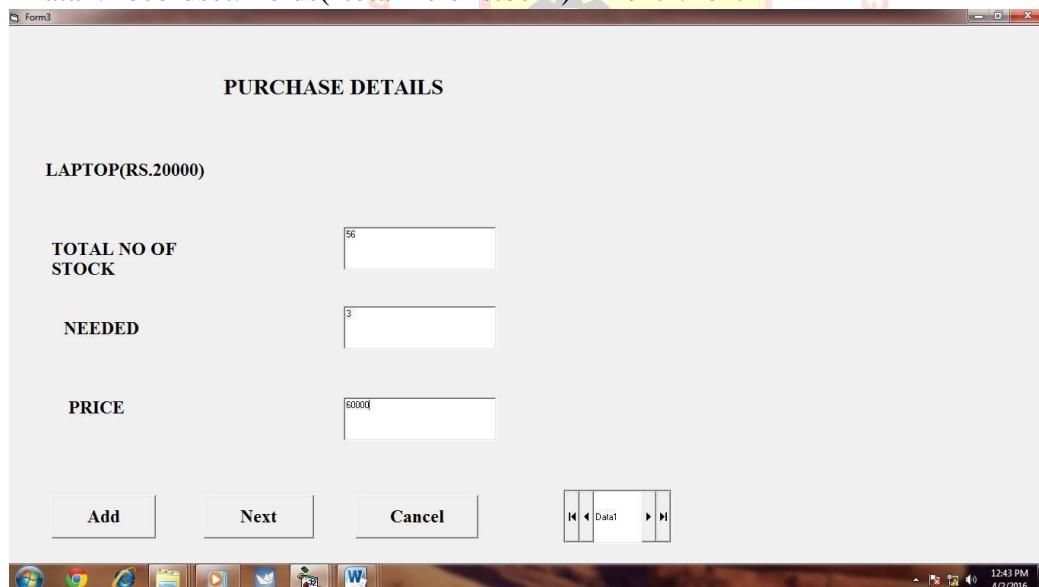
Form3(Laptop purchase details)

Coding:

```

Private Sub Command1_Click()
Data1.Recordset.AddNew
Data1.Recordset.Fields("total no of stock") = Text2.Text

```

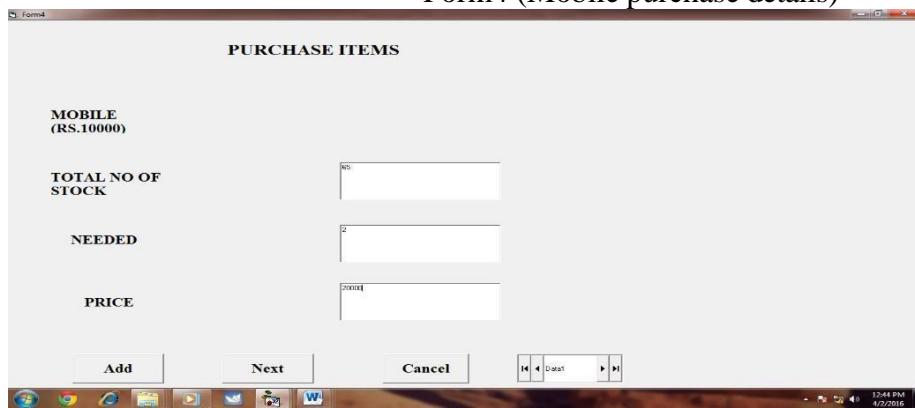


```

Data1.Recordset.Fields("needed") = Text3.Text
Data1.Recordset.Fields("price") = Text4.Text
Data1.Recordset.Update
End Sub
Private Sub Command2_Click()
Form4.Show
End Sub
Private Sub Command3_Click()
End
End Sub

```

Form4 (Mobile purchase details)

Coding:

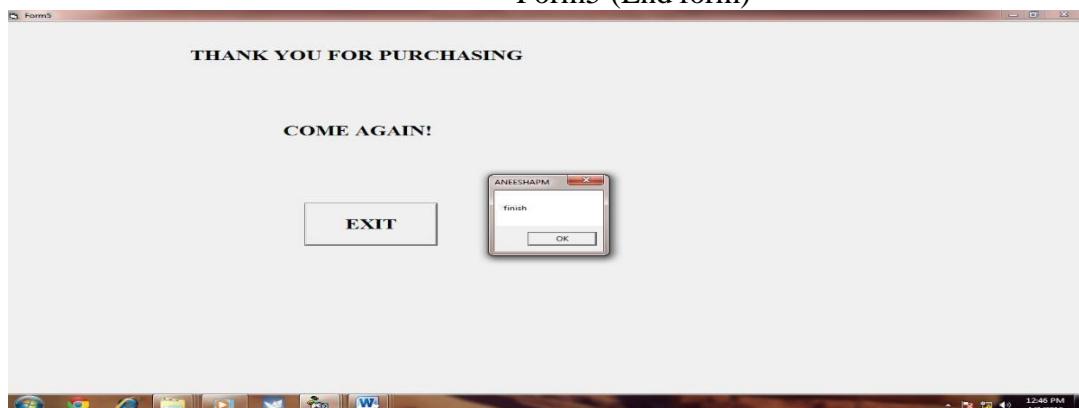
```

Private Sub Command1_Click()
Data1.Recordset.AddNew
Data1.Recordset.Fields("total no of stock") = Text1.Text
Data1.Recordset.Fields("needed") = Text2.Text
Data1.Recordset.Fields("price") = Text3.Text
Data1.Recordset.Update
End Sub
Private Sub Command2_Click()
Form5.Show
End Sub
Private Sub Command3_Click()
End
End Sub

```



Form5 (End form)

Coding:

```

Private Sub Command1_Click()
MsgBox "finish"
Form6.Show
End Sub

```

## 6. TESTING:

<b>Test case ID: Test_01</b>					
Test priority					
(Low/Medium/High):Medium					
Module name: login					
Test title :verify login with valid username and password					
Precondition: user has invalid username and password					
S.N O	TEST STEPS	EXPECTED RESULTS	ACTUAL RESULTS	STATUS	NOTES
1	Provide valid User name	User should Be able to login	The user is able to move to next Entry	Success	-
2	Provide valid password	User should be Able to Login	The user is able To login Successfully	Success	Incase of wrong Password was given an error Message box was displayed
3	Click login	User should be able to navigate to next page after validation	User name and password is validated and next page is displayed	Success	Incase user gives wrong entry the sign in page remains active
4	Click signup	User should be able to navigate to next page where user enters his credentials	User navigates to the signup page where his user name and password is validated	success	-

## RESULT:

The Stock monitoring system was designed and implemented successfully.

## I. QUIZ SYSTEM

### 1. PROBLEM STATEMENT

Online Quiz System has to be developed for conducting a quiz. The number of participants should be of 30. The duration for quiz is 30 minutes so a timer has to be kept which should show time duration. The number of questions should be 40 chosen randomly from the database in 3 different areas namely. i) Information Technology ii) logical reasoning and iii) General Knowledge. The questions should be of objective type with multiple options. For each correct answer the participant will receive 1Point and wrong answer .25Point will be deducted for each wrong answer. At the end of the quiz the user score will be displayed. Once a Student answered for question he can't changes his answer later.

### 2. LOGIN

#### 2.1 Brief Description

The use case describes how a Participant logs into the Quiz System

#### 2.2 Flow of Events

##### 2.2.1 Basic Flow

This use case starts when the Participant wishes to Login to the Quiz System

1. The System requests that the participant enter his/her name and password
2. The participant enters his/her name and password
3. The System validates the entered name and password and logs the participant into the System

##### 2.2.2 Alternative Flows

Invalid Name/Password

If, in the Basic flow, the participant enters an invalid name and/or password, the system displays an error message. The participant chooses to either return to the beginning of the Basic flow or cancel the login, at which point the use case ends.

##### 2.3 Special Requirements

The Number of participants should be 30

##### 2.4 Pre-Conditions

None

##### 2.5 Post-Conditions

If the use case was successful, the participant is now logged into the system. If not, the system State is unchanged.

##### 2.6 Extension Points

None

### 3. Select Options to Answer

#### 3.1 Brief Description

This use case allows the Participant to select appropriate answer to the question.

#### 3.2 Flow of Events

##### 3.2.1 Basic flow

This use case starts when the participant logs into the system. The Participant will be given a set of questions with answers in multiple options. The participant selects the answer from the set of options. Once the participant answered for a question next question will be displayed.

##### 3.2.1 Alternate Flow

Once the time slot allotted for the participant is over scorecard will be displayed.

#### 3.3 Special Requirements

None

### **3.4 Pre-Conditions**

The Participant must login into the System for answering the question

### **3.5 Post-Conditions**

If the use case was successful, the participant will be able to view the next question.

### **3.6 Extension Points**

None

## **4. Display Score Card**

### **4.1 Brief Description**

The use case gives the scorecard along with No of Correct and wrong answers

### **4.2 Flow of Events**

#### **4.2.1 Basic Flow**

Once the participant had completed answering to the questions within the stipulated time, he/she will be given a scorecard with final score.

#### **4.2 Alternative Flow**

None

### **4.3 Special Requirements**

None

### **4.4 Pre-Conditions**

The participant should have completed the questions within stipulated time.

### **4.5 Post-Conditions**

If the use case was successful, the participants can logout from the system

### **4.6 Extension Points**

None

## **5. Winner List**

### **5.1 Brief Description**

This use case gives winner list based on the scores obtained by the participant

### **5.2 Flow of Events**

#### **5.2.1 Basic Flow**

This use case gives the coordinator the winner list based on the scored obtained

#### **Alternative Flow**

If the coordinator logged into the system even before the quiz was over he will empty List.

### **5.3 Special Requirements**

None

### **5.4 Pre-Conditions**

If all the participants had answered for the questions then the winner list has to be generated.

### **5.5 Post-Conditions**

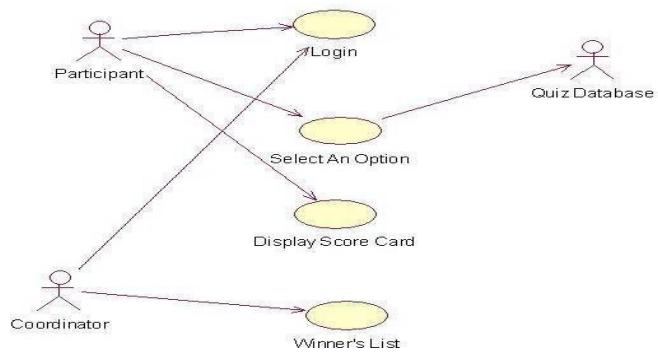
None

### **5.6 Extension Points**

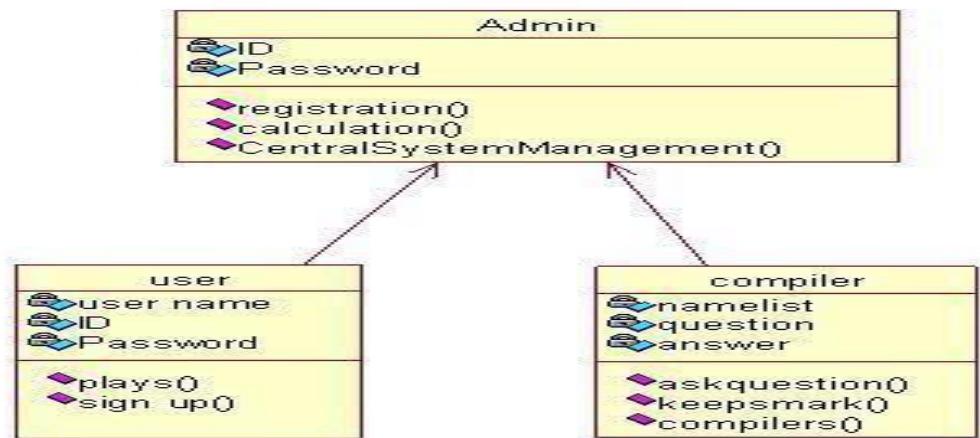
None

## 6. SYSTEM DESIGN:

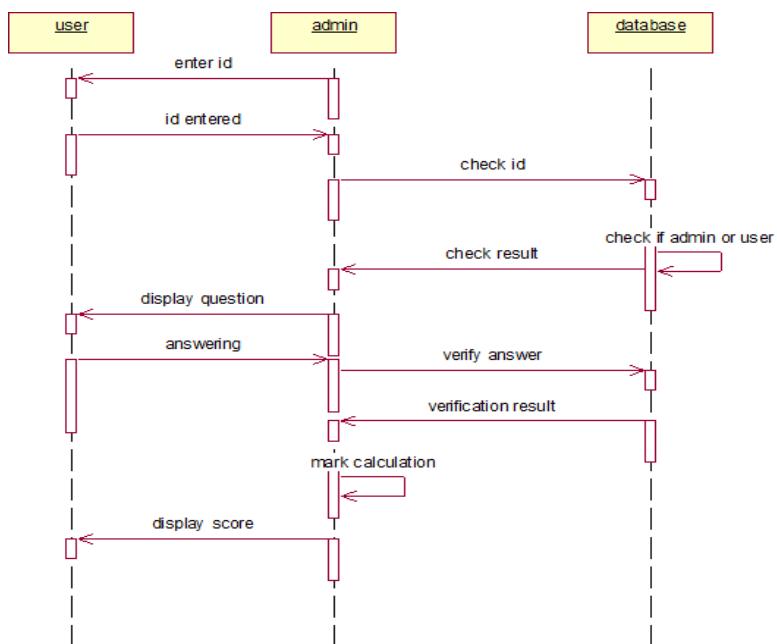
### 6.1 USECASE DIAGRAM:



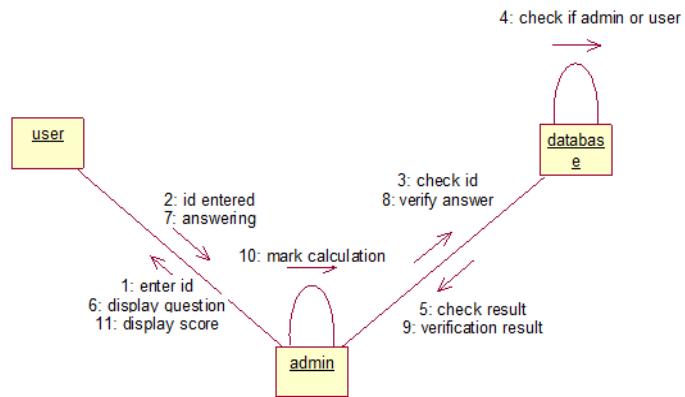
### 6.2 CLASS DIAGRAM:



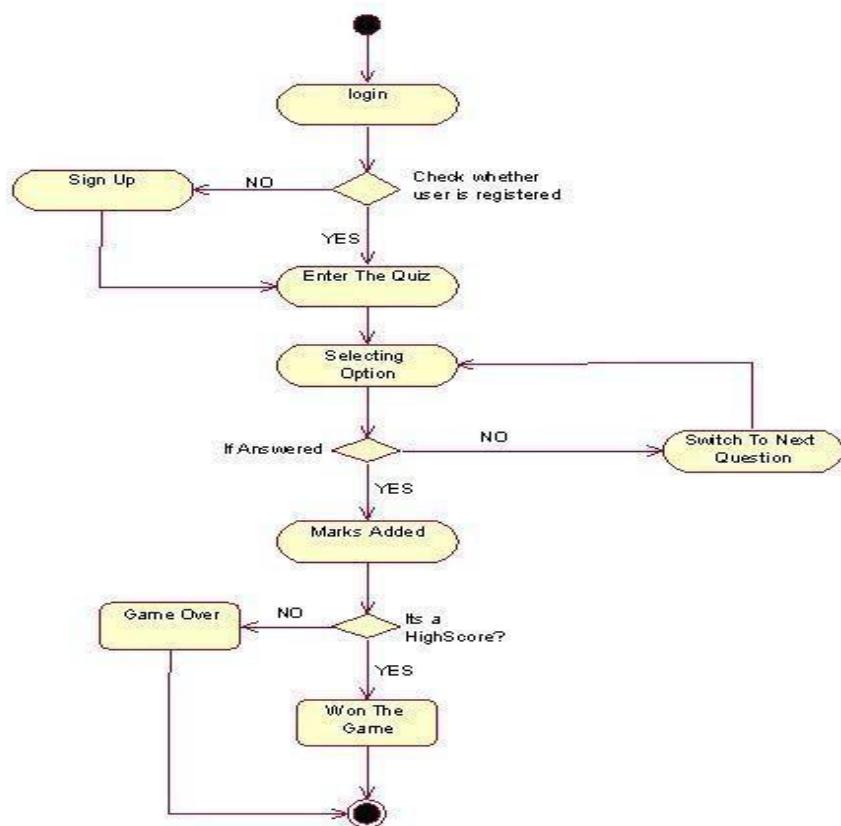
### 6.3 SEQUENCE DIAGRAM:



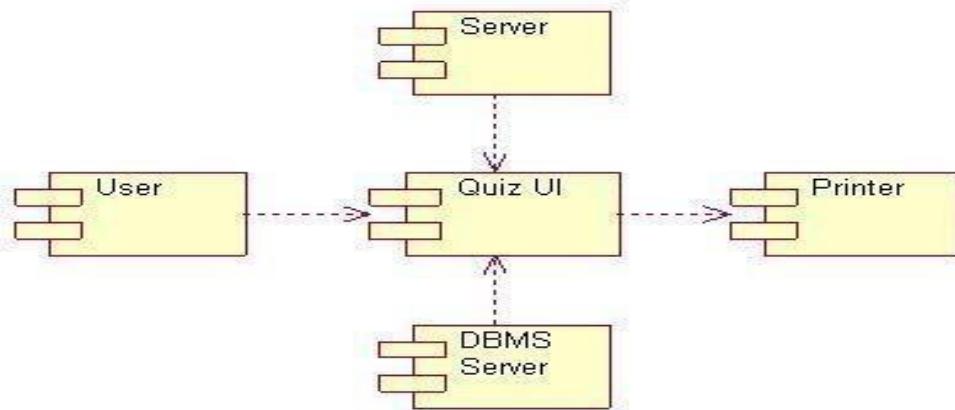
#### 6.4 COLLBORATION DIAGRAM:



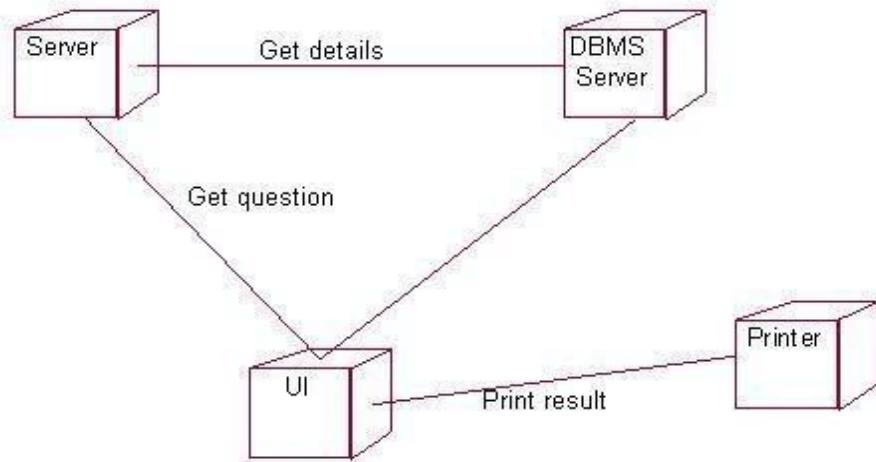
#### 6.5 ACTIVITY DIAGRAM:



### 6.6 COMPONENT DIAGRAM:



### 6.7 DEPLOYMENT DIAGRAM:



### 7. DATABASE DESIGN

Database Name: quiz

Table Name : mark

Fields	Data type
Name of competitor	Text
regno	Text
mark	Integer

## 8. IMPLEMENTATION:

Form1 (Login form)



### Coding:

```
Private Sub Command1_Click()
If Text1.Text = "abcd" And Text2.Text = "1234" Then
Form2.Show
Else
MsgBox ("wrong userid or password")
End If
End Sub
```

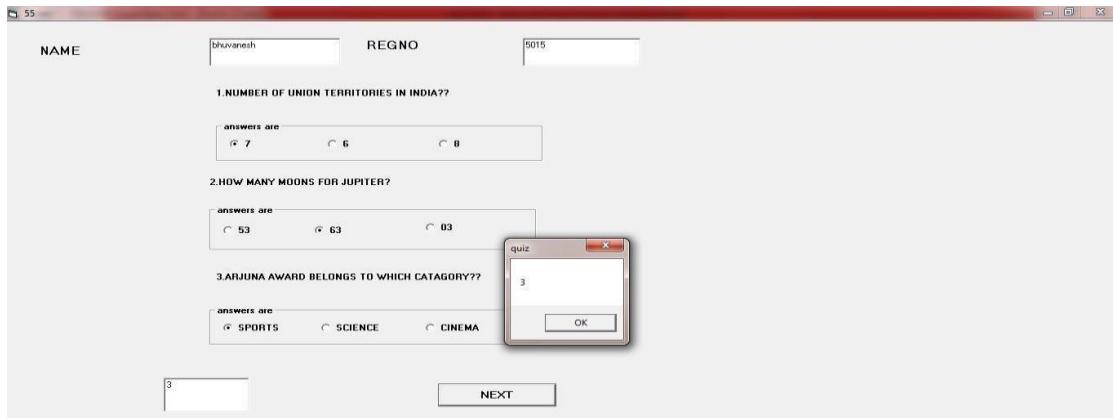
```
Private Sub Image1_Click()
End Sub
```

```
Private Sub Picture1_Click()
End Sub
```

```
Private Sub Picture2_Click()
End Sub
```

```
Private Sub Command2_Click()
End
End Sub
```

## Form2(Quiz)

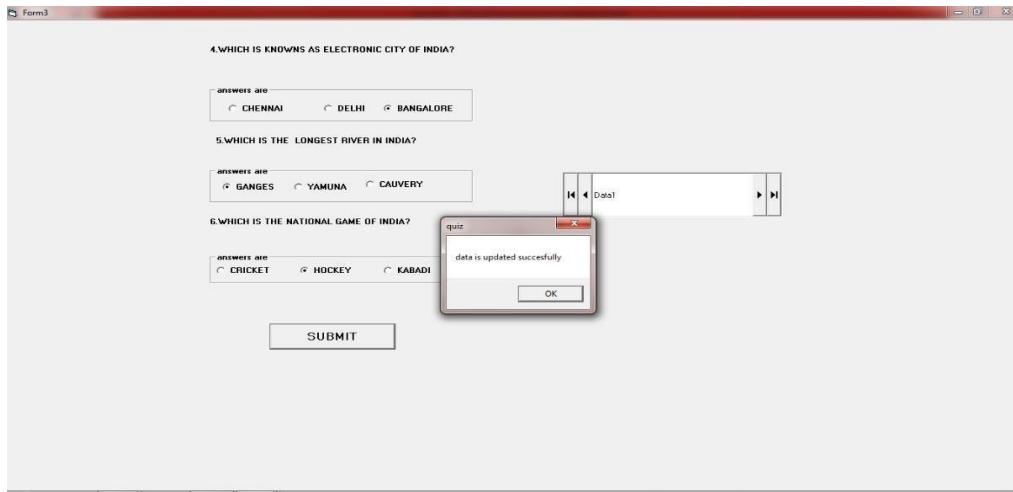
Coding:

```

Private Sub Command1_Click()
If Option13.Value = True Then
Text1.Text = Text1.Text + 1
End If
If Option2.Value = True Then
Text1.Text = Text1.Text + 1
End If
If Option4.Value = True Then
Text1.Text = Text1.Text + 1
End If
MsgBox (Text1.Text)
Form3.Show
End Sub

```

## Form3(Quiz cont...)



Coding:

```

Dim s
Private Sub Command1_Click()
If Option15.Value = True Then
Form2.Text1 = Form2.Text1 + 1
End If
If Option1.Value = True Then
Form2.Text1 = Form2.Text1 + 1
End If
If Option5.Value = True Then
Form2.Text1 = Form2.Text1 + 1
End If
MsgBox (Form2.Text1)
Data1.Recordset.AddNew
Data1.Recordset.Fields("name of competitor") = Form2.Text2
Data1.Recordset.Fields("regno") = Form2.Text3
Data1.Recordset.Fields("mark") = Form2.Text1
Data1.Recordset.Update
MsgBox ("data is updated successfully")
Form4.Show
End Sub

```

```

Private Sub Form_Load()
s = Form2.Text1
mark = s
End Sub

```



Form4(Display marks)

Form4

name of competitor	regno	mark
shubham	15	6
pratik	21	5
anushka	3	6
baskar	13	5
prashna	47	9
anil prakash	7	4
deepak	17	5
afsa	1	5
bhagavat	14	4
dhruv	13	5
esu	31	2
bhuvan	15	6
bhuvaneeth	905	6
total	1	1

[|] Data | [|]

## 9. TESTING:

<b>Test case ID: Test_01</b>					
Test priority(Low/Medium/High):Medium					
Module name: login					
Test title :verify login with valid username and password					
S.NO	TEST STEPS	EXPECTED RESULTS	ACTUAL RESULTS	STATUS	NOTES
1	Provide valid User name	User should Be able to login	The user is able to move to next Entry	Success	-
2	Provide valid password	User should be Able to Login	The user is able To login Successfully	Success	Incase of wrong Password was given an error Message box was displayed
3	Click login	User should be able to navigate to next page after validation	User name and password is validated and next page is displayed	Success	Incase user gives wrong entry the sign in page remains active
4	Click signup	User should be able to navigate to next page where user enters his credentials	User navigates to the signup page where his user name and password is validated	success	-

### RESULT:

The Quiz system was designed and implemented successfully.

## J. E-MAIL CLIENT SYSTEM

### **1 PROBLEM STATEMENT**

The main objective of this document is to illustrate the requirements of the project **E-mail Client System**. The document gives the detailed description of the both functional and non functional requirements proposed by the client. The document is developed after a number of consultations with the client and considering the complete requirement specifications of the given Project. The final product of the team will be meeting the requirements of this document.

### **2. PRODUCT PERSPECTIVE**

The proposed Inventory. This System will provide a search functionality to facilitate the search of resources. This search will be based on various categories viz. results to search various categories simultaneously. Further the Student can add/update/remove the resources and the resource users from the system.

#### **2.1 PRODUCT FEATURES**

The features that are available to the telephone directory are:

- Selection of client.
- Chat with server application
- More than one client at a time can chat.

#### **2.2 User Classes and Characteristics**

There are various kinds of users for the product. Usually web products are visited by various users for different reasons. The users include: Students who will be using the above features by accessing

#### **2.3 Operating Environment**

The product will be operating in windows environment. Also it will be compatible with the IE 6.0. Most of the features will be compatible with the Mozilla firefox & Opera 7.0 or higher version. The only requirement to use this online product would be the internet connection.

#### **2.4 User Documentation**

The product will include user manual. The user manual will include product overview, complete configuration of the used software (such as Linux OS), technical details, backup procedure and contact information which will include email address. There will be no online help for the product at this moment. The product will be compatible with the Internet Explorer 6.0 or higher. The databases will be created in the Microsoft access

#### **2.5 Assumptions and Dependencies**

The product needs following third party product.

- Unix operating system

### **3. System Features**

#### **3.1. Database – Storage**

##### **3.1.1. Description and Priority**

Proposed Database is intended to store, retrieve, update, and manipulate information related to university which include

- Selection of client.
- Chat with server application
- More than one client at a time can chat.

##### **3.1.2. Stimulus / Response Sequences**

The system will check for validity of login .If the Login and password are valid, the response to this action is the administrator will be able to modify, view, add, deleting and all other functions

that can be performed on the database able to modify, view, add, deleting and all other functions that can be performed on the database.

### **3.2. Functional Requirements**

This section gives the list of Functional and non functional requirements which are applicable to the Student Marks Analyzing System

#### **3.2.1 Interface Requirements**

This section describes how the software interfaces with other This section describes how the software interfaces with other software products or users for input or output.

##### **3.2.1.1 UserInterfaces**

Describes how this product interfaces with the user GUI Describes the graphical user interface if present. This section should include a set of screen dumps or mockups to illustrate user interface features.

###### **1. Description**

The user interface must be customizable by the administrator

###### **2. Criticality**

This issue is essential to the overall system. All the modules provided with the software must fit into this graphical user interface and accomplish to the standard defined.

###### **3. Technical issues**

In order to satisfy this requirement the design should be simple and all the different interfaces should follow a standard template. There will be the possibility of changing colors and images, plus switching between interfaces with the minimum impact for the users.

###### **4. Risks**

To reduce the circumstances under which this requirement might not able to be satisfied, all the designers must have been developed web sites previously and they must be aware of html restriction and cross browsers implementations before starting the designing. In order to reduce the probability of this occurrence the entire design team will be trained in basic html development and macromedia fireworks, this tool will be used instead of Photoshop.

###### **5. Dependencies with other requirements**

All user interfaces should be able to interact with the user management module and a part of the interface must be dedicated to the login/logout module

### **4. Non Functional Requirements**

#### **4.1. User Interfaces**

#### **4.2. Hardware Interfaces**

Server Side:

Operating System: Windows 9x/xp ,Windows ME

Processor: Pentium 3.0 GHz or higher

RAM: 256 Mb or more

Hard Drive: 10 GB or more

Client side:

Operating System: UNIX.

Processor: Pentium III or 2.0 GHz or higher.

RAM: 256 Mb or more

### **5. Other Nonfunctional Requirements**

#### **5.1 Performance Requirements**

The proposed system that we are going to develop will be used as the Chief performance system within the different documentation purpose. Therefore, it is expected that the database would perform functionally all the requirements that are specified by the university.

### 5.2 Safety Requirements

The database may get crashed at any certain time due to virus or operating system failure. Therefore, it is required to take the database backup

### 5.3 Security Requirements

We are going to develop a secured database for the university .There are different categories of users namely teaching staff, administrator, library staff , students etc.,Depending upon the category of user the access rights are decided .It means if the user is an administrator then he can be able to modify the data, delete, append etc., All other users other than library staff only have the rights to retrieve the information about database.

### Software Quality Attributes

The Quality of the database is maintained in such a way so that it can be very user friendly to all the users of the database

### 5.4 Hardware Constraints

The system requires a database in order to store persistent data. The database should have backup capabilities.

### 5.5 Software Constraints

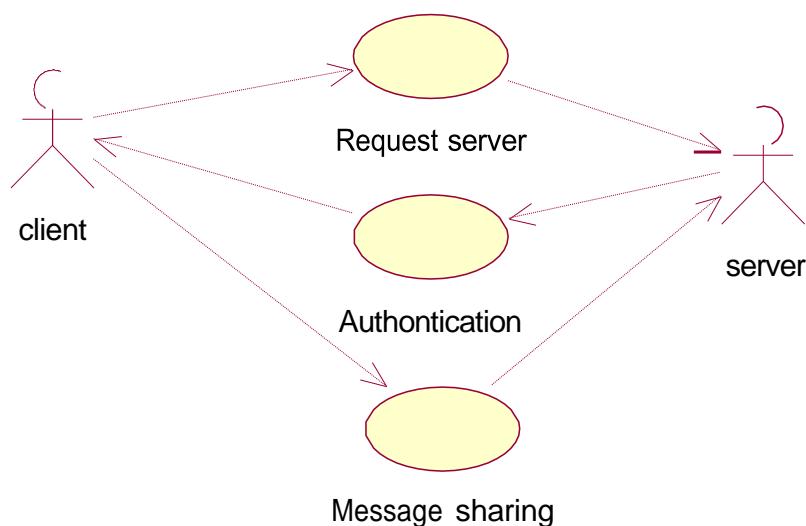
The development of the system will be constrained by the availability of required software such as web servers, database and development tools.

### 5.6 Design Constraints

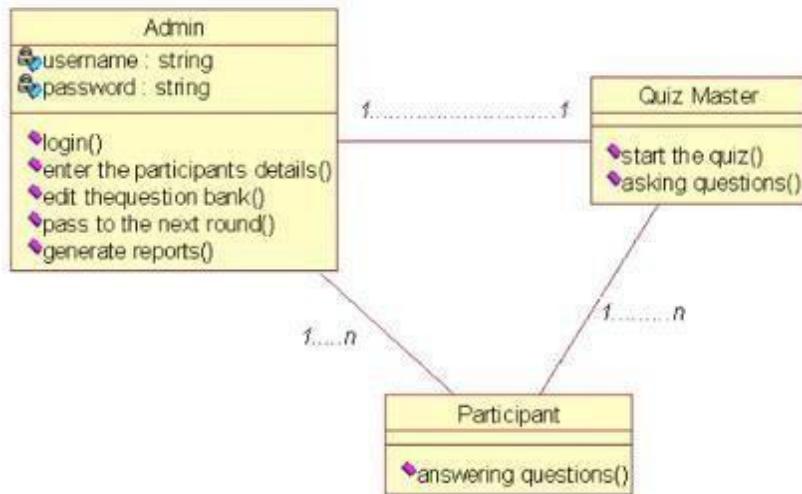
The system must be designed to allow web usability. That is, the system must be designed in such a way that will be easy to use and visible on most of the browsers.

## 6. DESIGN:

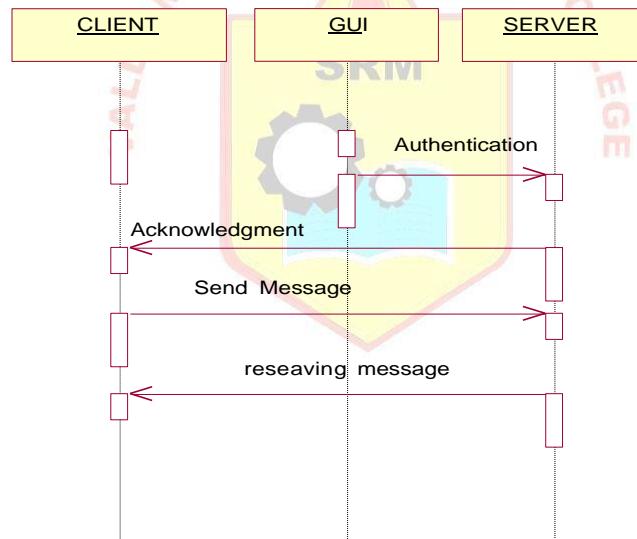
### 6.1 USE CASE DIAGRAM



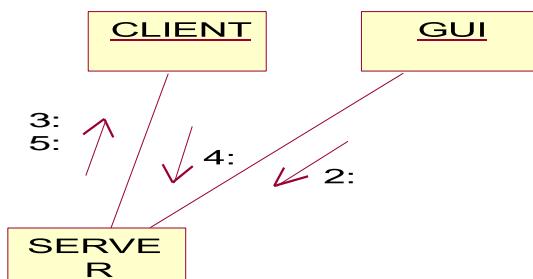
## 6.2 CLASS DIAGRAM:



## 6.3 SEQUENCE DIAGRAM



## 6.4 COLLABORATION DAIGRAM



1. Send request to server
2. Authentication verify
3. Acknowledgment send to client
4. message passing
5. Message reseaving

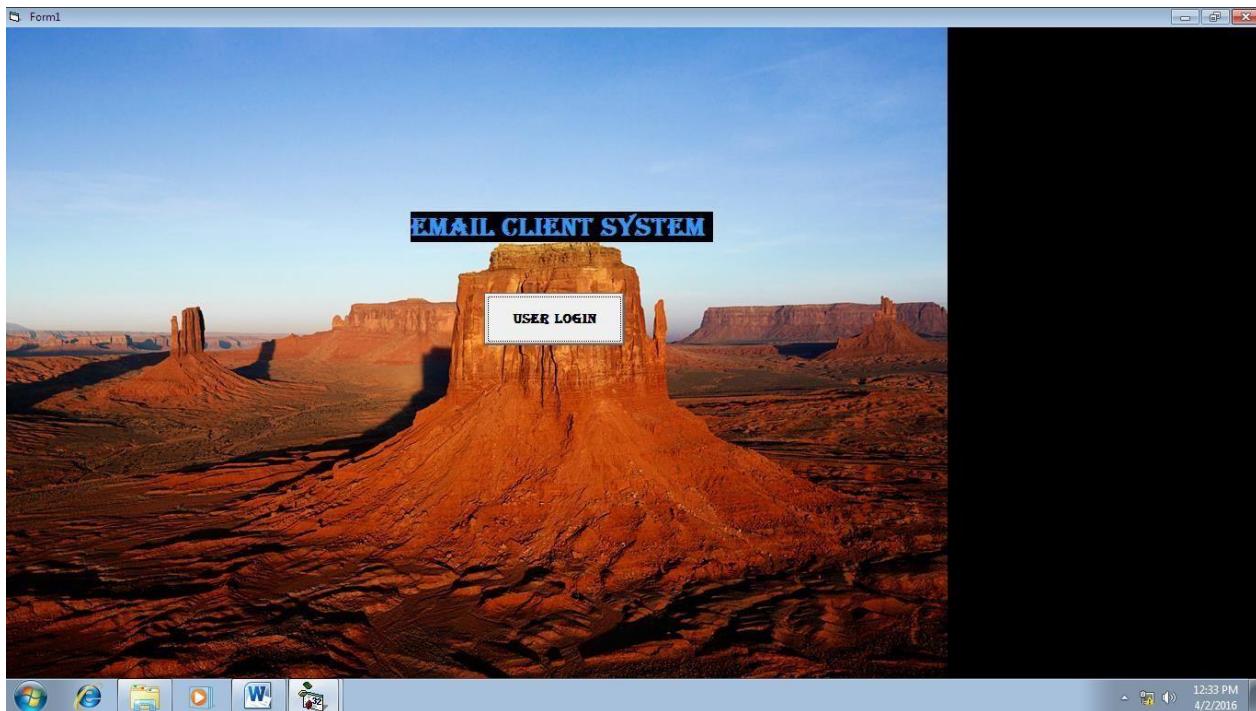
## 7. DATABASE DESIGN

Database Name: email  
 Table Name : data

Fields	Data type
to	Text
message	Text

## 8. IMPLEMENTATION:

Form1

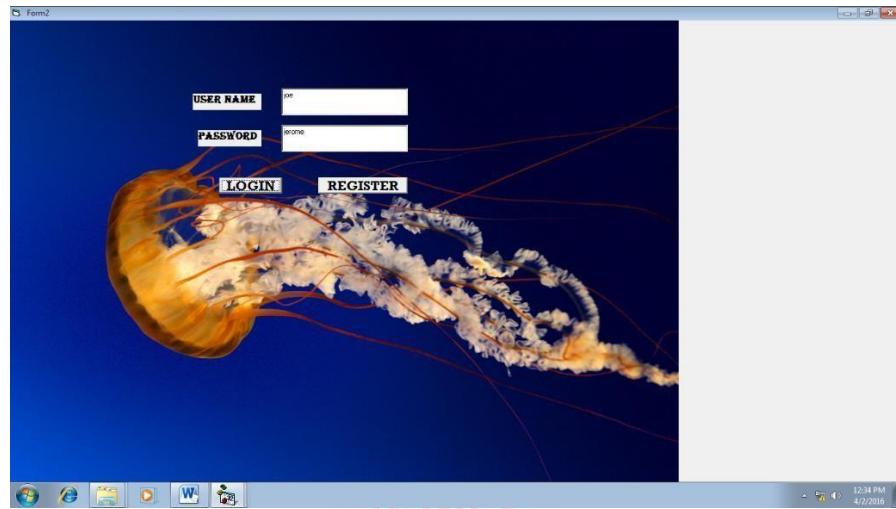


### Coding:

```
Private Sub Command1_Click()
If Text1.Text = "joe" And Text2.Text = "jerome" Then
Form3.Show
Else
MsgBox "invalid password"
End If
End Sub
```

```
Private Sub Command2_Click()
Form4.Show
End Sub
```

Form2

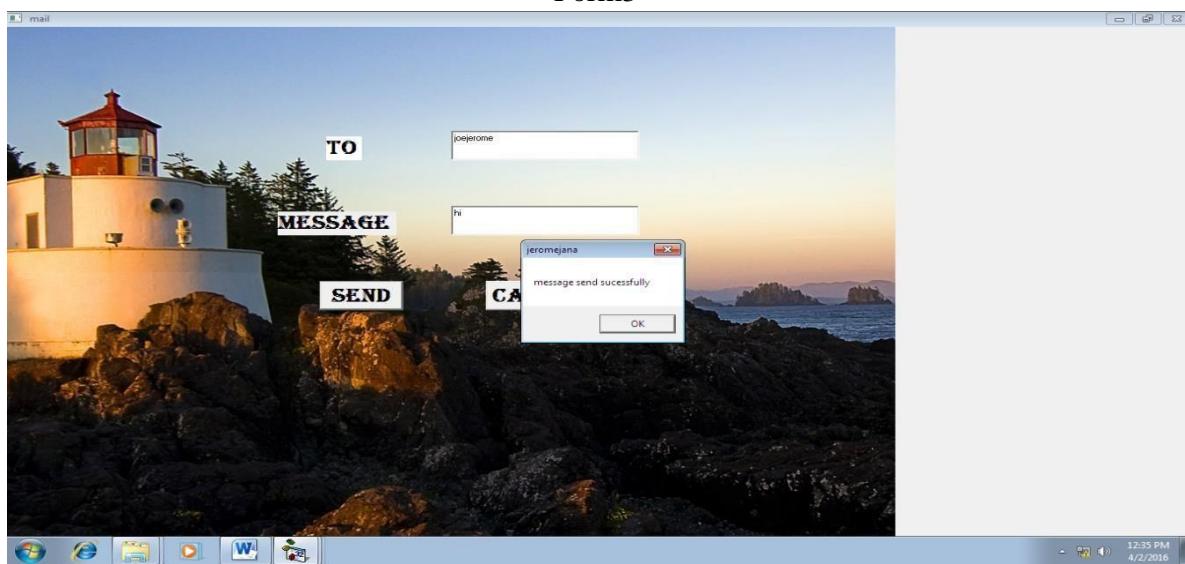
Coding:

```
Private Sub Command1_Click()
If Text1.Text = "joe" And Text2.Text = "jerome" Then
Form3.Show
Else
MsgBox "invalid password"
End If
End Sub
```

```
Private Sub Command2_Click()
Form4.Show
End Sub
```



Form3

Coding:

```
Private Sub Command1_Click()
Data1.Recordset.AddNew
```

```

Data1.Recordset.Fields("to") = Text1.Text
Data1.Recordset.Fields("message") = Text2.Text
If Text1.Text = "joejerome" Then
MsgBox "message send sucessfully"
Else
MsgBox "invalid id"
End If
End Sub

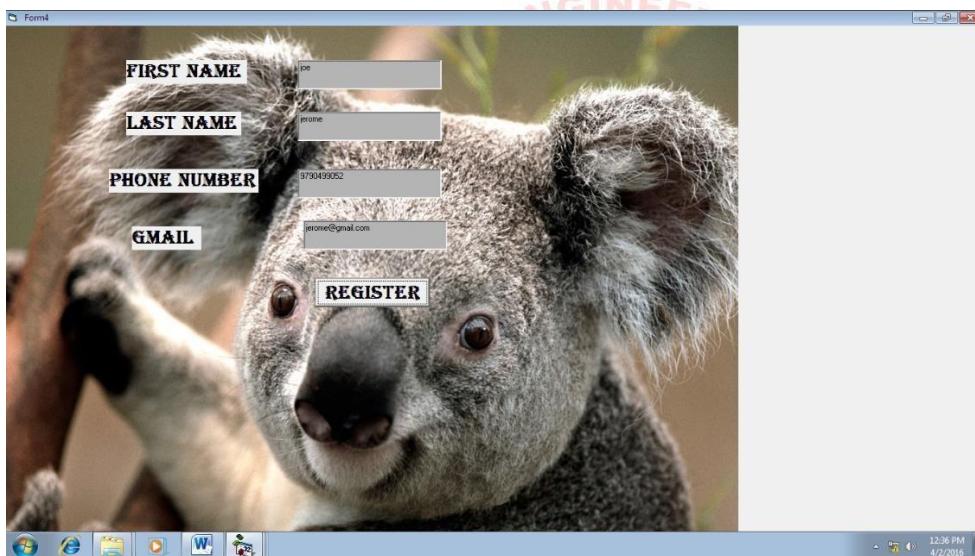
```

```

Private Sub Command2_Click()
MsgBox "sending canceled"
End Sub

```

Form4

Coding:

```

Private Sub Command1_Click()
Data1.Recordset.AddNew
Data1.Recordset.Fields("first_name") = Text1.Text
Data1.Recordset.Fields("last_name") = Text2.Text
Data1.Recordset.Fields("phone_number") = Text3.Text
Data1.Recordset.Fields("gmail") = Text4.Text
Data1.Recordset.Update
MsgBox "Inserted successfully"
End Sub

```

## 9. TESTING:

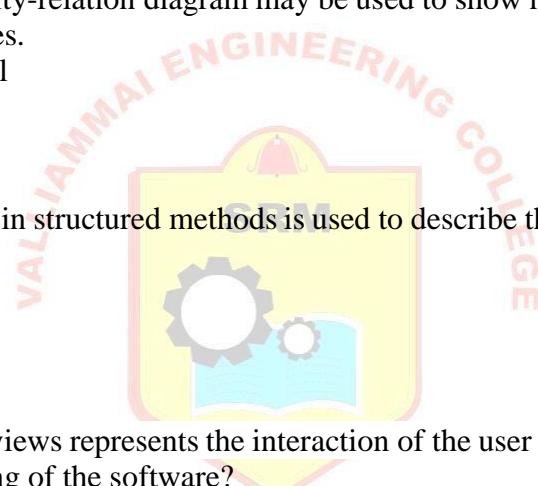
<b>Test case ID: Test_01</b>					
Test priority(Low/Medium/High):Medium					
Module name: login					
Test title :verify login with valid username and password					
Precondition: user has invalid username and password					
S.N O	TEST STEPS	EXPECTED RESULTS	ACTUAL RESULTS	STATUS	NOTES
1	Provide valid User name	User should Be able to login	The user is able to move to next Entry	Success	-
2	Provide valid password	User should be Able to Login	The user is able To login Successfully	Success	Incase of wrong Password was given an error Message box was displayed
3	Click login	User should be able to navigate to next page after validation	User name and password is validated and next page is displayed	Success	Incase user gives wrong entry the sign in page remains active
4	Click signup	User should be able to navigate to next page where user enters his credentials	User navigates to the signup page where his user name and password is validated	success	-

## RESULT:

The Email client system was designed and implemented successfully.

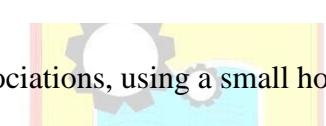
### ADDITIONAL VIVA – VOCE QUESTIONS

1. .... is a sub discipline of computer Science that attempts to apply engineering principles to the creation, operation, modification and maintenance of the software components of various systems.
  - A) Computer Engineering
  - B) Hardware Engineering
  - C) Software Engineering
  - D) Component Engineering
2. Software maintenance refers to the support phase of software development which includes.
  - A) Correction
  - B) Adaption
  - C) Enhancement
  - D) All the above
3. In a....., entity-relation diagram may be used to show how some entities in a system are composed of other entities.
  - A) data-processing model
  - B) composition model
  - C) classification model
  - D) process model
4. A/An ..... in structured methods is used to describe the logical data, structured being used.
  - A) data-flow model
  - B) structured model
  - C) classification model
  - D) entity-relation model
5. Which of the following views represents the interaction of the user with the software but tells nothing about the internal working of the software?
  - A. Use case diagram
  - B. Activity diagram
  - C. Class diagram
  - D. All of the above
6. What are the notations for the Use case Diagrams?
  - A. Use case
  - B. Actor
  - C. Prototype
  - D. Use case and Actor
7. Which among the following can be heuristic for Use case diagram?
  - A. The product can be made actor
  - B. Never name actors with noun phrases
  - C. Name Use cases with verb phrases
  - D. All of the mentioned
8. Which of the following are the valid relationships in Use Case Diagrams
  - A. Generalization
  - B. Include



- C. Extend  
D. All of the mentioned
9. Which of the following UML diagrams has a static view?  
A. Collaboration  
B. Use case  
C. State chart  
D. Activity
10. The essential characteristics of an object that distinguish it from all other kinds of objects and thus provide crisply defined conceptual boundaries, relative to the perspective of the viewer is called  
A. Encapsulation  
B. Modularity  
C. Hierarchy  
D. Abstraction
11. Which is a model static data structure?  
A. Object diagrams  
B. Class diagrams  
C. Activity diagrams  
D. Interaction diagrams
12. Where the class diagrams are not convenient?  
A. simple interactions model  
B. the vocabulary of a system model  
C. simple collaborations model  
D. logical database schema model
13. An association may be  
A. unary  
B. binary  
C. ternary or n-ary  
D. All of the above
14. Which of the following specifies how many instances of one class may relate to a single instance of an associated class?  
A. Multiplicity  
B. Association  
C. Degree  
D. None of the above
15. The attribute(s) is/are associated with the association is called  
A. Link attribute  
B. Derived attribute  
C. Multi-valued attribute  
D. None



16. Which of the following is a name that uniquely identifies one end of an association?
- Label name of the link
  - Role name
  - Link attribute name
  - None
17. The role name is a
- Derived attribute
  - Link attribute
  - Attribute
  - All of the above
18. Which of the following is the “part-whole” or “a-part-of” relationship in which objects representing the component of something are associated with an object representing the entire assembly?
- Generalization
  - Specialization
  - Aggregation
  - None
19. Aggregations are drawn like associations, using a small hollow ————— indicating the assembly end of the relationship.
- 
20. Inheritance is a ————— relationship between two classes.
- “is-a”
  - Part of
  - Both a and b
  - None
21. Which things are dynamic parts of UML models?
- Structural things
  - Behavioral things
  - Grouping things
  - Annotational things
22. Interaction Diagram is a combined term for
- Sequence Diagram + Collaboration Diagram
  - Activity Diagram + State Chart Diagram
  - Deployment Diagram + Collaboration Diagram
  - None of the mentioned

23. If you are working on real-time process control applications or systems that involve concurrent processing, you would use a  
 A. Activity diagram  
 B. Sequence diagram  
 C. State chart diagram  
 D. Object diagram
24. Forward Engineering is possible for an Activity Diagram especially if the context of the diagram is \_\_\_\_\_  
 A. an operation  
 B. a workflow  
 C. a class  
 D. a use case
25. Realization of a use case is specified by \_\_\_\_\_  
 A. a collaboration  
 B. a component  
 C. a node  
 D. an activity
26. Which of these is correct?  
 A. Artifacts instances and types have same names  
 B. Artifact names and instances are underlines  
 C. All of the mentioned  
 D. None of the mentioned
27. What refers to the value associated with a specific attribute of an object and to any actions or side?  
 A.Object  
 B. State  
 C.Interface  
 D.None of the mentioned
28. How many diagrams are here in Unified Modelling Language?  
 A.six  
 B. seven  
 C.eight  
 D.nine
29. Which of the following determines the state diagram?  
 A. The UML notation for specifying finite automata is the state diagram  
 B. In state diagrams, states are represented by rounded rectangles  
 C. Both A and B  
 D. None of the mentioned
30. Which of the statements state the name compartment?  
 A.The first compartment is the name compartment



- B. It contains the state name; State names are optional and may be path names  
 C. The name compartment can never be omitted  
 D. The first compartment is the name compartment, It contains the state name; State names are optional and may be path names
31. ...., sometimes called glass-box testing, is a test case design method that uses the control structure of the procedural design to derive test cases.  
 A) White-box testing  
 B) Control structure testing  
 C) Black-box testing  
 D) Gray-box testing
32. .... is black-box testing method that divides the input domain of a program into classes of data from which test cases can be derived.  
 A) Condition testing  
 B) Graph-based testing  
 C) Equivalence partitioning  
 D) loop testing
33. Boundary value analysis is a test design technique that complements .....  
 A) Condition testing  
 B) Graph-based testing  
 C) Equivalence partitioning  
 D) loop testing
34. In..... , once errors in individual tasks and in system behavior have been isolated, testing shifts to time related errors.  
 A) Task testing  
 B) Inter task testing  
 C) Behavioral testing  
 D) System testing
35. In software engineering,\_\_\_\_\_ is a general repeatable solution to a commonly occurring problem in software design. It is not a finished that can be transformed directly into code.  
 A. Design pattern  
 B. Pattern  
 C. Both (a) and (b)  
 D. Either (a) or (b)
36. The UML defines a\_\_\_\_\_ as “a contract or obligation of a classifier”. They are related to the obligations of an object in terms of its behavior.  
 A. Responsibility  
 B. Pattern  
 C. Object  
 D. Interconnection

37. Doing responsibilities of an object include:

- A. Doing something itself
- B. Initiating action in other objects
- C. Controlling and coordinating activities in other objects
- D. All of the above

38. Knowing responsibilities of an object include

- A. Private encapsulated data
- B. Related data
- C. Things it can derive or calculate
- D. All of the above

39. Why are patterns important?

- A. Capture expert design knowledge
- B. Capture design accessible to both novices and other experts
- C. Both (a) and (b)
- D. None of the above

40. What benefits does pattern provide?

- A. Novice designers can benefit from learning solution patterns that experts use, without needing design experience
- B. Expert designers can benefit from studying patterns too: they can broaden their repertoire of patterns and deepen their understanding of the patterns they already know
- C. Both (a) and (b)
- D. None of the above

41. \_\_\_\_\_ is the process of finding out where something went wrong and correcting the code to eliminate the errors or bugs that cause unexpected results.

- A. Debugging
- B. Coding
- C. Documentation
- D. Review

42. \_\_\_\_\_ result from incorrectly constructed code, such as an incorrectly typed keyword or some necessary punctuation omitted.

- A. Logical error
- B. Language/Syntax error
- C. Runtime error

- D. None of the above
43. \_\_\_\_\_ occur and are detected as the program is running, when a statement attempts an operation that is impossible to carry out.
- Syntax error
  - Logical error
  - Runtime error
  - All of the above
44. \_\_\_\_\_ occur when code doesn't perform the way you intended. The code might be syntactically valid and run without performing any invalid operations and yet produce incorrect results.
- Logic errors
  - Syntax errors
  - Both (a) and (b)
  - None of the above
45. Quality assurance testing consists of
- Error based testing
  - Scenario based testing
  - Both (a) and (b)
  - Either (a) or (b)
46. \_\_\_\_\_ technique search a given class's method for particular clues of interests, then describe how these clues should be tested.
- Error based testing
  - Scenario based testing
  - Unit testing
  - Sanity testing
47. \_\_\_\_\_ concentrates on what the user does, not what the product does.
- Scenario-based testing
  - Usage based testing
  - Error based testing
  - Either (a) or (b)
48. The extent of testing a system is controlled by
- Risks involved
  - Limitations on resources
  - Deadlines
  - All of the above

49. The concept of the \_\_\_\_\_ is used to represent a system whose inside workings are not available for inspection.
- White box testing
  - Black box testing
  - Grey box testing
  - Beta testing
50. \_\_\_\_\_ assumes that the specific logic is important and must be tested to guarantee the system's proper functioning.
- Black box testing
  - White box testing
  - Sanity testing
  - Integration testing

