**Chapter 1**

**INTRODUCTION**

**1.1 Introduction**

Innovative technologies together with mobility have increased the requirement to have more protected and reliable access through predefined gateways. Many organizations are trying to identify accurate, safe, and reliable techniques to protect access rights to their existing services or operation. Biometrics is one answer to these concerns.

Biometrics, especially in information technology, encompasses methods to analyze physical and behavioural identities to extract unique features for identification or monitoring purposes. Various physical features including faces, eyes, fingers, hands, veins, ears and teeth can be used by this technology, and characteristics such as gaits or voice patterns are also being investigated and analysed as part of the wider biometrics field. Biometrics offers a secure method of access to sensitive services and obviates the need to carry a token, card or to remember several passwords. Biometric techniques also reduce the risk of lost, forgotten or copied passwords, stolen tokens or over the shoulder attacks, yet despite these obvious benefits, most biometric techniques are not pervasive in everyday life.

There are some significant reasons for this. The cost of deployment of many techniques is very high; potentially requiring specialist analytical software and machines with the computing power to run it on. There is a lack of standardization of many methods, and the wide variance of algorithms results in different performance levels from comparable equipment. Additionally, end users may refuse to use some types of biometric identification due to possible hygiene misunderstandings, cultural differences or ethical issues.

The exception to this antipathy towards biometrics is fingerprint recognition (hereafter referred to as FR); a well-known technique to identify individuals by comparing fingerprint features with a pre-defined template which most people are familiar with nowadays. FR is widely used today in places such as airports and the legal system, and it is built in to devices such as laptops. More work has been found within the literature, which aims to quantify the best methods and algorithms of FR than any other biometric system; however there is still not a categorical standard algorithm for FR systems. Despite this, identification or authentication through FR still has three main advantages(Maltoni, Maio et al. 2009, Newman 2010).

* Low cost of deployment (cost effective)
* Simple to implement and use
* User must be physically available at the point of identification or verification.

This technique can be used for employee monitoring or in the legal system for criminal identification and most significantly, for time and attendance schemes. Monitoring student attendance has become a prime concern for Universities. Many universities use paper- based or smart card systems to check the students’ attendance. Other projects have been trying to design an attendance system for universities without taking into account recent issues; however this project, unlike the existing schemes, is trying to improve data accuracy by adding a fingerprint-based register and using a series of techniques to provide a reliable, optimal, and accurate identification procedure.

**1.2 Aim of the Project**

Designing a student attendance management system based on fingerprint recognition and faster one to many identification that manages records for attendance in Universities.

**1.3 Motivation and Challenges**

Every organization whether it be an educational institution or business organization, it has to maintain a proper record of attendance of students or employees for effective functioning of organization. Designing a better attendance management system for students so that records be maintained with ease and accuracy was an important key behind motivating this project. This would improve accuracy of attendance records because it will remove all the hassles of roll calling and will save valuable time of the students as well as teachers. Image processing and fingerprint recognition are very advanced today in terms of technology. It was our responsibility to improve fingerprint identification system.

**1.4 Using Biometrics**

* Biometrics is automated methods for recognizing a person based on their physiological or behavioral characteristic.
* Some of the features measured are face, fingerprints, hand geometry, handwriting, iris, retinal, vein, and voice.
* Biometric data are separate and distinct from personal information.
* Biometric templates cannot be reverse-engineered to recreate personal information and they cannot be stolen and used to access personal information.



**Figure 1:** General Architecture of biometrics

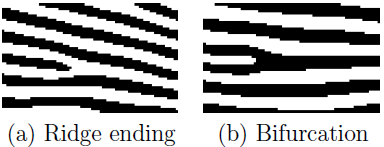
Biometrics refers to metrics related to human characteristics. Biometrics authentication (or realistic authentication)is used in computer science as a form of identification and control. It is also used to identify individuals in groups that are under [surveillance](https://en.wikipedia.org/wiki/Surveillance).

Biometric identifiers are then distinctive, measurable characteristics used to label and describe individuals Biometric identifiers are often categorized as physiological versus behavioral characteristics. Physiological characteristics are related to the shape of the body. Examples include, but are not limited to [fingerprint](https://en.wikipedia.org/wiki/Fingerprint), palm veins, [face recognition](https://en.wikipedia.org/wiki/Facial_recognition_system), [DNA](https://en.wikipedia.org/wiki/DNA), [palm print](https://en.wikipedia.org/wiki/Palm_print), [hand geometry](https://en.wikipedia.org/wiki/Hand_geometry), [iris recognition](https://en.wikipedia.org/wiki/Iris_recognition), [retina](https://en.wikipedia.org/wiki/Retinal_scan) and odour/scent. Behavioral characteristics are related to the pattern of behavior of a person, including but not limited to [typing rhythm](https://en.wikipedia.org/wiki/Keystroke_dynamics), [gait](https://en.wikipedia.org/wiki/Gait_analysis),and voice.Some researchers have coined the term behaviometrics to describe the latter class of biometrics.

Biometric Identification Systems are widely used for unique identification of humans mainly for verification and identification. Biometrics is used as a form of identity access management and access control. So use of biometrics in student attendance management system is a secure approach. There are many types of biometric systems like fingerprint recognition, face recognition, voice recognition, iris recognition, palm recognition etc. In this project, we used fingerprint recognition system.

**1.4.1 What are fingerprints?**

A fingerprint is the pattern of ridges and valleys on the surface of a fingertip. The endpoints and crossing points of ridges are called minutiae. It is a widely accepted assumption that the minutiae pattern of each finger is unique and does not change during one’s life. Ridge endings are the points where the ridge curve terminates, and bifurcations are where a ridge splits from a single path to two paths at a Y-junction. Figure 1 illustrates an example of a ridge ending and a bifurcation. In this example, the black pixels correspond to the ridges, and the white pixels correspond to the valleys.



**Figure 2**: Example of a ridge ending and a bifurcation

When human fingerprint experts determine if two fingerprints are from the same finger, the matching degree between two minutiae pattern is one of the most important factors. Thanks to the similarity to the way of human fingerprint experts and compactness of templates, the minutiae-based matching method is the most widely studied matching method.

**1.4.2 Why Fingerprints?**

Fingerprints are considered to be the best and fastest method for biometric identification. They are secure to use, unique for every person and do not change in one’s lifetime. Besides these, implementation of fingerprint recognition system is cheap, easy and accurate up to satisfactory.

Fingerprint recognition has been widely used in both forensic and civilian applications. Compared with other biometrics features, fingerprint-based biometrics is the most proven technique and has the largest market shares. Not only it is faster than other techniques but also the energy consumption by such systems is too less.

**1.4.3 Fingerprint recognition system for attendance management**

Managing attendance records of students of an institute is a tedious task. It consumes time and paper both. To make all the attendance related work automatic and on-line, we have designed an attendance management system which could be implemented in Universities. It uses a fingerprint identification system developed in this project. This fingerprint identification system uses existing as well as new techniques in fingerprint recognition and matching. A new one to many matching algorithm for large databases has been introduced in this identification system.

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 Introduction**

Research was carried out to identify the requirements of any biometric system, and specifically of a fingerprint recognition system for use within the University environment, before comparing fingerprint recognition suitability with that of other biometric techniques. A choice of fingerprint recognition device had to be considered, and consideration given to the role of identification or verification, so that the best method of operation could be applied. Approaches to analyzing the captured image were also researched and an approach was chosen which could be tested in the context of this project with currently available equipment. A measure of performance had to be discovered and decided upon, before benefits and concerns of fingerprint recognition systems could be considered.

**2.2 Problem Statement**

Survey and analysis of the current monitoring methods has shown that the majority

of lecturers use a paper-based attendance method to keep attendance records while

only some universities use the wall mounted RFID swipe card system.

Problems that have been discovered in using the wall mounted swipe card are:

• The swipe card system is not available in all rooms

• Lecturer cannot access collected data

• The system can be fooled by students (“buddy swiping” an absent student or skipping the session after swiping).

Problems that have been discovered in using paper based registers are:

• Paper based registers are not uploaded to a centralised system, so the   
data is lost for analysis.

• Time taken for data collection impacts on lecturing time.

• The system can be fooled by students “buddy-signing” on behalf of   
absent students.

Consequently, this project proposes Fingerprint’s Recognition as a method to overcome these problems.

**2.3 Why are fingerprints unique?**

What makes fingerprints such a brilliant way of telling people apart is that they are virtually unique: fingerprints develop through an essentially random process according to the code in your DNA (the genetic recipe that tells your body how to develop). Because the environment in the womb also has an effect, even the prints of identical twins are slightly different.The tiny friction ridges on the ends of our fingers and thumbs make it easier to grip things. By making our fingers rougher, these ridges increase the force of friction between our hands and the objects we hold, making it harder to drop things. You have fingerprints even before you're born. In fact, fingerprints are completely formed by the time you're seven months old in the womb. Unless you have accidents with your hands, your fingerprints remain the same throughout your life.

**2.4 Features of fingerprints**

Fingerprints are identified by both macro and micro features.The macro features of a fingerprint include:

• Ridge patterns

• Ridge pattern area

• Core point

• Delta point

• Type lines

• Ridge count

The micro features of a fingerprint are made up of minutia points.

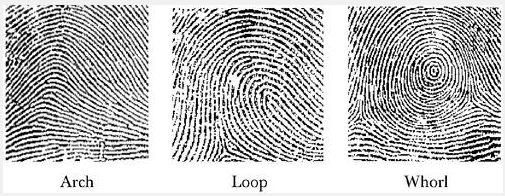
• Type

• Orientation

• Spatial frequency

• Curvature

• Position

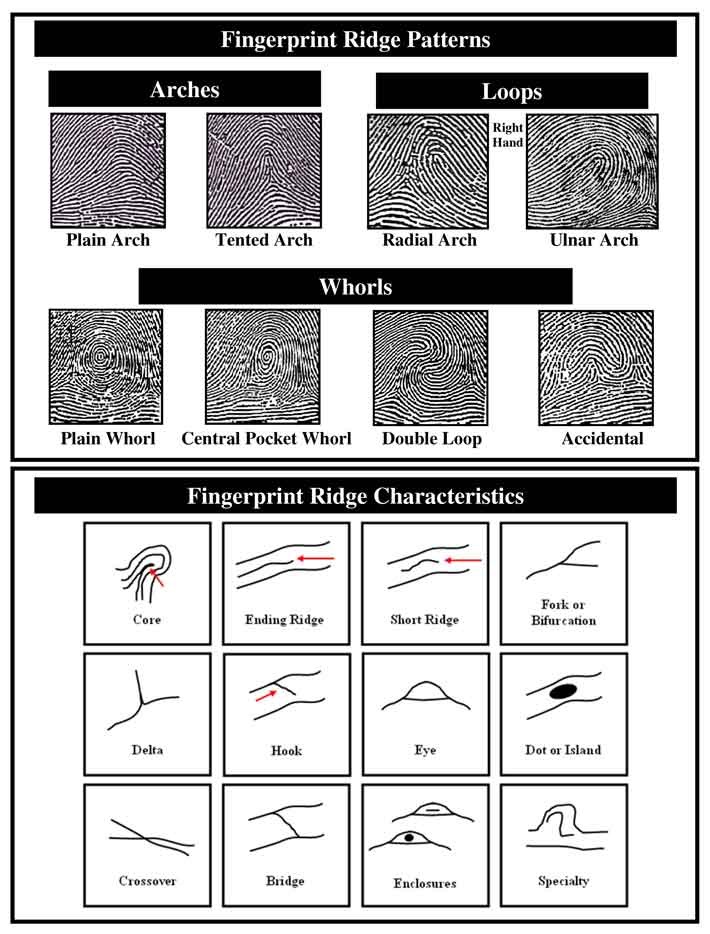


**Figure 3:** Ridge Patterns

Arches account for approximately 5% of the ridge pattern in a given population. Arches are different from loops in that arches are more open curves. Arches can also form a subgroup called tented arches. In a tented arch, the arch angle is much more obtuse than in a normal arch.

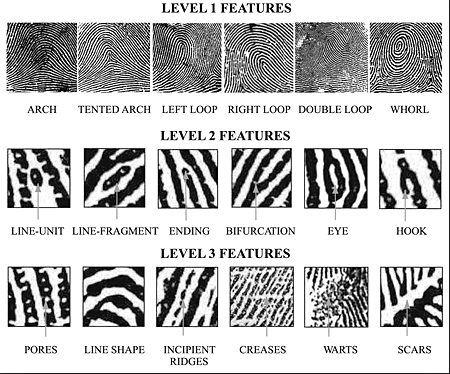
Loops account for approximately 60% of the ridge patterns in a given population. Loops may slant left or right, or be presented as a double loop. A double loop has both a left and right loop, conforming to each other's outline.

Whorls account for approximately 35% of the ridge patterns in a given population. Whorls are defined by at least one ridge making a complete circle.



**Figure 4:** Different types of fingerprint ridge patterns

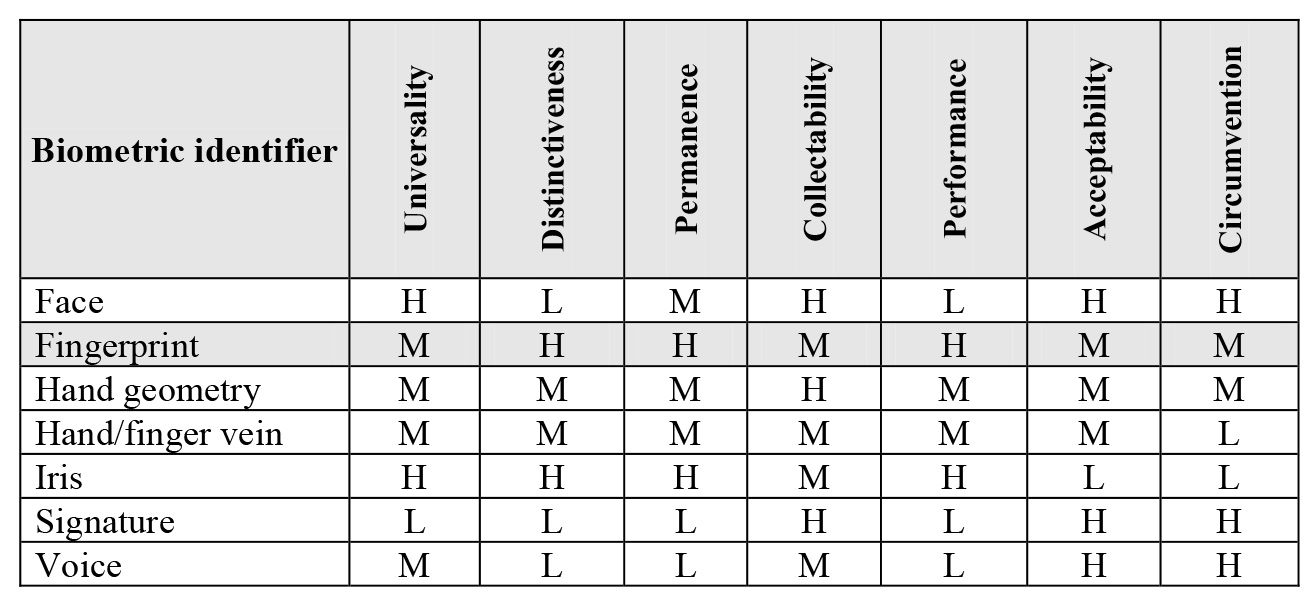
**2.5 Features of Fingerprints based on their importance**



**Figure 5:** Features of fingerprints based on their importance

**2.6 CRITICAL COMPARISON of all biometrics**

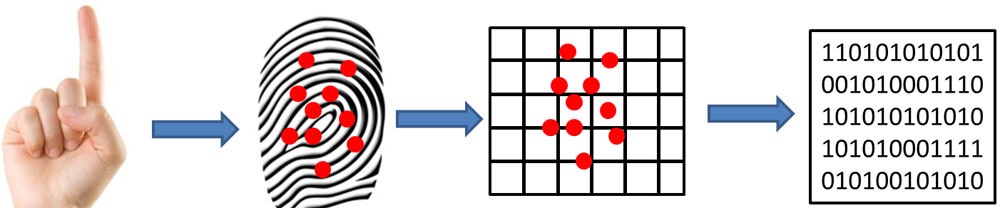
A comparison of fingerprint techniques with the other biometrics methods is given here. This comparison is completely based on the scientific literature but examples are given in an attempt to elucidate the findings (Newman 2010, Kothavale, Markworth et al. 2004). Table 1 shows a comparison of each technique’s suitability (low, medium or high) against Newman’s seven concerns.



**Table** **1**: Comparison of Biometric Techniques

**2.7 How does a fingerprint scanner work ?**

A fingerprint scanner system has two basic jobs, it needs to get an image of your finger, and it needs to determine whether the pattern of ridges and valleys in this image matches the pattern of ridges and valleys in pre-scanned images.Only specific characteristics, which are unique to every fingerprint, are filtered and saved as an encrypted biometric key or mathematical representation. No image of a fingerprint is ever saved, only a series of numbers (a binary code), which is used for verification. The algorithm cannot be reconverted to an image, so no one can duplicate your fingerprints.



**Figure 6:** Working of a Fingerprint Scanner

**CHAPTER 3**

**DESIGN**

**3.1. SYSTEM ANALYSIS**

After analyzing the requirements of the task to be performed, the next step is to analyze the problem and understand its context. The first activity in the phase is studying the existing system and other is to understand the requirements and domain of the new system. Both the activities are equally important, but the first activity serves as a basis of giving the functional specifications and then successful design of the proposed system. Understanding the properties and requirements of a new system is more difficult and requires creative thinking and understanding of existing running system is also difficult, improper understanding of present system can lead diversion from solution.

**3.1.1. EXISTING system**

In the conventional method attendance is taken in different methods,this may be by calling their names or by taking their sign.This system is not accurate.In some of the cases RFID is used for the purpose of attendance but this is also not so accurate because the card can be swiped by any person but not only the true identity.

**3.1.2. Proposed System**

To replace the current traditional attendance system by providing faster, accurate, and efficient system using fingerprint scanner. In the given system we are using a fingerprint scanner to first read and store the fingerprints of the student and while the verification these are used for matching fingerprints and the attendance is updated if fingerprints are matched.The feautures of the fingerprints which are given by the scanner are used for the matching of fingerprints.

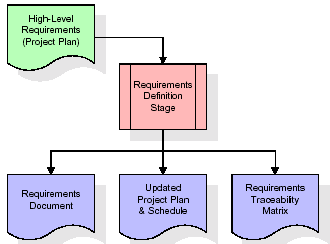
* 1. **SDLC**

SDLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software. Stages in SDLC are

* Requirement Gathering
* Analysis
* Designing
* Coding
* Testing
* Maintenance
  + 1. **REQUIREMENTS GATHERING STAGE**

The requirements gathering process takes as its input the goals identified in the high-level requirements section of the project plan. Each goal will be refined into a set of one or more requirements. These requirements define the major functions of the intended application, define operational data areas and reference data areas, and define the initial data entities. Major functions include critical processes to be managed, as well as mission critical inputs, outputs and reports.

A user class hierarchy is developed and associated with these major functions, data areas, and data entities. Each of these definitions is termed a Requirement. Requirements are identified by unique requirement identifiers and, at minimum, contain a requirement title and textual description.



**Figure 7:** Requirements Gathering

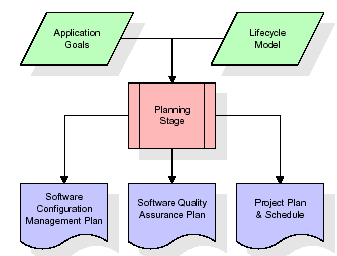
The Systems Development Life Cycle (SDLC), or Software Development Life Cycle in [systems engineering](http://en.wikipedia.org/wiki/Systems_engineering), [information systems](http://en.wikipedia.org/wiki/Information_systems) and [software engineering](http://en.wikipedia.org/wiki/Software_engineering), is the process of creating or altering systems, and the models and [methodologies](http://en.wikipedia.org/wiki/Methodologies) that people use to develop these systems. In software engineering the SDLC concept underpins many kinds of [software development methodologies](http://en.wikipedia.org/wiki/Software_development_methodologies). These methodologies form the framework for planning and controlling the creation of an information system the [software development process](http://en.wikipedia.org/wiki/Software_development_process).

These requirements are fully described in the primary deliverables for this stage: the Requirements Document and the Requirements Traceability Matrix (RTM). The requirements document contains complete descriptions of each requirement, including diagrams and references to external documents as necessary. Note that detailed listings of database tables and fields are not included in the requirements document. The title of each requirement is also placed into the first version of the RTM, along with the title of each goal from the project plan.

The purpose of the RTM is to show that the product components developed during each stage of the software development lifecycle are formally connected to the components developed in prior stages. In the requirements stage, the RTM consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this hierarchical listing, the RTM shows that each requirement developed during this stage is formally linked to a specific product goal. In this format, each requirement can be traced to a specific product goal, hence the term requirements traceability.

* The outputs of the requirements definition stage include the requirements document, the RTM, and an updated project plan.
* Feasibility study is all about identification of problems in a project.
* No. of staff required to handle a project is represented as Team Formation, in this case only modules are individual tasks will be assigned to employees who are working for that project.
* Project Specifications are all about representing of various possible inputs submitting to the server and corresponding outputs along with reports maintained by administrator.
  + 1. **ANALYSIS STAGE**

The planning stage establishes a bird's eye view of the intended software product, and uses this to establish the basic project structure, evaluate feasibility and risks associated with the project, and describe appropriate management and technical approaches.



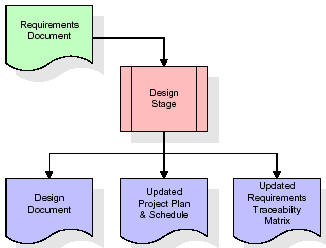
**Figure 8:** Analysis stage

The most critical section of the project plan is a listing of high-level product requirements, also referred to as goals. All of the software product requirements to be developed during the requirements definition stage flow from one or more of these goals. The minimum information for each goal consists of a title and textual description, although additional information and references to external documents may be included.

The outputs of the project planning stage are the configuration management plan, the quality assurance plan, and the project plan and schedule, with a detailed listing of scheduled activities for the upcoming Requirements stage, and high level estimates of effort for the out stages.

* + 1. **DESIGNING STAGE**

The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary. These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input.

****

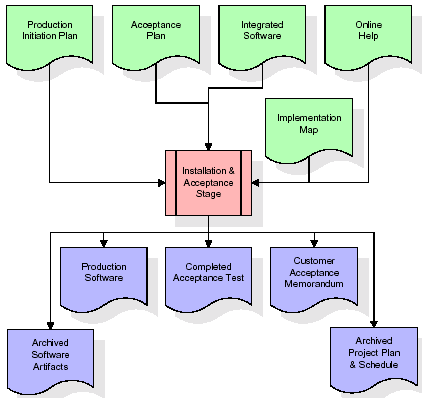
**Figure 9:** Design stage

When the design document is finalized and accepted, the RTM is updated to show that each design element is formally associated with a specific requirement. The outputs of the design stage are the design document, an updated RTM, and an updated project plan.

* + 1. **DEVELOPMENT (CODING) STAGE**

The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artifacts will be produced. Software artifacts include but are not limited to menus, dialogs, and data management forms, data reporting formats, and specialized procedures and functions. Appropriate test cases will be developed for each set of functionally related software artifacts, and an online help system will be developed to guide users in their interactions with the software.

The RTM will be updated to show that each developed artifact is linked to a specific design element, and that each developed artifact has one or more corresponding test case items. At this point, the RTM is in its final configuration. The outputs of the development stage include a fully functional set of software that satisfies the requirements and design elements previously documented, an online help system that describes the operation of the software, an implementation map that identifies the primary code entry points for all major system functions, a test plan that describes the test cases to be used to validate the correctness and completeness of the software, an updated RTM, and an updated project plan.



**Figure 10:** Development (Coding) Stage

The primary outputs of the installation and acceptance stage include a production application, a completed acceptance test suite, and a memorandum of customer acceptance of the software. Finally, the PDR enters the last of the actual labor data into the project schedule and locks the project as a permanent project record. At this point the PDR "locks" the project by archiving all software items, the implementation map, the source code, and the documentation for future reference.

**3.2.5. MAINTENANCE**

Outer rectangle represents maintenance of a project, Maintenance team will start with requirement study, understanding of documentation later employees will be assigned work and they will undergo training on that particular assigned category. For this life cycle there is no end, it will be continued so on like an umbrella (no ending point to umbrella sticks).

**3.3 DATA FLOW DIAGRAMS**

A data flow diagram is graphical tool used to describe and analyze movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. The physical data flow diagrams show the actual implements and movement of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams.. Each component in a DFD is labeled with a descriptive name. Process is further identified with a number that will be used for identification purpose.

The development of DFD’S is done in several levels. Each process in lower level diagrams can be broken down into a more detailed DFD in the next level. The lop-level diagram is often called context diagram. It consist a single process bit, which plays vital role in studying the current system. The process in the context level diagram is exploded into other process at the first level DFD.

The idea behind the explosion of a process into more process is that understanding at one level of detail is exploded into greater detail at the next level. This is done until further explosion is necessary and an adequate amount of detail is described for analyst to understand the process.

Data flow diagrams can be used to provide a clear representation of any business function. The technique starts with an overall picture of the business and continues by analyzing each of the functional areas of interest. This analysis can be carried out in precisely the level of detail required.

The technique exploits a method called top-down expansion to conduct the analysis in a targeted way. As the name suggests, Data Flow Diagram (DFD) is an illustration that explicates the passage of information in a process. A DFD can be easily drawn using simple symbols. Additionally, complicated processes can be easily automated by creating DFDs using easy-to-use, free downloadable diagramming tools. A DFD is a model for constructing and analyzing information processes.

A DFD is also known as a “bubble Chart” has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So it is the starting point of the design to the lowest level of detail. A DFD consists of a series of bubbles joined by data flows in the system.

Several rules of thumb are used in drawing DFD’S are here. Process should be named and numbered for an easy reference. Each name should be representative of the process. The direction of flow is from top to bottom and from left to right. Data traditionally flow from source to the destination although they may flow back to the source. One way to indicate this is to draw long flow line back to a source. An alternative way is to repeat the source symbol as a destination. Since it is used more than once in the DFD it is marked with a short diagonal. When a process is exploded into lower level details, they are numbered. The names of data stores and destinations are written in capital letters. Process and dataflow names have the first letter of each work capitalized.

A DFD typically shows the minimum contents of data store. Each data store should contain all the data elements that flow in and out.

Salient features of DFD’S are as follows

The DFD shows flow of data, not of control loops and decision are controlled considerations do not appear on a DFD. The DFD does not indicate the time factor involved in any process whether the dataflow take place daily, weekly, monthly or yearly. The sequence of events is not brought out on the DFD.

Data cannot move directly from an outside source to a data store, a process, which receives, must move data from the source and place the data into data store

* A data store has a noun phrase label.
* Data cannot move direly from a source to sink it must be moved by a process
* A source and /or sink has a noun phrase land

A Data Flow has only one direction of flow between symbols. It may flow in both directions between a process and a data store to show a read before an update. The later is usually indicated however by two separate arrows since these happen at different type. A join in DFD means that exactly the same data comes from any of two or more different processes data store or sink to a common location. A data flow cannot go directly back to the same process it leads. There must be at least one other process that handles the data flow produce some other data flow returns the original data into the beginning process.

1. A Data flow to a data store means update (delete or change).
2. A data Flow from a data store means retrieve or use.

A data flow has a noun phrase label more than one data flow noun phrase can appear on a single arrow as long as all of the flows on the same arrow move together as one package.

* 1. **UML DIAGRAMS:**

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

UML is a general-purpose visual modeling language that is used to specify, visualize, construct, and document the artifacts of the software system. It will provide vocabulary and rules for communications and function on conceptual and physical representation. So it is modeling language.

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

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

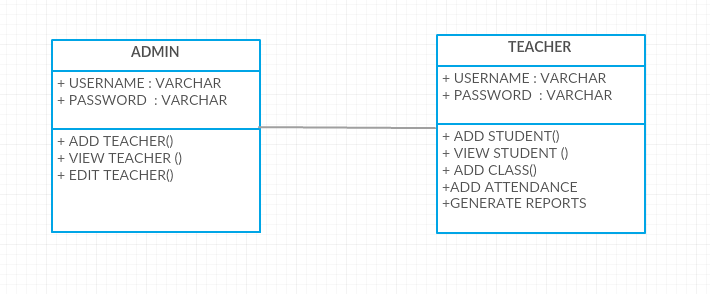
UML includes both graphical and textual representation. It makes easy to visualize the system and for better understanding.UML models can be directly connected to a variety of programming languages and it is sufficiently expressive and free from any ambiguity to permit the direct execution of models.UML provides variety of documents in addition raw executable codes.



**Figure 11:** Modeling a System Architecture using views of UML

**3.4.1. CLASS DIAGRAM**

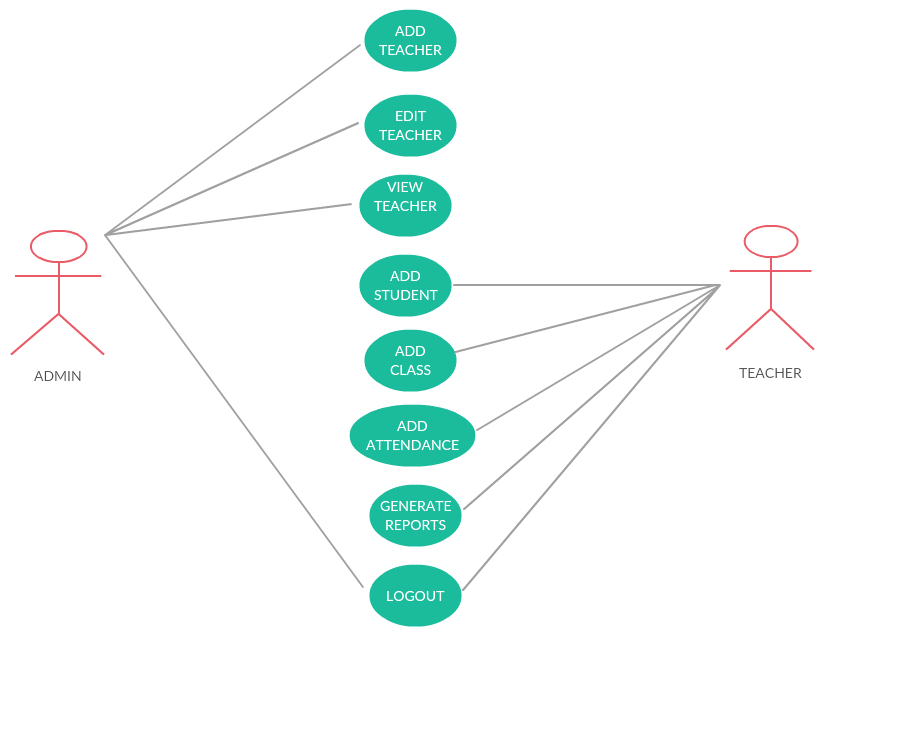
Class diagrams are widely used to describe the types of objects in a system and their relationships. Class diagrams model class structure and contents using design elements such as classes, packages and objects. Class diagrams describe three different perspectives when designing a system, conceptual, specification, and implementation. These perspectives become evident as the diagram is created and help solidify the design. Class diagrams are arguably the most used UML diagram type. It is the main building block of any object oriented solution. It shows the classes in a system, attributes and operations of each class and the relationship between each class. In most modeling tools a class has three parts, name at the top, attributes in the middle and operations or methods at the bottom. In large systems with many classes related classes are grouped together to create class diagrams. Different relationships between diagrams are show by different types of Arrows.



**Figure 12:** Class Diagram

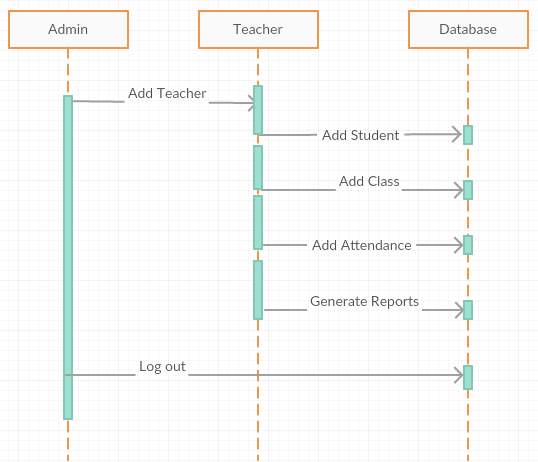
**3.4.2. USE-CASE DIAGRAM**

A use case is a set of scenarios that describing an interaction between a user and a system.  A use case diagram displays the relationship among actors and use cases.  The two main components of a use case diagram are use cases and actors. An actor represents a user or another system that will interact with the system you are modeling.  A use case is an external view of the system that represents some action the user might perform in order to complete a task.

**Figure 13:** Use Case

**3.4.3. SEQUENCE DIAGRAM**

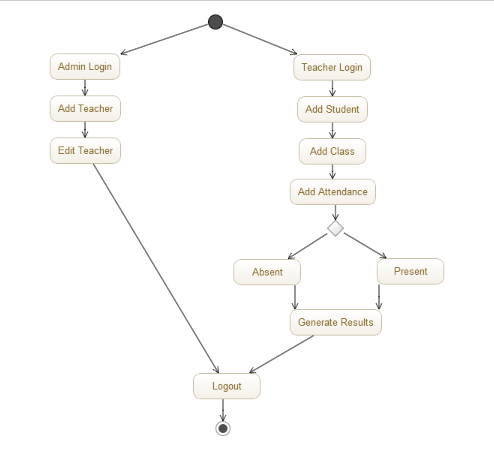
Sequence diagrams in UML shows how object interact with each other and the order those interactions occur. It’s important to note that they show the interactions for a particular scenario. The processes are represented vertically and interactions are show as arrows. This article explains the purpose and the basics of Sequence diagrams.



**Figure 14:** Sequence Diagram

**3.4 ACTIVITY DIAGRAM**

Activity diagrams describe the workflow behavior of a system.  Activity diagrams are similar to state diagrams because activities are the state of doing something.  The diagrams describe the state of activities by showing the sequence of activities performed.  Activity diagrams can show activities that are conditional or parallel.

****

**Figure 15:** Activity Diagram

**CHAPTER 4**

**IMPLEMENTATION**

**4.1. INTRODUCTION**

implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and it’s constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods. Implementation is one of the most important tasks in project is the phase in which one has to be cautions because all the efforts undertaken during the project will be very interactive. Implementation is the most crucial stage in achieving successful system and giving the users confidence that the new system is workable and effective. Each program is tested individually at the time of development using the sample data and has verified that these programs link together in the way specified in the program specification. The computer system and its environment are tested to the satisfaction of the user.

The implementation phase is less creative than system design. It is primarily concerned with user training, and file conversion. The system may be requiring extensive user training. The initial parameters of the system should be modifies as a result of a programming. A simple operating procedure is provided so that the user can understand the different functions clearly and quickly. The different reports can be obtained either on the inkjet or dot matrix printer, which is available at the disposal of the user.

The proposed system is very easy to implement. In general implementation is used to mean the process of converting a new or revised system design into an operational one.

**4.2. SOFTWARE REQUIREMENT SPECIFICATION**

Software requirements specification is a description of a software system to be developed. It lays out functional and non-functional requirements, and may include a set of use cases that describe user interactions that the software must provide. It is a structured collection of information that embodies the requirements of a system.

A Software Requirements Specification (SRS) – a [requirements specification](http://en.wikipedia.org/wiki/Requirements_specification) for a [software system](http://en.wikipedia.org/wiki/Software_system) is a complete description of the behavior of a system to be developed. It includes a set of [use](http://en.wikipedia.org/wiki/Use_case) cases that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. [Nonfunctional requirements](http://en.wikipedia.org/wiki/Non-functional_requirements) are requirements which impose constraints on the design or implementation (such as [performance engineering](http://en.wikipedia.org/wiki/Performance_engineering) requirements, [quality](http://en.wikipedia.org/wiki/Quality_%2528business%2529) standards, or design constraints).System requirements specification is a structured collection of information that embodies the requirements of a system.

A [business analyst](http://en.wikipedia.org/wiki/Business_analyst), sometimes titled [system analyst](http://en.wikipedia.org/wiki/System_analyst), is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions. Within the [systems development lifecycle](http://en.wikipedia.org/wiki/Systems_development_life_cycle) domain, the BA typically performs a liaison function between the business side of an enterprise and the information technology department or external service providers. Projects are subject to three sorts of requirements. [Business requirements](http://en.wikipedia.org/wiki/Business_requirements) describe in business terms what must be delivered or accomplished to provide value.

Product requirements describe properties of a system or product (which could be one of several ways to accomplish a set of business requirements.)Process requirements describe activities performed by the developing organization. For instance, process requirements could specify .Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization.

The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

**4.2.1. ECONOMIC FEASIBILITY**

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs. The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, There is nominal expenditure and economical feasibility for certain

**4.2.2**. **Operational Feasibility**

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization’s operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits. The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

**4.2.3. TECHNICAL FEASIBILITY**

Earlier no system existed to cater to the needs of ‘Secure Infrastructure Implementation System’. The current system developed is technically feasible. It is a web based user interface for audit workflow at NIC-CSD. Thus it provides an easy access to .the users. The database’s purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security.

Earlier no system existed to cater to the needs of ‘Secure Infrastructure Implementation System. The current system developed is technically feasible. It is a web based user interface for audit workflow at NIC-CSD. Thus it provides an easy access to .the users. The database’s purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security.

**4.3. SYSTEM REQUIREMENTS**

* + 1. **HARDWARE REQUIREMENTS**

System : Pentium IV 2.4 GHz

Hard Disk : 40 GB

Floppy Drive : 1.44 Mb

Monitor : 15 VGA Color

Mouse : Logitech

Ram : 512 Mb

Fingerprint Scanner :1

* + 1. **SOFTWARE REQUIREMENTS**

Operating system : Windows 7

       Technology Used : C#, Microsoft .NET

Backend Used : SQL Server 2005

**4.4. Microsoft.NET framework**

**4.4.1. FEATURES OF .NET**

Microsoft .NET is a set of Microsoft software technologies for rapidly building and integrating XML Web services, Microsoft Windows-based applications, and Web solutions. The .NET Framework is a language-neutral platform for writing programs that can easily and securely interoperate. There’s no language barrier with .NET: there are numerous languages available to the developer including Managed C++, C#, Visual Basic and Java Script. The .NET framework provides the foundation for components to interact seamlessly, whether locally or remotely on different platforms. It standardizes common data types and communications protocols so that components created in different languages can easily interoperate.

“.NET” is also the collective name given to various software components built upon the .NET platform. These will be both products (Visual Studio.NET and Windows.NET Server, for instance) and services (like Passport, .NET My Services, and so on).

1. **THE .NET FRAMEWORK**

The .NET Framework has two main parts:

1. The Common Language Runtime (CLR).

2. A hierarchical set of class libraries.

The CLR is described as the “execution engine” of .NET. It provides the environment within which programs run. The most important features are

* Conversion from a low-level assembler-style language, called Intermediate Language (IL), into code native to the platform being executed on.
* Memory management, notably including garbage collection.
* Checking and enforcing security restrictions on the running code.
* Loading and executing programs, with version control and other such features.
* The following features of the .NET framework are also worth description:

1. **MANAGED CODE**

The code that targets .NET, and which contains certain extra Information - “metadata” - to describe itself. Whilst both managed and unmanaged code can run in the runtime, only managed code contains the information that allows the CLR to guarantee, for instance, safe execution and interoperability.

1. **MANAGED DATA**

With Managed Code comes Managed Data. CLR provides memory allocation and Deal location facilities, and garbage collection. Some .NET languages use Managed Data by default, such as C#, Visual Basic.NET and JScript.NET, whereas others, namely C++, do not. Targeting CLR can, depending on the language you’re using, impose certain constraints on the features available. As with managed and unmanaged code, one can have both managed and unmanaged data in .NET applications - data that doesn’t get garbage collected but instead is looked after by unmanaged code.

1. **COMMON TYPE SYSTEM**

The CLR uses something called the Common Type System (CTS) to strictly enforce type-safety. This ensures that all classes are compatible with each other, by describing types in a common way. CTS define how types work within the runtime, which enables types in one language to interoperate with types in another language, including cross-language exception handling. As well as ensuring that types are only used in appropriate ways, the runtime also ensures that code doesn’t attempt to access memory that hasn’t been allocated to it.

1. **COMMON LANGUAGE SPECIFICATION**

The CLR provides built-in support for language interoperability. To ensure that you can develop managed code that can be fully used by developers using any programming language, a set of language features and rules for using them called the Common Language Specification (CLS) has been defined. Components that follow these rules and expose only CLS features are considered CLS-compliant.

1. **THE CLASS LIBRARY**

.NET provides a single-rooted hierarchy of classes, containing over 7000 types. The root of the namespace is called System; this contains basic types like Byte, Double, Boolean, and String, as well as Object. All objects derive from System. Object. As well as objects, there are value types. Value types can be allocated on the stack, which can provide useful flexibility. There are also efficient means of converting value types to object types if and when necessary.

The set of classes is pretty comprehensive, providing collections, file, screen, and network I/O, threading, and so on, as well as XML and database connectivity.

The class library is subdivided into a number of sets (or namespaces), each providing distinct areas of functionality, with dependencies between the namespaces kept to a minimum.

1. **LANGUAGES SUPPORTED BY .NET**

The multi-language capability of the .NET Framework and Visual Studio .NET enables developers to use their existing programming skills to build all types of applications and XML Web services. The .NET framework supports new versions of Microsoft’s old favorites Visual Basic and C++ (as VB.NET and Managed C++), but there are also a number of new additions to the family. Visual Basic .NET has been updated to include many new and improved language features that make it a powerful object-oriented programming language. These features include inheritance, interfaces, and overloading, among others.

Visual Basic also now supports structured exception handling, custom attributes and also supports multi-threading. Visual Basic .NET is also CLS compliant, which means that any CLS-compliant language can use the classes, objects, and components you create in Visual Basic .NET.

Managed Extensions for C++ and attributed programming are just some of the enhancements made to the C++ language. Managed Extensions simplify the task of migrating existing C++ applications to the new .NET Framework. C# is Microsoft’s new language. It’s a C-style language that is essentially “C++ for Rapid Application Development”. Unlike other languages, its specification is just the grammar of the language. It has no standard library of its own, and instead has been designed with the intention of using the .NET libraries as its own. Microsoft Visual J# .NET provides the easiest transition for Java-language developers into the world of XML Web Services and dramatically improves the interoperability of Java-language programs with existing software written in a variety of other programming languages. Active State has created Visual Perl and Visual Python, which enable .NET-aware applications to be built in either Perl or Python. Both products can be integrated into the Visual Studio .NET environment. Visual Perl includes support for Active State’s Perl Dev Kit. Other languages for which .NET compilers are available include

* FORTRAN
* COBOL
* Eiffel

|  |  |
| --- | --- |
| ASP.NET  XML WEB SERVICES | Windows Forms |
| Base Class Libraries | |
| Common Language Runtime | |
| Operating System | |

**Figure 16**: **.**Net Framework

C#.NET is also compliant with CLS (Common Language Specification) and supports structured exception handling. CLS is set of rules and constructs that are supported by the CLR (Common Language Runtime). CLR is the runtime environment provided by the .NET Framework; it manages the execution of the code and also makes the development process easier by providing services C#.NET is a CLS-compliant language. Any objects, classes, or components that created in C#.NET can be used in any other CLS-compliant language. In addition, we can use objects, classes, and components created in other CLS-compliant languages in C#.NET .The use of CLS ensures complete interoperability among applications, regardless of the languages used to create the application.

1. **CONSTRUCTORS AND DESTRUCTORS**

Constructors are used to initialize objects, whereas destructors are used to destroy them. In other words, destructors are used to release the resources allocated to the object. In C#.NET the sub finalize procedure is available. The sub finalize procedure is used to complete the tasks that must be performed when an object is destroyed. The sub finalize procedure is called automatically when an object is destroyed. In addition, the sub finalize procedure can be called only from the class it belongs to or from derived classes.

1. **GARBAGE COLLECTION**

Garbage Collection is another new feature in C#.NET. The .NET Framework monitors allocated resources, such as objects and variables. In addition, the .NET Framework automatically releases memory for reuse by destroying objects that are no longer in use. In C#.NET, the garbage collector checks for the objects that are not currently in use by applications. When the garbage collector comes across an object that is marked for garbage collection, it releases the memory occupied by the object.

1. **OVERLOADING**

Overloading is another feature in C#. Overloading enables us to define multiple procedures with the same name, where each procedure has a different set of arguments. Besides using overloading for procedures, we can use it for constructors and properties in a class.

1. **MULTITHREADING:**

C#.NET also supports multithreading. An application that supports multithreading can handle multiple tasks simultaneously, we can use multithreading to decrease the time taken by an application to respond to user interaction.

1. **STRUCTURED EXCEPTION HANDLING**

C#.NET supports structured handling, which enables us to detect and remove errors at runtime. In C#.NET, we need to use Try…Catch…Finally statements to create exception handlers. Using Try…Catch…Finally statements, we can create robust and effective exception handlers to improve the performance of our application.

1. **THE .NET FRAMEWORK**

The .NET Framework is a new computing platform that simplifies application development in the highly distributed environment of the Internet.

1. **OBJECTIVES OF. NET FRAMEWORK**
2. To provide a consistent object-oriented programming environment whether object codes is stored and executed locally on Internet-distributed, or executed remotely.
3. To provide a code-execution environment to minimizes software deployment and guarantees safe execution of code.
4. Eliminat1es the performance problems.

There are different types of application, such as Windows-based applications and Web-based applications.

**4.4.2. FEATURES OF SQL-SERVER**

The OLAP Services feature available in SQL Server version 7.0 is now called SQL Server 2000 Analysis Services. The term OLAP Services has been replaced with the term Analysis Services. Analysis Services also includes a new data mining component. The Repository component available in SQL Server version 7.0 is now called Microsoft SQL Server 2000 Meta Data Services. References to the component now use the term Meta Data Services.

The term repository is used only in reference to the repository engine within Meta Data Services

SQL-SERVER database consist of six types of objects. They are:

1. TABLE

2. QUERY

3. FORM

4. REPORT

5. MACRO

**4.5. IMPLEMENTATION MODULES**

**4.5.1 Admin Module**

Admin module is totally about administration rights and will help one in adding various numbers of administrators such as teachers etc and offers them necessary access permissions. The main responsibility of admin module includes help desk support managing various functionalities of administrators such as adding faculty, edit faculty information and view the same.

**4.5.2 Teacher Module**

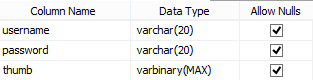
Teacher module consists of many functions. The student is added, viewed and edited by the teacher. The admin has no control over the student details. The admin can just add a teacher where the teacher has the sole authority over the student. After the student is added, the teacher needs to add the class for which the attendance needs to be updated. Teacher stores the fingerprints of the students while adding the student and uses them in the verification process. The student’s attendance is updated when the fingerprints match from the database.

**4.5.3 Reports Generation**

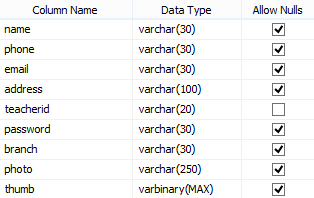
After the attendance is updated, the reports are to be generated for finding the defaulter’s list. In this module we can get the attendance percentage of each student either overall attendance or in a specific subject. These reports can be used for analyzing the students average attendance percentage and filter the students who are lacking attendance.

**4.6 TABLES**

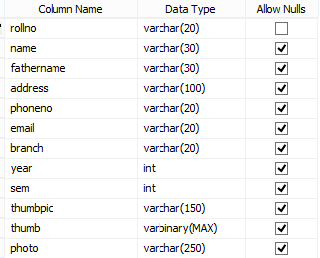
**Table 2**: Admin Table



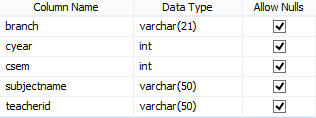
**Table 3**: Teacher Table



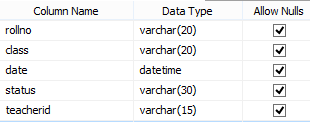
**Table 4:** Student Table



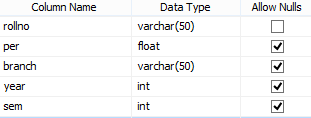
**Table 5:** Classes Table



**Table 6:** Attendance Table



**Table 7:** Reports Summary Table



**CHAPTER 5**

**SYSTEM TESTING**

In order to conduct a testing the list of requirements was used to find out which function must be examined. Additionally, it was important to prepare testing for the user interface and other components. This chapter as a testing section can be identified that the application:

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

**5.1. TYPES OF TESTS**

**A. Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**B. Integration testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

**C. Functional test**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**D. System Test**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

1. **White Box Testing**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

1. **Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**E. UNIT TESTING**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

**Test strategy and approach**

Field testing will be performed manually and functional tests will be written in detail.

**Test objectives**

* The details entered by the user must be entered into the database properly.
* The tweets pertaining to a certain keyword are to be retrieved from twitter and those tweets should prove to be valid.
* The result showing the sentiment of the tweets should be correct.

**Features to be tested**

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

**F. INTEGRATION TESTING**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**G. ACCEPTANCE TESTING**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

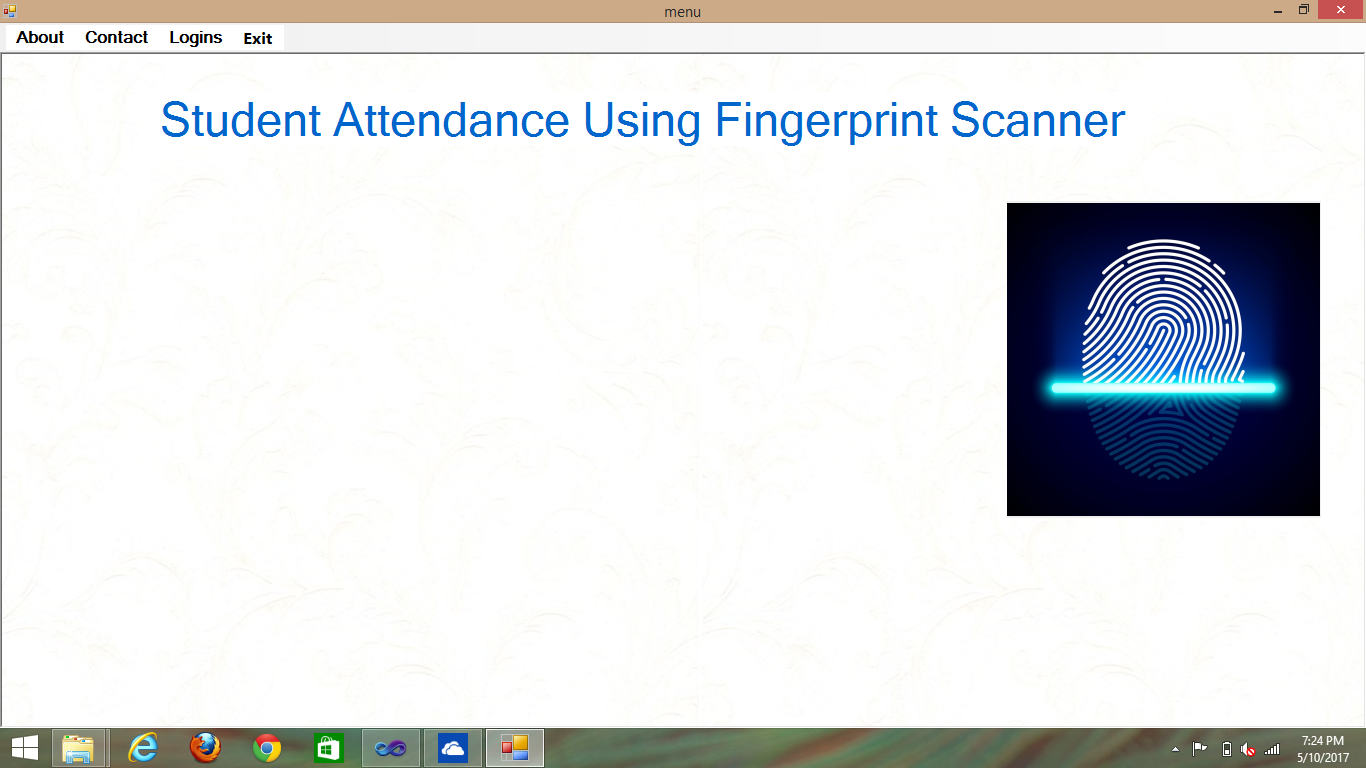
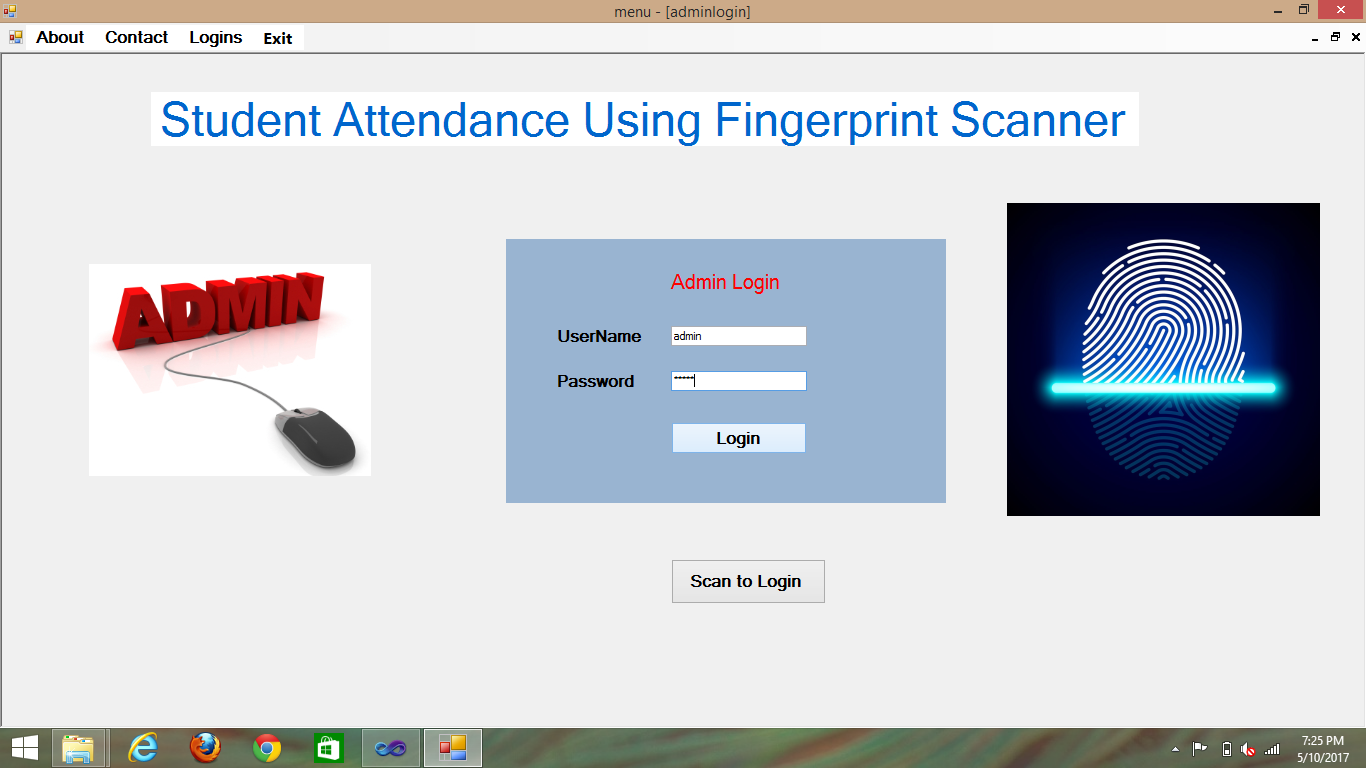
**CHAPTER 6**

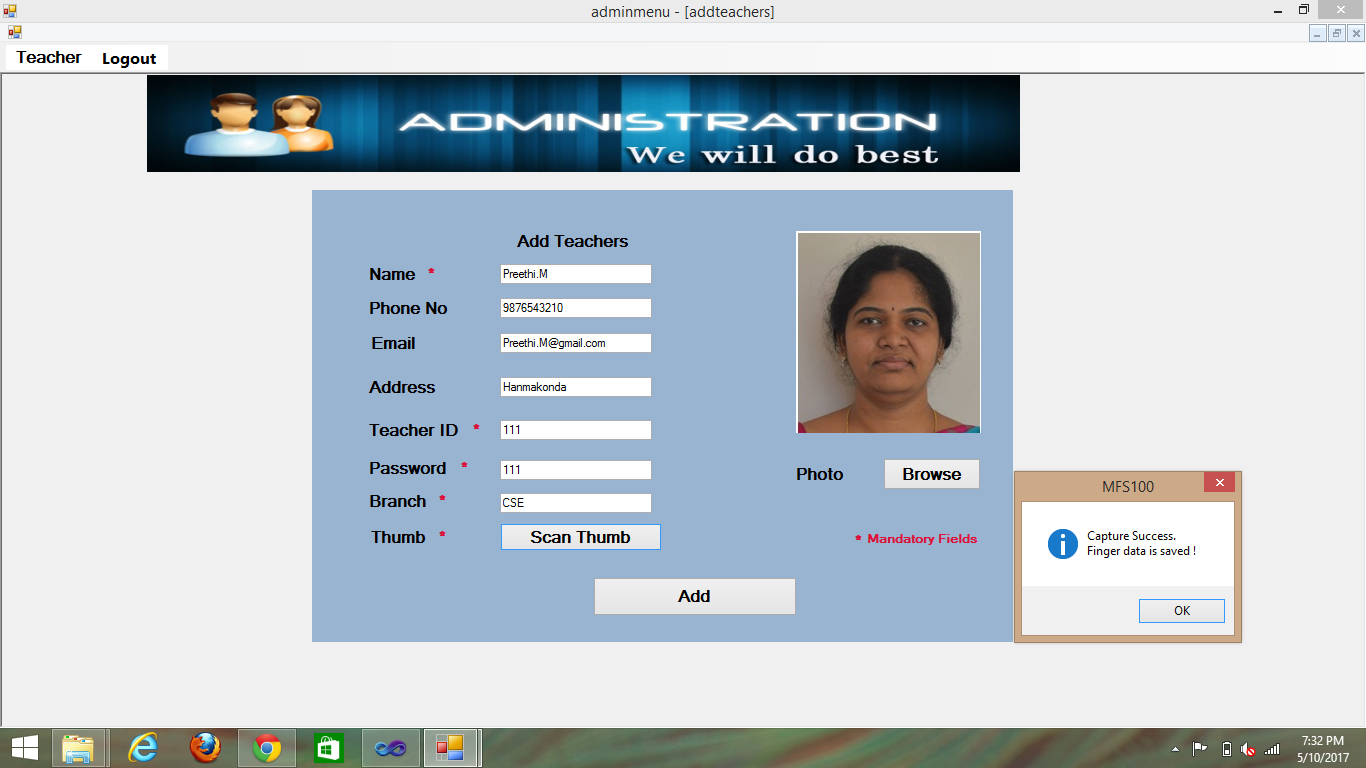
**RESULTS**

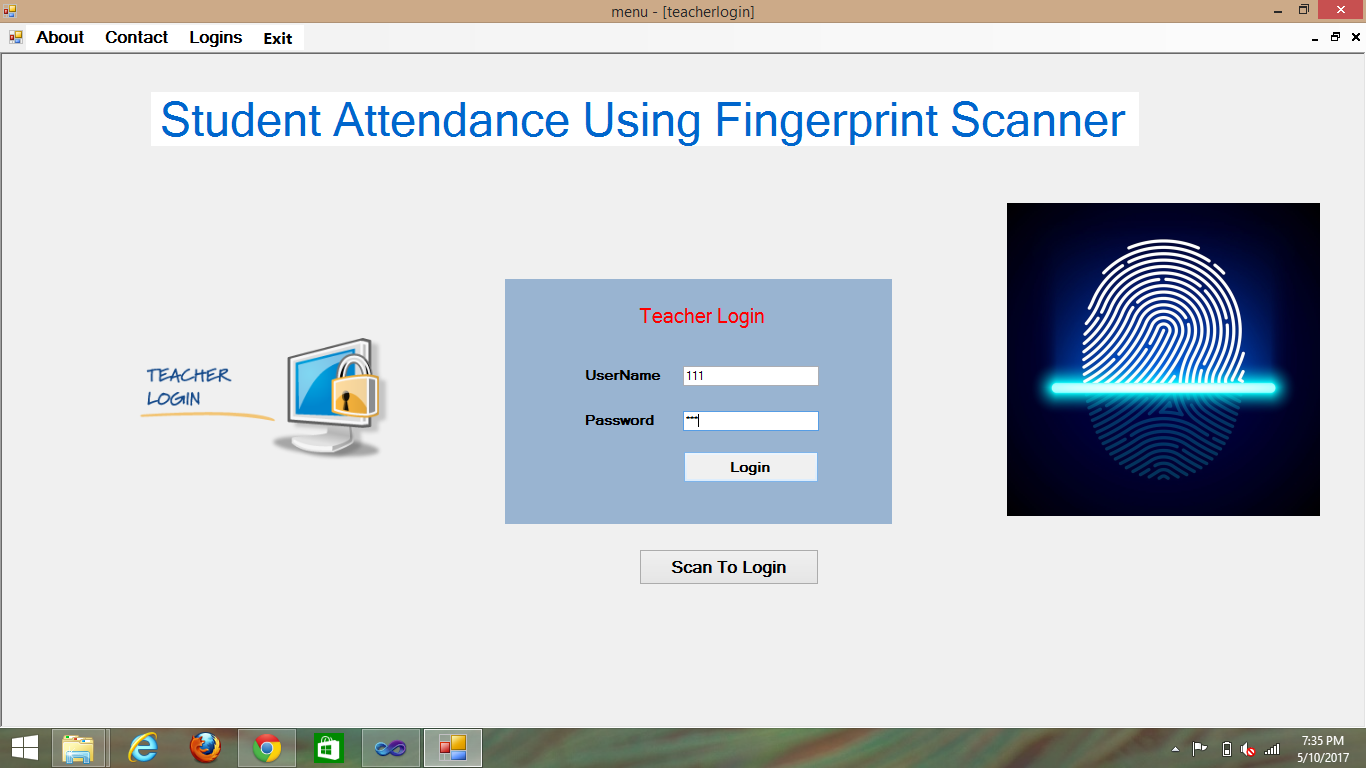
There are two logins

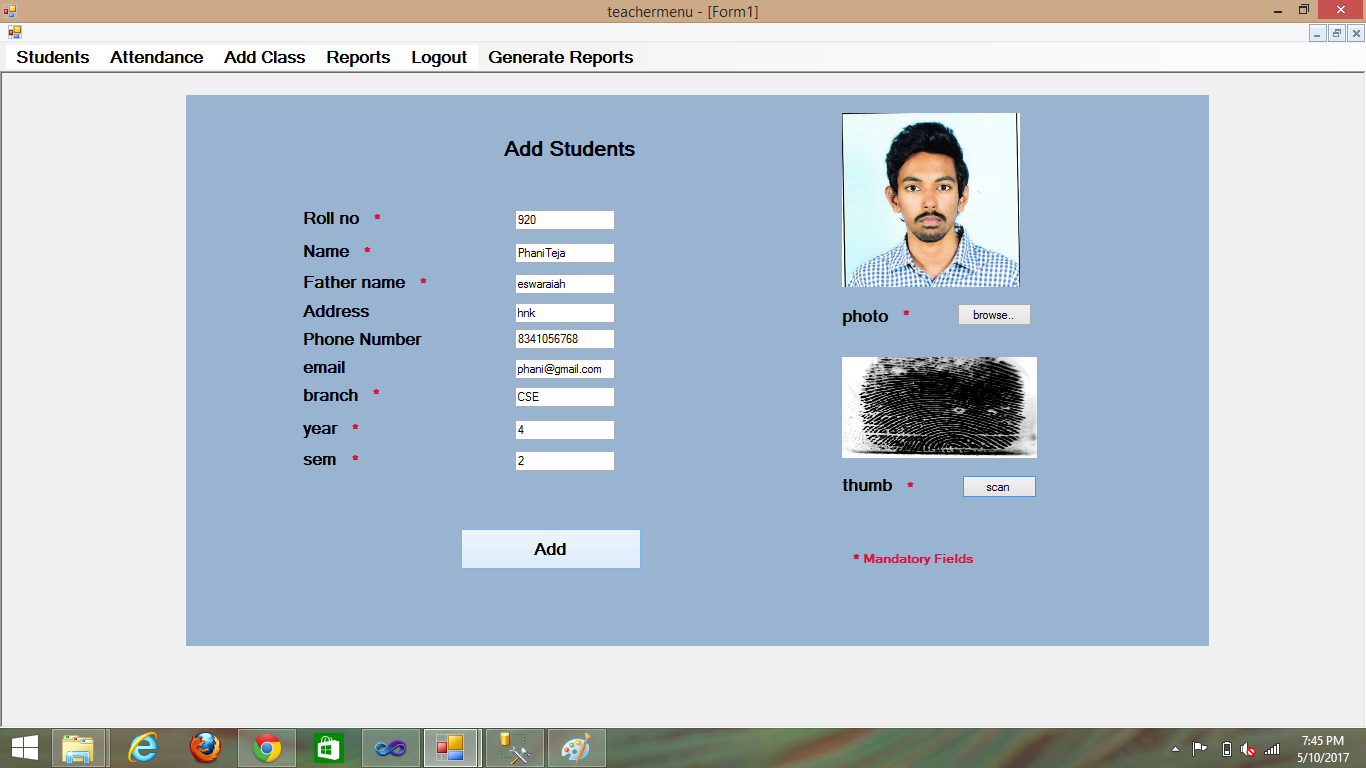
1. ADMIN LOGIN

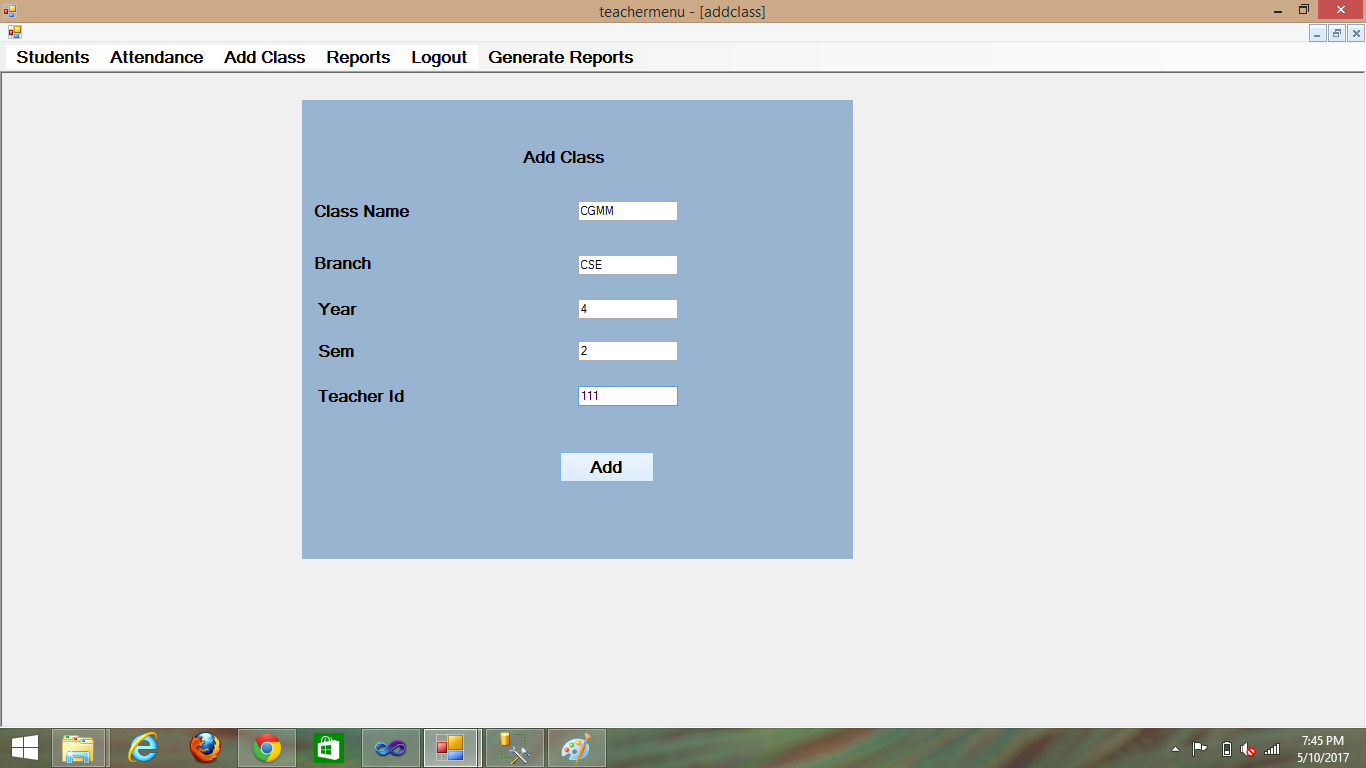
2. STUDENT LOGIN

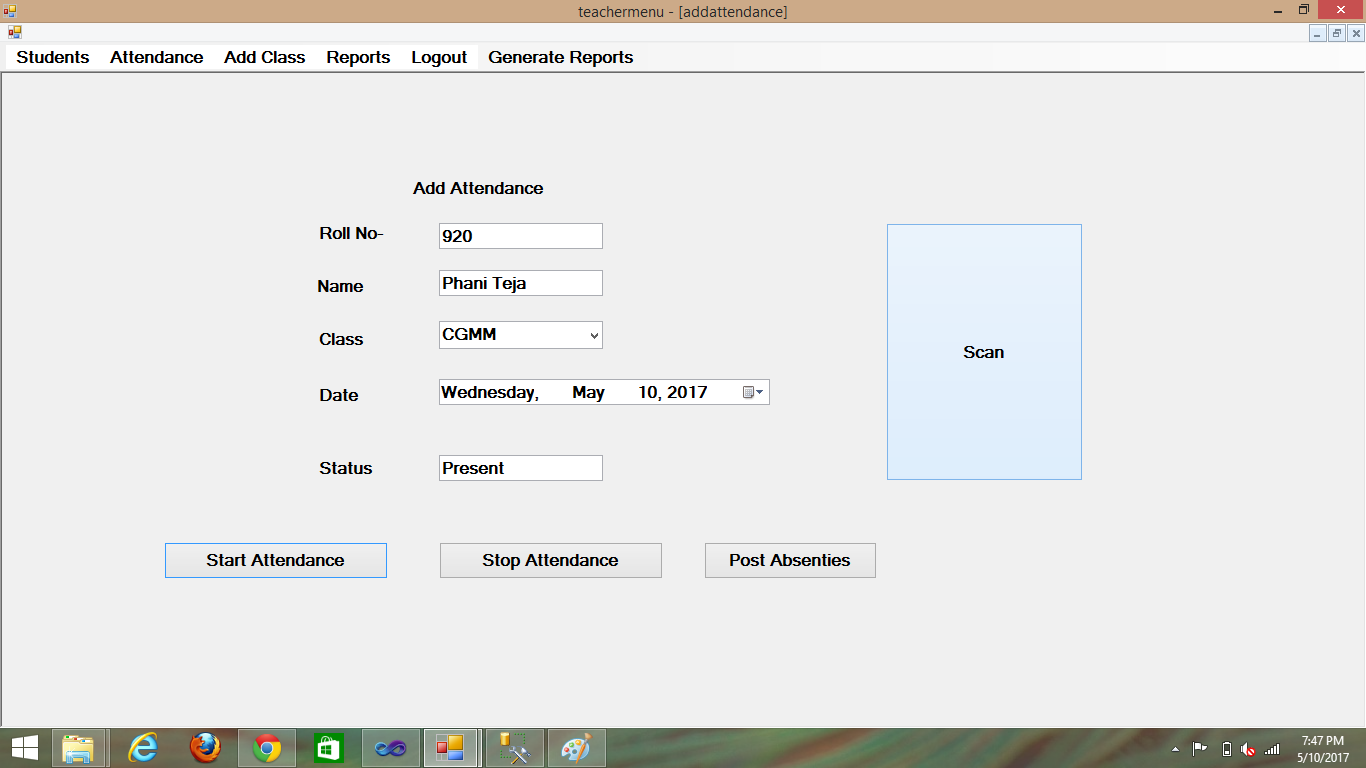
**HOME**

**Adding a teacher**

** Teacher Login**

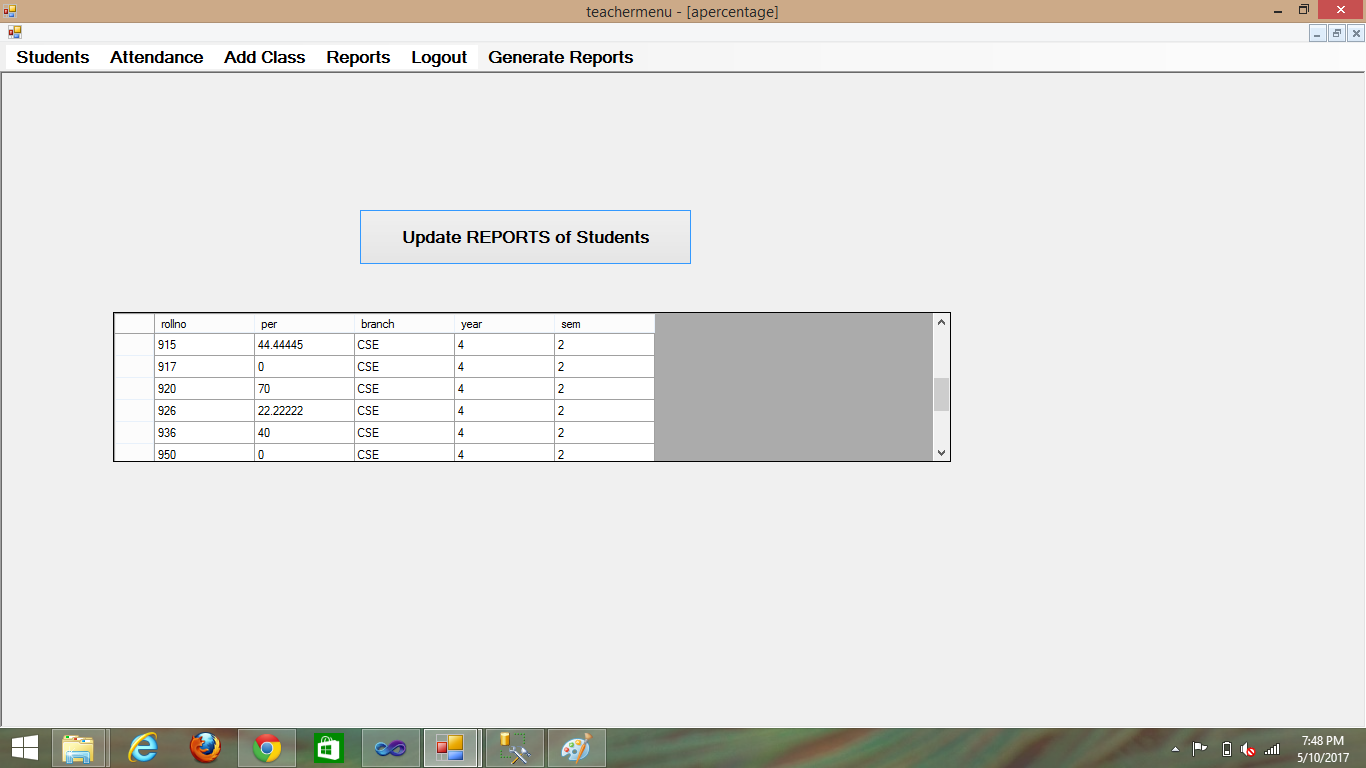
**Adding student**

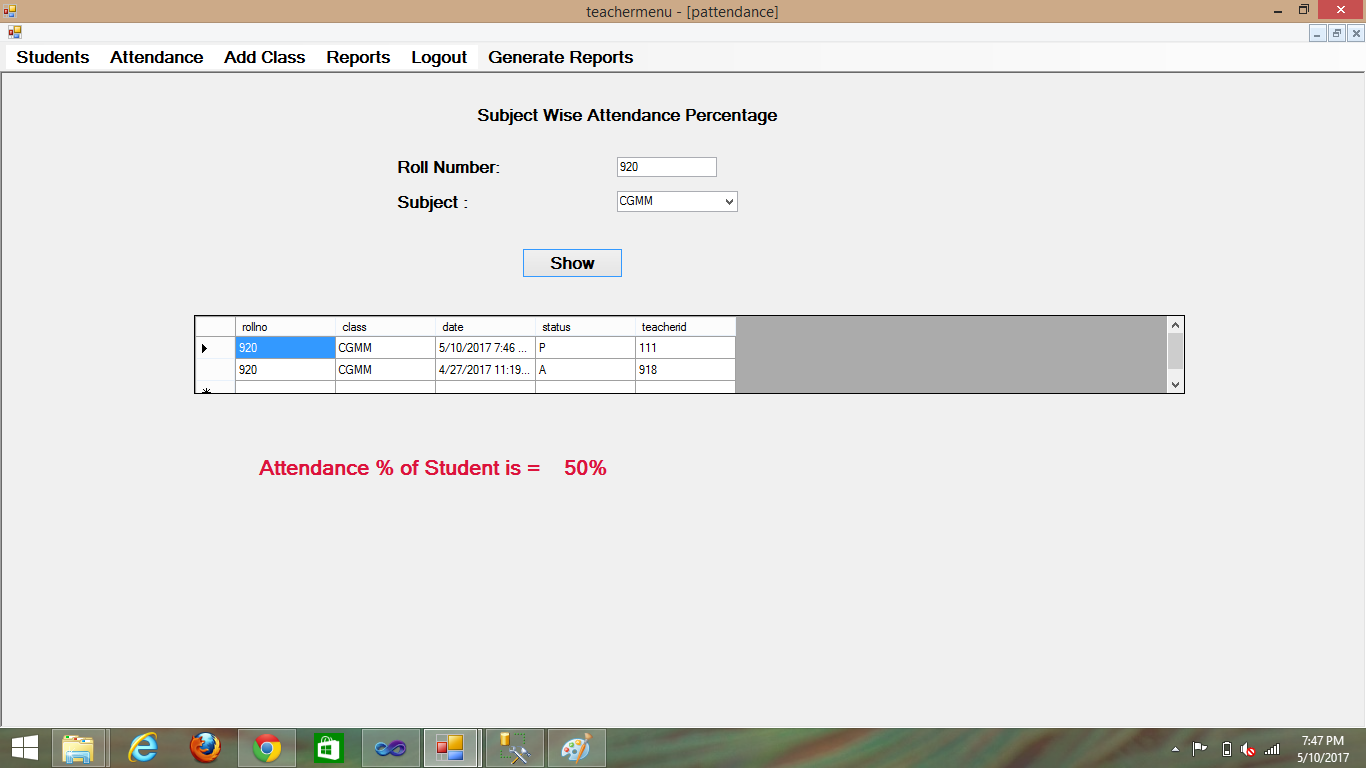
** Adding a class**

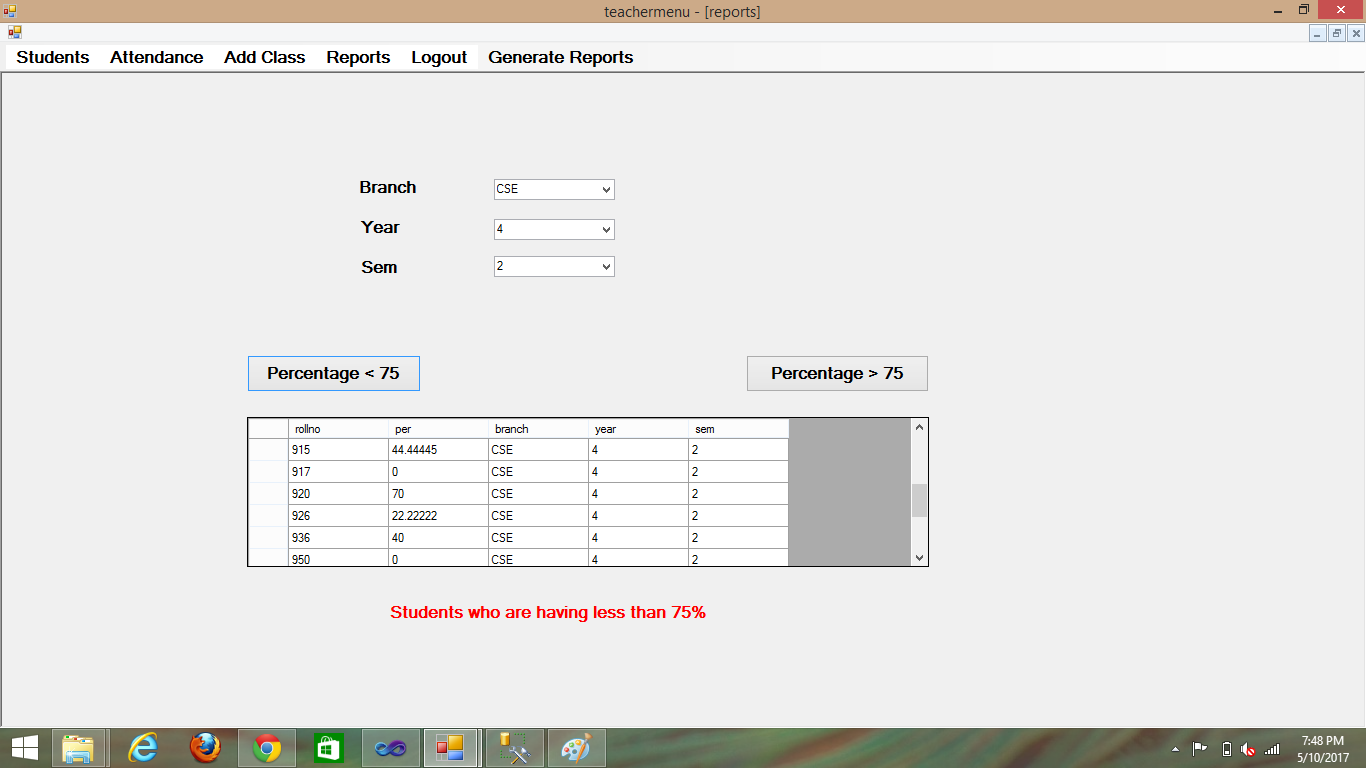
****

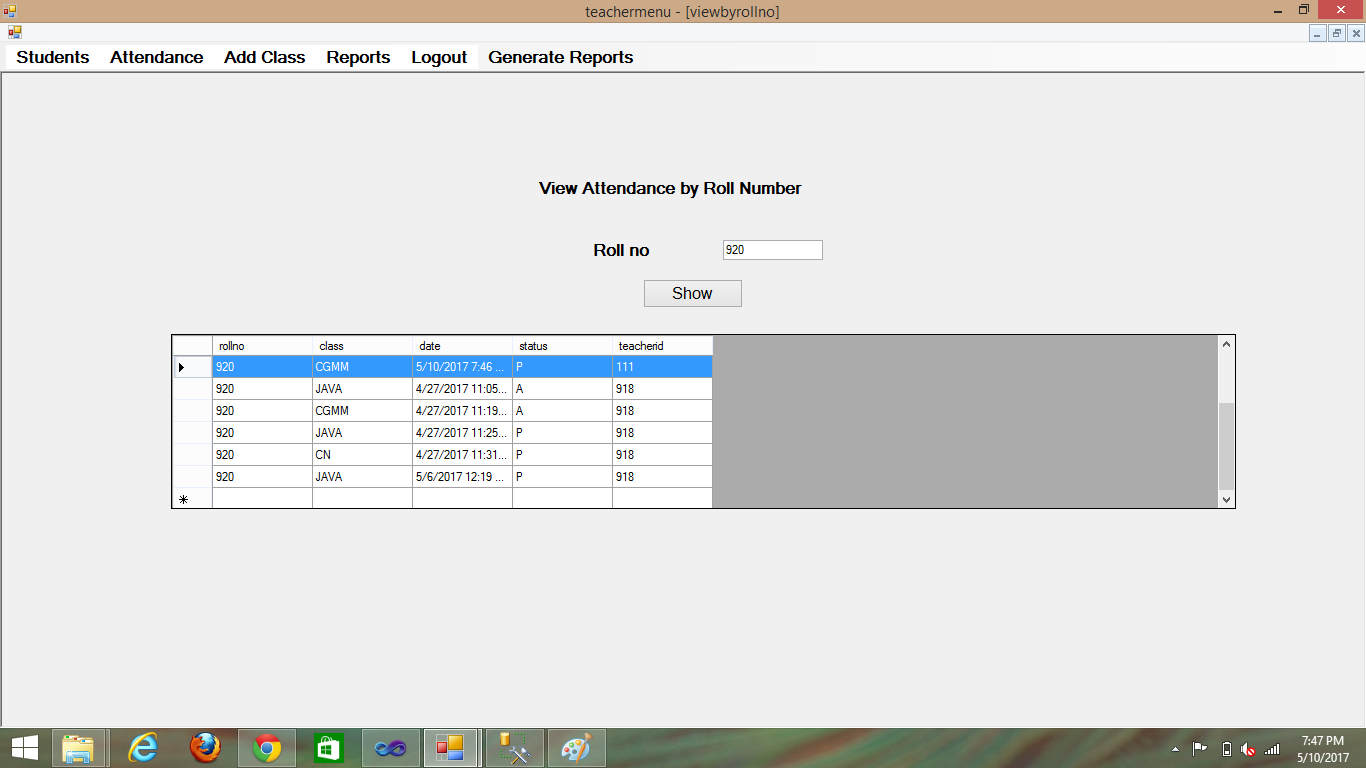
**Updating the attendance by using fingerprint scanner**

**GENERATION OF REPORTS**





****

****

****

**CHAPTER 7**

**CONCLUSION AND FUTURE SCOPE**

The Fingerprint-based Student Attendance Register set out to overcome the drawbacks of the current attendance system, which can be fooled by “buddy swiping” of absent students RFID card or signing the register sheet on behalf of absentee students within a university. This project has been implemented in four phases including create datasets, apply matching process to datasets, develop user interface, and finally test and evaluation in order to provide all deliverables.

An application was designed within C# under the Windows platform to capture the fingerprint image and then to verify a student who swipes his fingerprint against those stored values. The delivered application with a simple interface uses the Matching method and algorithms to compare fingerprints and meets the project requirements. Moreover, it has potential to be employed as student monitoring system in universities.

The recommendation from this project is that using Fingerprints as the mode of biometric for attendance is definitely valuable and provides high rate of accuracy along with low computational complexity.

**FUTURE SCOPE**

This project mainly comprised of development of attendance management system and fingerprint identification system. This project presented is window framework using which attendance management can be made automated. This can be extended to web application. A general implementable approach to attendance management can be proposed using LAN. Further, an idea for using portable devices (Wireless Fingerprint Scanner) along with wireless LAN or mobile 3G network was suggested. The limitations of using wireless LAN is that wireless devices are costly and implementation is hard because the wireless devices work in small area.