Employee Management System with Biometric Attendance

Project II Report

**Submitted in the partial fulfilment of requirement
For the award of the degree of**

Bachelor of Technology (B.Tech.)

In

Computer Science and Engineering

Under the Supervision of:
Er. Pooja Saini
Assistant Professor
CSE Dept.

Submitted By:
Divya Kashyap (2315015)
Rahul Rana (2315029)
Sahil Aggrawal (2315035)
Ullas Sood (2315049)

Department of Computer Science and Engineering
Ambala College of Engineering and Applied Research
(Affiliated to Kurukshetra University, Kurukshetra)
APRIL - 2019

CERTIFICATE

This is to certify that **Divya Kashyap (2315015), Rahul Rana (2315029), Sahil Aggrawal (2315035) and Ullas Sood (2315049)** studying in Ambala College of Engineering and Applied Research, Devsthali, (Batch: 2015-2019) has completed their project II entitled **"Employee Management System with Biometric Attendance"** at Ambala College of Engineering and Applied Research, Devsthali under my supervision during 8th semester.

Er. Pooja Saini

Project Supervisor

DECLARATION OF STUDENT

We hereby declare that the work which is present in this major project report entitled **"Employee Management System with Biometric Attendance"** in the partial fulfilment for the award of the degree of Bachelor of Technology and submitted to the Department of Computer Science and Engineering of Ambala College of Engineering and Applied Research, Devsthali affiliated to Kurukshetra University, Kurukshetra is an authentic record of us, carried out during a period of **JAN 2019 – APR 2019,** under the supervision of **Er. Pooja Saini, Assistant Professor (CSE).**

The matter presented in this project report has not been submitted by us for the award of any other degree of this or any other institute/university.

Date: _____________

2315015
(Divya Kashyap)

2315029
(Rahul Rana)

2315035
(Sahil Aggrawal)

2315049
(Ullas Sood)

PREFACE

With large number of work opportunities due to growing technologies, the Human workforce has increased. Thus there is a need of a system which can handle the data of such a large number of Employees. Employee Management System with Biometric Attendance is an application, developed to maintain the details of employees working in any organization. This system allows the administrator to add new employees, edit employee details, add salary information of employee and accept or reject leave requests from the employee. The Employee on the other hand can view his/her salary details, attendance details as well as can apply for leaves and also can view old leaves taken by him/her. The system is also capable of marking attendance of employees on daily basis and uses a Biometric approach which makes the process of marking attendance faster and more efficient.

Chapter 1 It contains the **Introduction**. This chapter provides the details regarding the introduction of the area of project, project itself, project scope, project objective along with complete SRS document and details regarding the components used in the system.

Chapter 2 It contains the **requirement analysis** of the system. This chapter describes the many type of requirements falls in the category such as use case diagrams, data flow diagrams & ER diagrams as well as the functional and non-functional requirements.

Chapter 3 It contains the **detailed design** of the system. It is very important to understand the whole design of the system according to every perspective to make coding efficient and easy. This chapter contains employer interface, admin interface etc...

Chapter 4 It contains the **implementation and testing** of the system. This chapter contains implementation, testing & user feedback of the project.

Chapter 5 It contains the **results** visuals of the System. This Chapter simply contains the pictures of the actual System developed and also gives a tour of the whole system.

At the end, the conclusion of the whole report is shown along with references and appendices.

ACKNOWLEDGEMENT

Engineers in all disciplines must acquire knowledge of project making. Student, in particular, will find ‘project making’ as an integral part of their studies that will infuse the spirit of doing practical work in them.

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible whose constant guidance crowned our efforts with success.

We sincerely express our deep gratitude to the management of our college for giving us liberty to choose and to work on the most relevant project i.e. **“Employee Management System with Biometric Attendance”**. We are thankful to **Er. Manjit Singh** (HOD CSE) for ensuring that we have a smooth environment at the college and lab. At the very outset we would like to offer our never ending thanks to our project supervisor **Er. Pooja Saini** (Assistant Professor, CSE) who helped us with our project from the beginning till the end. Her continuous surveillance over our work allowed us to work more efficiently.

Divya Kashyap (2315015)
Rahul Rana (2315029)
Sahil Aggrawal (2315035)
Ullas Sood (2315049)

ABSTRACT

An Employee Management System consists of crucial work-related and important personal information about an employee. It is like an online inventory of all employees of an organization. Employees are the strength of any organization, and it is more so in case of a growing business. It is crucial to handle this aspect of your business well. A good employee management system can actually make a world of difference to an organization, especially true in case of start-ups and small businesses, where the focus should be on growing the business more than anything else.

Employee Management System is used to keep track of your company's or field's employees with the help of a tool. This is beneficial for an organization as well as for employees. This feature-rich, fully customizable and scalable application makes it easier and more efficient to monitor and manage employees. This application enables the business organization to make more efficient use of their resources while saving time and money both.

List of Figures

List of Tables

Table of Contents

Chapter 1 Introduction

1.1 Purpose

Manual handling of employee information poses a number of challenges [1]. The use of paper work in handling the processes could lead to human error and not forgetting the fact that this is time consuming. These problems can be tackled by designing and implementing "**Employee Management System with Biometric Attendance**". The project is aimed at setting up employee information system about the attendance and status of the employee in order to help monitor the performance and achievements of the employee through a password protected system. This system will maintain employee information in a database by fully privacy and authority access. Using **Biometric attendance system** allows employees to clock in and out using their fingerprint. This can be done with the help of Fingerprint Module and Arduino microcontroller. This fingerprint sensor module will make adding fingerprint, detection and verification super simple. Pairing a biometric device with attendance software allows for more efficient attendance processing. Rather than spending hours adding up manually submitted clock-in and clock-out times, a biometric enabled attendance software will allow you to automatically import data into your system. This will significantly cut down on processing time and will eliminate errors derived from manually inputting information.

1.2 Objective

Employee Management System is a versatile application, developed to maintain the details of employees working in any organization. The objective of this project is to provide a comprehensive approach towards the management of employee information. This will be done by designing and implementing a Biometric enabled Employee Management System that will bring up a major paradigm shift in the way that employee information is handled.

The objectives of this system include:

- Design of a standalone Employee management system to fulfil requirements such as keeping record of employees working in organization, recording salary details, managing leaves taken by employees and attendance of employees.

- Well-designed database to store employee information.
- A user friendly front-end for the user to interact with the system.
- Easy and secure marking of attendance using biometric approach.
- Generic System that can be easily modified to meet any organizations need.

1.3 Project Scope

As the objective stated, the system is not made for any specific type of organization rather it is a generic system that contains all the basic features an employee management system [2] need to have. Hence, the scope of this project is limited to the following

Adding Employee profiles

Admin can enter employee details that are newly recruited as well as have the ability to view, edit as well as delete these details as and when needed.

Salary Information

The salary information of employees is stored and added as and when required.

Leave Requests

The system automates the leave taking procedure by allowing employees to request leaves and the admin to view and accept/reject those requests. The employee can also see his/her old leaves to help make decision about further leaves to take.

Attendance System

The most important aspect of any organization is to keep track of employee attendance which facilitates in making decisions about his/her salary and worthiness. Manually taking attendance is a cumbersome process and hence the system automates the process with use of biometric system.

Detail Access

Finally employees have the access to their information such as salary information, attendance information and leave information allowing them to make de3cisions based on them.

1.4 Benefits

Employee Management System is a ready source of information

It performs as a readily available source of information between the organization and the employee. Contact information, salary information, posts, work schedule, education information etc. is what most database systems consist of. Such information is readily available through HR software like Employee Management System.

Highly efficient system

Employee management system is highly efficient. An administrator of the organization can easily retrieve information about an employee whenever required, and that too on short notice. One can avoid making calls to the employee out on vacation just to retrieve an address to send an important letter.

Reliable accuracy

Since the information is mostly fed in by the employees themselves you can be sure the information is accurate since it's straight from the source. Moreover, an employee can access their information at any time. Therefore, he/she can keep it updated and correct mistakes, if any.

Updated data

The information added to the employee management system can be available for as long as an employee is working in a firm or if needed, longer than that. Also, if at any point in time, the employee data changes the employee they can make the alterations. As a result, obsolete Data is a rare find on such systems.

Employee management systems allow for confidentiality

Specific information about the employee can not only be set to be kept private from public viewing but can also be set to be kept private from anyone other than the admin of the software or the head of the organization. Therefore, it is safer to have an employee management system in an organization, big or small, than have your bank account information lying in some drawer of a dingy desk.

Improved business profits

The bottom line of an organization improves significantly by increasing employee productivity and quality of work.

Increased employee responsibility

Communicating realistic but challenging job expectations and making employees accountable for their decisions and actions result in noticeable improvements in employee tardiness, absences, and organizational commitment.

Ensure a better work environment

Implementation of a proper system goes a long way in providing a much better work environment.

This is said because the software will be empowered to gauge the levels of employee engagement at the workplace and even keep track of regular employee performance. The primary goal is to create a better place or a work environment where employees can thrive considerably and contribute largely.

Flexible management boundaries

The next big benefit that employers reap is the boost in management boundaries. This enables the development of a reward system where employees can take on more responsibility for the work they do, and managers can focus on several employees achieved goals rather than forced to concentrate on other details.

Benefits of using Biometric Attendance Software

Accountability By using a physical characteristic rather than simply using a swipe card or PIN, ensures that the employee is actually present. This avoids issues such as “buddy punching”, a term used to describe when other employees clock in and out for one another.

Efficiency Using a biometric time and attendance software allows for increased efficiency in multiple areas. First, employees don’t have to worry about remembering to bring in a punch card or remembering a PIN, so there is less time spent on recovering lost passwords and manually inputting an employee’s clock-in time. Second, when it comes to payroll, it is much easier to track in and out times by simply downloading the data from your biometric scanning device, and inputting it into your time and attendance software.

Profit A natural bi-product of increased accountability and efficiency is increased profit. By making employees more accountable to attendance times, you will increase productivity and only pay for time actually spent working. The same is true for increased efficiency. Creating a smoother process by using a **biometric time and attendance system**, will allow HR employees to work on more tasks.

1.5 Application Areas of Employee Management System

- Start-ups
- Business ventures
- IT firms
- Human Resource System
- Multi-National Companies (MNCs)

- Public limited company (PLC)
- Private company limited by shares (LTD)
- Unlimited Company

Chapter 2 Requirements Analysis

2.1 Requirements

As the goal of the application is ease of use and to provide an interactive interface, extensive research has been done to gain an insight into the needs and behaviors of various users. The working of the application is made convenient and easy to use for the end user. In employee management systems there are two parties who interact with the system, one of them are employees of the organization who work at the organization and other is the system admin that manages the system.

2.2 Functional Requirements

A good employee management systems [3] should have the following factors

Admin Requirements

- Ability to add new Employee information including his/her basic details, image and finger print for attendance system.
- Ability to view details of already enrolled employee.
- Ability to update details of the employee enrolled in the system.
- Ability to delete any employees details from the system.
- Ability to create salary records for employees.
- Ability to View, Accept or Reject Leave requests from the employees.
- Ability to add new admin for the system.
- Ability to change his/her own password for security.

Employee Requirements

- Ability to view his/her attendance record.
- Ability to view his/her Salary details.
- Ability to apply for single or multiple day leaves
- Ability to view old leaves and their status for making decision for new leaves.
- Ability to change their password.
- Ability to quickly mark their attendance using their fingerprints using biometric module.

2.3 Software Requirements

The system is built using the following software components

2.3.1 Java

Java is a general-purpose programming language that is class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to "bytecode" that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle) and released in 1995 as a core component of Sun Microsystems' Java platform. The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses.

Feature of Java

- Object Oriented
- Platform Independent
- Simple. Java is designed to be easy to learn.
- Secure
- Architecture-neutral
- Portable
- Robust
- Multithreaded
- High Performance
- Dynamic

2.3.2 WindowBuilder

WindowBuilder [8] is composed of SWT Designer and Swing Designer and makes it very easy to create Java GUI applications without spending a lot of time writing code. Use the WYSIWYG visual designer and layout tools to create simple forms to complex windows; the Java code will be generated for you. Easily add controls using drag-and-drop, add event handlers to your controls, change various properties of controls using a property editor, internationalize your app and much more.

WindowBuilder is built as a plug-in to Eclipse and the various Eclipse-based IDEs (RAD, RSA, MyEclipse, JBuilder, etc.). The plug-in builds an abstract syntax tree (AST) to navigate the source code and uses GEF to display and manage the visual presentation.

2.3.3 MySQL

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons

- MySQL is released under an open-source license. So you have nothing to pay to use it.
- MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
- MySQL uses a standard form of the well-known SQL data language.
- MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
- MySQL works very quickly and works well even with large data sets.
- MySQL is very friendly to PHP, the most appreciated language for web development.
- MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
- MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

The MySQL Database is being used in this system as the Backend which is used to store the data of the system.

2.4 Hardware Requirements

The Biometric Attendance feature of the system requires the following hardware to be attached with the system.

2.4.1 Arduino Uno r3

The Arduino Uno [6] is a microcontroller board based on the ATmega328. It has 20 digital input/output pins (of which 6 can be used as PWM outputs and 6 can be used as analog inputs), a 16 MHz resonator, a USB connection, a power jack, an in-circuit system programming (ICSP) header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer (or appropriate wall power adapter) with a USB cable or power it with a AC-to-DC adapter or battery to get started.

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features an ATmega16U2 programmed as a USB-to-serial converter. This auxiliary microcontroller has its own USB bootloader, which allows advanced users to reprogram it.

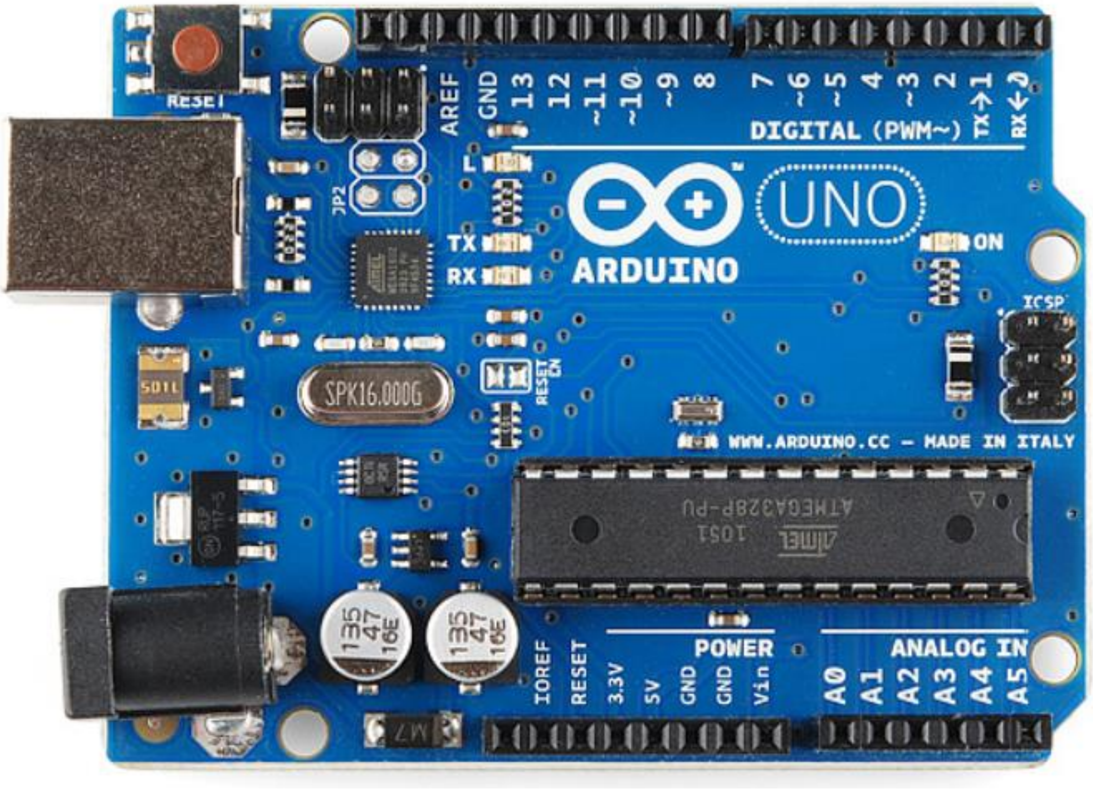


Figure 2-1 Arduino Uno r3

Features of Arduino Uno r3

- Microcontroller: ATmega328
- Operating Voltage: 5V
- Input Voltage (recommended): 7-12V
- Input Voltage (limits): 6-20V

- Digital I/O Pins: 14 (of which 6 provide PWM output)
- Analog Input Pins: 6
- DC Current per I/O Pin: 40 mA
- DC Current for 3.3V Pin: 50 mA
- Flash Memory: 32 KB of which 0.5 KB used by bootloader
- SRAM: 2 KB (ATmega328)
- EEPROM: 1 KB (ATmega328)
- Clock Speed: 16 MHz

2.4.2 Finger Print Sensor r305

Finger print sensor module [5] with TTL UART interface for direct connections to microcontroller UART or to PC through MAX232 / USB-Serial adapter. The user can store the finger print data in the module and can configure it in 1:1 or 1: N mode for identifying the person. The FP module can directly interface with 3v3 or 5v Microcontroller. A level converter (like MAX232) is required for interfacing with PC serial port.

Features

- Integrated image collecting and algorithm chip together, All-in-one
- Fingerprint reader can conduct secondary development, can be embedded into a variety of end products
- Low power consumption, low cost, small size, excellent performance
- Professional optical technology, precise module manufacturing techniques
- Good image processing capabilities, can successfully capture image up to resolution 500 dpi

Specifications

- Fingerprint sensor type: Optical
- Static indicators: 15KVBacklight: bright green
- Interface: USB1.1/UART (TTL logical level)
- RS232 communication baud rate: 4800BPS~115200BPS changeable
- Dimension: 55*32*21.5mm
- Image Capture Surface 15—18(mm)

- Verification Speed: 0.3 sec
- Scanning Speed: 0.5 sec
- Character file size: 256 bytes
- Template size: 512 bytes

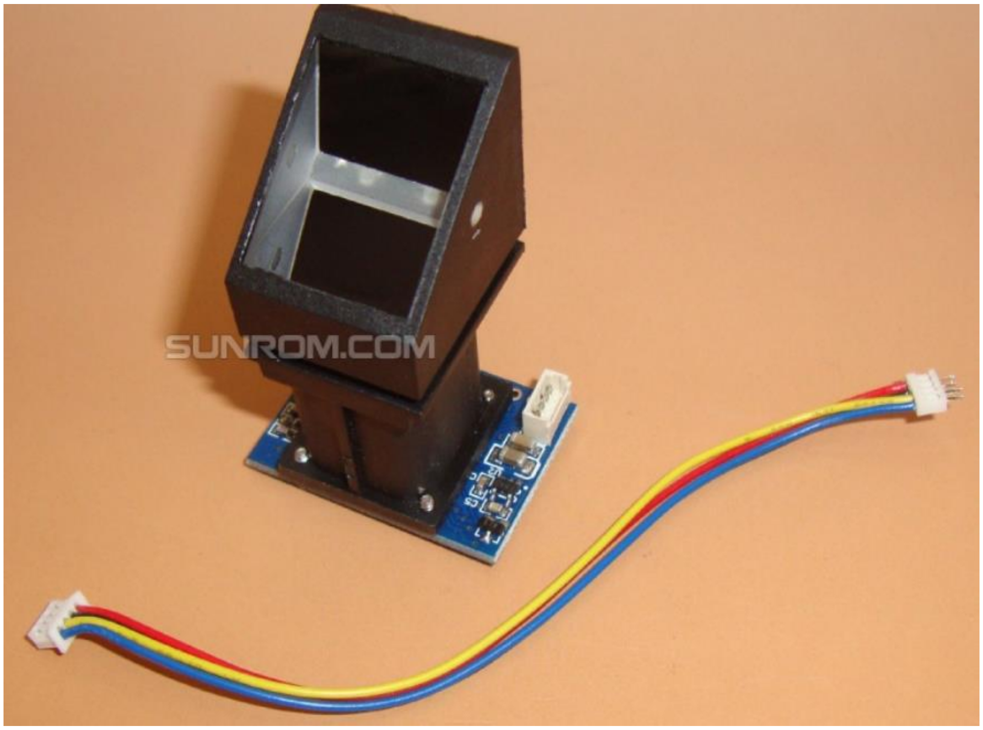


Figure 2-2 Finger Print sensor r305

2.5 Non-Functional Requirements

In system engineering and requirements, non-functional requirements are requirements which specify criteria that can be used to judge the operation of system, rather than specific behaviors. Non- functional requirements are often called qualities of a system. Other terms for non-functional requirements are “constraints”, ”quality attributes”, “quality goals” and “quality of service requirements” . Qualities, i.e. non-functional requirements can be divided into 2 main categories:

Execution qualities, such as security and usability, are observable at runtime

Evolution qualities, such as extensibility and scalability, embody in the static structure of the software system. The non-functional requirements in our projects are

Performance Requirements

The System is expected to perform at least 20hr. continuously during a day without being interrupted by any errors. The functions of the system are expected to work in real-time that is, there should be minimum or no time lag between issuing a command to system and display of the result.

Security Requirements

The system developed should be as secure as possible. Intrusions in the system as well as loopholes are to be avoided. Furthermore only the authorized group of users should have the privilege to access and change information.

Reliability

This system must be highly robust. Amount of computing resources and code required to perform function extent to which program performs with required precision

Correctness - extent to which program satisfies specifications, fulfils user's mission and objective.

Efficiency - Flexibility - effort needed to modify operational program

Interoperability- effort needed to couple one system with another

Testability - effort needed to test to ensure performs as intended

Reusability

This requirement is present, as this system will interact with user.

Extent to which it can be reused in another application.

Effort required to learn, operate, prepare input, and interpret output.

Resource Utilization

The System will be developed so that it uses all type of functions and fulfil all the requirements of the user who uses it. The class hierarchy and pre inbuilt library functions which is completely utilized in the system designing and coding.

Maintainability

Effort required to locate and fix an error during operation

Time The project should be completed within the stipulated time period

Cost The cost involved in marketing the project should be less.

Chapter 3 Design Specification

The system is designed using Object Oriented Methodologies. The design of the system is based on UML Diagrams such as Class Diagram and Sequence Diagrams. This Chapter introduces the UNL diagrams used in the project along with the Diagrams itself made for the project. The Chapter also shows the database design using an Extended Entity-Relationship Diagram and at the end describes the Hardware interfacing of Finger Print Module.

3.1 Class Diagram

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

In the diagram, classes are represented with boxes that contain three compartments:

- The top compartment contains the name of the class. It is printed in bold and centered, and the first letter is capitalized.
- The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lowercase.
- The bottom compartment contains the operations the class can execute. They are also left-aligned and the first letter is lowercase.

In the design of a system, a number of classes are identified and grouped together in a class diagram that helps to determine the static relations between them. With detailed modeling, the classes of the conceptual design are often split into a number of subclasses.

3.1.1 Classes in the System

- **StaffMember**

  This class is the base class for all the members of the organization. It contains all the basic attributes and methods that are common to all the members of the organization and the system stores it.

- **Admin**
  The Admin class is a derived class from the StaffMember class. The Admin class has special privileges in the system including adding employee details, creating admins, updating details, inserting salary information and accepting and rejecting leave requests.
- **Employee**
  The Employee class represents all the employees in the organization. The employee can do basic operations like viewing attendance information, salary information and applying for leaves. This class is also derived from the class StaffMember.
- **SerialTest**
  This class is responsible for the communication of finger print module with the system. The class has the responsibility to initiate, communicate and close the finger print sensor when needed.
- **CheckTime**
  The CheckTime class is responsible to check timing to open and close the attendance portal for the system as well as perform periodic updates to the database for attendance record.
- **MainWindow**
  This class is the heart of User Interface of the system, containing all UI components and is responsible for creation and deletion of objects during Run time.
- **Attendance**
  This class shows and loads the Attendance window in the system and interact with database using other objects.

3.1.2 Relationship among Classes

- Admin class inherits StaffMembers class.
- Employee class inherits StaffMembers class.
- CheckTime class inherits MainWindow class.
- SerialTest class inherits MainWindow class.
- MainWindow class is associated with Attendance view class.
- MainWindow class is associated with Admin class.

- MainWindow class is associated with Employee class.

3.1.3 Class Diagram of the System

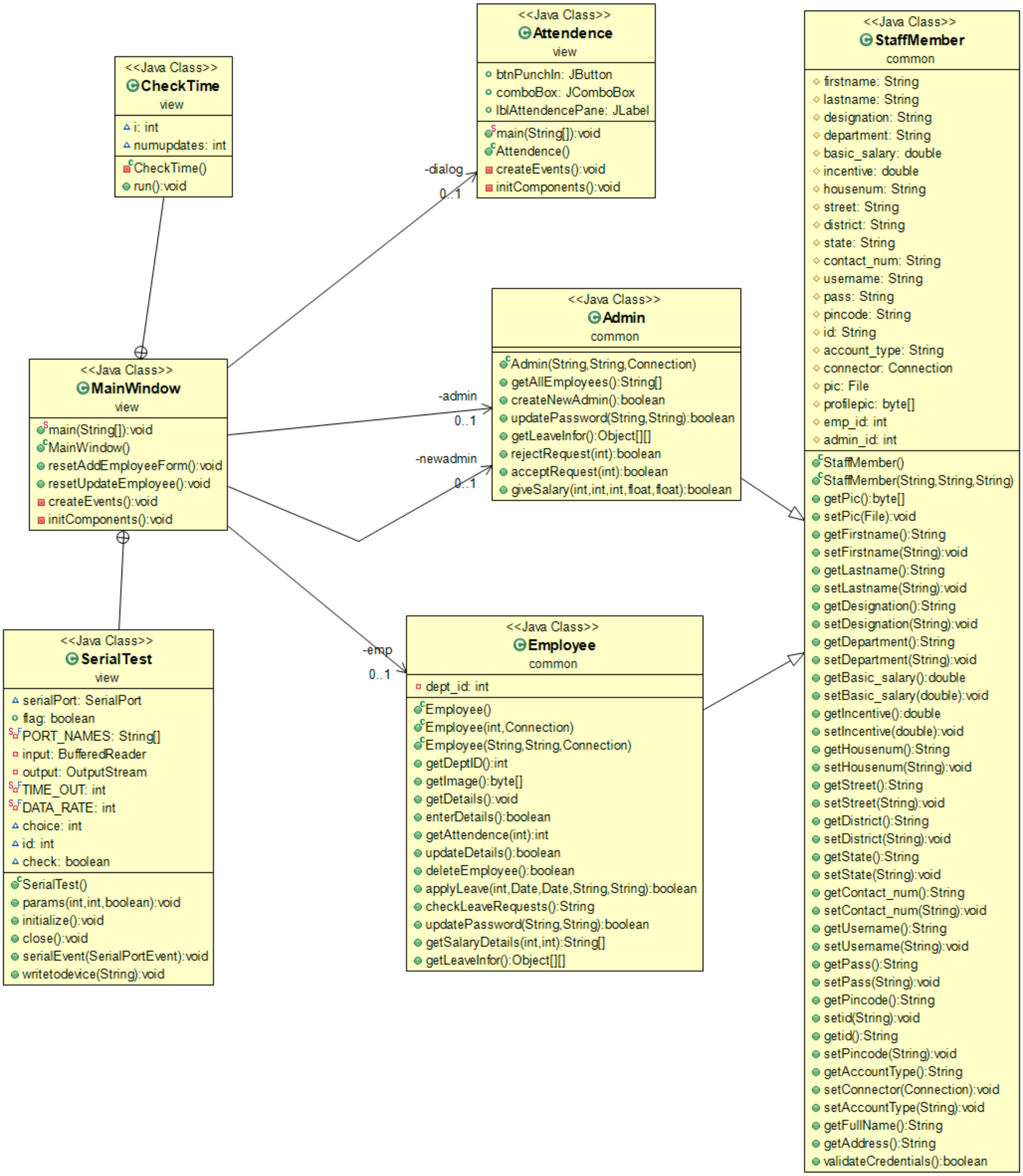


Figure 3-1 Class Diagram of System

3.2 Sequence Diagram

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called **event diagrams** or **event scenarios**.

A sequence diagram shows, as parallel vertical lines (*lifelines*), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

All the Important interactions in the system are shown below using sequence diagrams for each.

3.2.1 Login

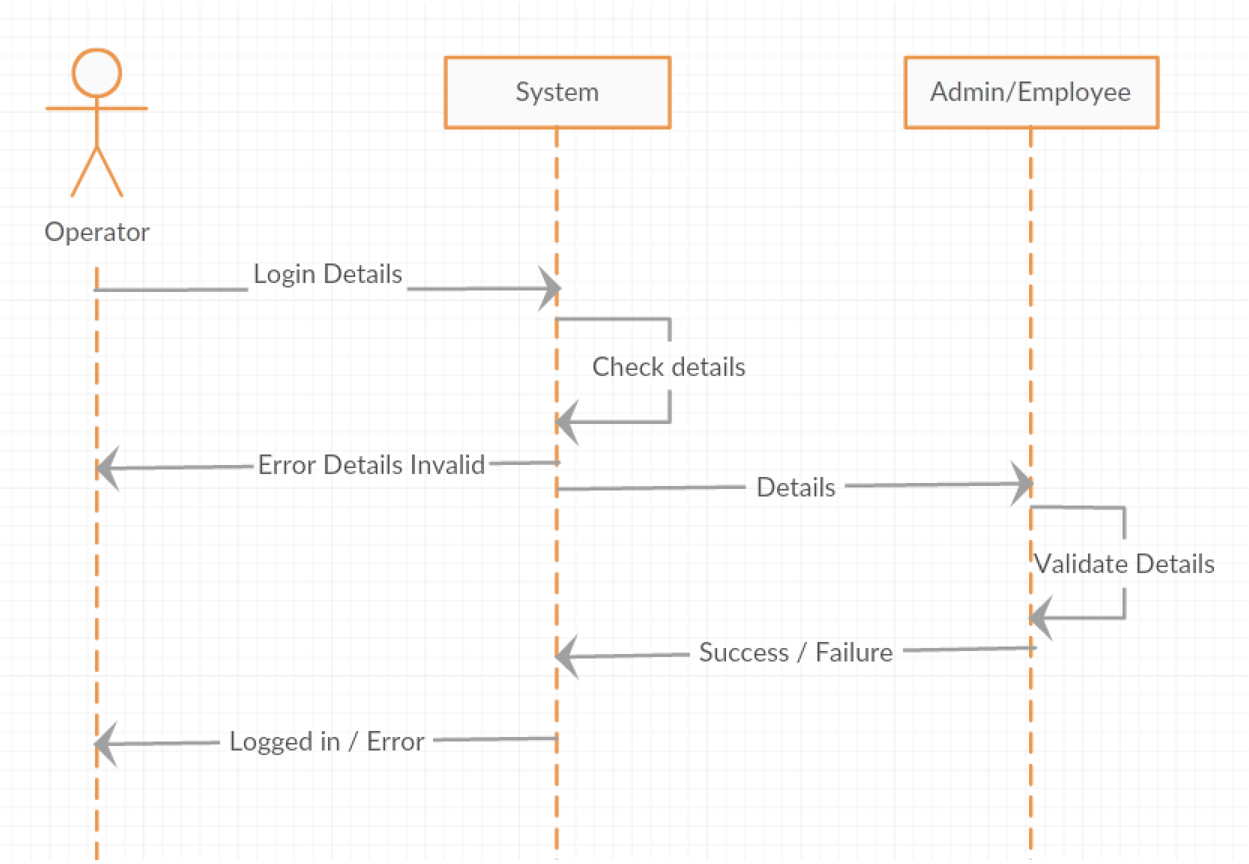


Figure 3-2 Login Sequence

Description

- The operator enters login details.

- The system first checks if details are entered or not.
- If not entered returns error.
- If entered, the Admin/Employee object takes over and check the details.
- If correct details are entered appropriate screen is displayed else an error message is shown.

3.2.2 Add New Employee

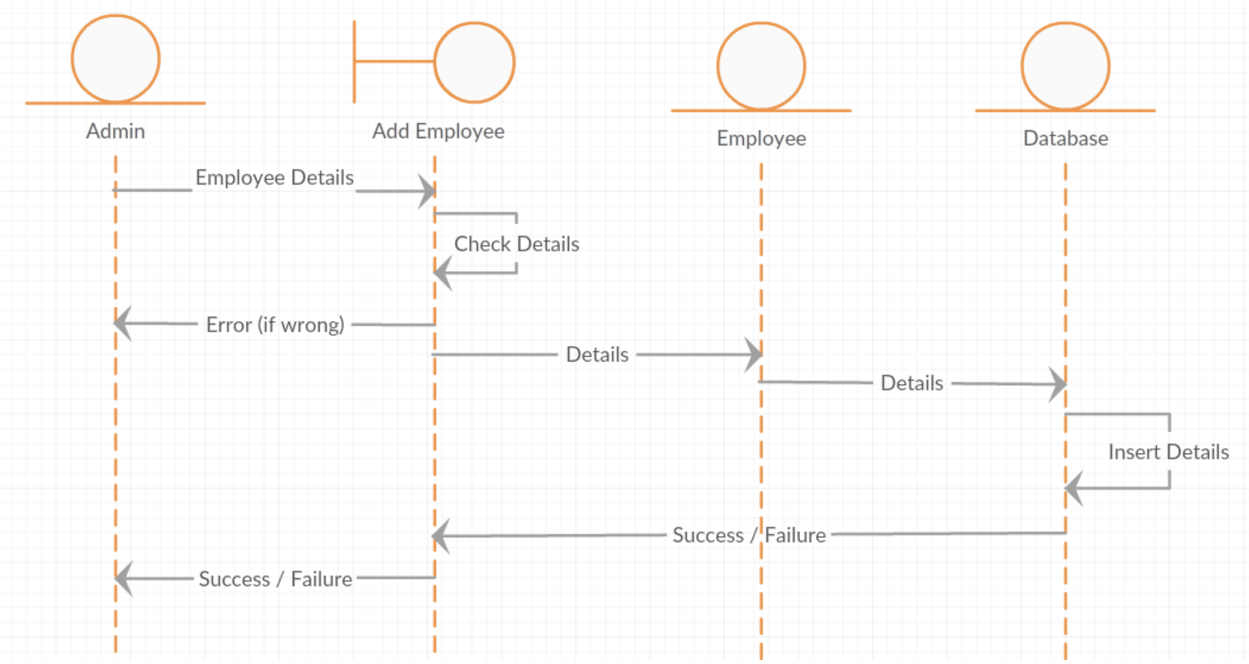


Figure 3-3 Add new Employee Sequence

Description

- The Admin enters new Employee details
- The system checks basic details validity.
- If found incorrect returns error.
- If found correct, new Employee is created in the system and its details are entered in the database.
- If successfully entered returns success message else returns error message.

3.2.3 Finger Print Enrolment

Description

- The Admin selects new enrolment option.
- The system initiate enrolment process.

- If cannot be initiated, system returns error.
- Else system initiates sensor and enrolment is carried out.
- If successfully enrolled, returns success messages else returns error message.

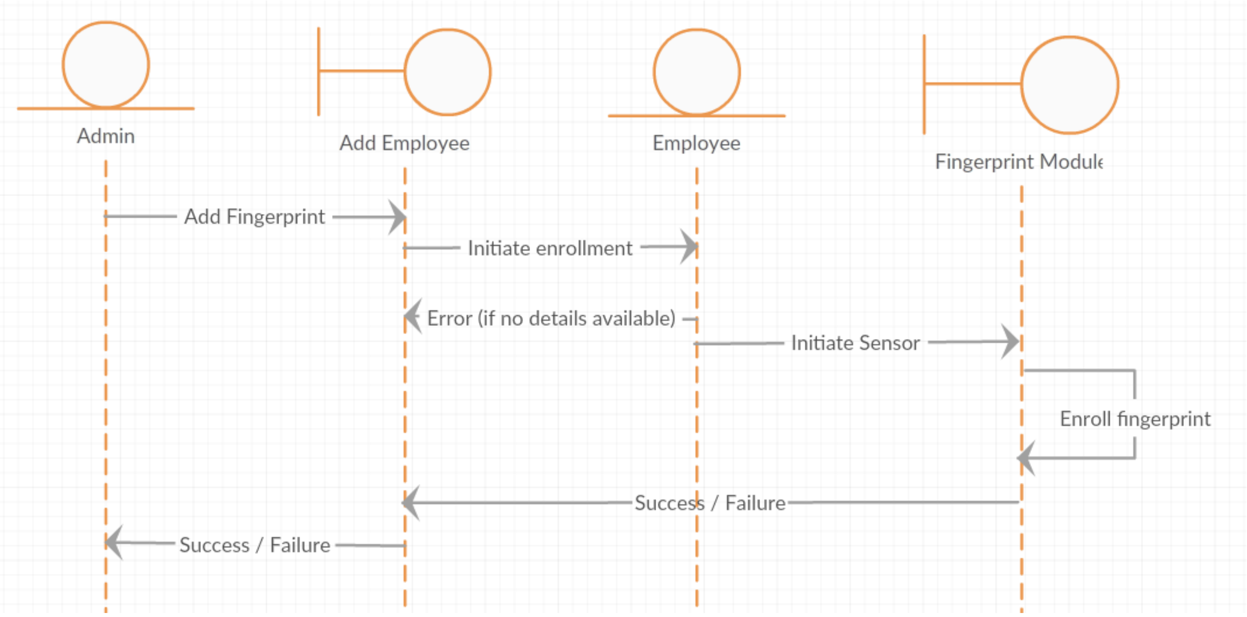


Figure 3-4 Finger Print Enrolment Sequence

3.2.4 Update Employee Details

Description

- The Admin selects employee to update details.
- The system returns the selected employee details.
- Admin enters new updated details
- System performs basic validity checks on new details.
- If incorrect, returns error.
- If found correct, Employee details are updated in the system.
- If successfully updated returns success message else returns error message.

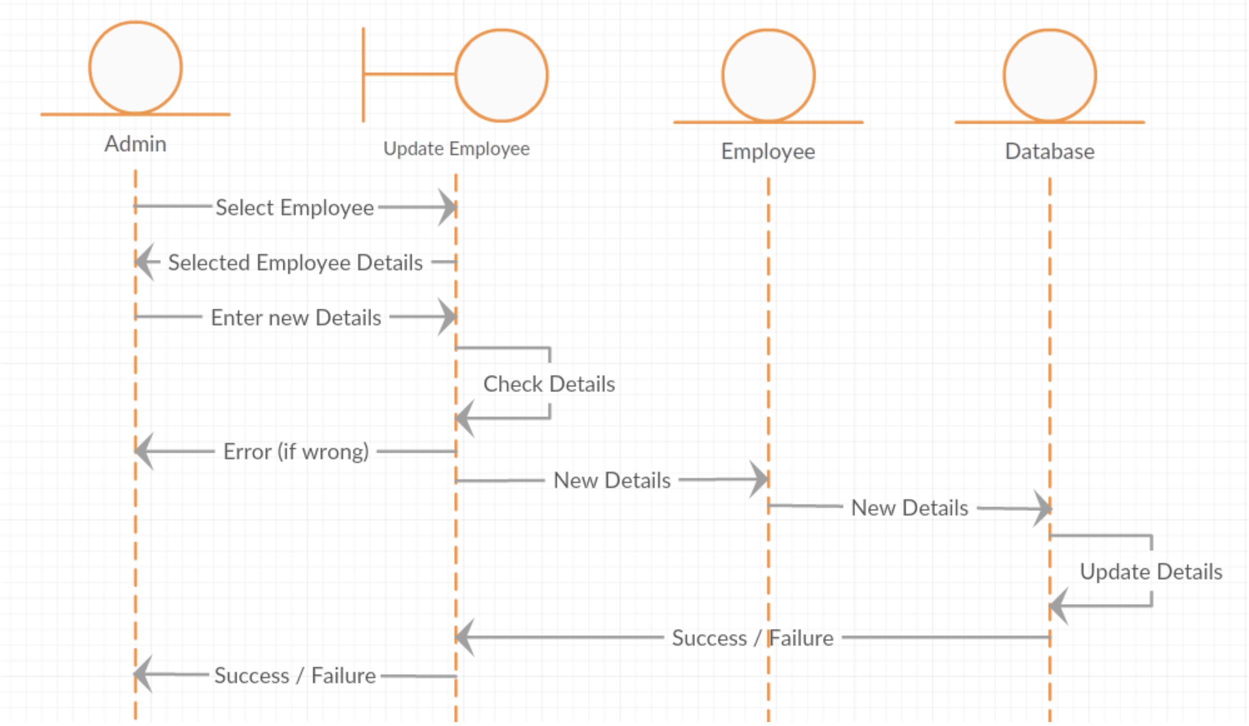


Figure 3-5 Employee Details update Sequence

3.2.5 Delete Employee Details

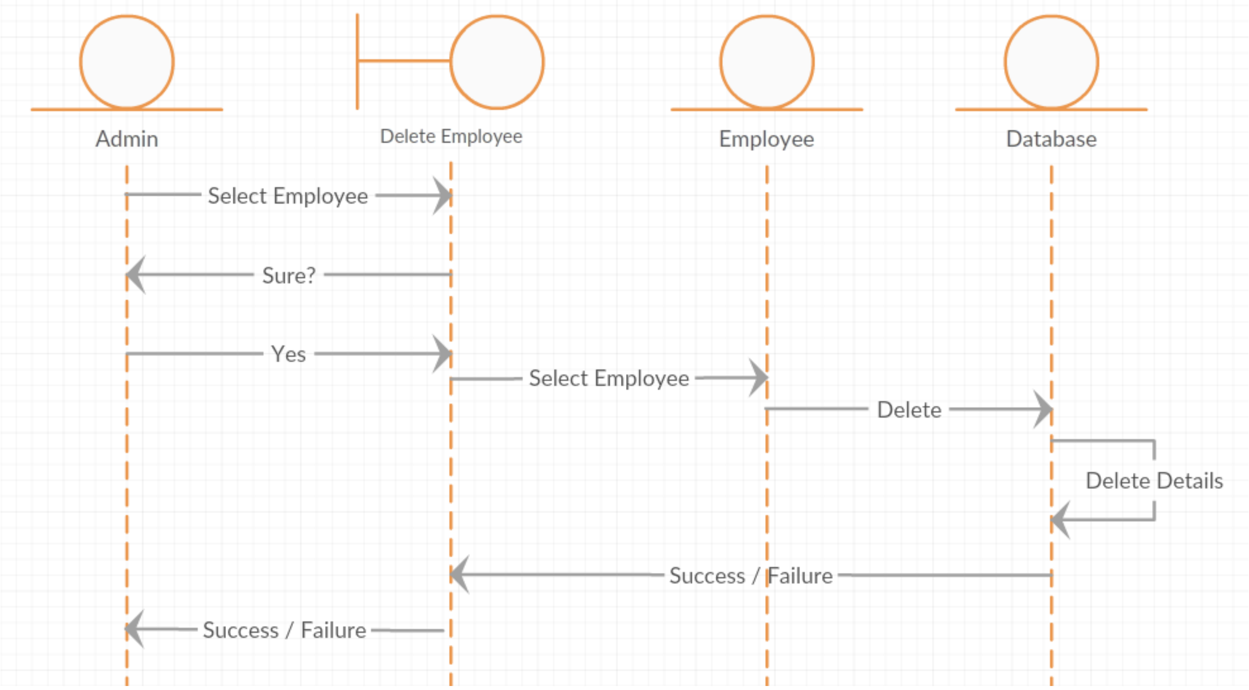


Figure 3-6 Delete Employee Sequence

Description

- The Admin selects employee to delete.
- The system prompts if he is sure or not.
- If sure system selects the employee from database.
- System performs Delete operation.
- If Employee deleted successfully, returns success messages else returns error message.

3.2.6 Salary Portal Admin

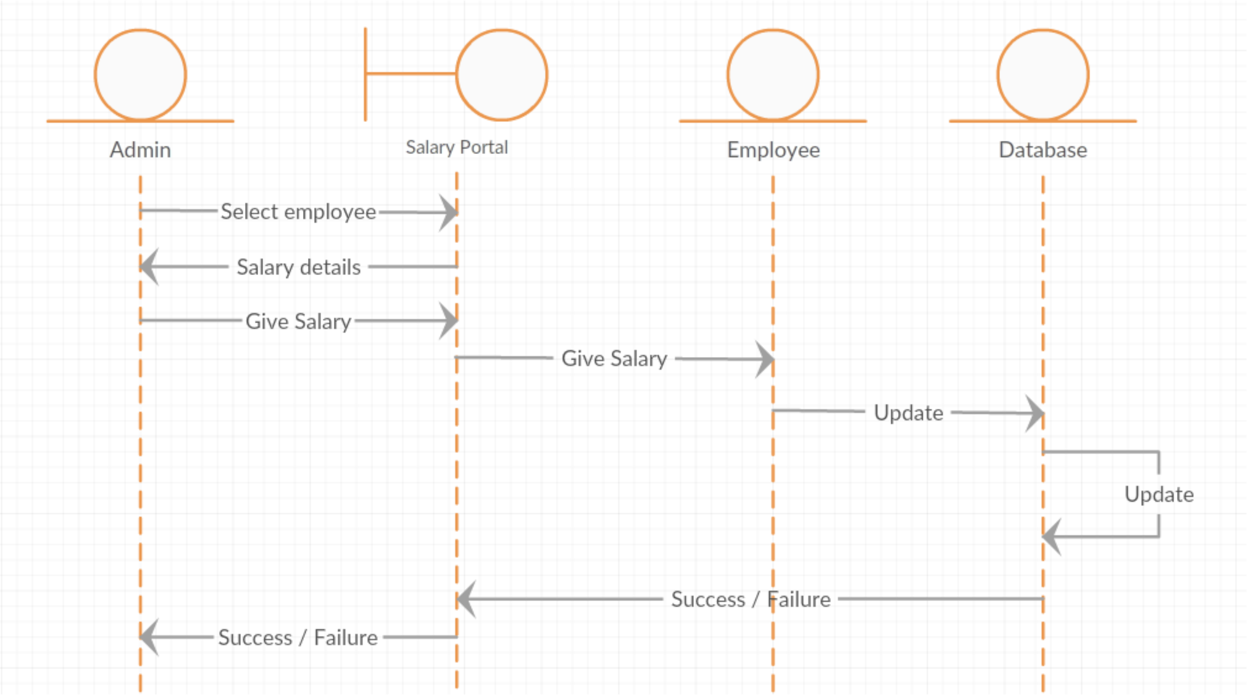


Figure 3-7 Salary portal Admin Sequence

Description

- The Admin selects employee to give salary.
- The system retrieves the salary information of the employee.
- Admin adjusts the details as required and submit it to system.
- System performs Update in the database.
- If successfully updated, returns success messages else returns error message.

3.2.7 Leave Requests Admin Portal

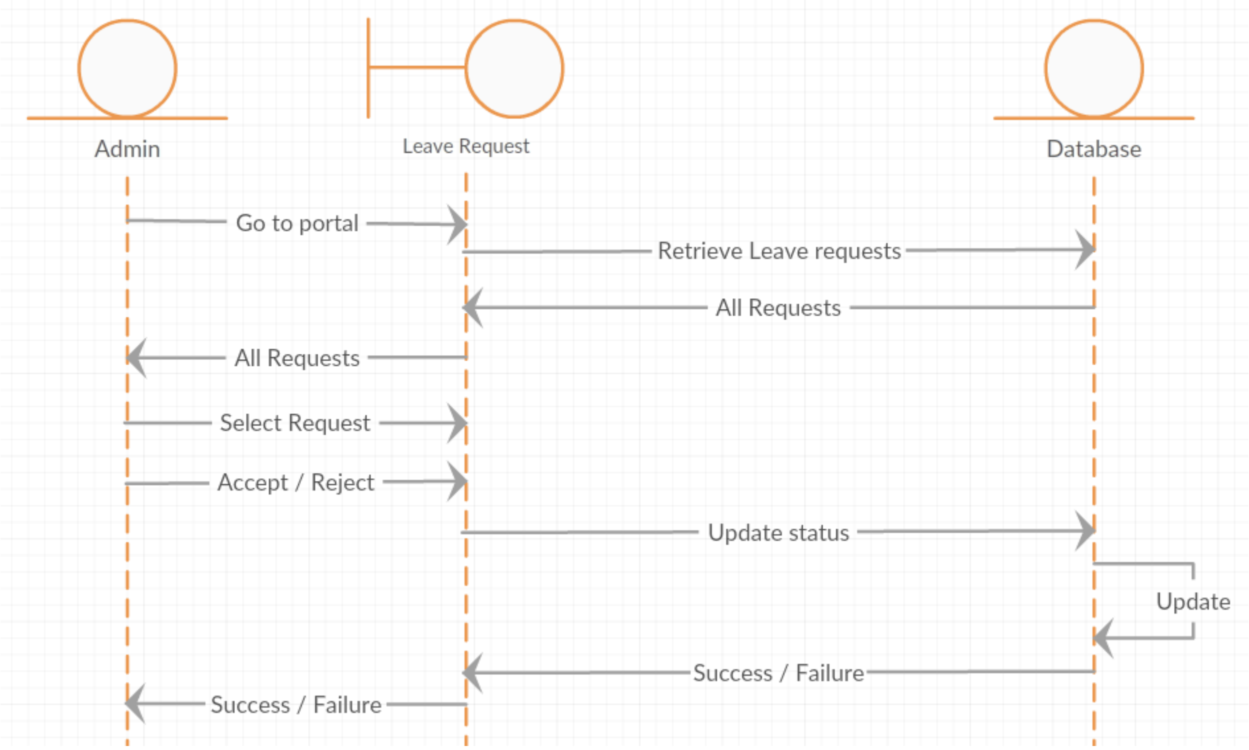


Figure 3-8 Leave request Admin Sequence

Description

- The Admin goes to leave requests portal.
- The system retrieves all leave request and sort them from latest.
- Admin selects request to accept or reject.
- Admin accepts or rejects the request.
- System updates the selected leave request status.
- If successfully updated, returns success messages else returns error message.

3.2.8 Add New Admin

Description

- The Admin enter new admin details.
- The system performs basic checks on details.
- If found incorrect returns error message.
- Else system inserts new admin details into the system.

- If successfully created new admin, returns success messages else returns error message.

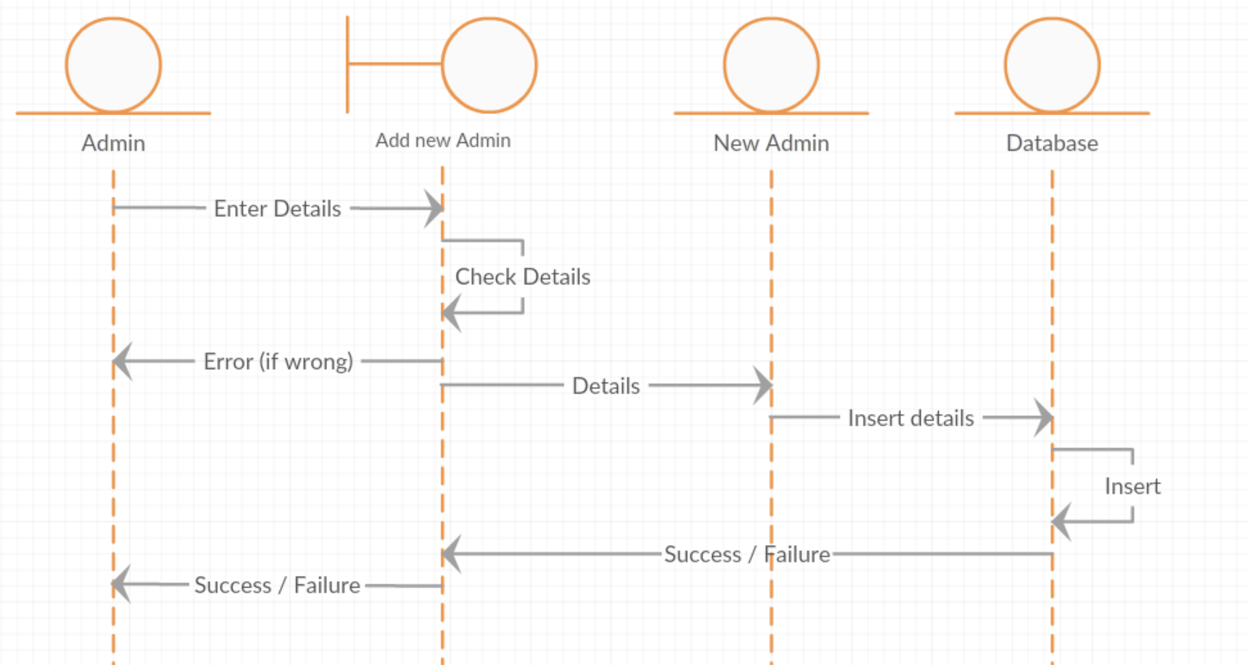


Figure 3-9 Add New Admin Sequence

3.2.9 Attendance Details Check

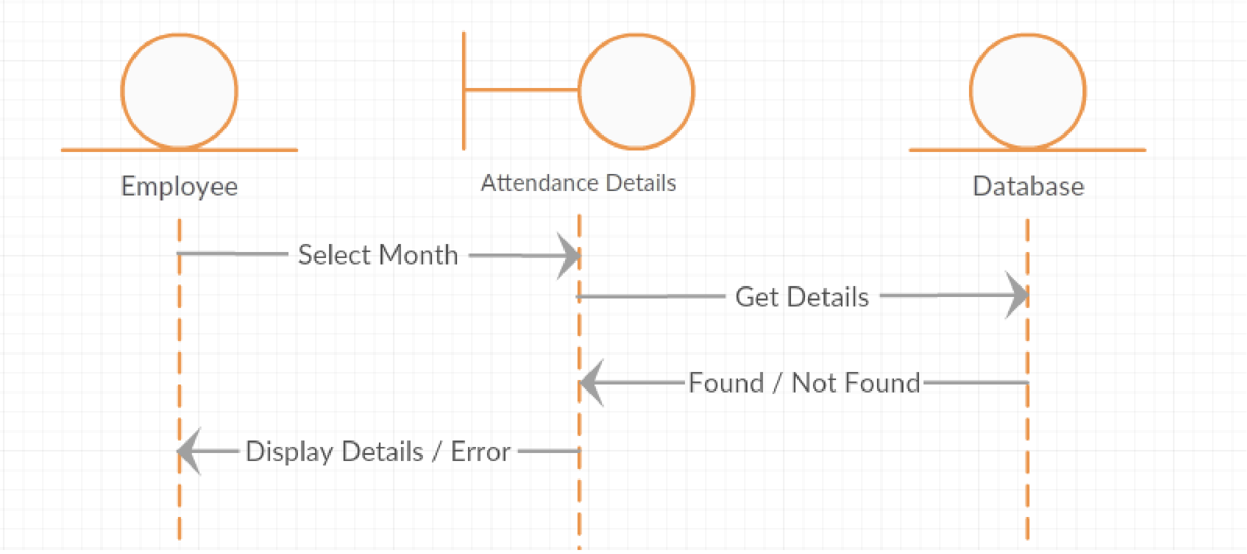


Figure 3-10 Attendance Details Check Sequence

Description

- The Employee selects month for Attendance details.
- The system retrieves the Attendance information for given month.

- If details found, returns salary details else returns error message.

3.2.10 Apply for Leave

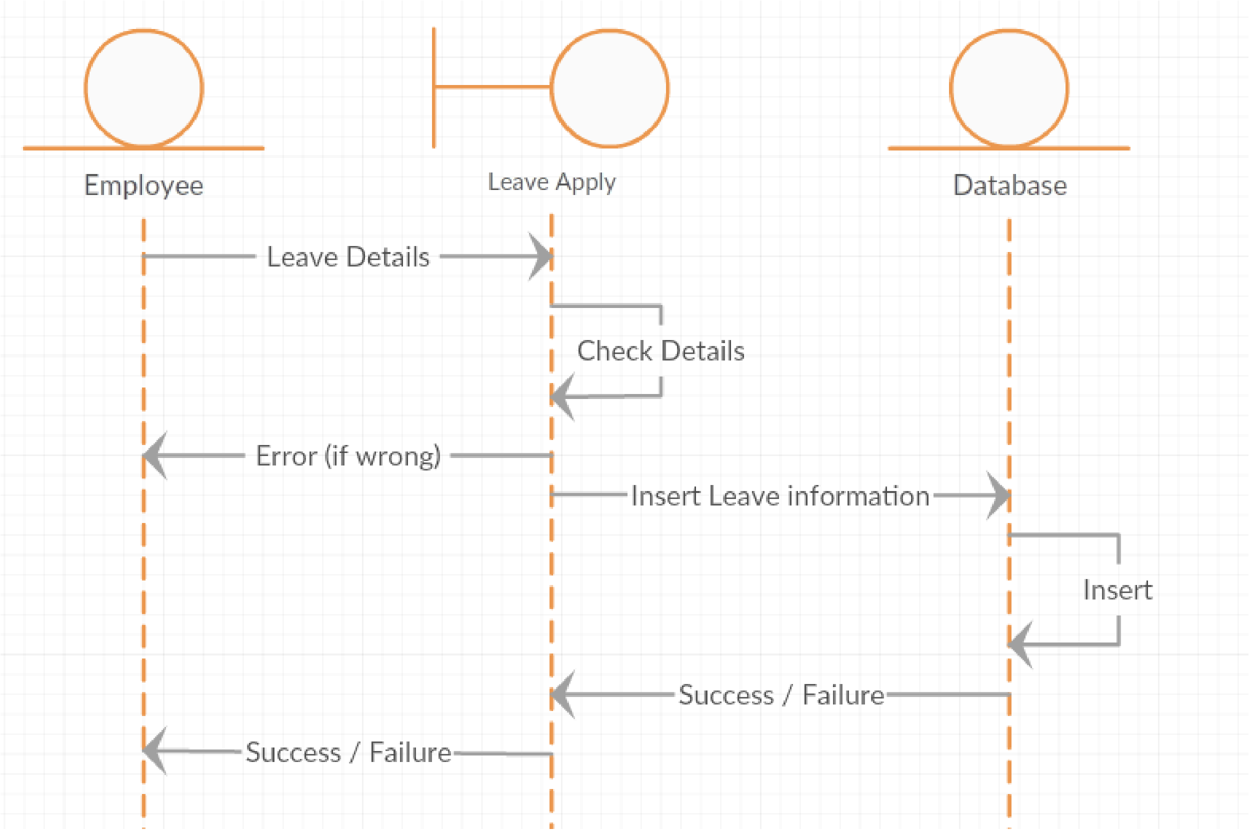


Figure 3-11 Apply for Leave Sequence

Description

- The Employee enters leave details.
- The system checks the entered details.
- If incorrect returns error message.
- Else system inserts the leave request in the system with pending status.
- If Request added successfully, returns success messages else returns error message.

3.2.11 Salary Details Check

Description

- The Employee selects month for salary description.

- The system retrieves the salary information for given month.
- If details found, returns salary details else returns error message.

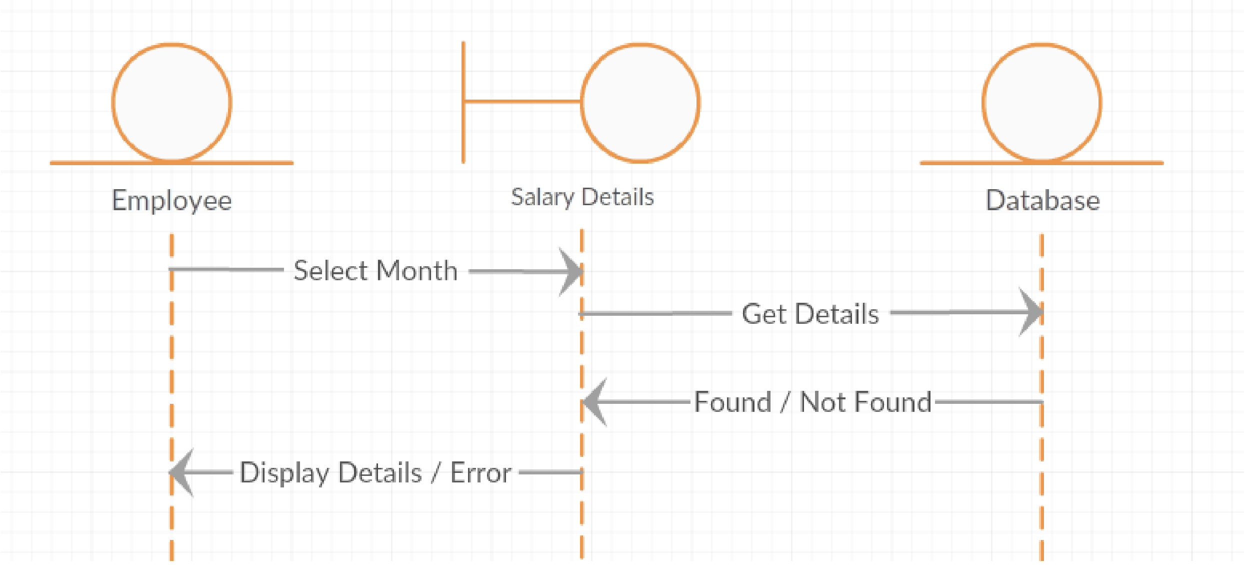


Figure 3-12 Salary Details Check Sequence

3.2.12 Change Password

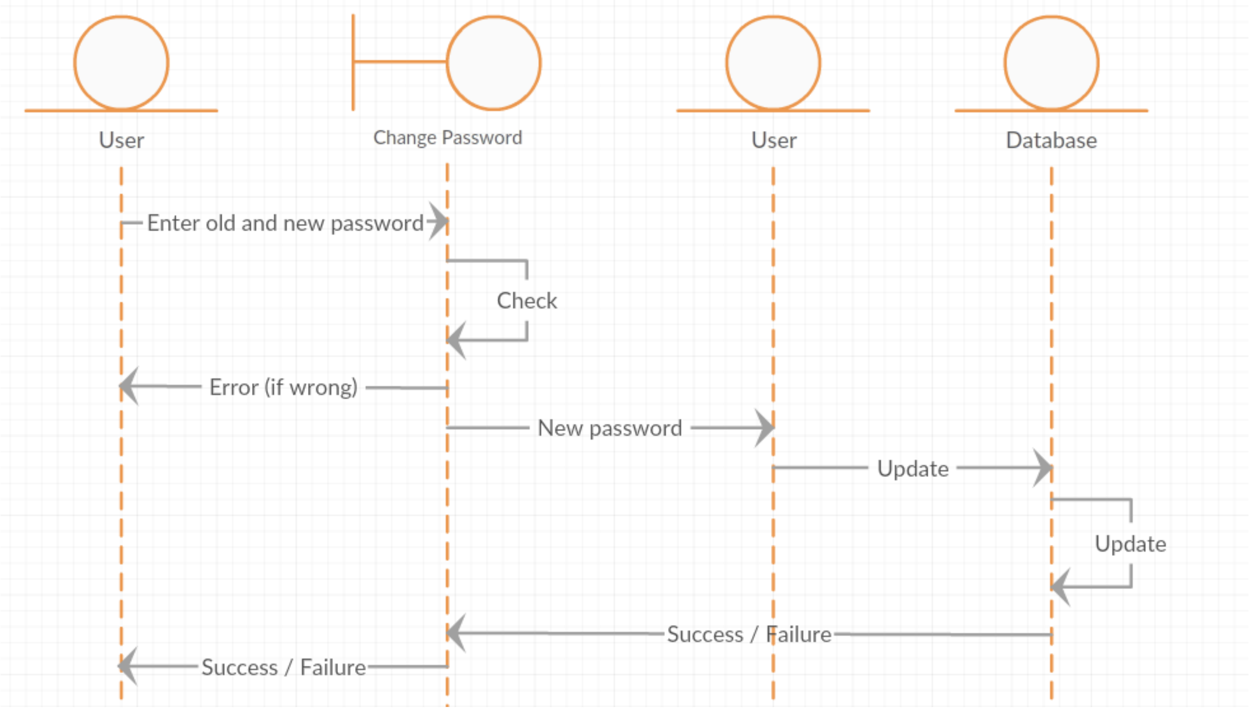


Figure 3-13 Change Password Sequence

Description

- The user enters old and new password.
- The system checks the entered details.
- If incorrect returns error message.
- Else system updates the user password.
- If Updated successfully, returns success messages else returns error message.

3.2.13 Attendance Marking

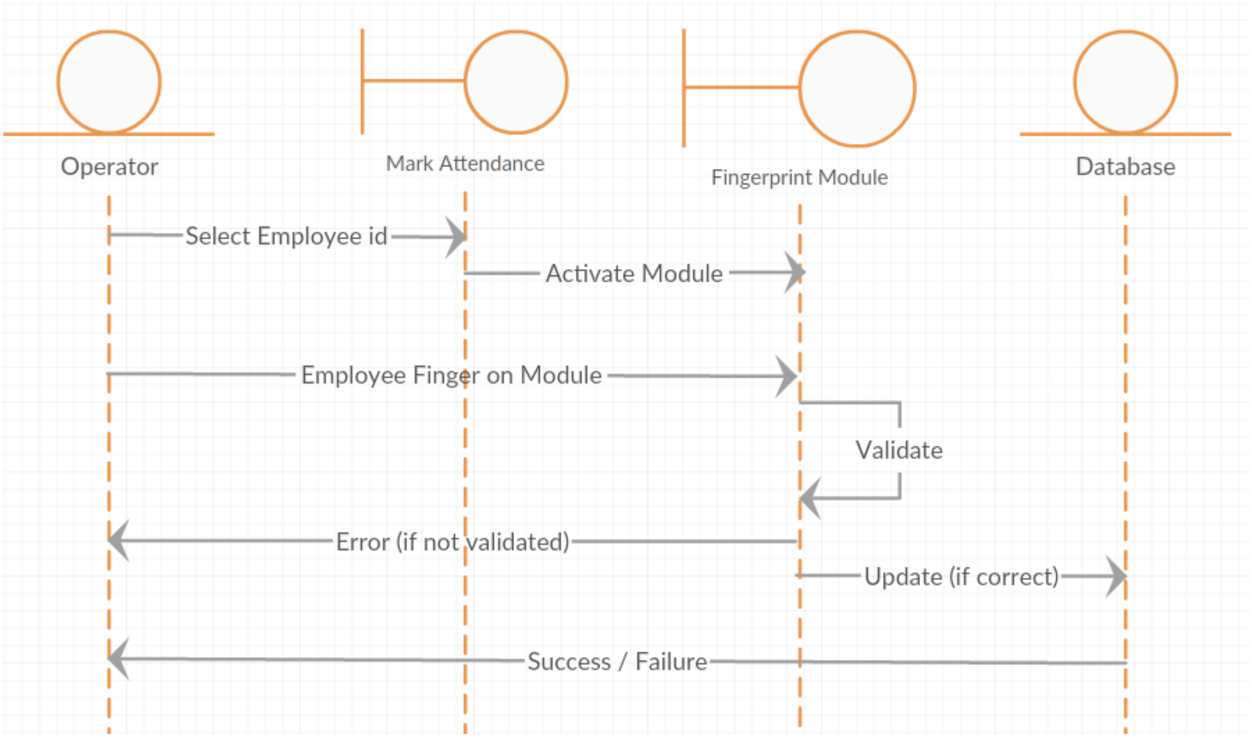


Figure 3-14 Attendance Marking Sequence

Description

- The user selects employee id.
- The system activates module and wait for finger print to be entered.
- User puts finger on the module.
- Module validates the print.
- If incorrect returns error message.
- Else system updates attendance status.
- If Updated successfully, returns success messages else returns error message.

3.3 Database Design

The ER or (Entity Relational Model) is a high-level conceptual data model diagram. Entity-Relation model is based on the notion of real-world entities and the relationship between them. \ ER modeling helps you to analyze data requirements systematically to produce a well-designed database. So, it is considered a best practice to complete ER modeling before implementing your database.

Features of ER Diagram

- ER model allows you to draw Database Design
- It is an easy to use graphical tool for modeling data
- Widely used in Database Design
- It is a GUI representation of the logical structure of a Database
- It helps you to identifies the entities which exist in a system and the relationships between those entities

ER diagram of the project is as follows

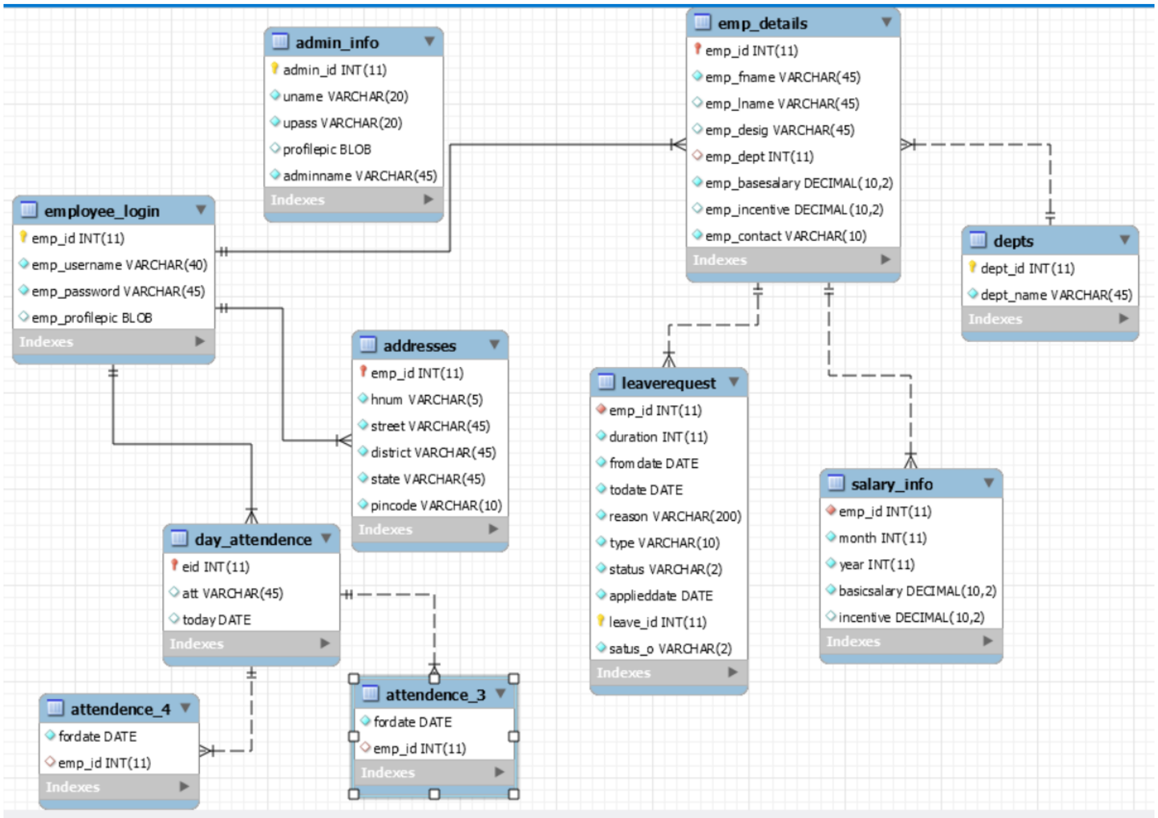


Figure 3-15 ER Diagram

3.4 Hardware Interfacing

As the system has Biometric support, it uses a simple microcontroller enables finger print module for attendance marking. The microcontroller used is an Arduino uno r3 connected with a r305 finger print module [7], [9], [10]. The Arduino communicates with the application using serial connection between the system and Arduino. The module is only activated when it is needed. The following diagram shows the interfacing of Arduino with r305 module.

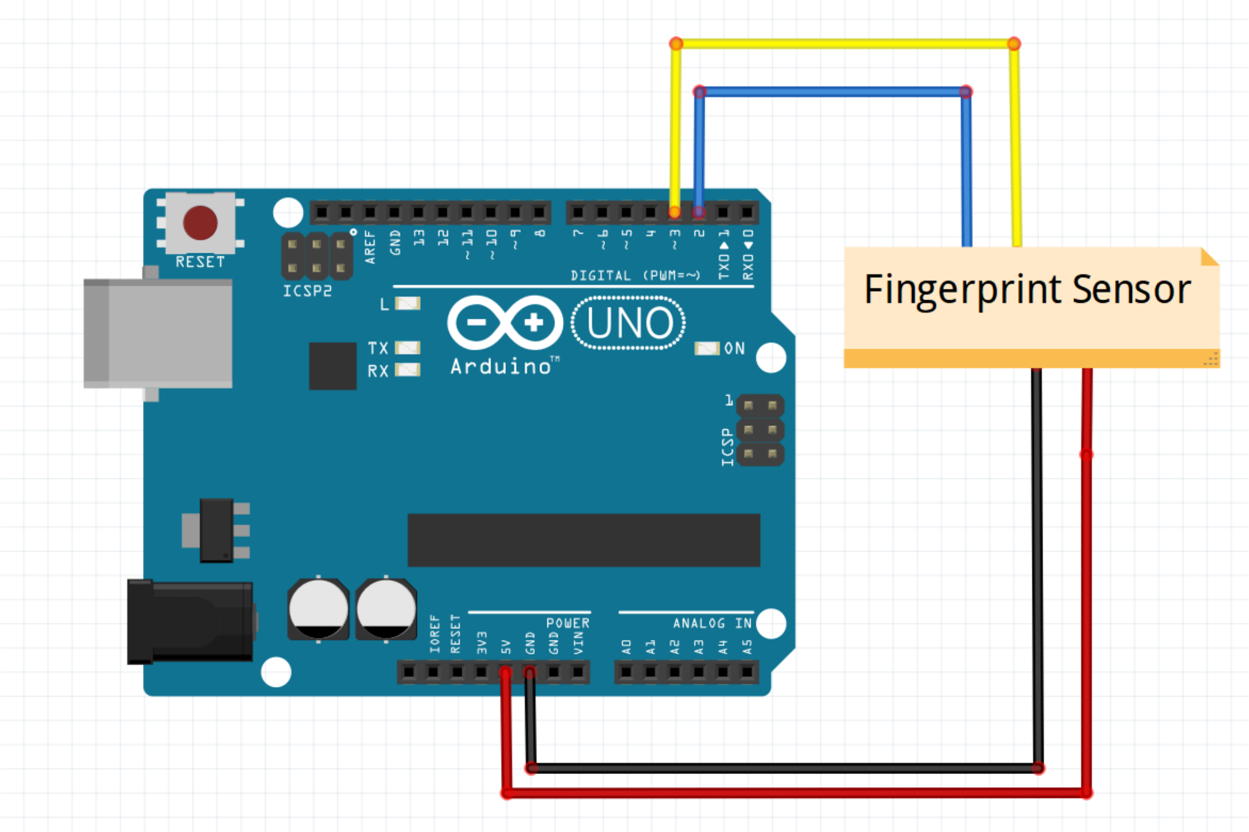


Figure 3-16 Arduino-r305 interfacing

Description

- The red wire is the power supply to the r305 module of 5V.
- The black wire is the ground wire.
- The Blue wire represents RX pin of the module and receives data from Arduino via it.
- The yellow wire represents the TX pin of the module and sends data to the Arduino via it.

Chapter 4 Testing

Software testing is the process of verifying a system with the purpose of identifying any errors, gaps or missing requirement versus the actual requirement. It involves execution of a software component or system component to evaluate one or more properties of interest.

Software testing also helps to identify errors, gaps or missing requirements in contrary to the actual requirements. It can be either done manually or using automated tools. Some prefer saying Software testing as a White Box and Black Box Testing. In this chapter we look at various testing's performed for the project.

4.1 Black Box Testing

Black Box Testing, also known as Behavioral Testing, is a software testing method in which the internal structure/design/implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional.

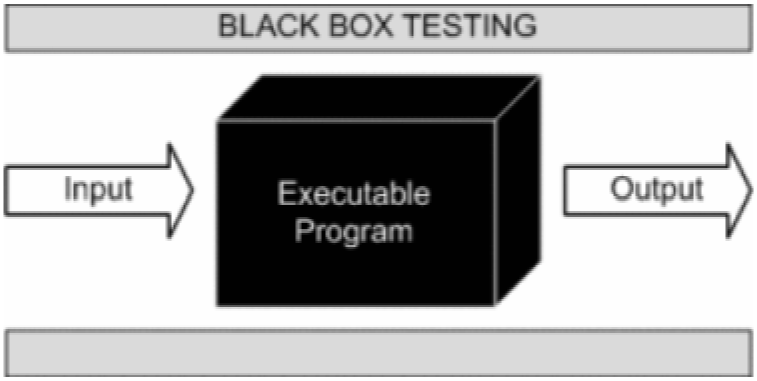


Figure 4-1 Black Box testing

This method is named so because the software program, in the eyes of the tester, is like a black box; inside which one cannot see. This method attempts to find errors in the following categories:

- Incorrect or missing functions
- Interface errors
- Errors in data structures or external database access
- Behavior or performance errors

- Initialization and termination errors

4.1.1 Techniques in Black Box Testing

Following are some techniques that can be used for designing black box tests.

Equivalence Partitioning

It is a software test design technique that involves dividing input values into valid and invalid partitions and selecting representative values from each partition as test data.

Boundary Value Analysis

It is a software test design technique that involves the determination of boundaries for input values and selecting values that are at the boundaries and just inside/ outside of the boundaries as test data.

Cause-Effect Graphing

It is a software test design technique that involves identifying the cases (input conditions) and effects (output conditions), producing a Cause-Effect Graph, and generating test cases accordingly.

4.2 White Box Testing

White Box Testing (also known as Clear Box Testing, Open Box Testing, Glass Box Testing, Transparent Box Testing, Code-Based Testing or Structural Testing) is a software testing method in which the internal structure/design/implementation of the item being tested is known to the tester. The tester chooses inputs to exercise paths through the code and determines the appropriate outputs. Programming know-how and the implementation knowledge is essential. White box testing is testing beyond the user interface and into the nitty-gritty of a system.

This method is named so because the software program, in the eyes of the tester, is like a white/transparent box; inside which one clearly sees.

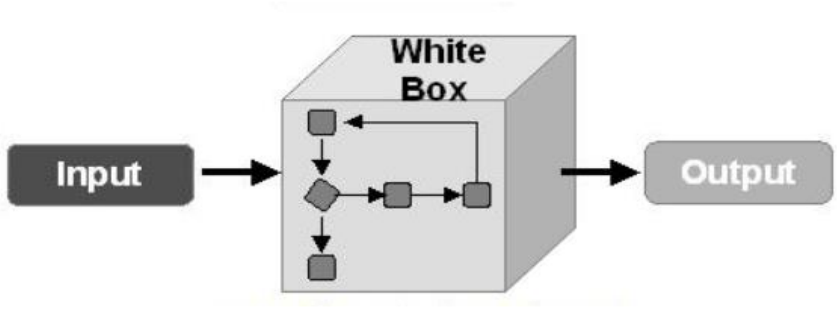


Figure 4-2 White Box Testing

4.3 Hardware Testing

Hardware testing is usually more detailed and thorough than verification and software testing. Testing is needed to ensure that every component of a system is operating as it should, and that the system is performing exactly in accordance with the specific local requirements.

Testing measures that are followed include for the given semi-autonomous car are as follows.

- applying 'non-operating' tests to ensure that equipment can stand up to expected levels of physical handling, such as transit drop tests
- examining if appropriate any code 'hard wired' in hardware (this code is sometimes known as firmware) to ensure its logical correctness and to ensure that appropriate standards are followed
- applying functional tests to determine whether the test criteria have been met
- applying qualitative assessments to determine whether the test criteria have been met
- conducting 'load tests', simulating as closely as possible a variety of 'real life' conditions and using or exceeding the amounts of data that could be expected in an actual situation
- verifying that 'what goes in' is 'what comes out', by entering known data and checking that the output agrees with the input

4.4 Testing of the Project

As the whole system is Object Oriented in nature, we have opted to perform testing by developing test cases such that they traverse and pop up every error that is there in the project exposing maximum bugs. These are discussed below

4.4.1 Functionality Testing

Functionality involves to check the functions of our system whether it is working properly or not. It involves the following

- Wrong inputs to perform negative testing.
- Checking that the system does not insert incorrect data in database.
- Verify all UI elements are reachable and functional.
- Verify external hardware is interfaced correctly.
- Verify incorrect and no inputs are not accepted by the system.

- Verify suitable error messages are shown to the user.

Table 4-1 shows the test cases for login screen which ensures that no unauthorized access is given to any user. Table 4-2 shows the testing for attendance marking system ensuring marking of attendance is done properly without any spoofing. Table 4-3 show testing of addition of employee screen which ensures no incorrect information is entered about an employee. Table 4-4 shows testing of leave request screen ensuring no invalid leave request is made.

Test Cases for Login Screen

Test Case ID	Inputs			Expected Result	Actual Result	Pass/Fail
	username	password	type			
TC-01	-	-	admin	Error	Error	Pass
TC-02	-	-	employee	Error	Error	Pass
TC-03	admin	-	admin	Error	Error	Pass
TC-04	admin	admin	admin	Success	Success	Pass
TC-05	admin	admin	employee	Error	Error	Pass
TC-06	sahil123	sahil	employee	Success	Success	Pass
TC-07	sahil123	-	employee	Error	Error	Pass
TC-08	sahil123	sahil	admin	Error	Error	Pass

Table 4-1 Test Cases for Login Screen

Test Cases for Attendance Marking

Test Case ID	Inputs			Expected Result	Actual Result	Pass/Fail
	Employee id	Finger Print	Time			
TC-01	-	-	-	Error	Error	Pass
TC-02	1	-	before 8 am	Error	Error	Pass
TC-03	1	-	after 10 am	Error	Error	Pass
TC-04	1	correct	8 am to 10 am	Success	Success	Pass
TC-05	2	incorrect	8 am to 10 am	Error	Error	Pass

Table 4-2 Test Cases for Attendance Marking

Test Cases of Addition of Employee Details

Test Case ID	Inputs						Expected Result	Actual Result	Pass/Fail
	First name	Username	Department	salary	incentive	Contact Number			
TC-01	-	User	Accounts	20000	1000	9896198961	Error	Error	Pass
TC-02	user	-	Sales	50000	3000	9896198961	Error	Error	Pass
TC-03	user	user1	-	60000	5000	9897198961	Error	Error	Pass
TC-04	user	user1	Develop	-	4000	9896198871	Error	Error	Pass
TC-05	user	user1	Accounts	80000	-	9896568961	Error	Error	Pass
TC-06	user	user1	Sales	60000	4000	-	Error	Error	Pass
TC-07	user	user1	Develop	20000	6000	abcder	Error	Error	Pass
TC-08	user	user1	Develop	-100	6000	9896194661	Error	Error	Pass
TC-09	user	user1	Develop	20000	-1233	9896194661	Error	Error	Pass
TC-10	user	user1	Develop	40000000	6000	9896194661	Error	Error	Pass
TC-11	user	user1	Develop	40000	400000	9896194661	Error	Error	Pass
TC-12	user	user1	Develop	40000	2000	9896194661	Success	Success	Pass

Table 4-3 Test Cases for Adding Employee

Test Cases for Applying Leave Requests

Test Case ID	Inputs				Expected Result	Actual Result	Pass/Fail
	Leave Duration	From date	To date	Reason			
TC-01	-	4/14/2019	4/15/2019	Work purpose	Error	Error	Pass
TC-02	Single Day	-	4/15/2019	Work purpose	Error	Error	Pass
TC-03	Single Day	4/14/2019	-	Work purpose	Error	Error	Pass
TC-04	Single Day	4/15/2019	10/15/2019	Work purpose	Error	Error	Pass
TC-05	Single Day	4/16/2019	4/17/2019	-	Error	Error	Pass
TC-06	Single Day	4/16/2019	4/17/2019	Work purpose	Success	Success	Pass
TC-07	Multiple Days	-	4/20/2019	Work purpose	Error	Error	Pass
TC-08	Multiple Days	4/16/2019	-	Work purpose	Error	Error	Pass
TC-09	Multiple Days	4/16/2019	4/20/2019	-	Error	Error	Pass
TC-10	Multiple Days	4/16/2019	4/20/2019	Work purpose	Success	Success	Pass

Table 4-4 Test Cases for Applying Leave Requests

4.4.2 Hardwire Code Testing

There are many components of the code that make this system. The code consist to interaction with the Digital pins, the use fingerprint sensors and serial communication with Arduino all are important aspects of the system. All these are thoroughly tested to ensure minimum faults in the working of the system.

Interaction with Digital pins

The digital pins are major part of the Arduino and the system. The pins are used to send appropriate signals to the finger print sensor as well as to the serial communication channel. Any reading from the sensor or to the sensor are sent via the digital pins. To ensure correct signal is sent to the correct point in the sensor or component, the points are hardwired in the code itself.

Chapter 5 Results

This chapter deals with the resultant system and shows various parts of the system and their functionality respectively. This chapter can be used by the end user as a user manual to navigate through the software and make him/her acquitted with the system.

5.1 Login / Welcome Screen

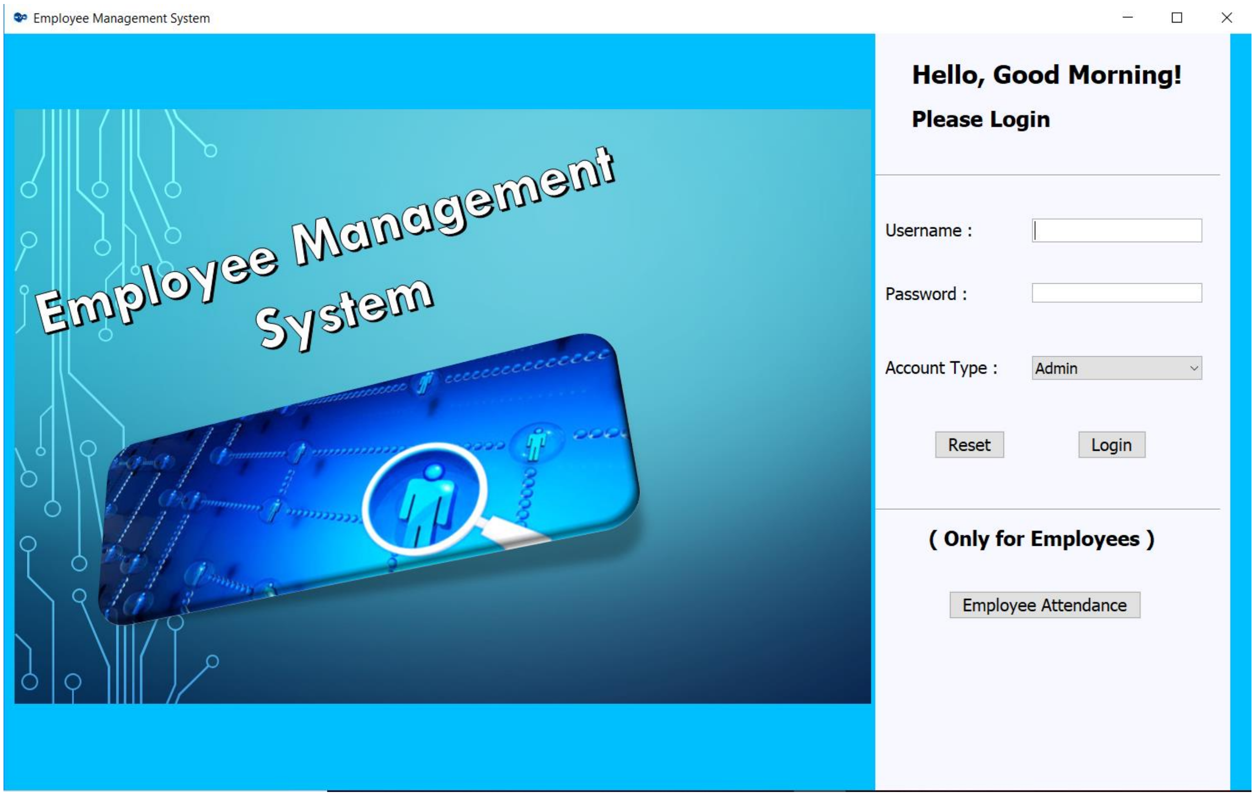
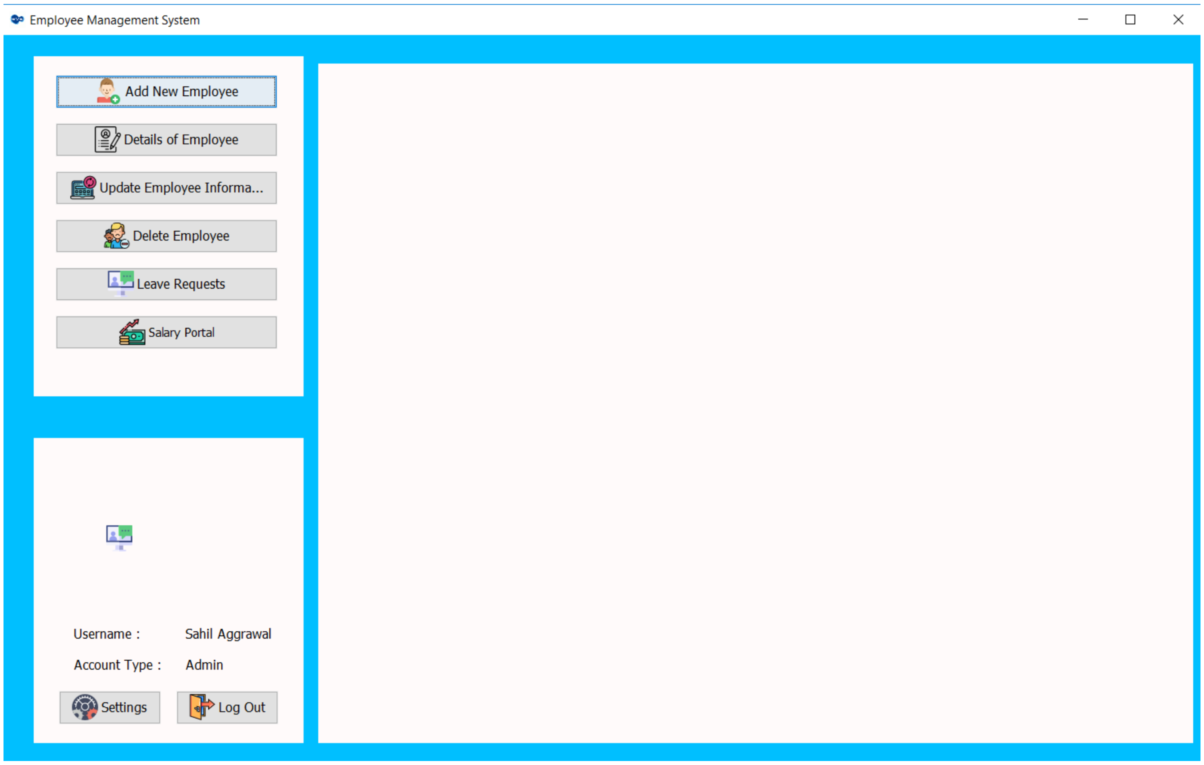


Figure 5-1 Login Screen

- This is the welcome/login screen of the project.
- In this frame the administrator as well as employee can login.
- Employee Attendance portal opens up by clicking on the “Employee Attendance” button.
- Wish Label is dynamic and wishes according to the time of the system i.e. “Good Morning” between 6AM-12AM, “Good Afternoon” between 12PM-4PM and “Good Evening” between 4PM-8PM.
- The Combo Box can be used to switch between the account types either “Admin” or “Employee”.

5.2 Admin Portal



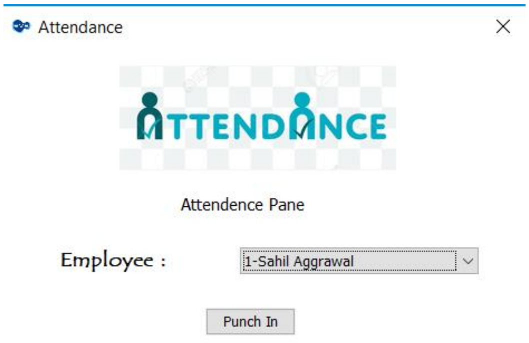
Username : Sahil Aggrawal

Account Type : Admin

Figure 5-2 Admin Portal

- This is the admin screen in which the employee can perform various administrative tasks
- It contains various buttons which on clicking open different frames connected to it.
- It also contains panel in which administrator profile image, logout and settings button is shown.

5.3 Attendance Portal

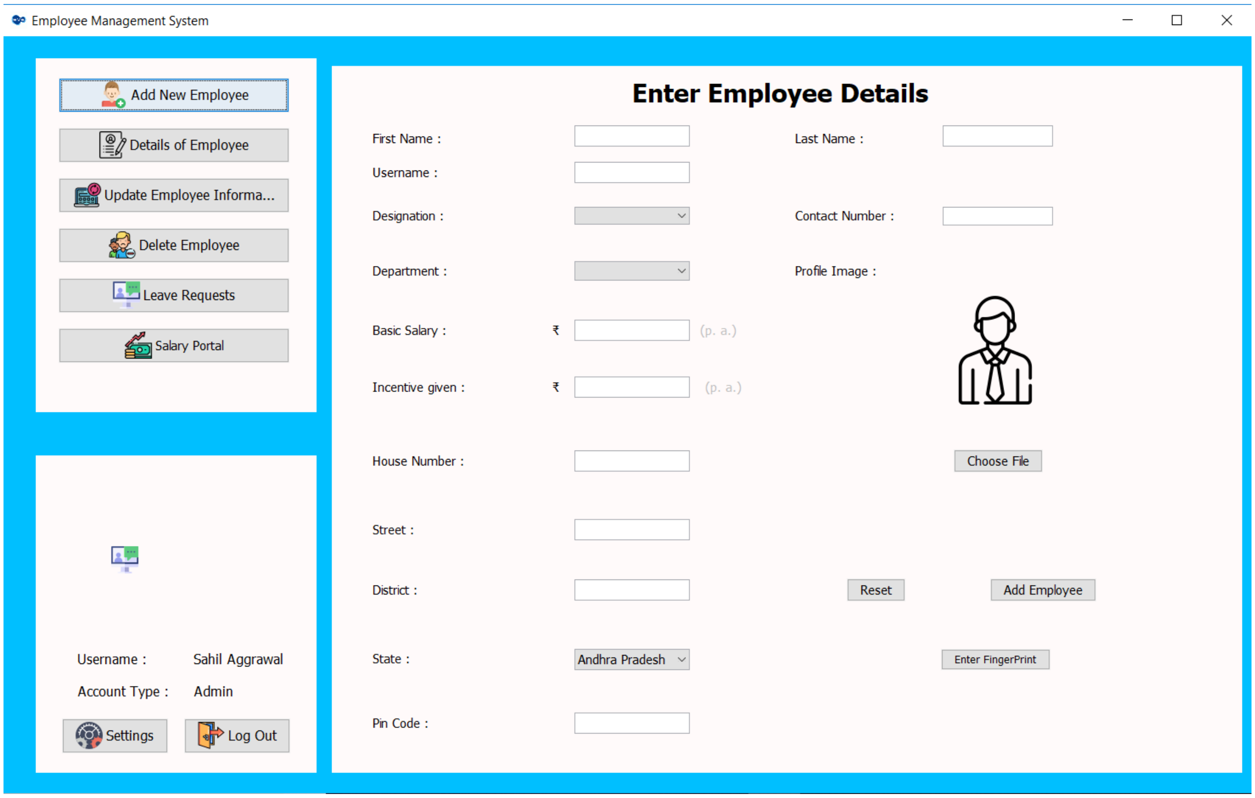


Employee : 1-Sahil Aggrawal

Figure 5-3 Attendance Portal

- This is the Attendance Portal Screen.
- Attendance portal automatically creates new table when month changes or year changes.
- It also updates the number of working days respective to each month according to which percentage attendance is counted for that respective month.
- Employee can punch his/her attendance by selecting their respective employee ID from the drop-down menu and clicking on “Punch In” button and then proceeding to with fingerprint module.

5.4 Add Employee Screen



Username : Sahil Aggrawal

Account Type : Admin

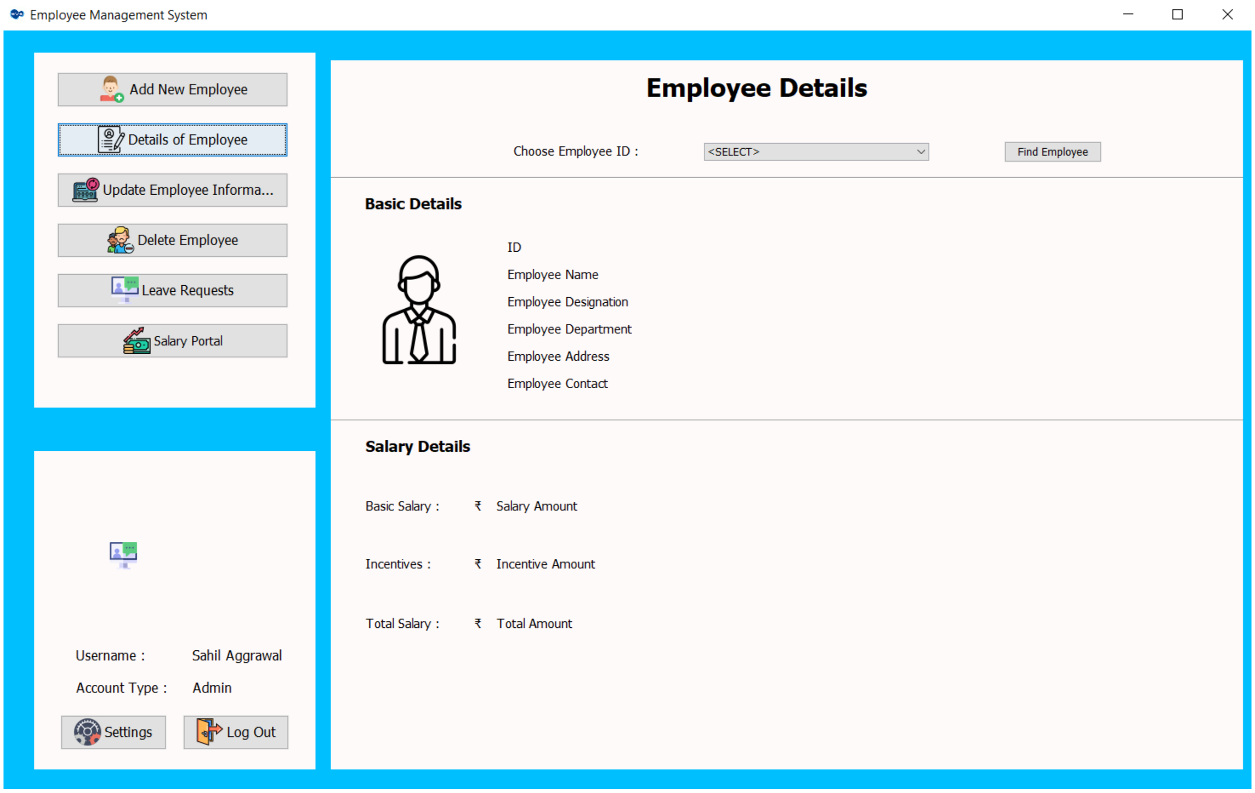
State : Andhra Pradesh

Figure 5-4 Add New Employee

- In this frame you can add new employee.
- It contains various text fields which are to be filled in with the appropriate data.
- Some of the fields are optional to be filled
- Administrator can also add employee image.

- Employee details mentioned are stored in the respective database.

5.5 Employee Details Screen



Employee Details

Choose Employee ID : <SELECT>

Basic Details

ID
Employee Name
Employee Designation
Employee Department
Employee Address
Employee Contact

Salary Details

Basic Salary : ₹ Salary Amount

Incentives : ₹ Incentive Amount

Total Salary : ₹ Total Amount

Username : Sahil Aggrawal

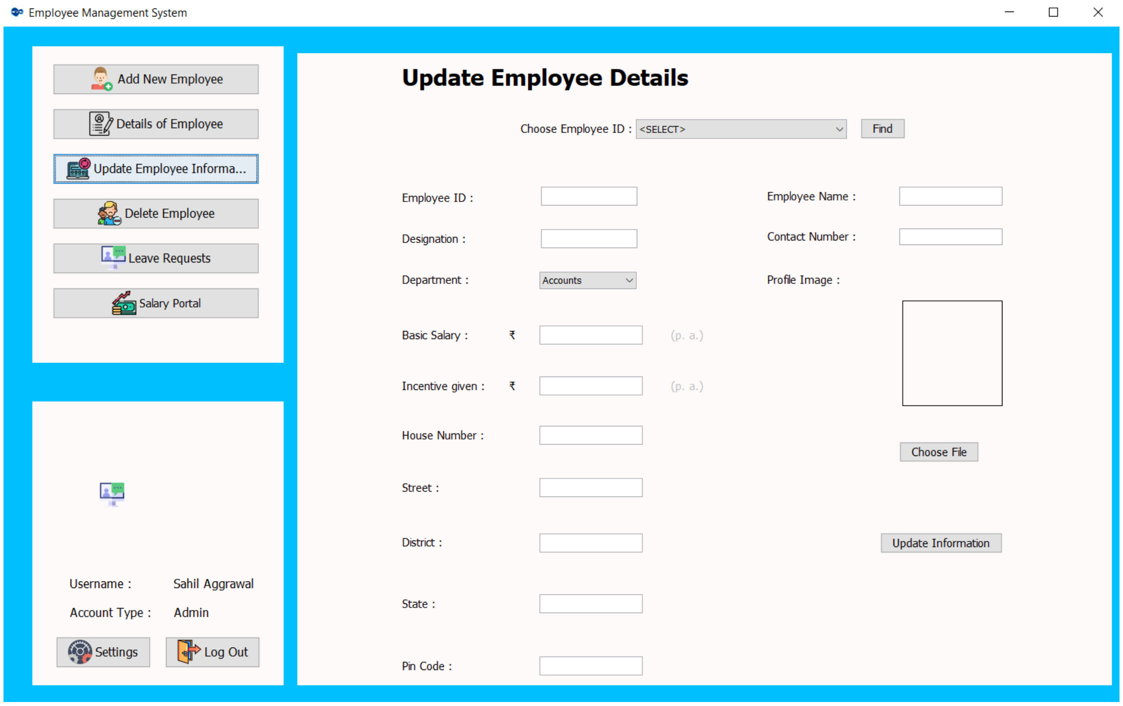
Account Type : Admin

Figure 5-5 Employee Details

- In this frame you can retrieve the details of any employee stored in the database.
- Salary Statement for the current month and all other is also shown in this frame only.
- Travelling allowance is calculated according to the attendance of the employee for that respective month.

5.6 Update Employee Details

- This frame is used to update the employee details.
- You can change all the details of an employee excluding Employee ID and Employee Name.
- You can also update the profile picture of the employee.
- If profile picture is not set then it is automatically set to default employee icon.



Choose Employee ID : <SELECT>

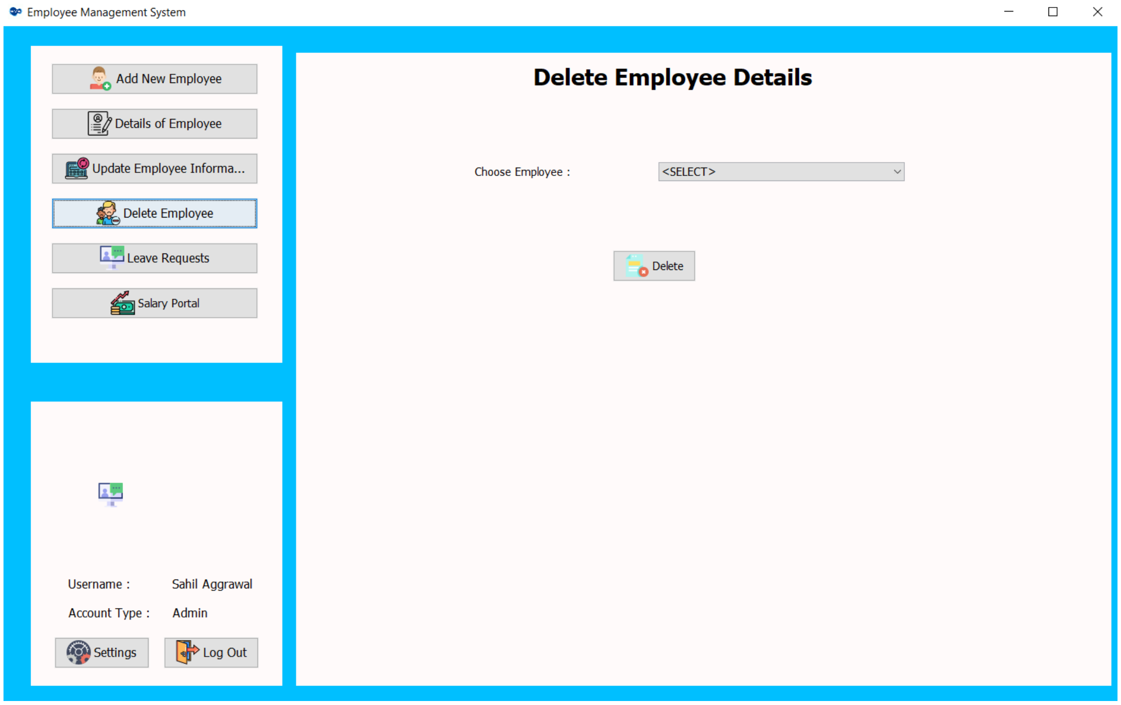
Department : Accounts

Username : Sahil Aggrawal

Account Type : Admin

Figure 5-6 Update Employee Details

5.7 Delete Employee Screen



Choose Employee : <SELECT>

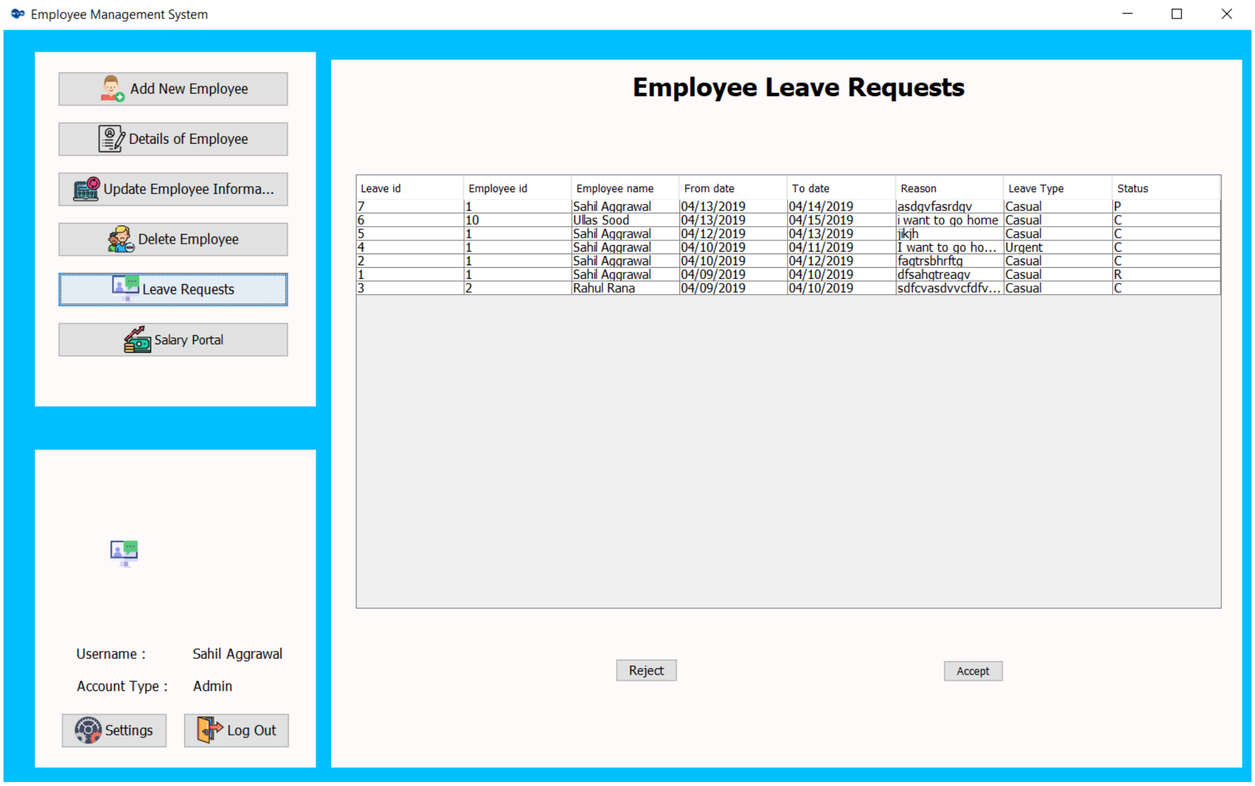
Username : Sahil Aggrawal

Account Type : Admin

Figure 5-7 Delete Employee

- This frame is used to delete employee’s information.
- By selecting respective employee from drop-down menu and clicking on “Delete” button.

5.8 Employee Leave Request Screen



Employee Leave Requests

Leave id	Employee id	Employee name	From date	To date	Reason	Leave Type	Status
7	1	Sahil Aggrawal	04/13/2019	04/14/2019	asdgvfasrdgv	Casual	P
6	10	Ullas Sood	04/13/2019	04/15/2019	i want to go home	Casual	C
5	1	Sahil Aggrawal	04/12/2019	04/13/2019	jikjh	Casual	C
4	1	Sahil Aggrawal	04/10/2019	04/11/2019	I want to go ho...	Urgent	C
2	1	Sahil Aggrawal	04/10/2019	04/12/2019	fagtrsbhrftg	Casual	C
1	1	Sahil Aggrawal	04/09/2019	04/10/2019	dfsahgtreagv	Casual	R
3	2	Rahul Rana	04/09/2019	04/10/2019	sdfcvasdvvcfdfv...	Casual	C

Username : Sahil Aggrawal

Account Type : Admin

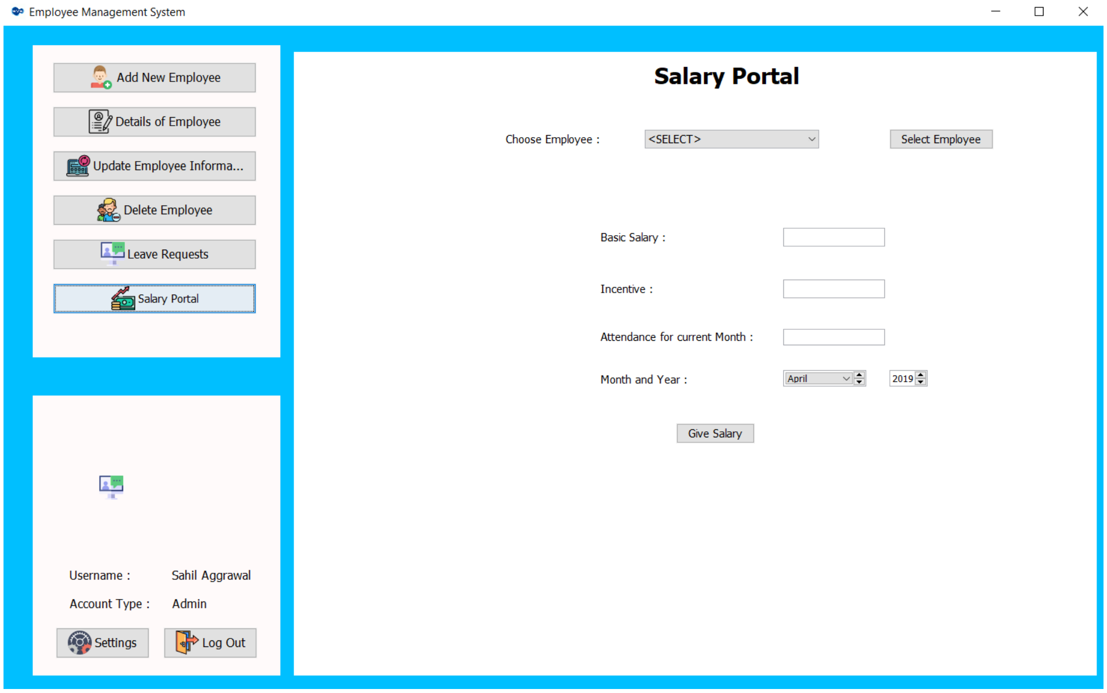
Figure 5-8 Leave Requests Screen

- This frame displays all the leave requests made by employees till date in order where latest are shown on the top.
- The admin can select any leave request that does not have its status set to ‘C’.
- The admin can make the choice to accept or reject the leave request

5.9 Salary Portal

- The admin can enter salary information for an employee from here.
- Admin selects the employee and all his/her details are retrieved from the database and shown to the admin
- The admin can then make a choice to alter the salary based on the Attendance record for the month.

- After all the changes admin has made, the salary will be added to the database.



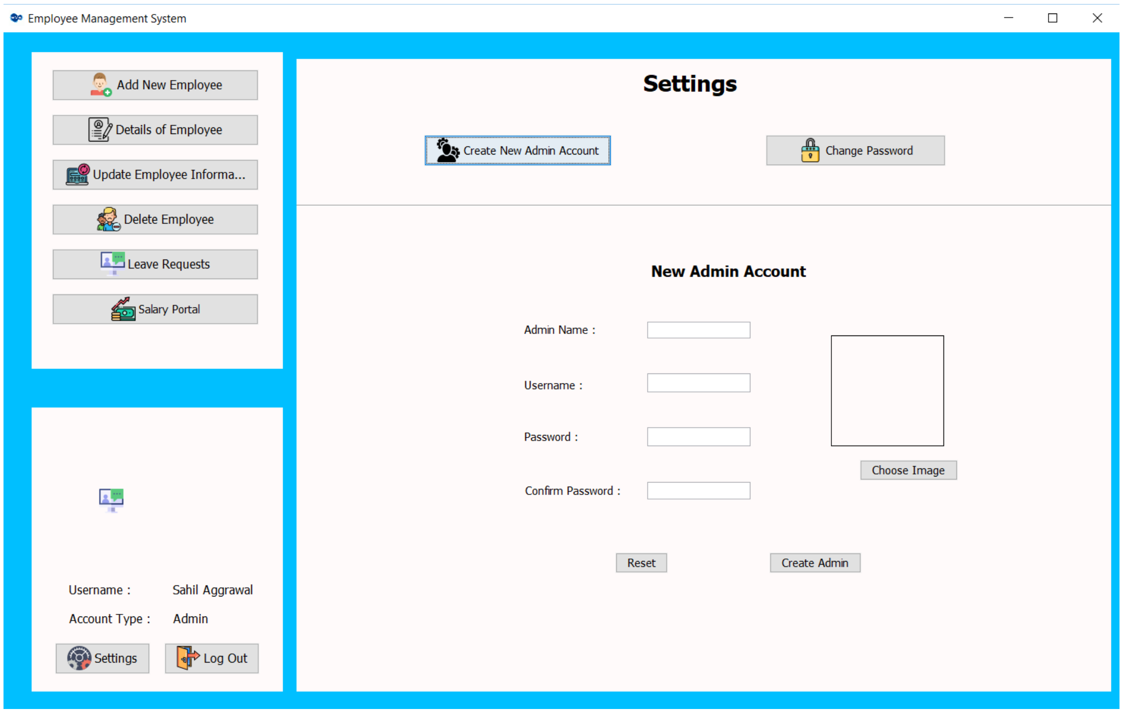
Username : Sahil Aggrawal

Account Type : Admin

Month and Year : April 2019

Figure 5-9 Salary Portal

5.10 Create New Admin Account



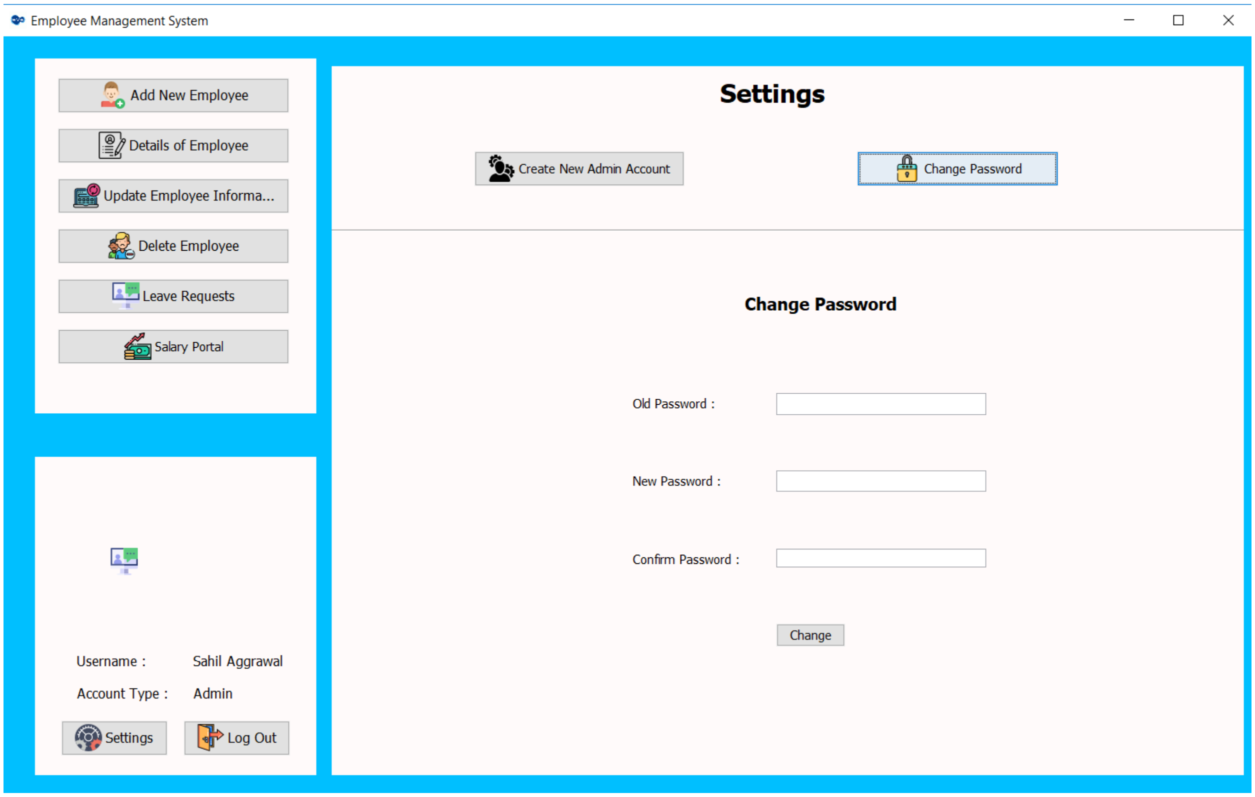
Username : Sahil Aggrawal

Account Type : Admin

Figure 5-10 Create new Admin account

- In this frame administrator can create new administrator account.
- It also contains re-captcha so that security measure is taken care of.
- Strength of new password is also checked.
- New password should be at least of 8 characters and contain at least special digit, at least 1 numerical digit.

5.11 Change Password



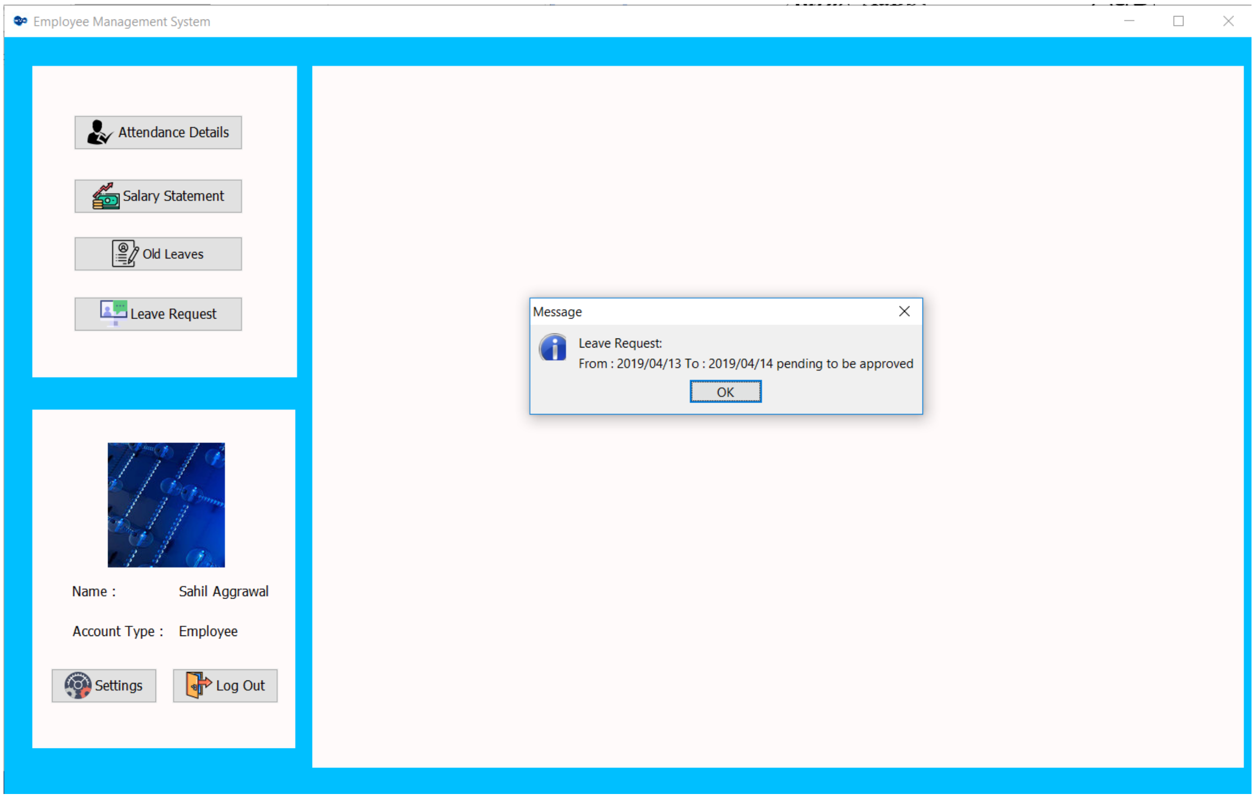
Username : Sahil Aggrawal

Account Type : Admin

Figure 5-11 Change admin Password

- In this frame administrator can change account password.
- New password should be at least of 8 characters and contain at least special digit, at least 1 numerical digit.
- Password entered in old password text field should match the password stored in the database.
- Password gets change by clicking on the “Change” button.

5.12 Employee Portal



Leave Request:
From : 2019/04/13 To : 2019/04/14 pending to be approved

Name : Sahil Aggrawal

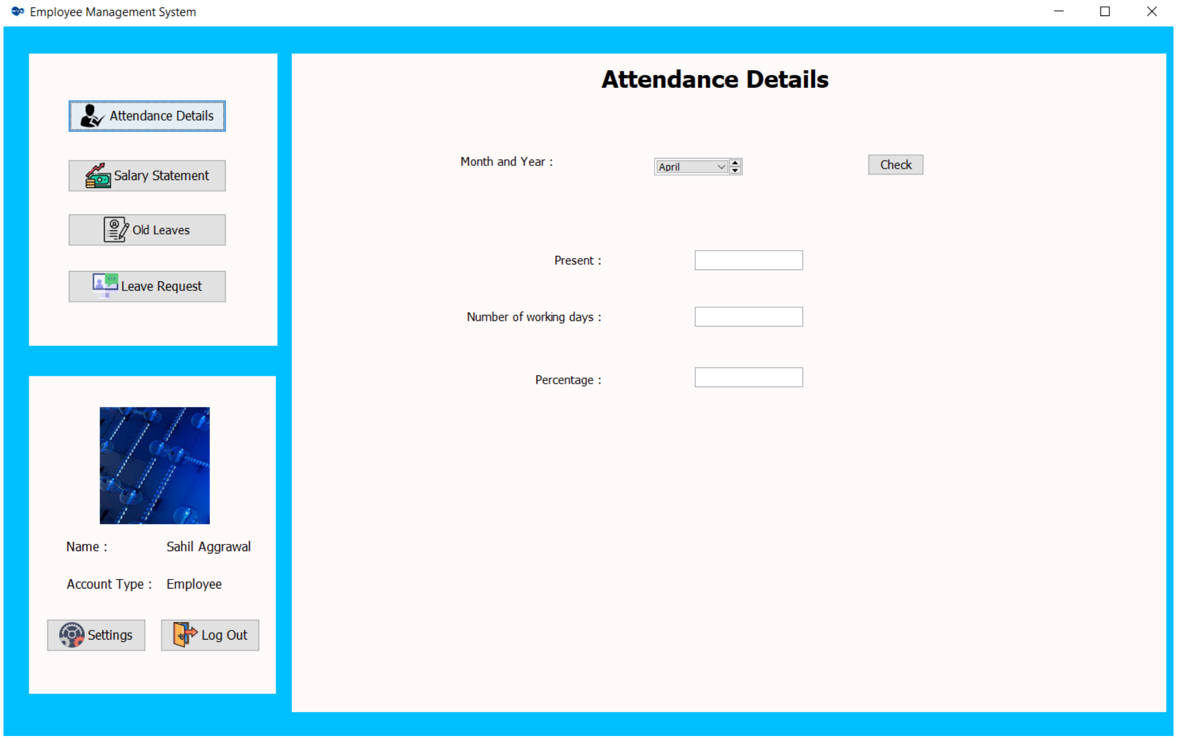
Account Type : Employee

Figure 5-12 Employee Portal

- This is the frame which is shown when an employee login in the application.
- This frame contains various buttons through which employee can perform various task accessible to him/her.
- As the employee logs in, if he or she has entered any leave requests, the status of that request is shown to the employee.
- Once the leave request has been accepted or rejected this pop-up message is disabled.
- “Logout” button is used to logout from the employee portal back to login /welcome screen.

5.13 Attendance Details

- This frame shows the attendance details of the respective employee.
- Employee can check his/her details for the current month and past months also.



Attendance Details

Month and Year : April

Present :

Number of working days :

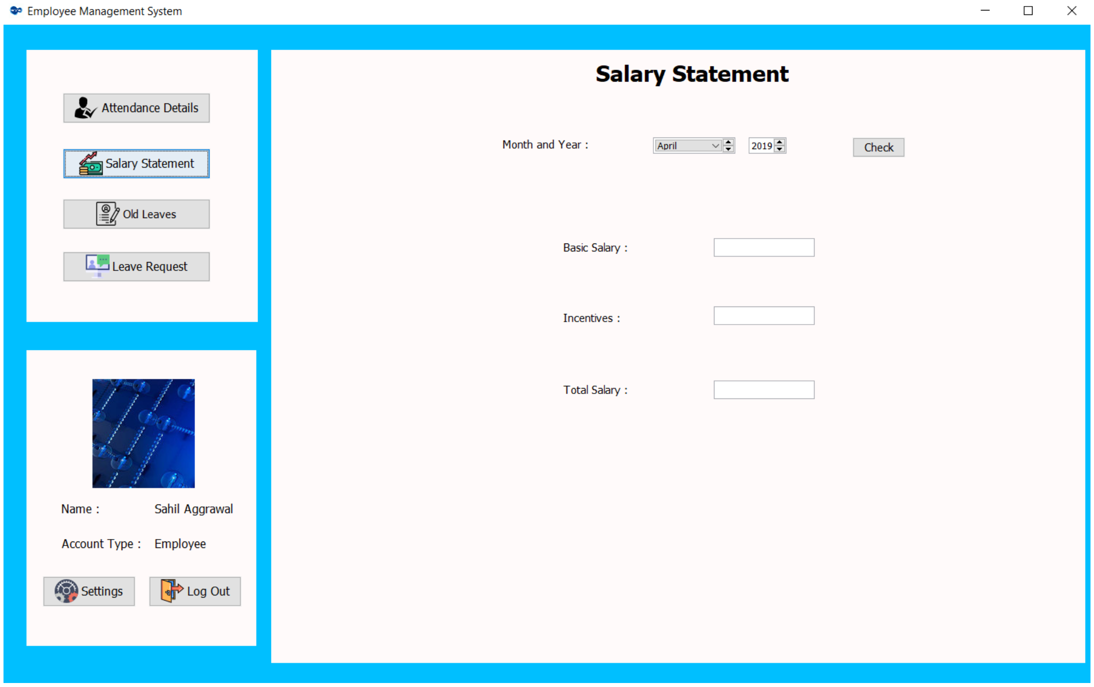
Percentage :

Name : Sahil Aggrawal

Account Type : Employee

Figure 5-13 Attendance Details

5.14 Salary Statement



Salary Statement

Month and Year : April 2019

Basic Salary :

Incentives :

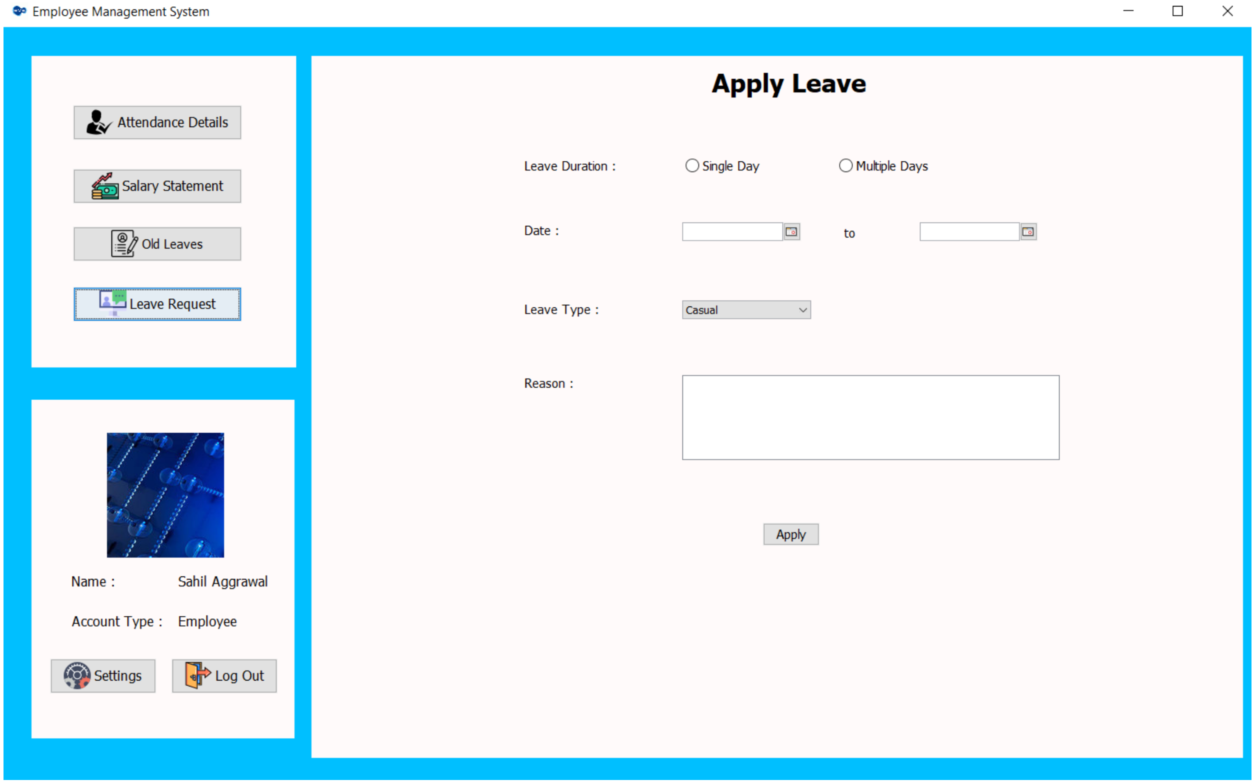
Total Salary :

Name : Sahil Aggrawal

Account Type : Employee

Figure 5-14 Salary Statement

5.15 Apply Leave



Leave Duration : ○ Single Day ○ Multiple Days

Date : to

Leave Type : Casual

Reason :

Name : Sahil Aggrawal

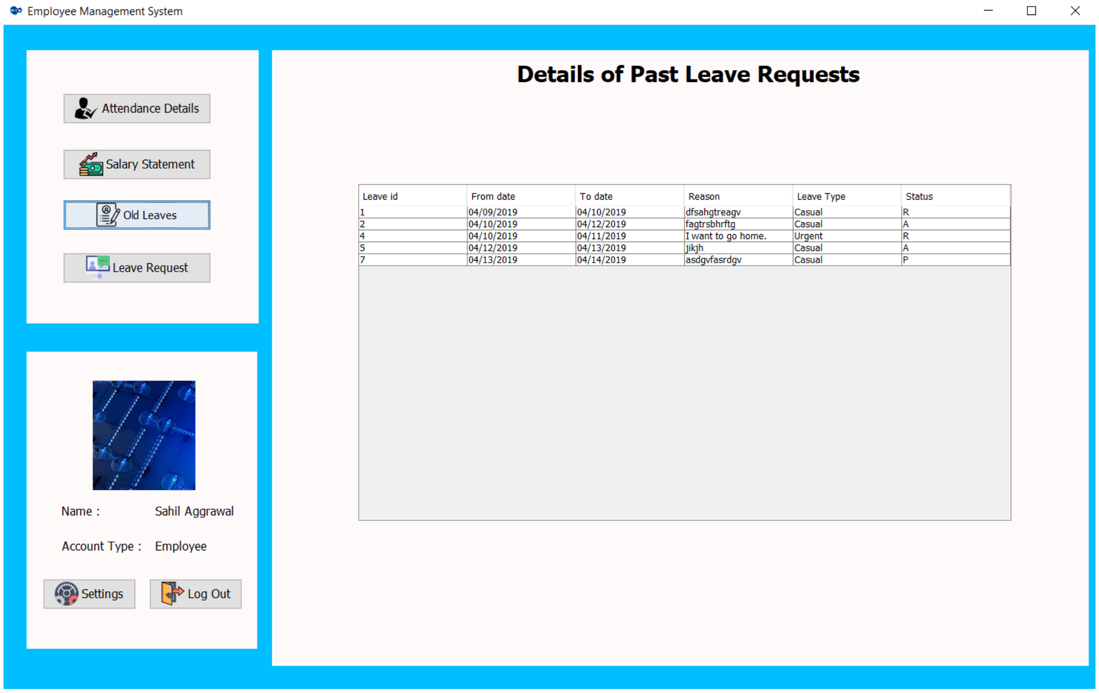
Account Type : Employee

Figure 5-15 Apply Leave

- In this frame user can apply leave.
- Leave can be of single day or multiple day.
- Leave can be of three types: urgent leave and causal leave.
- It is optional for the employee to mention the reason for the leave.
- Leave is applied by pressing the “Apply” button.

5.16 Details of Past Leave Requests

- In this employee can see his/her past leave requests.
- All leaves from the beginning are shown
- The leave information here can be used by the employee to make decision for making further leave requests.



Details of Past Leave Requests

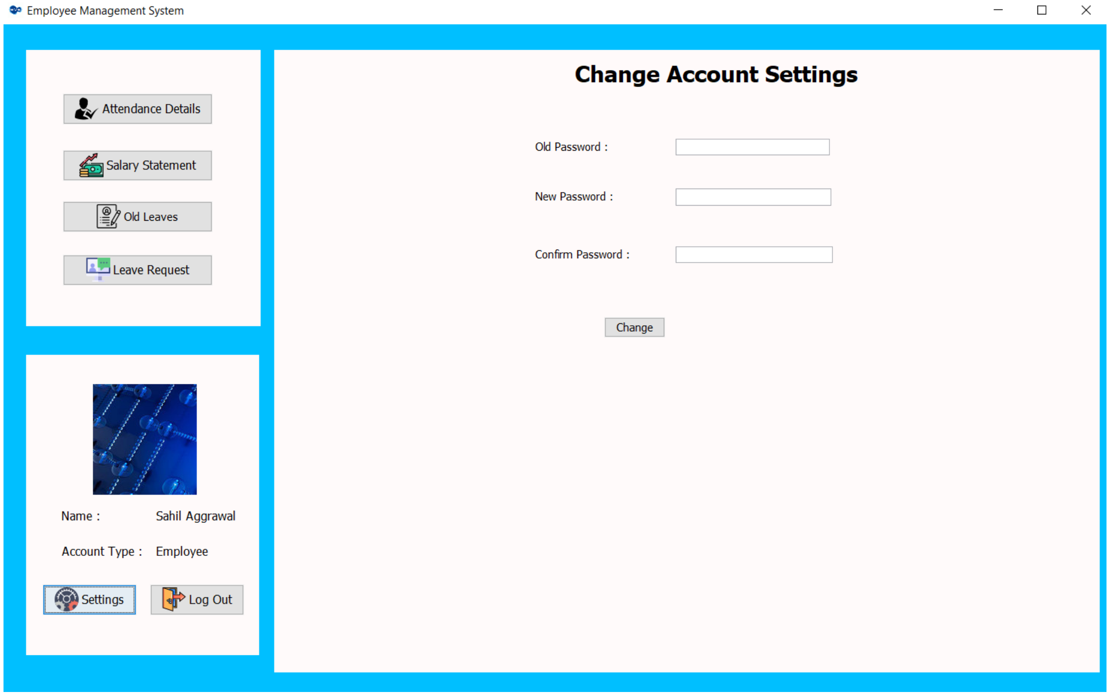
Leave id	From date	To date	Reason	Leave Type	Status
1	04/09/2019	04/10/2019	dfsahgtreagv	Casual	R
2	04/10/2019	04/12/2019	fagtrsbhrftg	Casual	A
4	04/10/2019	04/11/2019	I want to go home.	Urgent	R
5	04/12/2019	04/13/2019	jikjh	Casual	A
7	04/13/2019	04/14/2019	asdgvfasrdgv	Casual	P

Name : Sahil Aggrawal

Account Type : Employee

Figure 5-16 Details of Past Leaves

5.17 Change Employee Account Password



Change Account Settings

Old Password :

New Password :

Confirm Password :

Name : Sahil Aggrawal

Account Type : Employee

Figure 5-17 Change Employee Account Password

- In this frame employee can change account password.
- New password should be at least of 8 characters and contain at least special digit, at least 1 numerical digit.
- Password entered in old password text field should match the password stored in the database.
- Password gets change by clicking on the “Change” button.

Conclusion

This application provides a computerized version of employee management system which will benefit the employees as well as the administrative staff of the organization. It makes entire process online where employee can apply leaves, administrator can generate reports and check on employee’s performance. It also has a facility for employee login where employee can login and can see status of salary slip issued as well request for leave or give some suggestions. It has a facility of administrator login where administrator can add employees and also give necessary suggestion to HR team and also add info about workshops or events happening in the organization or nearby firms in the online notice board. With the additional feature of marking attendance using the biometric sensor make this system more unique and eloquent in its performance making the attendance procedure in an organization super simple. The project offers all the most required features an organization need for the employee management and can be adopted swiftly to any particular company domain.

Future Scope

The following possible extensions can be made to our work to further improve and extend the functionalities of the system.

- Addition of more portals for admin as well as employee to see detailed information
- Addition of ability of the system to generate reports in the form of pdf files rather than showing details in the software itself.
- Addition of marking of attendance in both halves of the day so that accurate record of attendance can be kept
- online work submission facility
- a feature of group chat where employees can discuss various issues of organization can be added to this project thus making it more interactive more user
- Use of machine learning technology to further enhance the attendance system via face recognition

References

[1] Jay Thaker, “Employee management system for startups”, Assessed on JAN 20, 2019. Available at: https://www.sumhr.com/employee-management-system-startupssmbs-free-worries/

[2] Rocket HRMS, “Employee management system”, Assessed on JAN 29, 2019, Available at: https://www.pockethrms.com/employee-management-system/

[3] IntoWeb Softwares, “Humar Resource System”, Assessed on FEB 3, 2019, Available at: http://www.intoweb.com/hr/module_employees.php

[4] Nazmus Sakib, “Employee Management System System”, Assessed on FEB 7, 2019, Available at: https://www.slideshare.net/MonotheistSakib/employee-management-system-62900052

[5] Vega Robot Kit, “r305 Fingerprint module”, Assessed on FEB 10, 2019, Available at: http://www.vegarobokit.com/index.php?route=product/product&product_id=638

[6] Pololu Cooperation, “Arduino Uno r3”, Assessed on FEB 11, 2019, Available at: https://www.pololu.com/product/2191

[7] Theory Circuits, “Fingerprint module interfacing with arduino”, Assessed on FEB 15, 2019, Available at: http://www.theorycircuit.com/fingerprint-sensor-scanner-arduino/

[8] Eclipse Cooperation, “WindowBuilder”, Assessed on MAR 12, 2019, Available at: https://www.eclipse.org/windowbuilder/

[9] Arduino Cooperation, “Interfacing Java with arduino”, Assessed on MAR 18, 2019, Available at: https://playground.arduino.cc/interfacing/java/

[10] Arduino Cooperation, “Serial Communication in arduino”, Assessed on APR 6, 2019, Available at: https://www.arduino.cc/reference/en/language/functions/communication/serial/

Appendix A

This section contains the Arduino code for interfacing Fingerprint module r305

```
#include <Adafruit_Fingerprint.h>

// On Leonardo/Micro or others with hardware serial, use those! #0 is green wire, #1 is white
// uncomment this line:
// #define mySerial Serial1

// For UNO and others without hardware serial, we must use software serial...
// pin #2 is IN from sensor (GREEN wire)
// pin #3 is OUT from arduino  (WHITE wire)
// comment these two lines if using hardware serial
SoftwareSerial mySerial(2, 3);

Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

uint8_t id;

void setup() {
  // put your setup code here, to run once:
Serial.begin(9600);
  while (!Serial);  // For Yun/Leo/Micro/Zero/...
  delay(100);

  // set the data rate for the sensor serial port
  finger.begin(57600);

  Serial.println("Select");
}

uint8_t readnumber(void) {
  uint8_t num = 0;

  while (num == 0) {
    while (! Serial.available());
    num = Serial.parseInt();
  }
  return num;
}

void loop() {
  // put your main code here, to run repeatedly:
  int choice = readnumber();
  if(choice == 1){
      Serial.println("Ready, type ID");
      id = readnumber();
      if (id == 0) {// ID #0 not allowed, try again!
         return;
      }

      while (1){
        int x = getFingerprintEnroll();
        if(x == -1)
          Serial.println("Cannot store!");
         break;
      }
  }
  else if(choice == 2){
    int i = 0;
    Serial.println("Checking");
    while(1){
      int p = getFingerprintIDez();
      if(p > 0){
        Serial.println("Found match");
        break;
      }
      else{
        if(i > 50){
          Serial.println("No match found!");
          break;
        }
        i++;
      }
      delay(50);

    }
  }
}

uint8_t getFingerprintID() {
  uint8_t p = finger.getImage();
  switch (p) {
    case FINGERPRINT_OK:
      Serial.println("Image taken");
      break;
    case FINGERPRINT_NOFINGER:
      Serial.println("No finger detected");
      return p;
    case FINGERPRINT_PACKETRECIEVEERR:
      Serial.println("Communication error");
      return p;
    case FINGERPRINT_IMAGEFAIL:
      Serial.println("Imaging error");
      return p;
    default:
      Serial.println("Unknown error");
      return p;
  }

  // OK success!

  p = finger.image2Tz();
  switch (p) {
    case FINGERPRINT_OK:
      Serial.println("Image converted");
      break;
    case FINGERPRINT_IMAGEMESS:
      Serial.println("Image too messy");
      return p;
    case FINGERPRINT_PACKETRECIEVEERR:
      Serial.println("Communication error");
      return p;
    case FINGERPRINT_FEATUREFAIL:
      Serial.println("Could not find fingerprint features");
      return p;
    case FINGERPRINT_INVALIDIMAGE:
      Serial.println("Could not find fingerprint features");
      return p;
    default:
      Serial.println("Unknown error");
      return p;
  }

  // OK converted!
  p = finger.fingerFastSearch();
  if (p == FINGERPRINT_OK) {
    Serial.println("Found a print match!");
  } else if (p == FINGERPRINT_PACKETRECIEVEERR) {
    Serial.println("Communication error");
    return p;
  } else if (p == FINGERPRINT_NOTFOUND) {
    Serial.println("Did not find a match");
    return p;
  } else {
    Serial.println("Unknown error");
    return p;
  }

  // found a match!
  Serial.print("Found ID "); Serial.print(finger.fingerID);

  return finger.fingerID;
}

// returns -1 if failed, otherwise returns ID #
int getFingerprintIDez() {
  uint8_t p = finger.getImage();
  if (p != FINGERPRINT_OK)  return -1;

  p = finger.image2Tz();
  if (p != FINGERPRINT_OK)  return -1;

  p = finger.fingerFastSearch();
  if (p != FINGERPRINT_OK)  return -1;

  // found a match!
  Serial.print("Found ID " + (String) finger.fingerID + "\n");
  return finger.fingerID;
}

uint8_t getFingerprintEnroll() {

  int p = -1;
  Serial.print("Waiting for valid finger to enroll as #"); Serial.println(id);
  while (p != FINGERPRINT_OK) {
    p = finger.getImage();
    switch (p) {
    case FINGERPRINT_OK:
      Serial.println("Image taken");
      break;
    case FINGERPRINT_NOFINGER:
      break;
    case FINGERPRINT_PACKETRECIEVEERR:
      Serial.println("Communication error");
      break;
    case FINGERPRINT_IMAGEFAIL:
      Serial.println("Imaging error");
      break;
    default:
      Serial.println("Unknown error");
      break;
    }
```

```
}

// OK success!

p = finger.image2Tz(1);
switch (p) {
  case FINGERPRINT_OK:
    Serial.println("Image converted");
    break;
  case FINGERPRINT_IMAGEMESS:
    Serial.println("Image too messy");
    return p;
  case FINGERPRINT_PACKETRECIEVEERR:
    Serial.println("Communication error");
    return p;
  case FINGERPRINT_FEATUREFAIL:
    Serial.println("Could not find fingerprint features");
    return p;
  case FINGERPRINT_INVALIDIMAGE:
    Serial.println("Could not find fingerprint features");
    return p;
  default:
    Serial.println("Unknown error");
    return p;
}

Serial.println("Remove finger");
delay(2000);
p = 0;
while (p != FINGERPRINT_NOFINGER) {
  p = finger.getImage();
}

p = -1;
Serial.println("Place same finger again");
while (p != FINGERPRINT_OK) {
  p = finger.getImage();
  switch (p) {
  case FINGERPRINT_OK:
    Serial.println("Image taken");
    break;
  case FINGERPRINT_NOFINGER:
    Serial.print(".");
    break;
  case FINGERPRINT_PACKETRECIEVEERR:
    Serial.println("Communication error");
```

```
      break;
    case FINGERPRINT_IMAGEFAIL:
      Serial.println("Imaging error");
      break;
    default:
      Serial.println("Unknown error");
      break;
    }
  }

  // OK success!

  p = finger.image2Tz(2);
  switch (p) {
    case FINGERPRINT_OK:
      Serial.println("Image converted");
      break;
    case FINGERPRINT_IMAGEMESS:
      Serial.println("Image too messy");
      return p;
    case FINGERPRINT_PACKETRECIEVEERR:
      Serial.println("Communication error");
      return p;
    case FINGERPRINT_FEATUREFAIL:
      Serial.println("Could not find fingerprint features");
      return p;
    case FINGERPRINT_INVALIDIMAGE:
      Serial.println("Could not find fingerprint features");
      return p;
    default:
      Serial.println("Unknown error");
      return p;
  }

  // OK converted!

  p = finger.createModel();
  if (p == FINGERPRINT_OK) {
    Serial.println("Prints matched!");
  } else if (p == FINGERPRINT_PACKETRECIEVEERR) {
    Serial.println("Communication error");
    return p;
  } else if (p == FINGERPRINT_ENROLLMISMATCH) {
    Serial.println("Fingerprints did not match");
    return p;
  } else {
    Serial.println("Unknown error");
    return p;
  }

  p = finger.storeModel(id);
  if (p == FINGERPRINT_OK) {
    Serial.println("Stored");
  } else if (p == FINGERPRINT_PACKETRECIEVEERR) {
    Serial.println("Communication error");
    return p;
  } else if (p == FINGERPRINT_BADLOCATION) {
    Serial.println("Could not store in that location");
    return p;
  } else if (p == FINGERPRINT_FLASHERR) {
    Serial.println("Error writing to flash");
    return p;
  } else {
    Serial.println("Unknown error");
    return p;
  }
}
```