Project Report On

Monthly Employee Attendance Processing System

“A dissertation submitted in partial fulfillment of the requirements of Bachelor of Technology Degree in Computer Science and Engineering of the Maulana Abul Kalam Azad University of Technology (formerly known as West Bengal University of Technology) for the year 2015-2016”



Submitted by

Sukhpreet Singh Anand (14800112165)

Rangan Das (14800112125)

Rajan Kumar (14800112124)

Under the guidance of

Shri Abhijit Saha

Associate Professor

Department of Computer Science & Engineering

Future Institute of Engineering and Management

Department of Computer Science and Engineering

Future Institute of Engineering & Management

(Affiliated to Maulana Abul Kalam Azad University of Technology, West Bengal

Formerly known as West Bengal University of Technology)

Kolkata – 700150, WB

***Certificate of Approval***

This is to certify that this report of B.Tech. 8th Semester, entitled **“Monthly Employee Attendance Processing System (MEAPS)”** is a record of bona-fide work, carried out by **Sukhpreet Singh Anand, Rangan Das, Rajan Kumar** under my supervision and guidance.

In my opinion, the report in its present form is in partial fulfillment of all the requirements, as specified by the ***Future Institute of Engineering and Management*** and as per regulations of the ***Maulana Abul Kalam Azad University of Technology* (formerly known as *West Bengal University of Technology*).** In fact, it has attained the standard, necessary for submission. To the best of my knowledge, the results embodied in this report, are original in nature and worthy of incorporation in the present version of the report for B. Tech. program in Computer Science and Engineering in the year 2015-2016.

**Guide/Supervisor**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Mr. Abhijit Saha**

Department of Computer Science and Engineering

Future Institute of Engineering & Management

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Examiner(s) Head of the Department**

Computer Science and Engineering

Future Institute of Engineering & Management

**ACKNOWLEDGEMENT**

We would like to express our gratitude to several individuals without whom the project would not be the same. The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success.

First of all, we are sincerely grateful to our project guide Prof. Abhijit Saha for his guidance, inspiration and constructive suggestions towards helping us in the preparation of this project. Secondly, we also thank all our Faculty, Technical & Laboratory staffs and attendants of the Computer Science and Engineering Department who have assisted us towards the successful completion of the project. Last but not the least, we would like to thank our Head of the Department of Computer Science and Engineering, Prof. Tapas Roy for his humble support and also our Principal Sir, Mr. Aloke Ghosh for giving us this wonderful opportunity.

**Mr. Sukhpreet Singh Anand:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Roll: 14800112165

Registration: 121480110116 of 2012-2013

**Mr. Rangan Das:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   
Roll: 14800112125

Registration: 121480110076 of 2012-2013

**Mr. Rajan Kumar:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   
Roll: 14800112124

Registration: 121480110075 of 2012-2013

**PROJECT ABSTRACT**

Monthly Employee Attendance Processing System (MEAPS) is an application program that will automate the entire process of evaluating attendance of an employee working in any institution or organization and generating advanced reports based on their attendance, while providing options for manual overrides in necessary areas.

The aim of the project is to generate reports based on the performance of an employee of an institution or organization on the basis of his/her attendance as collected by the biometric scanner. The Monthly Employee Attendance Processing System (MEAPS) calculates the attendance of the employees of the organization based on its customized parameters and measures the performance of an employee in a properly organized way with the help of software applications.

The employee only has to provide his/her attendance using a biometric fingerprint scanner while entering and leaving the institution. The first step of the application software will be to send the employee's attendance data from the biometric fingerprint scanner, which is generated as an MS Excel sheet, to a relational SQL database. Now, from this SQL database, the application program will process the data gathered from the scanner. After processing of the data gathered from the scanner, the system owner can view the employee attendance reports in various formats. He/she can also attach their leave applications due to absence on any particular day.

The application software will have only one user: The system owner, that is, the head of the department.

The System owner has several roles in using the software:

1. His/her first role is to add the employees and add or update his/her details such as the Employee ID, Employee Name, Department, Joining Date. The system owner will also set the number of leaves such as Casual Leave (CL), Medical Leave (ML), Earn Leave (EL) to be assigned to a particular employee.
2. His/her second role is to assign all the holidays to be given an exception by the system including national holidays, religious holidays, other secular holidays and unofficial holidays and any lack of attendance on those days would be automatically ignored by the system.
3. His/her third role is to assign attendance exceptions on any particular day for the employees based on their leave applications or any other reasons related to the institution.
4. His/her fourth role will be to assign time exceptions on any particular day due to instances such as a half day, college events, etc. The new entry and exit time will be set for that day by the system owner.
5. His/her fifth role will be to add and view leave applications for the employees working in the organization for any particular day and also assigning the type of leave (CL, ML or EL).
6. His/her sixth role will be to view the leave status of all the employees. The leave status can be filtered to be viewed on a daily basis or a monthly basis. Using this option, the system owner can view all the days the employees have been present, absent or partially present in the organization.
7. His/her final role will be to generate reports based on his current leave status or based on leave taken between any two dates.

**CONTENTS**

**CHAPTER 1 INTRODUCTION**

* 1. **Objective 7**
  2. **Scope of the system 8**
  3. **Motivation 9**
  4. **Background 11**
  5. **Feasibility Study 14**
     1. **Technical Feasibility 14**
     2. **Operational Feasibility 16**
     3. **Economic Feasibility 17**

**CHAPTER 2 SOFTWARE REQUIREMENT SPECIFICATION 19**

**CHAPTER 3 SOFTWARE DEVELOPMENT PROCESS MODEL ADOPTED 23**

**CHAPTER 4 OVERVIEW 29**

**4.1 System Overview 29**

**4.1.1 Limitation of Existing System 32**

**4.2 Proposed System 34**

**4.2.1 Objectives of the proposed system 34**

**4.2.2 Users of the proposed system 35**

**CHAPTER 5 ASSUMPTION AND DEPENDENCIES 37**

**CHAPTER 6 TECHNOLOGIES 40**

**6.1 Tools used in Development 40**

**6.2 Development Environment 41**

**6.3 Software Interface 43**

**6.4 Hardware used 45**

**CHAPTER 7 DESIGN 47**

**7.1 Code Map 47**

**7.2 Data Flow Diagram 48**

**7.3 Entity Relationship Diagram 51**

**CHAPTER 8 DATA DICTIONARY 52**

**CHAPTER 9 TESTING 55**

**9.1 Unit testing 55**

**9.2 Integrity testing 58**

**CHAPTER 10 SNAPSHOTS 59**

**CHAPTER 11 CONCLUSIONS AND FUTURE SCOPE 65**

**CHAPTER 12 REFERENCES 67**

**1. INTRODUCTION**

Monthly Employee Attendance Processing System (MEAPS) is a real life software application that finds its use in a lot of places in this fast growing world. Nowadays, we find the use of biometric scanner in many small or large scale industries, where the organization installs a biometric scanner at the entrance and exit of the office and the employees or the members of the organizations enter and go out of the place with the help of this device. If the fingerprint of the employee matches with a previously recorded data, he/she is granted access to the secured part of the organization. So, biometric scanner, of any type, is a very essential security system in any part of an organization, be it office, home, bank, hospital, etc. and is required for authorization of an individual to a place of secrecy and security. With any biometric scanner, comes its data manipulation and report generation. On different models of such a device, come different ways of organizing data. Some biometrics collect and store data in Database format, some in Excel format, and their prices are according to their usage and needs. But those with data in Excel format, gets a hard time being manipulated by software applications because of their format of not being able to be collected in the form of direct databases. This is where MEAPS comes into action. It gets the data from the scanner in the Excel format, translates it into Database format, manipulate those data and hence generate reports based on them. So, due to the wide application of biometrics in real world in recent times, along with IT based software applications, MEAPS is a very useful tool for all grades of the industry which will help them to make secure access and grant them the power of secured authorization.

**1.1 Objective**

The objective of this project is to send the data from the biometric fingerprint scanner, which is generated as an MS Excel sheet, to a SQL database and process those data gathered from the scanner. Basically, the aim of the project is to generate reports based on the performance of an employee of an organization on the basis of his/her attendance as collected by the biometric scanner. The Monthly Employee Attendance Processing System (MEAPS) calculates the attendance of the employees of the organization based on its customized parameters and measures the performance of an employee in a properly organized way with the help of software applications.

The following are the objectives of the project:

1. **Cleaning up raw data**: Data from the spreadsheet is cleaned up and stored in a relational database.
2. **Storing data in a relational data store**: A database with pre-existing employee data will be used to store the cleaned up data.
3. **Developing of application program to process data**: Data will be batch processed using an application program that is run at regular intervals.
4. **Allow manual input in necessary areas:** In case of exceptions, the administrator can take over the automated processing or can define how the automated processing will occur by setting holiday exceptions, attendance exceptions and time exceptions.
5. **Provide advanced reporting and evaluation systems:** Storing data in an RDBMS will automatically provide advanced querying systems which can be used to generate reports by the user.

**1.2 Scope of the System**

The scope of MEAPS is also far beyond industries, they can be used in any type of organization for performing various actions like counting attendance, open doors, security purpose, etc. and they can be even used at home. So, MEAPS is a very useful tool in the present world due to its increased security and advance data processing features as well as its versatility in different environments.

This project deals with complex manipulations of real world data in a database. The application that will be developed will be a client side application which will connect to a local database. For efficiency, type-safety and ease of programming, an ORM, such as Entity Framework or LINQ to SQL classes may be used. The application frontend can be developed using Java or C# and the database such as MySQL or SQL Server is to be used.

The work that needs to be accomplished in order to deliver the project are as follows:

* Importing a spreadsheet into the database and processing the raw data.
* Developing a front end application to manipulate the processed data according to the requirements of the organization/institution.
* Generating reports using built in queries and further allowing custom queries.

**1.3 Motivation**

We came to know from our Head of the Department one day that the biometric scanner in our college needs a lot of manual labor during post processing. We were quite surprised by his words, because machines were supposed to reduce human effort and the biometric scanner attendance monitoring system was no exception. Out of curiosity, we asked our professor the reason behind his statement and then we came to know that the biometric scanner is not computerized in terms of generating output.

The biometric scanner is an authentication system is which fingerprint is used to detect the biometrics of an individual. In our college, Future Institute of Engineering and Management, we have a similar one at the entrance of our main building. When the office hours begin, employees, when entering college, gives a thumb impression on the scanner and the device records the time as the entry time of that person for that particular day. When leaving the college, the employee again gives a thumb impression on the scanner which denotes his exit time for that particular day. In this way, the biometric scanner records only the entry and exit time for a particular day against a particular employee, who has to be a previously registered user. The scanner generates a report at the end of a month. This report is a sheet of data generated in MS Excel format, filtered day wise and department wise in a month. The report is taken as a print out from the device and here begins the problem.

The aim of installing the biometric scanner is to regulate the attendance of the employees without much of a hassle. But the device fails to do so, most importantly in an organized and computerized manner. The data recorded cannot be processed by a computer, and only print out can be taken to show the various outputs. But our desired result is not the one we are getting in hand. We want to actually calculate the leaves taken by an employee and generate a report against his actions regarding his attendance. All these need a lot of manual labor, time and attention when done by human. If the entire system was computerized, we could have generated report which are error free and in no time.

In our present scenario, the Head of the Department is the one who gets this printed report straight from the scanner. He only gets the details of check in and check outs of the employees of his department, and do not have the access to the details of employees of other departments. He has to undergo lot of strenuous job to calculate the leave status of each employee at the end of each month based on the report generated by the biometric scanner.

The following are the reasons have acted as the motivation to take up this project.

* Even though an institution has multiple automated system, attendance related data is still processed manually. There does not exist a dedicated system to take care of the attendance, collect related data and assess performance.
* Managing attendance manually is a hectic and repetitive job and even though it sometimes requires manual intervention, a large part of the system can be automated. Most of it is essentially a waste of time, and can be made more efficient by optimizing the time spent on this task.
* Biometric and code scanners may occasionally provide erroneous data. The data can be analyzed and corrected with minimum manual intervention. The data which is present in form of a spreadsheet is transferred to a database and is manipulated by the application. Therefore, it learning to use spreadsheet application programs is not needed.
* The application will allow complicated querying allowing the departmental head or the administration to generate customized reports. This was necessary, but was not possible with the currently implemented system.
* The system will give more insight on employee performance.

This problem is the sole motivation behind undertaking this project. Since we have grown up hearing, ‘Charity begins at home’, we thought of doing something for our college, helping it to serve its members with better technologies and also giving a bit of relief to our beloved Head of the Departments professors. Also, with our skills we can serve the people with easier formatting and error free reports. Manual labor is always prone to errors and such things can lead to unpleasant relations among the controller and the employees. With MEAPS, one can surely avoid the interpersonal conflicts as well as the errors generated by miscalculation during manual work.

**1.4 Background**

With the ongoing problems of the current system, there are plenty of cases where the system administrators are being harassed. Most of which are basically due to errors generated while doing the calculations manually. Consistently for a few years’ problems came popping up due to this device. The main problem that aroused was the issue of reports with errors.

Let us see a number of possible cases with the error reports in which the employee is not fully responsible for the error:

1. The employee forgets to put his thumb in the biometric scanner when entering the college premises. But when leaving the premises, he manages to give his impression. So, in this case, the scanner will take his time of checking out as his entry time. And hence, his whole cycle of attendance monitoring is disrupted. But, since the biometric scanner is bound to accept the exit time within a particular time limit after the entry time, for example 8 office hours, the employee will be shown as absent or his record will be a faulty one, showing no exit time. Hence, the employee will be charged with a proper amount of penalty from his dedicated leaves.

It may so happen that the biometric scanner fails to get the thumb impression once, this same problem may occur, and the employee may not even know it! Thus this produces erroneous reports in the Excel sheet which in turns may hamper employee’s performance. Sometimes, in addition it may even happen that the person responsible for calculating the reports manually is dragged into interpersonal conflicts with the employees whose report is erroneous. He/she may even assume that the person in charge has done intentionally, which may or may not be the case.

1. The same situation may occur when the employee fails to put his thumb on the biometric scanner during leaving the college. In that case, his check out time is not recorded, thereby leading to results with errors in the office timings of that employee for that particular day. Also, the biometric scanner may fail to record this data due to its faulty activity. In both the cases, the employee will be having his day of work incomplete, as his check out time is invalid as after some designated hours the biometric will freshly record the data for the next day check in. So, his performance may be degraded and also like the previous case, he may become doubtful of the supervisor, which is not likely for any organization.
2. In relation to these cases, the supervisor who is in charge of calculating the leave and report for the employees for a particular department, who in this case is the Head of the Department, may mistakenly or knowingly change the data and generate reports which do not show the truth. This creates integrity issues as well as faulty reports. The main problem with the present system is the chance of manipulation of data that are generated by the biometric scanner. If a system is designed where the data directly go from the biometric scanner to the database system and then the data are manipulated to give output, we will have very less chances of errors.
3. Again, there is a situation where an employee has to go out early due to official work and he checks out early. In this case, the supervisor will not be aware of the fact that the employee has gone off for an official work, and thus his leave or absence must be granted. In this case, the employee has to manually submit the absence cause to the supervisor. This again requires manual labor and in turn it adds more confusion.
4. Sometimes, it may so happen that due to special case, the employees need to be present on the day which was listed as a holiday, for example, Saturday and Sunday. This occurs in terms of emergency, special occasions or due to special cases. In such case, the employees must be given special credit or bonus due to their presence. This was not fully implemented in the present biometric system.
5. There are cases when the check in and checkout time of a particular day are changed due to some special case. For example, if the day is declared as a rainy day, the check-out timings are before the normal one or the check-in timings are after the normal one. But the biometric will not be aware of the special case, and will generate report based on the thumb impressions. So, for this case, the supervisor will have to maintain a written record of the day when the timings are adjusted.
6. An employee may have joined a date late than the starting of the month, in that case, his joining date will be treated as the starting date of the month and hence other relations are calculated. It’s not his/her fault if his joining date does not match the starting date of the month, and in this case if the cases are not matched the reports are erroneous.
7. Lastly, there may be sudden announcement of holiday or the day may be a listed holiday. So, as no employees will be present on that day, the biometric scanner will not record any data and hence the employees will be marked absent. If the supervisor is not aware of the listed holiday, then he will treat this as absence of the employee and hence the reports will not be carrying the true data. So, computerized handling of the data is required to eliminate all these problems.

Now these problems are persistent in the present system and needs severe attention. But there can be other problems which are directly or indirectly responsible by the employee. Let us cite a few examples regarding this issue:

1. An employee is responsible for any leave caused by any medical conditions and he/she is required to submit the required medical certificate in favor of his/her absence on that day. If the employee fails to submit it on due time, he will be treated as absentee on that particular day. Whereas, if the employee produces his/her report on the stipulated amount of time and his medical situations satisfy the conditions for granting him a medical leave, the employee must be considered in leave. But all these applications are not available in the present system, due to which confusions and confrontations arise leading to error prone results and inter personal conflicts.
2. An employee may have taken a special leave for a personal reason and he/she has not given his thumb impression for a period of time. But this is not known to the biometric scanner so it marks that person absent on those days. But during the calculation, the supervisor checks the leave applied and again correct his calculations. It may so happen that during the calculations, the supervisor forgets to check the leave applied and hence marks him/her absent, thereby generating reports with faults. So, it is advisable to have a system with automated data entry and calculation to avoid errors and confusions generated by manual errors.

These were some of the backlogs that have motivated us to endeavor this project and serve the people with better technologies. All these form the background of the undertaken project. These problems and drawbacks have paved the way towards thinking and innovation of automated software system where data is manipulated in and organized manner with the help of computers.

**1.5 Feasibility Study**

The feasibility test assesses the technical, operational and economic merits of the proposed project. The feasibility study is intended to be a preliminary review of the facts to see if it is worthy of proceeding to the analysis phase. From the systems analyst perspective, the feasibility analysis is the primary tool for recommending whether to proceed to the next phase or to discontinue the project.

The feasibility study is a management oriented activity. The objective of our feasibility study is to find out if an information system project can be done and to suggest possible alternative solutions.

**1.5.1 Technical Feasibility**

A study of resource availability that may affect the ability to achieve an acceptable system. This evaluation determines whether the technology needed for the proposed system is available or not.

* Can the work for the project be done with current equipment existing software technology & available personal?
* Can the system be upgraded if developed?
* If new technology is needed, then what can be developed?

This is concerned with specifying equipment and software that will successfully satisfy the user requirement. The technical needs of the system may include:

**Front-end and back-end selection:**

An important issue for the development of a project is the selection of suitable front-end and back-end. When we decided to develop the project we went through an extensive study to determine the most suitable platform that suits the needs of the organization as well as helps in development of the project.

The aspects of our study included the following factors.

**Front-end selection:**

1) It has a graphical user interface that is simple and user friendly.

2) Scalability and extensibility.

3) Flexibility.

4) Robustness.

5) According to the organization requirement and the culture.

6) Must provide excellent reporting features with good printing support.

7) Platform independent.

8) Easy to debug and maintain.

9) Event driven programming facility.

10) Front end must support some popular back end like SQL server 2014.

According to the above stated features we selected ASP.NET using Visual Studio 2015 environment as the front-end for developing our project.

**Back-end Selection:**

1) Multiple user support.

2) Efficient data handling.

3) Provide inherent features for security.

4) Efficient data retrieval and maintenance.

5) Stored procedures.

6) Popularity.

7) Operating System compatible.

8) Easy to install.

9) Various drivers must be available.

10) Easy to implant with the Front-end.

According to above stated features we selected SQL Server 2014 as the backend.

The technical requirements of the project are very less than one can even imagine. With an installed biometric in an organization, only a master computer with the pre-installed application is required to have a fully functional MEAPS. It does not require a server computer, because it is not being used in the web. MEAPS is intentionally meant for a single master user, or specifically a very small group of master users who can have the access to the data generated by the biometric and operate on them. This is basically due to security purpose, as more is the user of the application there are more chances of losing data integrity and the data is more prone to manipulation. So, technical feasibility is very high for this project.

**1.5.2 Operational Feasibility**

It is mainly related to human organizations and political aspects. The points to be considered are:

* What changes will be brought with the system?
* What organization structures are disturbed?
* What new skills will be required? Do the existing staff members have these skills? If not, can they be trained in due course of time?

MEAPS is a software application which has high operational feasibility as it is very easy to maintain and operate on. As stated earlier, MEAPS is not targeted towards a very large number of users over internet, but instead it aims on technological professionals that deal with the management activities of the organization. So, MEAPS will not be used by the large mass of common people, but by the people who are strong technologically. But this is not the limitations of MEAPS, instead it provides a very easy-to-operate-on interface along with guided instructions so that it never become any confusion for the users. Even if the supervisor(s) is not a technical person, the instructions and easy layouts can guide him to proper functioning of data and generate desired outputs. This denotes that the operational feasibility of the project is very high.

The program has been thoroughly checked and rechecked for bugs, exceptions and errors. The program has been tested to solve the problems related to the operation. It takes into account the requirements identified in the requirements analysis phase of system development and fulfils its objective as per the methods specified in the scope of the definition.

To ensure success, desired operational outcomes are imparted during design and development. These include such design-dependent parameters such as reliability, maintainability, supportability, usability, reducibility, disposability, sustainability, affordability and others. These parameters considered at the early stages of design if desired operational behaviors are to be realized.

**1.5.3 Economic Feasibility**

Economic feasibility depends on the cost of operation of the entire project. With the basic structure of MEAPS, the cost of installation and proper functioning requires negligible amount of money. The main cost is the installation of biometric scanner, without which the whole project cannot operate. But the biometric scanner is not included as a component for this project. MEAPS comes into action when there is a pre-installed biometric scanner in an organization. So, in particular, MEAPS has basically no cost at all. Only a computer or a very few ones are required where the entire software application will be installed. The project does not require an internet connection to run on, so there is no cost associated with web server hosting. The project thus requires the most basic requirements of a complete software application and its economic feasibility is very high.

**2. SOFTWARE REQUIREMENT SPECIFICATION (SRS)**

An SRS is basically an organization's understanding (in writing) of a customer or potential client's system requirements and dependencies at a particular point in time (usually) prior to any actual design or development work. It's a two-way insurance policy that assures that both the client and the organization understand the other's requirements from that perspective at a given point in time.

The SRS document itself states in precise and explicit language those functions and capabilities a software system (i.e., a software application, an e-Commerce Web site, and so on) must provide, as well as states any required constraints by which the system must abide. The SRS also functions as a blueprint for completing a project with as little cost growth as possible. The SRS is often referred to as the "parent" document because all subsequent project management documents, such as design specifications, statements of work, software architecture specifications, testing and validation plans, and documentation plans, are related to it.

It's important to note that an SRS contains functional and nonfunctional requirements only; it doesn't offer design suggestions, possible solutions to technology or business issues, or any other information other than what the development team understands the customer's system requirements to be.

A well-designed, well-written SRS accomplishes four major goals:

* It provides feedback to the customer. An SRS is the customer's assurance that the development organization understands the issues or problems to be solved and the software behavior necessary to address those problems. Therefore, the SRS should be written in natural language, in an unambiguous manner that may also include charts, tables, data flow diagrams, decision tables, and so on.
* It decomposes the problem into component parts. The simple act of writing down software requirements in a well-designed format organizes information, places borders around the problem, solidifies ideas, and helps break down the problem into its component parts in an orderly fashion.
* It serves as an input to the design specification. As mentioned previously, the SRS serves as the parent document to subsequent documents, such as the software design specification and statement of work. Therefore, the SRS must contain sufficient detail in the functional system requirements so that a design solution can be devised.
* It serves as a product validation check. The SRS also serves as the parent document for testing and validation strategies that will be applied to the requirements for verification.

SRSs are typically developed during the first stages of "Requirements Development," which is the initial product development phase in which information is gathered about what requirements are needed--and not. This information-gathering stage can include onsite visits, questionnaires, surveys, interviews, and perhaps a return-on-investment (ROI) analysis or needs analysis of the customer or client's current business environment. The actual specification, then, is written after the requirements have been gathered and analyzed.

The SRS for the Monthly Employee Attendance Processing System is discussed as follows:

* **Purpose:**  This SRS describes the software functional and non-functional requirements for the Monthly Employee Attendance Processing System (MEAPS). This document is intended to be used by the members of the project team that will implement and verify the correct functioning of the system. Unless otherwise noted, all requirements specified here are high priority and committed for release of the project.

The Monthly Employee Attendance Processing System (MEAPS) will permit the system owner to automate the whole process of processing attendance of employees taken via a biometric fingerprint scanner and generating reports from them as well as their leave status. A detailed project description is available in the MEAPS Objective and Scope of the system (Chapter 1.1 and 1.2). The section in that chapter lists the features that are scheduled for full or partial implementation in this release.

* **User functional requirements:**

1. Interfaces: For any software application, we require a platform in which the software is based on. We have used ASP .NET as our platform with the help of which the interfaces are designed. For this project, a number of interfaces are required. As we are not dealing with any client-server system here, there is only one interface to handle. The interface contains 9 pages, each with back button, which leads to the previous page, and the next button, which leads to the next page. The interfaces contain scope for data entry for each employee and has special fields for accepting and rejecting leave application, medical certificates, etc. Also, there is an interface for editing the special time table or schedule in case of change in check in and check out times. Also, there is one entry field for listing the national or listed holidays. The master user just has to click a few buttons to generate the report card of the employees as well as calculate their performance regarding attendance. All these easy-to-use interface provides great simplicity to the project.
2. Functional Capabilities: The entire software application is very simple and provides excellent fully functional interfaces. The windows have an easy functional flow system, and going from one page to another is easy. So, the user will not have any difficulty in navigating through all the pages of the software all at once. The data is taken from the Excel format to the Database format internally, and there is no loss of data in this transformation process. The data fields on the interface maintains data integrity and hence has constraints based on the format required to be typed on the fields. For example, if one field requires date to be entered, the format in which it has to be stored in the database, is mentioned to the user for easier use, and hence there is no data redundancy and useless data in the database.
3. Performance Levels: The entire software application does not require any internet to work on. The performance and speed of the application do not require any access to internet and hence the performance increases. Also, its catchy user interface is easier to use and hence makes work simple. The performance of an application is generally based on the complexity of its design and the time required during its use. And both of these are very well balanced in case of the MEAPS.
4. Data Structures/Elements: A wide range of data structures are used in the project, because it requires a large number of data to be worked on. We have used Microsoft SQL database to store our data, which collects the data in Excel format, coverts it into SQL and stores them in an organized way, providing easier chances of access and manipulation, thereby reducing data redundancy and inconsistency.

* **Non Functional requirements:**

1. Safety: The foremost thing that a software application needs is safety. The user needs to feel safe while using the application and hence all his entry must be secured. If anyone requires a password to log into a system, the user must feel at ease while typing his password and he/she must be able to trust the system. In this case, the password may not be visible while typing, and need to be stored securely so that others cannot get access to his portal. Also, there must be features like opportunity of changing the password in case the user forgets it. This can be done with the help of his/her email and phone number as well as security questions. Now coming to the data entry and output section, all the reports generated are confidential and requires authorized access, and hence this is something fundamental requirement. Since MEAPS is not using internet, there is very less chance of virus getting into the system and also there are no scope of cookies and unwanted caches. This will make the system run faster.
2. Reliability: The entire software application must be reliable and trustworthy. It must not give a feel to the user that the system is not secure and the data entered are providing security issues. Most important aspect of reliability in a software is that it’s proper functioning. A user will never want the entire system to be crashed. Also, the system getting slow while working with other background tasks is also not desired. The software is built on a platform that consumes as less space in the user’s PC memory as possible. And even if the system crashes due to power failure and other issues, the application must be restored back onto the previous state of work from where it was crashed. This will make the application much more efficient and popular for the users. Also, the user will not want another application to be built if his/her operating system gets upgraded to a newer version. So, one application one platform is what the fundamental goal should be.
3. Security/Privacy: Security comes side by side with privacy, and hence a user will only want his data to be secured. A master user of a particular department may not be able to see the data of another department. Coming to typing safe passwords, to extra security over cross platform intruders, the project has to be robust and secured.
4. Quality: Quality is what the users want primarily. If a client has no technical knowledge, he/she will firstly stress on the basic design of the application and then the functionality of it. The basic design and GUI must be very simple yet classy as well as unique. Nobody like the same design repeated for popular applications. There has to be something new and exciting. So, MEAPS will have interactive interface and detailed instructions of properly using the application. The functionality issues have already been discussed previously. The main focus being security, safety and functionality of the software that needs to be taken delicate care of.
5. Constraints and Limitations: Coming to any software system, there has to be some limitations or the other. The constraints are what make the work challenging. The entire project is based on customized constraints that are needed for a particular organization. Since we are dealing with the attendance system of an organization, the entry and exit time of the employees may vary from one place to another and so does the list of holidays, special leaves, and the system of calculating the casual, earn and medical leaves. Here we are taking the example of a college and so accordingly we have fixed our fields in the database. And also these are subjected to change anytime. Limitations also include the system being not a cloud application and so it requires space in the hard disk. As there are many such, the constraints provide more scope of improvement for the project.

The SRS of the project shows what the user or the client want from the system and their description is discussed above in details.

**3. SOFTWARE DEVELOPMENT PROCESS MODEL ADOPTED**

In software engineering, a software development methodology (also known as a software development life cycle, software development process) is a splitting of software development work into distinct phases or stages containing activities with the intent of better planning and management. It is often considered a subset of the systems development life cycle. The methodology may include the pre-definition of specific deliverables that are created and completed by a project team to develop or maintain an application.

Common methodologies include waterfall, prototyping, iterative and incremental development, spiral development, rapid application development, extreme programming and various types of agile methodology. Some people consider a life-cycle "model" a more general term for a category of methodologies and a software development "process" a more specific term to refer to a specific process chosen by a specific organization. For example, there are many specific software development processes that fit the spiral life-cycle model.

The approaches that have been used to develop this project are discusses as follows:

1. **The waterfall model**:

The development of this project is seen as a flowing downwards steadily. In this model, one phase has to be finalized before the process can progress to the next phase. So, we have chosen the Waterfall model for our process development.

The waterfall model is a sequential development approach, in which development is seen as flowing steadily downwards (like a waterfall) through several phases, typically:

1. Requirement analysis resulting in SRS.
2. Software design
3. Implementation
4. Testing
5. Integration
6. Deployment or Installation
7. Maintenance

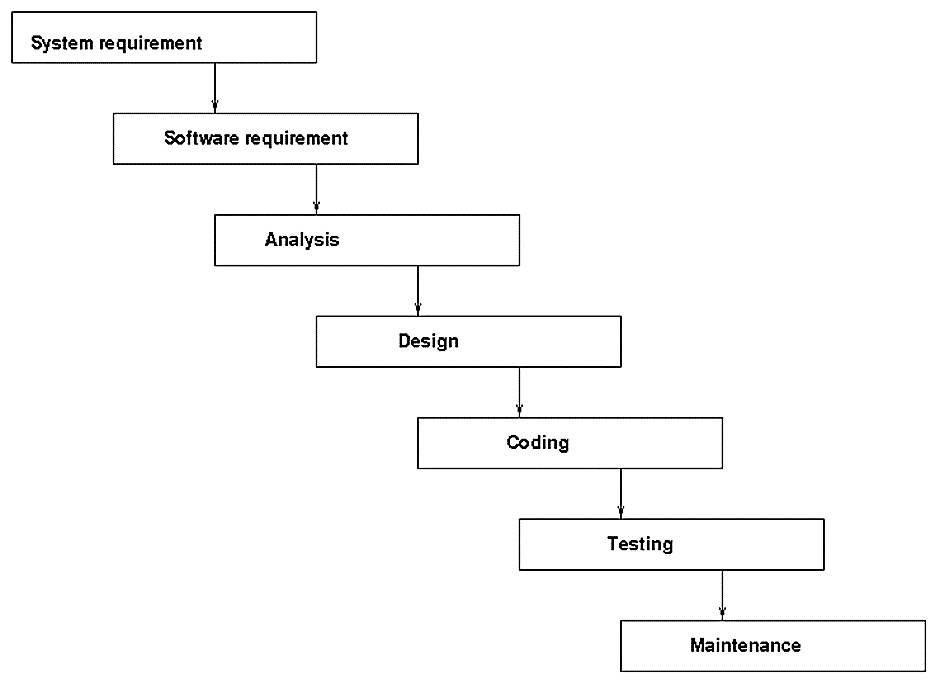


Fig 3.1: Diagram of Classical Waterfall Model.

Project is divided into sequential phases, with some overlap and splash back acceptable between phases.

Emphasis is on planning, time schedules, target dates, budgets and implementation of an entire system at one time.

Tight control is maintained over the life of the project via extensive written documentation, formal reviews, and approval/signoff by the user and information technology management occurring at the end of most phases before beginning the next phase.

The waterfall model used in the project follows the basic foundations: -

**Planning:** This is the initial evaluation of the software to be developed or in most cases; it is an attempt to upgrade an existing system.

**Requirement Analysis and Specification:** In this stage, the problem the new system is identified; its operational capabilities are defined and the resources needed for the system and maintenance goals are set.

**Component Implementation and Debugging:** codifies the preceding specifications into Operational source code implementations and validates their basic operation.

**Software Integration and Testing:** This is the stage where integrity of the software system configuration is tested and verified for consistency and completeness. All developed modules, interfaces and resources are verified against their specification.

**Documentation Revision and System Delivery:** At this stage, the technical know-how of the developed soft-ware system is documented as user’s guides.

**Deployment and Installation:** Installation of the software system into local computing environment, operating systems configuration and diagnostic test cases to ensure the workability of the system is done at this stage.

**Software Maintenance:** Maintaining the system in the environment in which it is created for is also an important aspect that must be taken into consideration.

**Iterative and incremental Model:** This model is developed in response to the shortcoming of waterfall model. It does not start with full specification requirement of a project, rather specify and implement some part of the software one at a time in other to review it at every step along the line to identify any further requirement. These processes have to be done again and again to produce a new requirement. As waterfall model is not practically possible, we used the iterative version of it to implement the project.

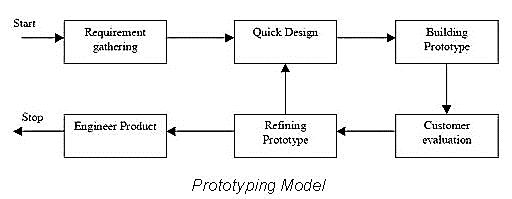


Fig 3.2: Diagram of Iterative Waterfall Model.

1. **Prototyping:**

Software prototyping is the development approach of activities during software development, the creation of prototypes, i.e., incomplete versions of the software program being developed.

The basic principles are:

* Not a standalone, complete development methodology, but rather an approach to handle selected parts of a larger, more traditional development methodology (i.e. incremental, spiral, or rapid application development (RAD)).
* Attempts to reduce inherent project risk by breaking a project into smaller segments and providing more ease-of-change during the development process.
* User is involved throughout the development process, which increases the likelihood of user acceptance of the final implementation.
* Small-scale mock-ups of the system are developed following an iterative modification process until the prototype evolves to meet the users’ requirements.
* While most prototypes are developed with the expectation that they will be discarded, it is possible in some cases to evolve from prototype to working system.
* A basic understanding of the fundamental business problem is necessary to avoid solving the wrong problems.

The following methodology will be implemented while working with the project based on the above mentioned principles:

* Since this application will work on data that is already present, creation of sample data will be the first work that is to be done. For this project, data will be generated as a simple Microsoft Excel spreadsheet. Errors will be intentionally incorporated in the data to implement a real life scenario.
* The next job would be to import the data to the database. Once a schema is made, a frontend application will be developed with basic report generation capabilities. Some more functionalities are to be added once the application succeeds in report generation.
* A basic prototype will be built on which the testing will be performed. The prototype will be tested and improved in cycles over time until a functional module is developed.
* Work will be divided for planning, UI and UX design, coding, testing and debugging. Multiple modules can be simultaneously developed together and then deployed as a part of a single project.
* With the core module completed, further optional modules can be implemented to refine the experience of the application.
* As the project progresses, necessary documentation of the project is to be done in the required format.

This hence describes the methodology using which the project is developed.

**4. OVERVIEW**

**4.1 System Overview**

Monthly Employee Attendance Processing System is a robust application to analyze, manage and manipulate employee attendance data and generate reports from them. All organizations and institutions have some systems that allows the administration to look up the attendance and leaves of the employees and other staffs. However, this comes with limitations that makes the processing inefficient and cumbersome.

In most cases, a biometric scanner, or an ID card scanner is employed to collect the attendance data. The report is generated in either in a spreadsheet format or a plain text format. This data is without a schema and is non-relational is nature and hence further calculations and reporting based on that data is to be done manually. Also, at times, there may be erroneous scanning where an employee's detail is scanned multiple times, or may not be scanned at all. There may also be numerous exceptions that a simple biometric scanner cannot take care of. In such cases manual processing of such data is less complicated. However, manual processing is boring and repetitive that can be automated, if the data from the biometric scanner is taken into a relational data store.

The monthly employee attendance processing system automates this entire process. At first, the raw data from the scanner is put in a table in a database where the data is corrected if possible and records with exceptions are marked for manual processing. In the next stage, all the leaves that are taken by the employees, faculties or staffs are manually specified. If any such leave is not manually specified, it will be subtracted from the remaining leaves of an employee. Leaves can be causal, medical or earn leaves, and in that order or preference, leaves are subtracted. If one type of leave is not available, the next one is checked. If they all become zero at some point, the balance goes into negative values if further leaves are taken. Balance leaves are moved over to the next year, however, casual leaves are not moved over to the next year. Also, a casual leave can be of a quarter day and other types of leaves are at least half day long. Also an employee must provide an application if a leave of over half a day is taken. These are the areas that require manual attention, even though the rest is automated.

Furthermore, exceptions like when a faculty or employee is on duty outside the premises of the campus, or when working hours have ended earlier is managed by the application based on manually entered data. The rest is processed manually. This data processing also gives further insight into employee performance, besides automating the attendance system.

 Furthermore, fine tuning the application can be done that will add multiple other features related to advanced querying of data and report generation. Custom queries can be made to the database, besides having built in queries that allows the administrator or the departmental head to quickly and efficiently generate reports on employee performance.

The Leave Calculation table is a table that is looked up to decide what part of a day is taken as a leave based on the entry and exit timings.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Entry →**  **Exit ↓** | **Before  9:50 AM** | **Between 9:50 AM and 10:00 AM** | **Between 10:00 AM and 10:30 AM** | **Between 10:30 AM and 1:30 PM** | **Beyond 1:30 PM** |
| **Before 1:30 PM** | 1 | 1 | 1 | 1 | - |
| **Between 1:30 PM and 3:30 PM** | ½ | ½ | ¼ + ½ | 1 | 1 |
| **Between 1:30 PM to 5:25 PM** | ¼ | ¼ | ¼ + ¼ | 1 | 1 |

Fig. 4.1.1 Leave calculation table

The procedure of calculation of leaves is given below.

|  |  |
| --- | --- |
| **Leave type** | **Durations** |
| Casual Leave (CL) | ¼ day, ½ day, 1 day |
| Medical Leave (ML) | ½ day, 1 day |
| Earn Leave (EL) | 1 day |

Fig. 4.1.2 Leave calculation chart

The flow of the project is explained below with the help of a flowchart :

**1. Initialization :**



Fig. 4.1.3 Flow of the project



Fig. 4.1.3 Flow of the project

**4.1.1 Limitation of the Existing System**

The present system and its drawbacks have motivated us to undertake this project. We cannot always blame the present system for its flaws, but those are the things that encourage people to build better technologies. There are a number of efficient working systems in different organizations which calculate attendance and generate reports based on the biometric scanner present. But the present case we are dealing with have no data connection between any software systems along with the biometric scanner. So the data collected by the biometric is stagnant and hence has no flow. The data generated by the present scanner is dumped into its memory every day in MS Excel format, and hence the data gets no chance of being manipulated to generate reports. Also, another drawback of the current system is that it functions monthly and not day wise. The sole user gets the detailed report at the end of every month. But if this has been generated day wise, it would have led to less errors in the reports and would have provided less data to work with at a time, thus reducing work pressure. Also, a machine is always prone to error, and hence the data collected by the biometric may or may not provide us with the correct data. This may happen when the data recorded is faulty, or post processing or matching of the fingerprint, retina etc. by the biometric generated faulty results. This leads to erroneous data and hence processing them needs more attention and man power which may lead to recursive errors. Also the exceptions as discussed in the Background section also needs attention while recording and hence the current system needs upgradation. The main problem behind the current system is the requirement of manual labor for calculations and hence its chances to incorporate errors in the results knowingly or unknowingly.

The main limitations of the existing system can be summarized as follows:

1. **Data Duplication**: The biometric system may create multiple duplicate entries for a single input.
2. **Errors in Defining Shifts**: An input made too early for a day may make an entry for the previous day.
3. **Manual Batch Processing of Data**: Processing large number of records in time consuming, tedious and exhausting.
4. **Limited/Broken Reports**: Manual processing of large amounts of data may introduce human error.
5. **Lack of Advanced Reporting and Evaluation**: Data, being stored in a non-relational way, is difficult to process to generate advanced reports.

**4.2 Proposed System**

**4.2.1 Objectives of the Proposed System**

The objective of the proposed upgraded system aims at eradicating the limitations of the current system. The entire software application tends to relief the users from the tenuous manual work during complex calculations with the help of software. The proposed system tends to eradicate erroneous data and hence produce better and flawless results. Keeping the SRS in mind, we need to focus on the efficiency of the system with the use of newer technologies. The modules have to be developed with keeping in mind all the restrictions and customization of the requirements of the projects. The main frame must be consisting of simple yet beautiful windows which are easy to glide through with proper instructions for specified fields. The project aims at simplifying the current system. This can be achieved by transmission of data from Excel format to the database format. The latter is the key to the success of the application. Proper database management techniques have to apply for properly organizing the data and manipulating them. The graphical user interface will be having easier access to these well-organized data and with the help of abstracted queries the user can use this data for other purposes. The goal of the proposed system is to have efficient purposes leading to at the end of the data organization. There is a need to generate basic reports from the master table. This report will contain employee attendance and leaves without taking into account public holidays or exceptions. Then it is required to refine this generated report by taking into account public holidays and exceptional cases. This objective will be critical and time consuming, and it would bring the project closer to a real life application. The next objective would be to carry forward one month’s information to another, one year’s information to the next. In this way, entire year will be covered month wise, and hence it will be easier to provide additional complex reports, surveys, certifications, etc.

The project is divided into the following modules which working together will solve the problem

1. **Data Importing and Correcting:** The module will take in data from the spreadsheet and will put the raw data in a temporary database table. Each record will be scanned for missing or extraneous data and appropriate corrections will be made. Corrections that requires user attention will be flagged. A fresh table will be generated from a clean dataset on which the processing will be done.
2. **Exception Handling:** Exceptional cases where the standard data from the biometric scanner will not suffice will have to be dealt with the head/administrator. This module will take in input and will process the exceptional records accordingly. Those records for which manual input is not provided will be forwarded to the Leave Calculation module.
3. **Leave Calculation:** This module looks after the number of leaves an employee has, the number that is to be subtracted and the number that is to be passed on to the forthcoming year.
4. **Report Generation:** This section will deal with the monthly report generation based on data analysis and calculation done by the previous modules.
5. **Custom Querying:** Furthermore, custom reports according to the user’s needs can be acquired by querying the database manually.

**4.2.2 Users of the Proposed System**

The application software is targeted for a specific master supervisor or audience or clients rather than aiming for a mass. But this system benefits the entire officials of an organization without getting publicized to be used by all the members of the company. This is like a secret machine that actually abstracts the technical knowhow of the process for the mass but serves each and every one of them in the most efficient manner in this regard. The different kinds of users of the proposed system are discussed below:

* 1. The employees who come to the organization as their duty and need their attendance monitored by a proper attendance governing system. In older times, people used to make written records of the attendance which are prone to errors and intentional data modifications. But since the world is moving towards everything being controlled by computers, the biometric scanner along with MEAPS serve the purpose of efficient and error free attendance regulation system. The employees need to give a thumb impression or scan his/her retina on the biometric scanner and the device hence generates the authorization procedure for that person. If the person giving the thumb impression can be allowed to be granted that particular access, he is signaled positive, and the biometric collects the required data for that person. For our system, the time of giving the thumb impression for that particular person for that particular day is recorded as the check in or check out time of that employee for that particular day. In this way, the employees use the system indirectly, and their data find their way into the software system where leaves and reports are calculated according to their performance based on attendance.

2. The master user is the one who has the right to have access to read, write, update and delete the data generated by the biometric scanner. His role is to check whether the records of the biometric is erroneous or not, and if yes then take necessary actions based on written records in back up and thereby modify the errors in that record. Next, he needs to use the software and hence calculate the leave status of each employee under his consideration, for example his department, and generate various reports based on that. The reports may include, the employee of the month/year, percentage of employees working for more hours, find out employees who are not dedicated and make other surveys too. Such reports are very important for an organization, to increase their quality in terms of employee management. The master user or supervisor needs to manipulate the data in the database with the help of GUI based on leave applications, listed holidays, exception cases, etc. which have already been discussed in the previous sections. And since this application does not need any web server to run, the user will have this installed in his/her personal computer along with the connection made with the biometric scanner.

**5. ASSUMPTIONS AND DEPENDENCIES**

In this project, very few assumptions need to be made since MEAPS is a real life application. We have shown that the requirement of data conversion from Excel format to SQL format is performed using a dummy table and MS Excel sheet. But it has however no linking with the actual biometric scanner on which the project is based. The physical connection between the biometric scanner installed in the organization and the database table on the user’s computer has not yet been possible. We have used dummy sheets of data by exactly matching the one that is being generated by the biometric scanner. But in reality, no such physical connection has been made between the biometric and the computer. This is one assumption we have made, to continue with the progress of the proposed system.

There are a few dependencies present on this project, the main one being the timings of entry and exit based on which different types of leaves are calculated against a particular grade of employee. Such timings and rules to determine the leave are unique for each organization and hence is customizable. So, we need to depend on the data provided by the organization on this regard.

The dependencies of the system on leaves assignment to the employees is given as follows:

**1.** Leave Types:

|  |  |
| --- | --- |
| **Leave type** | **Durations** |
| Casual Leave (CL) | ¼ day, ½ day, 1 day |
| Medical Leave (ML) | ½ day, 1 day |
| Earn Leave (EL) | 1 day |

Fig. 5.1 Leave calculation chart

**2.** The leave type is decided based on the duration and the application provided by the employee beforehand. If no application is provided, the default is CL.

**3.** Without applications, leaves get exhausted in the order CL, ML and then EL.

**4.** When all leaves are exhausted, leaves are recorded as Leave Without Pay (LWP).

**5.** If no attendance is recorded for the day, the employee is either absent, or is on external duty. If it is the latter, the employee is marked as full present.

**6.** On certain days, the default entry time and the exit time may be changed.

Furthermore, a leave calculation table based on the entry and exit times of the Employee is also assumed as per the schedule of the college this project has been initially made to be based upon. The leave calculation table is given as follows:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Entry →**  **Exit ↓** | **Before  9:50 AM** | **Between 9:50 AM and 10:00 AM** | **Between 10:00 AM and 10:30 AM** | **Between 10:30 AM and 1:30 PM** | **Beyond 1:30 PM** |
| **Before 1:30 PM** | 1 | 1 | 1 | 1 | - |
| **Between 1:30 PM and 3:30 PM** | ½ | ½ | ¼ + ½ | 1 | 1 |
| **Between 1:30 PM to 5:25 PM** | ¼ | ¼ | ¼ + ¼ | 1 | 1 |

Fig. 4.1.1 Leave calculation table

Also we are using ASP .NET as our framework and platform of work, so it is not platform independent and is targeted towards Windows users. Users of other systems like Linux, IOS, etc. cannot use MEAPS. This one being a basic limitation of the project. So assumption is made that the application will run on Windows and other users will have to wait for the application on their platform for a time being. One assumption is made that since we are not using cloud storage, we have ample amount of free memory in our system which runs the application quite fast as expected.

Although there are few limitations, assumptions and dependencies, the project is quite efficient in solving the purpose which it aims at.

**6. TECHNOLOGIES**

The entire project was done by hand and that required the use of several tools. The main problem, right off the bat was the issue of coherency, the program needed to have WORA or write once and run anywhere option for us, as most of us were in different development parts of the project.

Thus, the chapter carefully explains what we faced and how we face it technology wise.

**6.1 Tools used in Development**

There are number of tools available online and offline for developing software. Some of them which are used in the process of developing MEAPS are discussed below:

1. Design patterns i.e. UML tools and techniques: Microsoft Visio
2. Project planning and management: Microsoft Project
3. Effort estimation: Top-down, PERT
4. Programming languages: C#, XAML, SQL
5. Technologies, libraries and frameworks:
   * 1. **.NET Framework 4.5.1**
     2. **Entity Framework 5**
     3. **Windows Presentation Foundation**

### IDE: Microsoft Visual Studio 2015

### Databases/Data storages: Microsoft SQL Server, Microsoft Excel

### Application and web servers: Microsoft Excel 2016

### Web and graphic design: Adobe Illustrator

### Testing: Installation testing, Functional testing, Load testing, Performance profiling, Data integrity testing, Sanity testing, Regression testing, Automated testing

### Testing tools and frameworks: Microsoft Visual Studio Diagnostic Tools

### Unit testing: Microsoft Visual Studio

### Some of the delicate things we require for the project are as follows:

* Internet Connectivity and online storage is crucial for collaborated development on a project.
* A software development platform like .NET Framework will be used. The required tools are provided as a part of the software development kit.
* An Object Relation Mapper that is available for the platform used may be used to ease the database connectivity and data manipulation.
* Since this project is aimed to work as a real-life application, exhaustive testing is to be done and extensive time has to be dedication to testing.

Real life data from a biometric scanner is to be acquired for testing purposes.

**6.2 Development Environment**

Supported operating systems

* + Windows 10 (x86 and x64)
  + Windows 8.1 (x86 and x64)
  + Windows 8 (x86 and x64)
  + Windows 7 SP1 (x86 and x64)
  + Windows Server 2012 R2 (x64)
  + Windows Server 2012 (x64)
  + Windows Server 2008 R2 SP1 (x64)

Framework: .NET Framework 4.5

Database: SQL Server 2014 Express

Browser: None

IDE: Visual Studio 2015

This version of Visual Studio works best with Microsoft Edge or Internet Explorer 11. If Internet Explorer 11 is not installed on your computer, some features might not work as expected.

ORM: Entity Framework 5.0

NET Framework is a software framework developed by Microsoft that runs primarily on Microsoft Windows. It includes a large class library known as Framework Class Library (FCL) and provides language interoperability (each language can use code written in other languages) across several programming languages. Programs written for .NET Framework execute in a software environment (as contrasted to hardware environment), known as Common Language Runtime (CLR), an application virtual machine that provides services such as security, memory management, and exception handling. FCL and CLR together constitute .NET Framework.

FCL provides user interface, data access, database connectivity, cryptography, web application development, numeric algorithms, and network communications. Programmers produce software by combining their own source code with .NET Framework and other libraries. .NET Framework is intended to be used by most new applications created for the Windows platform. Microsoft also produces an integrated development environment largely for .NET software called Visual Studio.

.NET Framework started out as a proprietary framework, although the company worked to standardize the software stack almost immediately, even before its first release. Despite the standardization efforts, developers—particularly those in the free and open-source software communities—expressed their uneasiness with the selected terms and the prospects of any free and open-source implementation, especially with regard to software patents. Since then, Microsoft has changed .NET development to more closely follow a contemporary model of a community-developed software project, including issuing an update to its patent that promises to address the concerns.

.NET Framework family also includes two versions for mobile or embedded device use. A reduced version of the framework, .NET Compact Framework, is available on Windows CE platforms, including Windows Mobile devices such as smartphones. Additionally, .NET Micro Framework is targeted at severely resource-constrained devices.

The Entity Framework is a set of technologies in ADO.NET that support the development of data-oriented software applications. Architects and developers of data-oriented applications have typically struggled with the need to achieve two very different objectives. They must model the entities, relationships, and logic of the business problems they are solving, and they must also work with the data engines used to store and retrieve the data. The data may span multiple storage systems, each with its own protocols; even applications that work with a single storage system must balance the requirements of the storage system against the requirements of writing efficient and maintainable application code.

The Entity Framework enables developers to work with data in the form of domain-specific objects and properties, such as customers and customer addresses, without having to concern themselves with the underlying database tables and columns where this data is stored. With the Entity Framework, developers can work at a higher level of abstraction when they deal with data, and can create and maintain data-oriented applications with less code than in traditional applications. Because the Entity Framework is a component of the .NET Framework, Entity Framework applications can run on any computer on which the .NET Framework (starting with version 3.5 SP1) is installed.

Microsoft SQL Server is a relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications which may run either on the same computer or on another computer across a network (including the Internet).

Microsoft markets at least a dozen different editions of Microsoft SQL Server, aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing applications with many concurrent users.

SQL Server uses as its primary query languages T-SQL and ANSI SQL.

**6.3 Software Interface**

As mentioned earlier, we have chosen Microsoft Visual Studio 2015 enterprise edition for the development and let us go in depth about the software used.

Visual Studio includes a host of visual designers to aid in the development of applications such as Windows Form Designers and Windows Presentation Foundation, etc

For this project, we have used Windows Presentation Foundation (WPF) for the system interface.

**Windows Presentation foundation:** WPF, which stands for Windows Presentation Foundation, is Microsoft's latest approach to a GUI framework, used with the .NET framework.

The WPF designer, codenamed Cider, was introduced with Visual Studio 2008. Like the Windows Forms designer, it supports the drag and drop metaphor. It is used to author user interfaces targeting Windows Presentation Foundation. It supports all WPF functionality including data binding and automatic layout management. It generates XAML code for the UI. The generated XAML file is compatible with Microsoft Expression Design, the designer-oriented product. The XAML code is linked with code using a code-behind model.

But what IS a GUI framework? GUI stands for Graphical User Interface, and you're probably looking at one right now. Windows has a GUI for working with your computer, and the browser that you're likely reading this document in has a GUI that allows you to surf the web.

A GUI framework allows you to create an application with a wide range of GUI elements, like labels, textboxes and other well-known elements. Without a GUI framework you would have to draw these elements manually and handle all of the user interaction scenarios like text and mouse input. This is a lot of work, so instead, most developers will use a GUI framework which will do all the basic work and allow the developers to focus on making great applications.

There are a lot of GUI frameworks out there, but for .NET developers, the most interesting ones are currently WinForms and WPF. WPF is the newest, but Microsoft is still maintaining and supporting WinForms. As you will see in the next chapter, there are quite a few differences between the two frameworks, but their purpose is the same: To make it easy to create applications with a great GUI.

**6.4 Hardware Used**

1. **Computer:**

The following computers were used in the making of the project:

1. Dell Inspiron 5547

CPU: Intel Core i5-4210U @ 1.7GHz

Memory: 8GB

Operating System: Microsoft Windows 10 Home

DirectX Version: 12

2. HP Pavilion 15-AB125AX

CPU: AMD A10-8700P @ 2.2GHz

Memory: 16GB

Operating System: Microsoft Windows 10 Enterprise

DirectX Version: 12

**2. Biometric Scanner:**

**Features:**

1. ZK optical sensor
2. X990 Color TFT screen with GUI interface for ease of use
3. Stores 3000 templates and 100000 transactions
4. Reads fingerprint and/or PINS
5. Optional integrated proximity or smart card reader
6. Built-in USB port allows for manual data transfer when network isn't available
7. Built-in serial and Ethernet ports
8. Multi-language support
9. SDK available for OEM customers and software developers
10. Optional: Wi-Fi, GPRS

**Technical Specification:**

1. User Capacity: 3000
2. Transaction Storage: 100000
3. Matching: 1: N or 1: 1
4. Communications: TCP/IP, USB
5. Identification time: =2 seconds
6. FAR: =1%
7. FRR: =0.001%
8. Operating Temperature: 0°C - 45°C
9. Operating Humidity: 20%-80%
10. Sensor: ZK Optical Sensor 500 DPI
11. Power Supply: 5V DC 2A
12. Card reader: Optional Reader (Proximity, Mifare)
13. Language: English
14. Optional: Built-in battery backup, approx. 3-4 hours’ continuous operation



Fig 6.4.3 Biometric Fingerprint Scanner

**7. DESIGN**

The following are the design models of MEAPS:

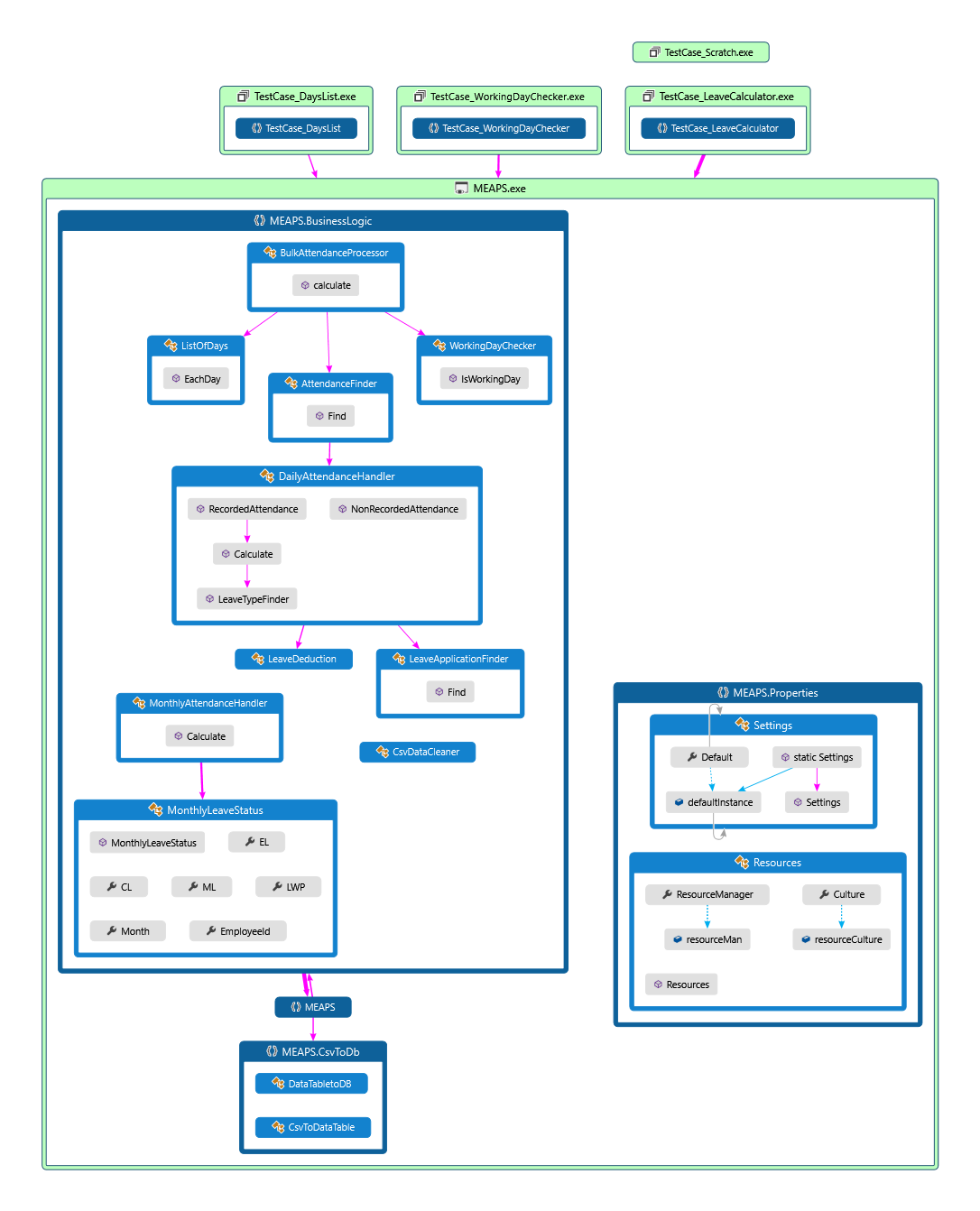
**7.1 Code Map**

Fig. 7.1.1 Code map Generated in Visual Studio

**7.2 Data Flow Diagram**

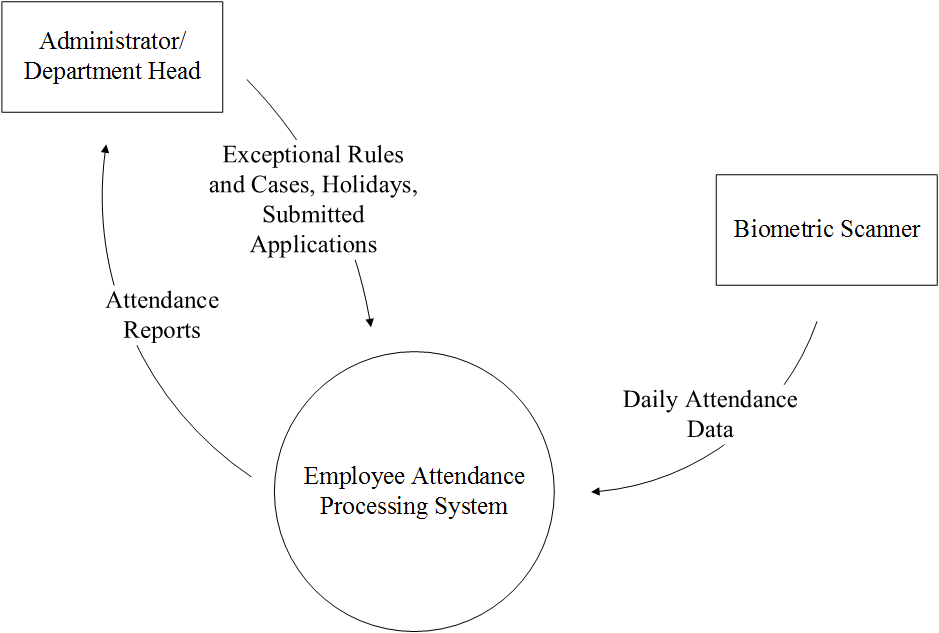
****

Fig 7.2.1 DFD Level 0

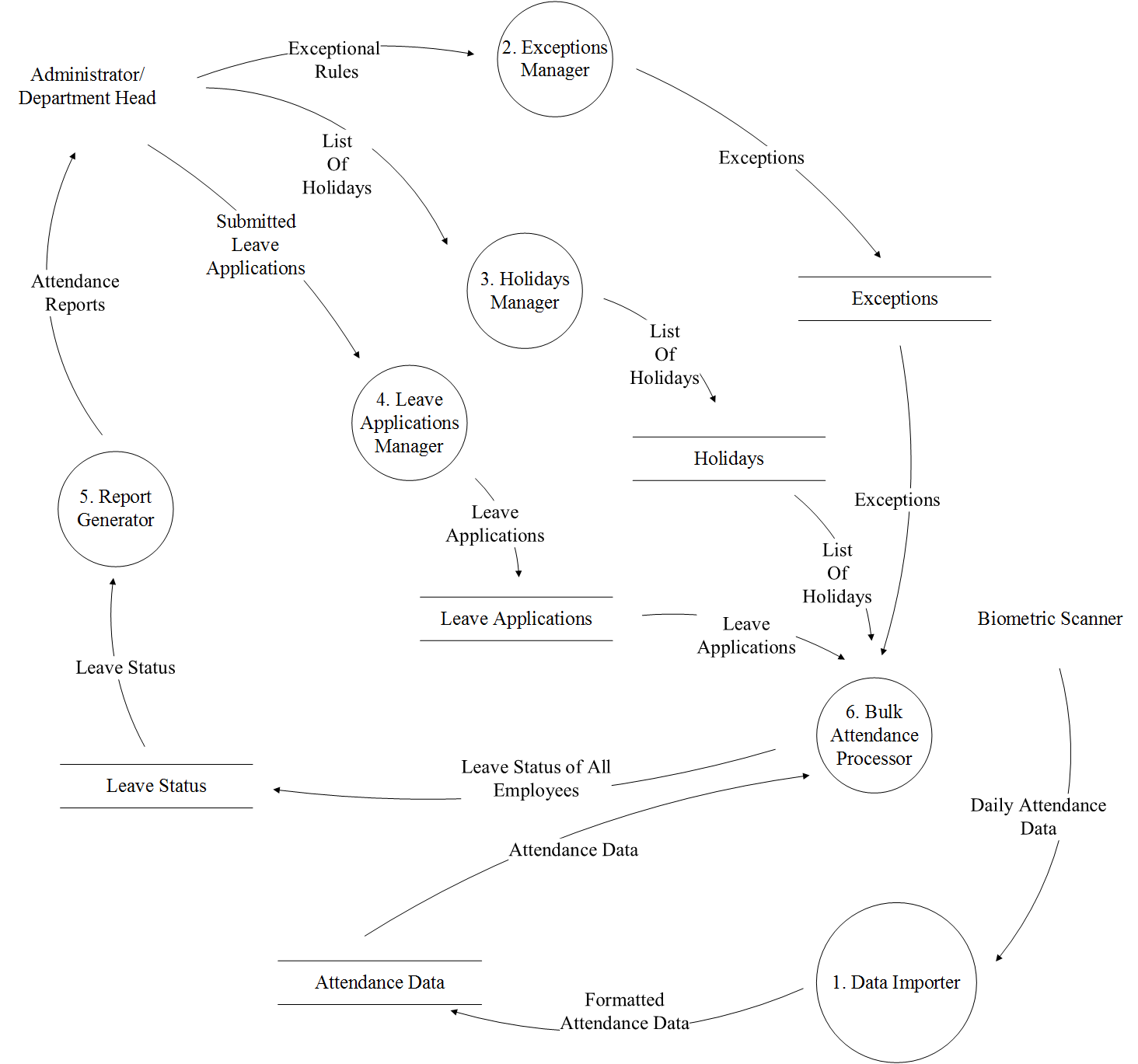


Figure 7.2.2 DFD Level 1

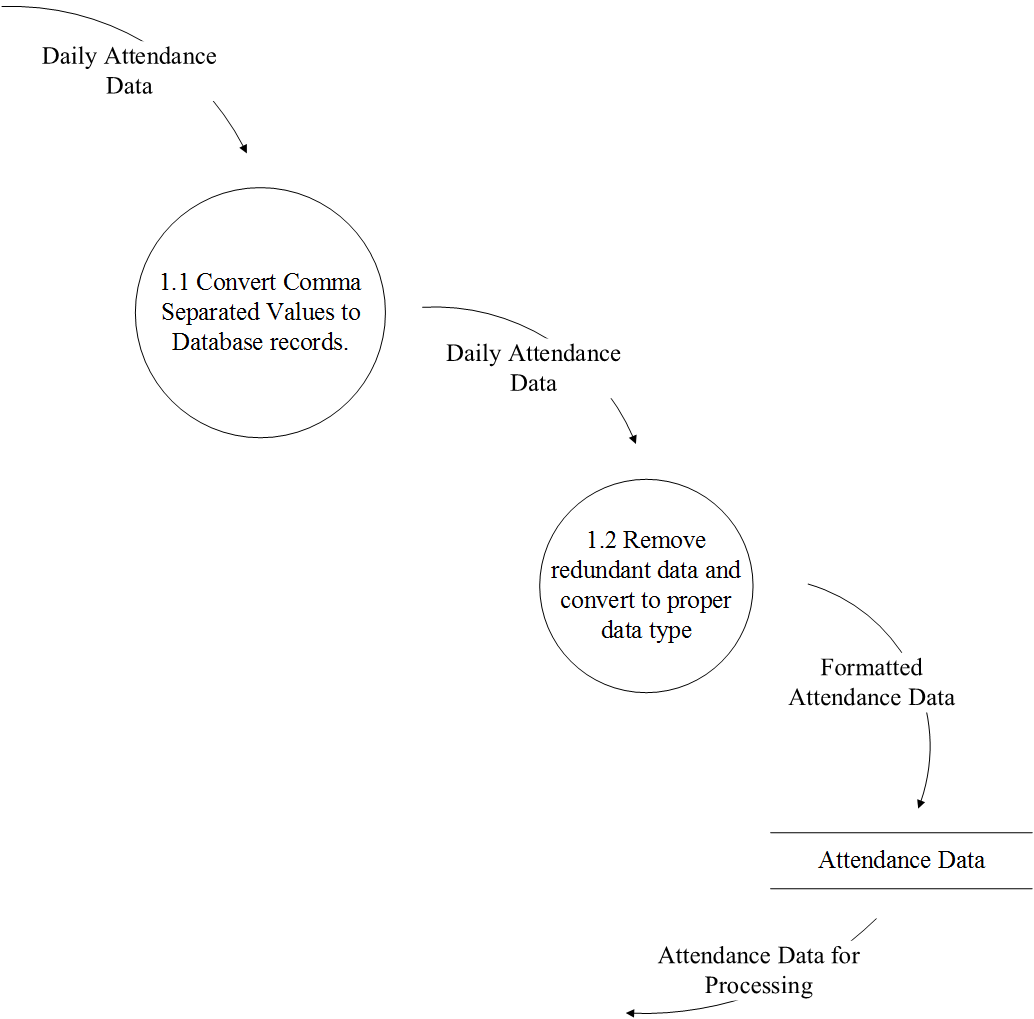


Fig 7.2.3 DFD Level 2

**7.3 Entity Relationship Diagram**

****

Fig. 7.3.1 Entity Relationship Diagram

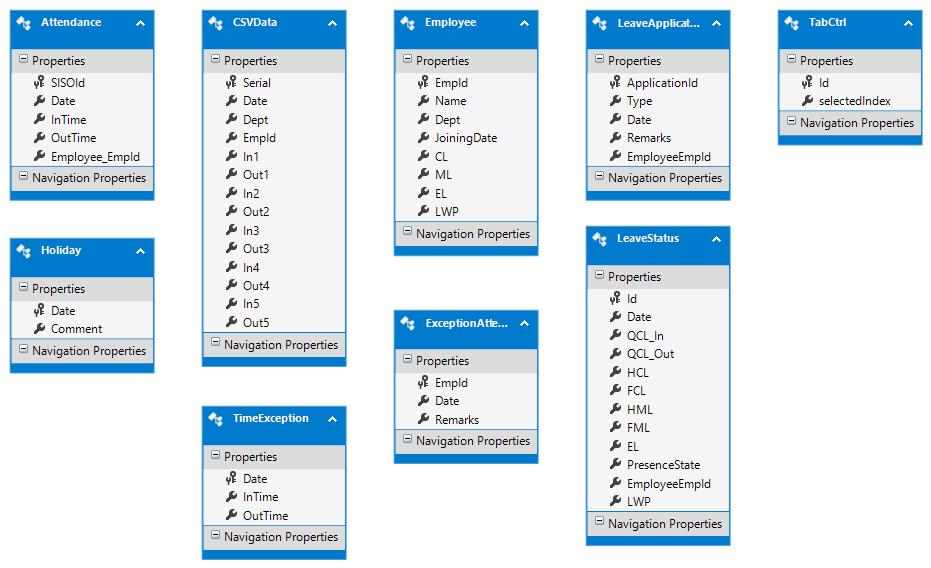
****

Fig. 7.3.2 Entity Designer Diagram in Visual Studio

**8. DATA DICTIONARY**

|  |  |  |
| --- | --- | --- |
| **Name** | **Data type** | **Allow nulls** |
| Serial | int | Unchecked |
| Date | nvarchar(MAX) | Unchecked |
| Dept | nvarchar(MAX) | Unchecked |
| EmpId | nvarchar(MAX) | Unchecked |
| In1 | nvarchar(MAX) | Unchecked |
| Out1 | nvarchar(MAX) | Checked |
| In2 | nvarchar(MAX) | Checked |
| Out2 | nvarchar(MAX) | Checked |
| In3 | nvarchar(MAX) | Checked |
| Out3 | nvarchar(MAX) | Checked |
| In4 | nvarchar(MAX) | Checked |
| Out4 | nvarchar(MAX) | Checked |
| In5 | nvarchar(MAX) | Checked |
| Out5 | nvarchar(MAX) | Checked |

Fig 8.1 CSV Data

|  |  |  |
| --- | --- | --- |
| **Name** | **Data Type** | **Allow Nulls** |
| SISOId | int | Unchecked |
| Date | datetime | Unchecked |
| InTime | time(7) | Unchecked |
| OutTime | time(7) | Unchecked |
| Employee\_EmpId | int | Unchecked |

Fig 8.2 Attendance

|  |  |  |
| --- | --- | --- |
| **Name** | **Data Type** | **Allow nulls** |
| EmpId | int | Unchecked |
| Name | nvarchar(MAX) | Unchecked |
| Dept | nvarchar(MAX) | Unchecked |
| JoiningDate | datetime | Unchecked |
| CL | int | Unchecked |
| ML | int | Unchecked |
| EL | int | Unchecked |

Fig 8.3 Employees

|  |  |  |
| --- | --- | --- |
| **Name** | **Data type** | **Allow nulls** |
| EmpId | int | Unchecked |
| Date | datetime | Unchecked |
| Remarks | nvarchar(MAX) | Unchecked |

Fig. 8.4 Exception Attendances

|  |  |  |
| --- | --- | --- |
| **Name** | **Data type** | **Allow nulls** |
| Date | datetime | Unchecked |
| Comment | nvarchar(MAX) | Unchecked |

Fig 8.5 Holidays

|  |  |  |
| --- | --- | --- |
| **Name** | **Data type** | **Allow nulls** |
| ApplicationId | int | Unchecked |
| Type | nvarchar(MAX) | Unchecked |
| Date | datetime | Unchecked |
| Remarks | nvarchar(MAX) | Unchecked |
| EmployeeEmpId | int | Unchecked |

Fig 8.6 Leave Application

|  |  |  |
| --- | --- | --- |
| **Name** | **Data Type** | **Allow nulls** |
| Id | int | Unchecked |
| Date | datetime | Unchecked |
| QCL\_In | int | Checked |
| QCL\_Out | int | Checked |
| HCL | int | Checked |
| FCL | int | Checked |
| HML | int | Checked |
| FML | int | Checked |
| EL | int | Checked |
| PresenceState | nvarchar(MAX) | Unchecked |
| EmployeeEmpId | int | Unchecked |

Fig 8.7 Leave statuses

|  |  |  |
| --- | --- | --- |
| **Name** | **Data type** | **Allow nulls** |
| Date | datetime | Unchecked |
| InTime | time(7) | Unchecked |
| OutTime | time(7) | Unchecked |

Fig 8.8 Time exception

**9. TESTING**

System Testing is an important stage in any system development lifecycle. Testing is a process of executing a program with the intention of finding errors. The importance of software testing and its implications with respect to software quality cannot be overemphasized. Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. A good test case is one that has a high probability of finding a yet undiscovered error.

Testing is the set of activities that can be planned in advance and conducted systematically. Different test conditions should be thoroughly checked and the bugs detected should be fixed. The testing strategies formed by the user are performed to prove that the software is free and clear from errors. To do this, there are many ways of testing the system’s reliability, completeness and maintainability.

**9.1 Unit Testing**

In the unit testing the analyst tests the program making up a system. The software units in a system are the modules and routines that are assembled and integrated to perform a specific function. In a large system, many modules on different levels are needed. Unit testing can be performed from the bottom up starting with the smallest and lowest level modules and proceeding one at a time. For each module in a bottom-up testing, a short program executes the module and provides the needed data.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tested by** | | Rangan Das, Sukhpreet Singh Anand and Rajan Kumar | |
| **Test Type** | | Unit Testing | |
| **Test Case No.** | | 1 | |
| **Test Case Name** | | Working Day Checker | |
| **Test Case Description** | | This module takes a date as input and matches it with the list of holiday dates records stored in the database to check whether the given day is a holiday or not. | |
| **Item(s) to be tested** | | | |
| 1 | Verification of the date entered as input with the holiday records in the database. If a match is found, return true or else false. | | |
| **Specifications** | | | |
| **Input** | | | **Expected Output/Result** |
| 1) 25th December, 2016  2) 2nd February, 2016 | | | 1) true  2) false |

|  |  |  |  |
| --- | --- | --- | --- |
| **Tested by** | | Rangan Das, Sukhpreet Singh Anand and Rajan Kumar | |
| **Test Type** | | Unit Testing | |
| **Test Case No.** | | 2 | |
| **Test Case Name** | | Leave Calculator | |
| **Test Case Description** | | This module takes Employee ID, date, entry and exit times as inputs, checks database for any leave applications and generates the type and the fraction of leave taken. | |
| **Item(s) to be tested** | | | |
| 1 | Create a record of the Leave Status table based on the input and calculate the different fields. Finally print all the values of the record. | | |
| **Specifications** | | | |
| **Input** | | | **Expected Output/Result** |
| 5000 1/1/2000  9:50 AM  5:30PM | | | ¼ CL In |

|  |  |  |  |
| --- | --- | --- | --- |
| **Tested by** | | Rangan Das, Sukhpreet Singh Anand and Rajan Kumar | |
| **Test Type** | | Unit Testing | |
| **Test Case No.** | | 3 | |
| **Test Case Name** | | Days List | |
| **Test Case Description** | | This module takes two dates and generates a list of all calendar dates between the two dates | |
| **Item(s) to be tested** | | | |
| 1 | List of days generated should correspond to the calendar being used. | | |
| **Specifications** | | | |
| **Input** | | | **Expected Output/Result** |
| 25th December, 2016  1stJanuary, 2016 | | | 12/25/2015 Friday  12/26/2015 Saturday  12/27/2015 Sunday  12/28/2015 Monday  12/29/2015 Tuesday  12/30/2015 Wednesday  12/31/2015 Thursday  1/1/2016 Friday |

**9.2 Integration Testing**

Integration testing is a systematic technique for constructing the program structure while conducting test to uncover errors associate with interfacing. Objectives are used to take unit test modules and built program structure that has been directed by design. The integration testing is performed for MEAPS to make it a complete system. After integration the project works successfully.

The process involves the testing of the deposit material to ensure that it is virus free, accessible and of the expected type.

The following checks are carried out as standard on all deposits received:

**9.2.1 Virus Check:** The software is checked using the very latest virus checking software

* + 1. **Media read check:** The material is read to ensure that all data can be retrieved successfully.
    2. **Compression check:** All compressed files are checked to ensure that they can be decompressed successfully.
    3. **Source code check:** A check is carried out to ensure that source code files have been deposited and that they are in human readable form.
    4. **Source code information check:** A sample of the source code files is examined to determine whether they contain features that aid readability, such as; modification histories, indentation, comments, meaningful variable and procedure names and meaningful file names.

**10. SNAPSHOTS**



Fig 10.1 Splash Screen

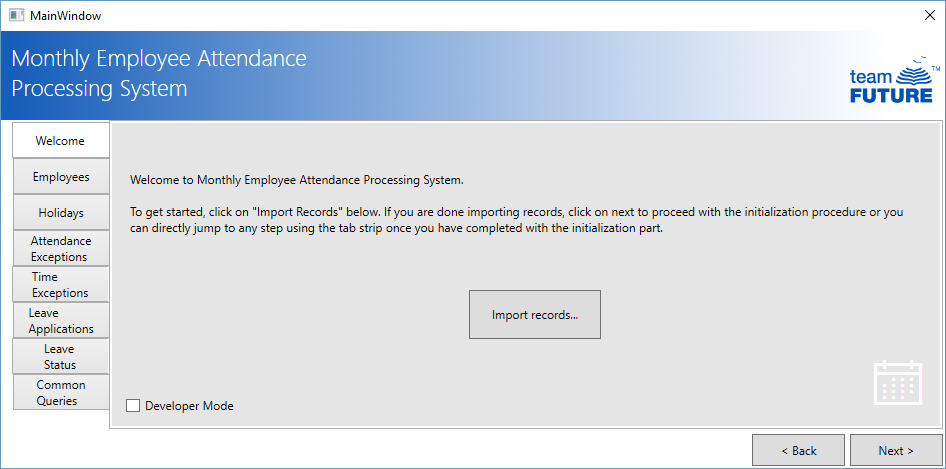


Fig 10.2 Welcome Screen

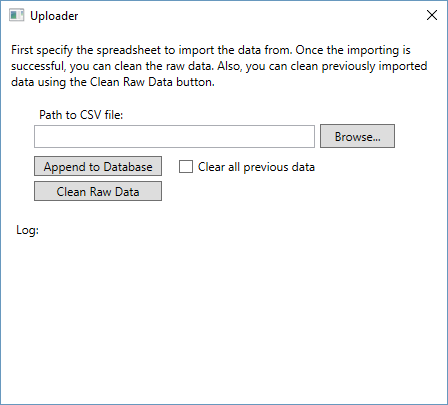


Fig 10.3 Import attendance from spreadsheet

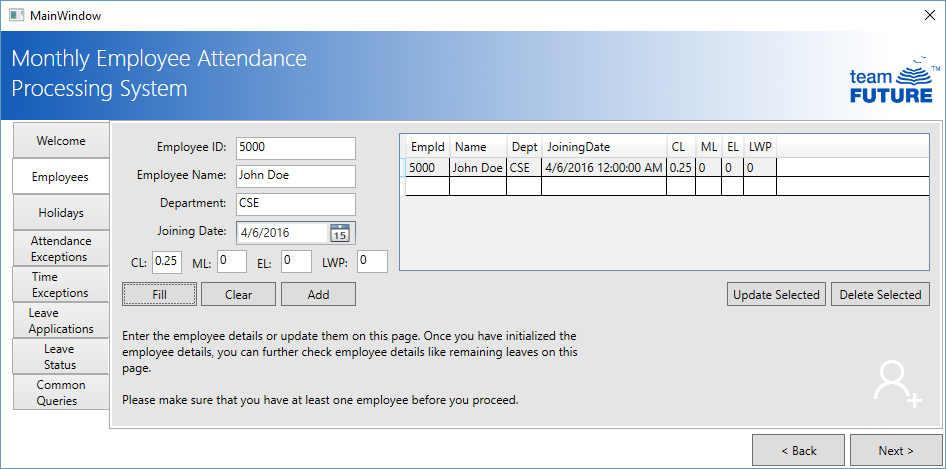


Fig 10.4 Employee data initialization

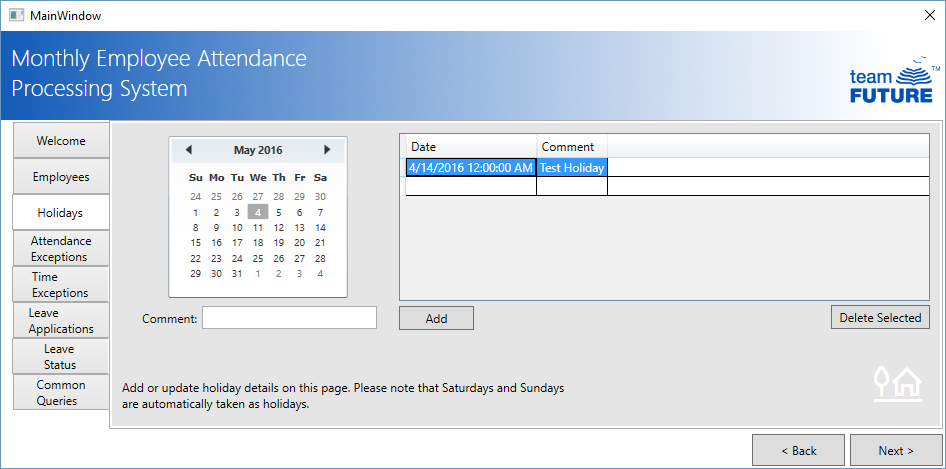


Fig. 10.5 Holiday details initialization

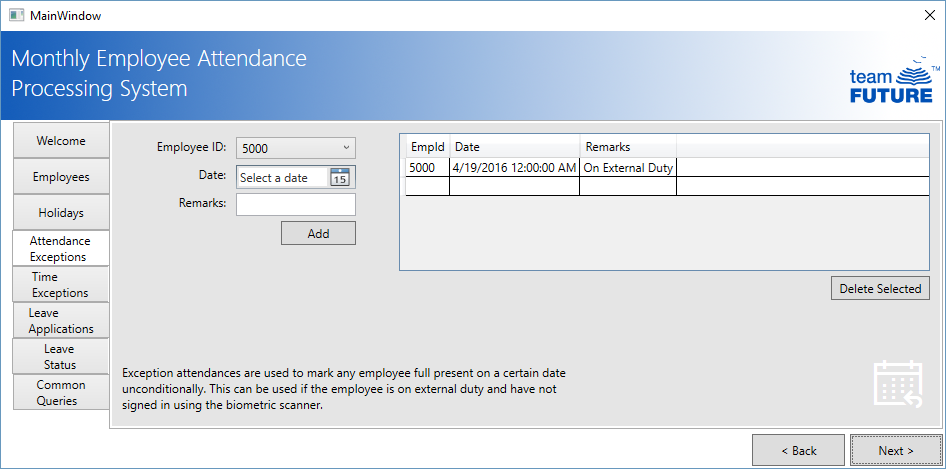


Fig 10.6 Exception attendance initialization

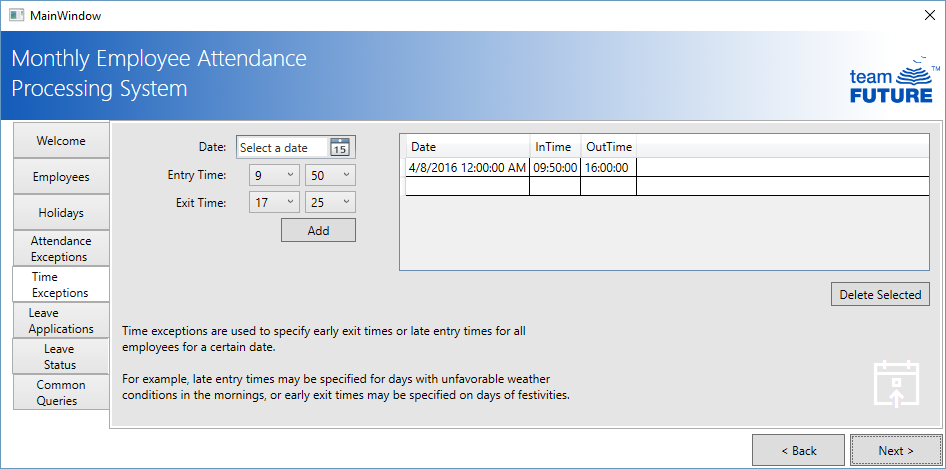


Fig 10.7 Time Exceptions initialization

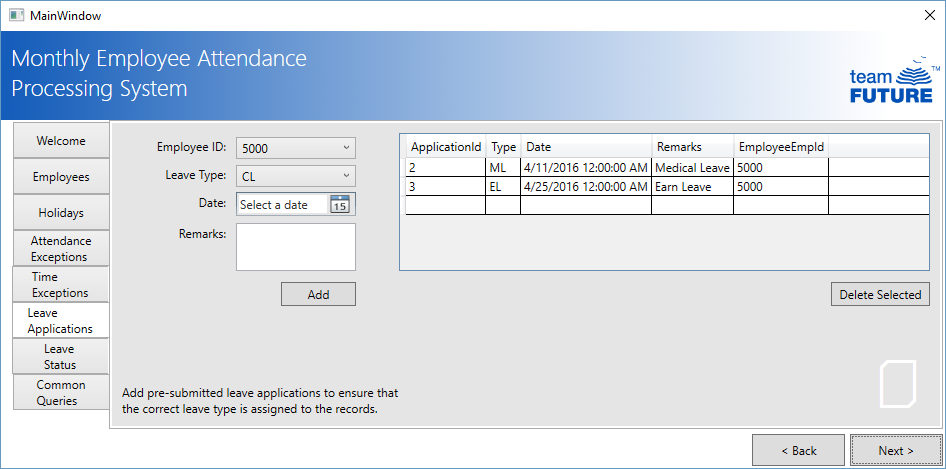


Fig 10.8 Leave Applications Entry

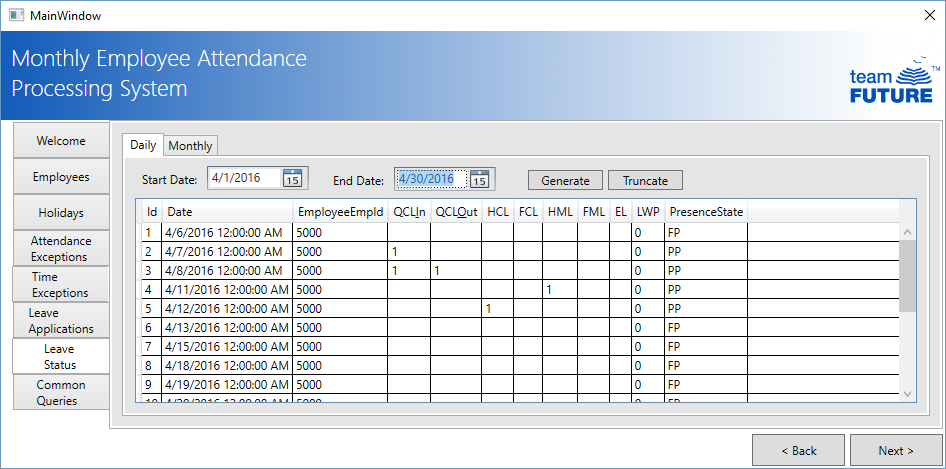


Fig 10.9 Daily leave status report

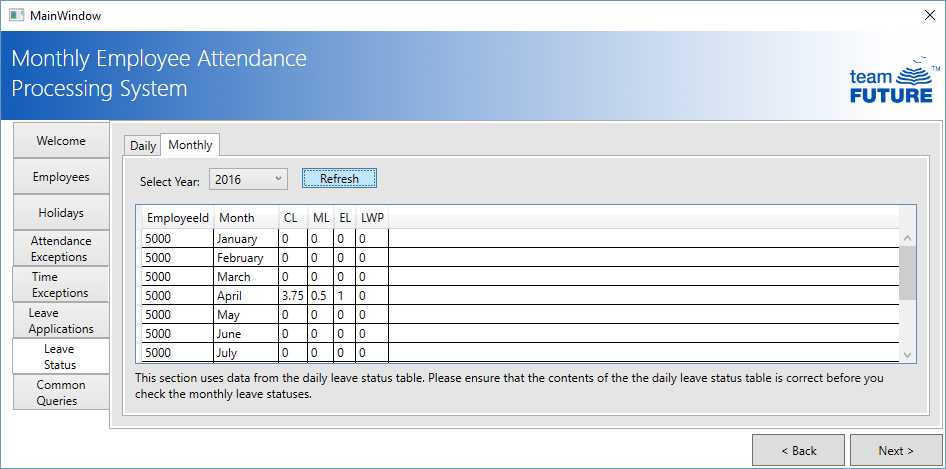


Fig 10.10 Monthly leave status report

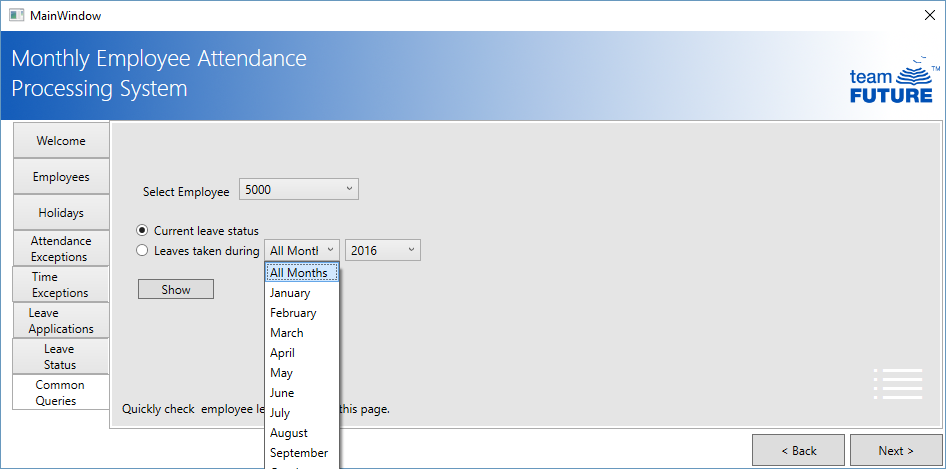


Fig 10.11 Common Queries

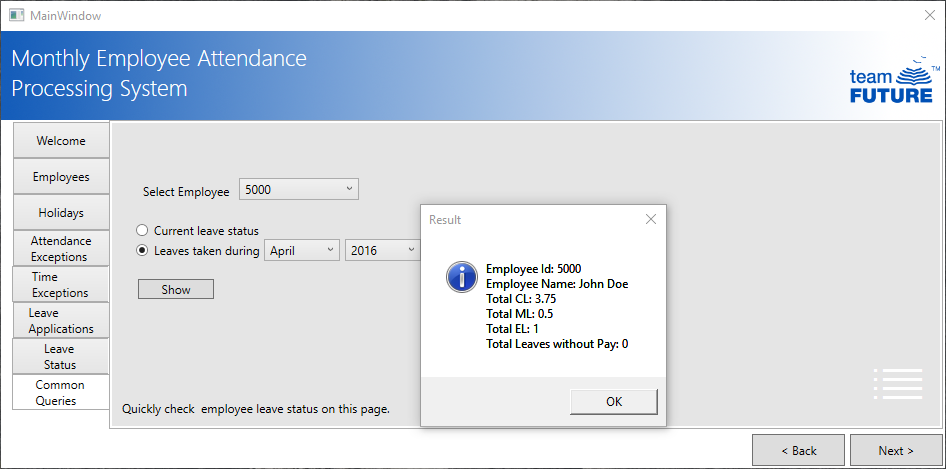


Fig 10.12 Common Queries

**11. CONCLUSION AND FUTURE SCOPE**

MEAPS is a real life software system and finds its use in any organization. The project demands extensive working hours. Planning and coding requires limited time, but debugging and testing will be time taking. Such working hours are to be set aside for the project.

Current real life data might not be available. Past data from the biometric scanner has to be acquired for testing.

MEAPS is an extensive robust software which finds its way in any kind of organization be it university, bank, college or even home. Security is the sole aim of installing MEAPS along with the biometric scanner. The biometric scanner is solely responsible for granting authorization to private places under high security. If the biometric scanner fails to identify the individual, he/she will not be granted the required access. Now, coming to the inside of the biometric, we are unable to perform operations involving modification of these data. This is where MEAPS comes into action. This software application connects the bridge between the biometric hardware system and the software based database system which enables us insert, update, modify and delete those data in the form of database format. This provides interesting GUI which enables the user with proper informative, functional and simple workflow. There are a lot of provisions for handling exception and special cases. This provide unique features and efficient functionalities. The unique features of this system are:

* 1. Cleaning up raw data:Data from the spreadsheet is cleaned up and stored in a relational database.
  2. Storing data in a relational data store:A database with pre-existing employee data will be used to store the cleaned up data.
  3. Developing of application program to process data:Data will be batch processed used an application program that is run at regular intervals.
  4. Allow manual input in necessary areas:In case of exceptions, the administrator can take over the automated processing or can define how the automated processing will occur.
  5. Provide advanced reporting and evaluation systems: Storing data in an RDBMS will automatically provide advanced querying systems.

These are the highlights of the proposed system but there are a few future scopes which are discussed below. There are provisions for sending repeated and regular informative emails and text messages to the employees based on their performance on attendance. Nowadays, the entire application system is based on security measures which allows features like sending OTPs, passwords and notifications via internet. All these can be added up in the system which enables efficient management of the desired data. Also, complex report generation and analysis of surveys can be done using MEAPS. It can also be made cross platform that is platform independent, so that the system works on any operating system.

**12. REFERENCES**

The following books and websites were referred to while this project was being made

• Professional C# 5.0 and the .NET 4.5 Framework 6th Edition by Andrew Troelsen

• PluralSight – Getting Started with Entity Framework 6.0

• Channel 9 Official Site – channel9.msdn.com

• Microsoft Virtual Academy - www.microsoftvirtualacademy.com

• Stack Overflow – www.stackoverflow.com

• Code Project – www.codeproject.com