



VAASAN AMMATTIKORKEAKOULU  
UNIVERSITY OF APPLIED SCIENCES

Manju Maharjan

AN INVENTORY AND HUMAN  
RESOURCE MANAGEMENT SYSTEM

Technology and Communication  
2018

## **ACKNOWLEDGEMENTS**

I would like to extend my sincere gratitude to my thesis instructor Principal Lecturer Pirjo Prosi for the full support and aid to complete my thesis. Also, I am thankful to the Heritage Art and Craft Pvt. Ltd. for providing me opportunity to conduct my thesis.

Last but not the least, I am grateful to my family members for their support and love in one way or another during my studies.

## ABSTRACT

Author	Manju Maharjan
Title	An InventoryAnd Human Resource Management System
Year	2018
Language	English
Pages	56
Name of Supervisor	Pirjo Prosi

---

The conventional method of keeping records in any business is very time consuming and tedious. Hence, the use of modern technology is on the rise. As a result, desktop and mobile applications are becoming popular among the users.

The main goal of this application was to develop a desktop inventory management system for an organization. This desktop based application is designed to administer the employees of an organization and its stock of inventories. This desktop based IMS is an automated system that can handle the information about the organization's employees, like their start and end times of work, their daily total working hours, monthly total working hours and stock left in the store.

This application allows the admin to view the working employees of the day, their starting times and as well as ending time. The admin can add or remove employees, add new inventories, view the details of an employee's like name, contact address, phone number and monthly salary. Netbeans IDE 8.0.2 was used for the front end and MySQL database was used to store the data for developing this application.

---

Keywords     Desktop Application, MySQL JAVA, Netbeans

# CONTENTS

ABSTRACT

ACKNOWLEDGEMENT

LIST OF ABBREVIATIONS

1	INTRODUCTION .....	1
1.1	BACKGROUND .....	1
1.2	Objective of Project .....	1
1.3	Problem Definition.....	2
2	REQUIREMENT IDENTIFICATION .....	3
2.1	System Analysis.....	3
2.1.1	Problem Analysis .....	3
2.1.2	Feasibility study .....	3
2.1.3	Technical feasibility .....	4
2.1.4	Economic Feasibility.....	4
2.1.5	Operational Feasibility .....	4
2.1.6	Schedule feasibility .....	5
2.1.7	IT infrastructure requirement and analysis .....	5
3	GENERAL ARCHITECTURE OF IMS.....	6
3.1	Component Architecture of IMS .....	6
4	SYSTEM DESIGN.....	8
4.1	Use Case Diagram.....	8
4.2	Deployment Diagram.....	10
4.3	ER Diagram .....	11
4.4	Class Diagram.....	12
4.5	Sequence Diagram .....	13
4.5.1	Admin login .....	14
4.5.2	Adding User .....	15
4.5.3	Deleting User .....	16
4.5.4	Editing User .....	17
4.5.5	Add Inventory .....	18
4.5.6	User login.....	19
5	SYSTEM DEVELOPMENT.....	20

5.1	Java Programming Language.....	20
5.2	MySQL .....	20
6	USER INTERFACE.....	22
6.1	Login Page .....	22
6.2	Home page .....	23
6.3	Add or remove user.....	24
6.4	Inventory .....	26
6.5	Time log .....	28
6.6	User Home Page .....	29
6.7	User time log.....	30
7	IMPLEMENTATION .....	31
7.1	Database Connection .....	31
7.2	Admin Login.....	32
7.3	User Login .....	33
7.4	Add user .....	34
7.5	Delete User.....	36
7.6	Update User Info.....	38
7.7	Add Inventory .....	40
7.8	Remove Inventory.....	41
7.9	Update Inventory .....	42
7.10	TimeLog check .....	43
7.11	TimeLog checkout .....	44
8	TESTING .....	45
9	SUMMARY AND CONCLUSION .....	48
	REFERENCES.....	49
	APPENDICES	

## LIST OF FIGURES AND TABLES

FIGURE 1: GENERAL ARCHITECTURE OF IMS .....	6
FIGURE 2: COMPONENT ARCHITECTURE OF IMS .....	7
FIGURE 3:ADMIN USE CASE DIAGRAM .....	9
FIGURE 4: USER USE CASE DIAGRAM .....	10
FIGURE 5:DEPLOYMENT DIAGRAM.....	11
FIGURE 6:E-R DIAGRAM .....	11
FIGURE 7: CLASS DIAGRAM .....	13
FIGURE 8: SEQUENCE DIAGRAM FOR LOGIN.....	14
FIGURE 9: ADDING NEW USER.....	15
FIGURE 10: DELETING USER.....	16
FIGURE 11: EDITING USER .....	17
FIGURE 12: ADDING INVENTORY .....	18
FIGURE 13: USER LOG IN .....	19
FIGURE 14: APPLICATION'S DATABASE .....	21
FIGURE 15: LOG IN.....	22
FIGURE 16: HOME PAGE .....	23
FIGURE 17: ADD EMPLOYEE.....	24
FIGURE 18: DELETE EMPLOYEE .....	25
FIGURE 19: UPDATE EMPLOYEE INFO .....	25
FIGURE 20: ADD INVENTORY .....	26
FIGURE 21: REMOVE INVENTORY .....	27
FIGURE 22: UPDATE INVENTORY INFO.....	27
FIGURE 23: TIME LOG SHOWING EMPLOYEE DAILY WORK HOUR .....	28
FIGURE 24: USER HOME PAGE.....	29
FIGURE 25: USER TIME LOG .....	30
TABLE 1: SOFTWARE REQUIREMENT .....	ERROR! BOOKMARK NOT DEFINED.
TABLE 2: HARDWARE REQUIREMENT.....	ERROR! BOOKMARK NOT DEFINED.
TABLE 3: DESKTOP APPLICATION TESTING.....	45
CODE SNIPPET 1: DATABASE CONNECTION .....	31
CODE SNIPPET 2: ADMIN LOGIN CODE.....	32
CODE SNIPPET 3: USER LOGIN CODE .....	33
CODE SNIPPET 4: ADD USER CODE.....	35
CODE SNIPPET 5: DELETE USER CODE .....	37
CODE SNIPPET 6: USER INFO MODIFY .....	39

<b>CODE SNIPPET 7: ADD NEW INVENTORY .....</b>	<b>40</b>
<b>CODE SNIPPET 8: DELETE INVENTORY .....</b>	<b>41</b>
<b>CODE SNIPPET 9: UPDATE INVENTORY .....</b>	<b>42</b>
<b>CODE SNIPPET 10: STARTING TIME .....</b>	<b>43</b>
<b>CODE SNIPPET 11: END TIME .....</b>	<b>44</b>

## **LIST OF ABBREVIATIONS**

IMS	Inventory Management System
SQL	Structured Query Language
IT	Information Technology
OS	Operating System
IDE	Integrated Development Environment
SDK	Software Development Kit
DB	Database
DBMS	Database Management System
RDBMS	Relational Database Management system
ER- Diagram	Entity-Relationship Diagram

# **1 INTRODUCTION**

The Inventory Management System is an automated system for maintaining the stock and humanresources management System means to manage the employees of the organization. The abbreviation IMS will refer to this application in this report.

This desktop based application is designed to keep track of employee and the amount of stock in an organization called-Heritage Art and Crafts Pvt. Ltd. This system has been developed for the effective utilization of modern technology in the organization. This application can handle lot of information about the organization's employees like their personal information, their starting and ending hours for a particular day of the month, the total daily and monthly working hours and the remaining stock.

## **1.1 BACKGROUND**

In this present world the use of computer applications has increased rapidly because of their effectiveness and efficiency in extracting and exchanging information. Communication has become more advanced and user friendly because of computer applications. Usage of such technology has made every day life as well as business life much easier and simpler. With just one click in the application we can know what is going on around our neighborhood and the globe. E-banking, online shopping, social networking, sports, finance and many more are just a click away because of these applications. Therefore, there are very high demands for efficient, responsive and interactive applications.

## **1.2 Objective of Project**

The main objective of the application is to convert the day to day working into a computerized working environment. This application will have the following objectives:

- To add and remove employees.
- To add and maintain employee details.
- To keep record of starting and ending time of work of employee on a certain day.

- To automatically generate total daily and monthly working hours of an employee.
- To check the stock remaining in the organization.

### **1.3 Problem Definition**

At the moment starting times and ending times of the employees are recorded manually on paper. This creates a lot of work at the end of every month when calculating the total working hours of each employee, and it may take hours to calculate the total hours depending upon the number of working employees. Also, it is a tedious and time consuming task to check all the inventories manually. Hence, this application will solve all these problems and save a lot of time and money.

## **2 REQUIREMENT IDENTIFICATION**

The organization staffs were consulted to learn about the actual problem in the recording of stock and managing staff data. Also, the manual working procedure of recording of stock was observed. The problems identified were searching for the particular stock, duplication of the same record and time consumption even on simple tasks related to inventory.

### **2.1 System Analysis**

In the system analysis phase the functional requirements of the system and the need of the end users are developed to meet the needs of the organization. The analysis phase refines the information gathered during the initiation phase by further identifying the specific activities and resources required to complete a project. There should be discussion between the management and the users regarding the security, design and development of the system. The goal of the system analysis is to determine the solutions for any arising problems.

Key considerations were taken during the study of the system under consideration and they are described in the following subchapters.

#### **2.1.1 Problem Analysis**

There was a manual paper system to record the information of the stock and employee data. Likewise, the records of stock were recorded on daily basis for regular expenses and others on non-regular basis. The problem was with storing the regular and non-regular records of stock. The other problem with the manual paper-based system was in searching for the past records, high chance of replication, high chance of lose and sorting the record.

#### **2.1.2 Feasibility study**

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

The primary objective of a feasibility study is to find out if an information system project can be done and to suggest possible alternate solutions. /1/

### **2.1.3 Technical feasibility**

A large part of determining resources should do with assessing technical feasibility. It considers the technical requirements of the proposed project, which are then compared to the technical capability of the organization. The system project is considered technically feasible if the internal technical capability is sufficient to support the project requirement.

### **2.1.4 Economic Feasibility**

Economic analysis could also be referred to as cost/benefit analysis which is used for evaluating the effectiveness of a new system. In an economic analysis, the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with cost. If benefits outweigh cost, then the decision is made to design and implement the system. /2/

An inventory management system is economically feasible to Heritage Art and Crafts to implement it in the organization as it does not need to spend extra amount for manual work for the same. This application is easy to maintain for the record of stock which is based on regular and non-regular usage that will ultimately help in future decisions.

### **2.1.5 Operational Feasibility**

Operational feasibility is a measure of how well a proposed system solves the problems and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirement analysis phase of system development.

IMS is operationally feasible in Heritage Art and Crafts as it is a user-friendly application for recording the information of the stock. The user needs only the basic knowledge of using a computer system and can easily run the system. The system also simplifies the task of recording the information that helps to store and retrieve the information from the past as well.

### 2.1.6 Schedule feasibility

Schedule feasibility is the probability of a project to be completed within its scheduled time limits, by a planned due date. IMS is a simple system with high value. So, it does not require much time to complete the project when handled by a person having good knowledge of programming and management. /3/

### 2.1.7 IT infrastructure requirement and analysis

IT infrastructure plays a vital role for any software development and the execution. So, the IT infrastructure requirement for any computer based system is very important to be analysed properly even before any system undergoes the development process. Hence, the IT infrastructure required for the development and execution of IMS was analysed properly before development. For the IMS the table 1 lists the components requirements:

**Table 1.** Software Requirement

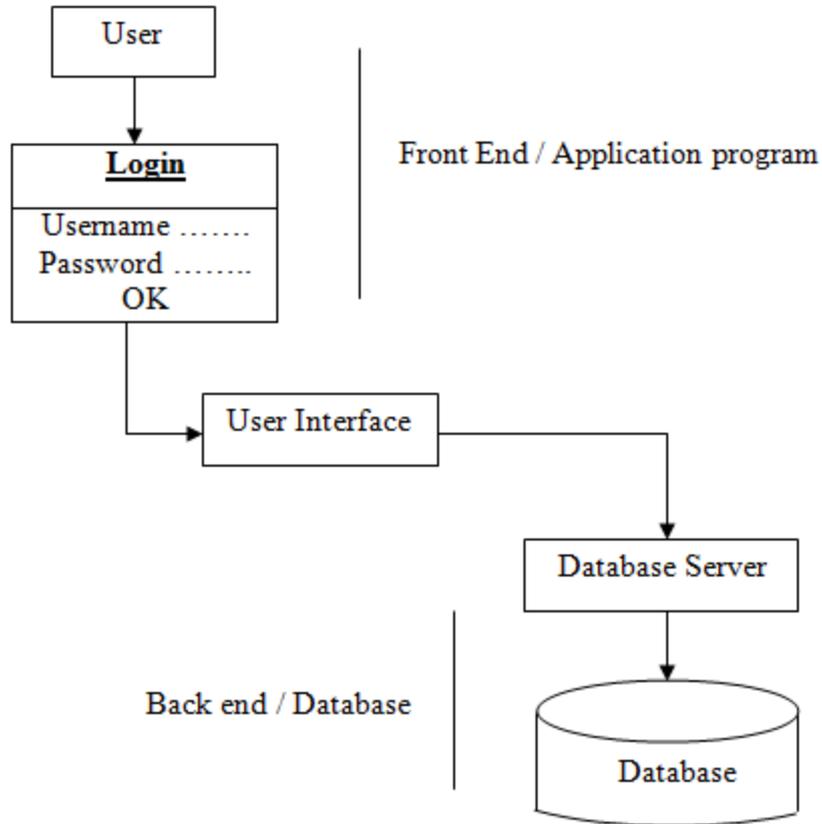
Software	Specification
Platform	Windows, Linux, UNIX based system
Runtime Environment	JRE (Java Runtime Environment)
Server	MySQL

**Table 2.** Hardware Requirement

Hardware	Specification
Processor	Dual Core or above
RAM	1 GB or above
Hard disk	40 GB or above

### 3 GENERAL ARCHITECTURE OF IMS

The general architecture of IMS is represented in the Figure1.



**Figure 1: General Architecture of IMS**

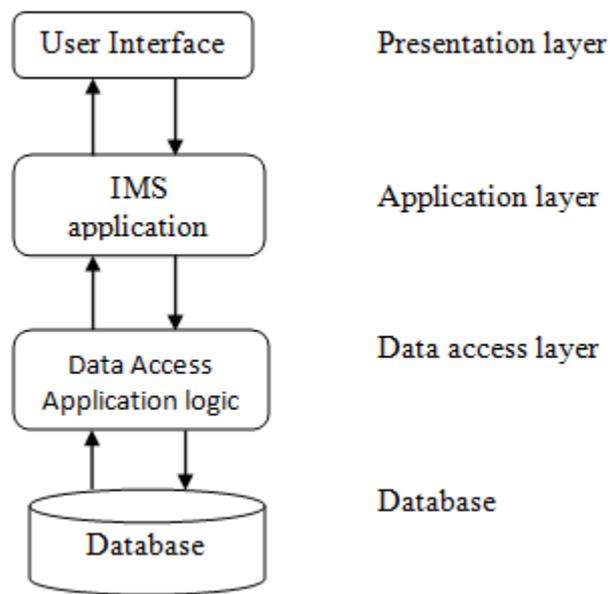
#### 3.1 Component Architecture of IMS

The component architecture consists of three layers. They are:-

- Presentation Layer
- Application Layer
- Data Access Layer

The presentation layer deals with the user end. The software is presented to the user. It displays the GUI of the system to the user.

First the application layer is the middle layer which acts as an intermediary between the user and the database. Then the data access layer is the logic for accessing the data of the software that is stored in the database. The component architecture of the IMS is shown in Figure 2.



**Figure 2: Component Architecture of IMS**

## 4 SYSTEM DESIGN

In this phase the module of the software is developed that satisfies the functional requirements or the purposed system. Here the specifications of the software's hardware, human and data resources are done. The output of this stage describes the new system as a collection of modules or subsystem.

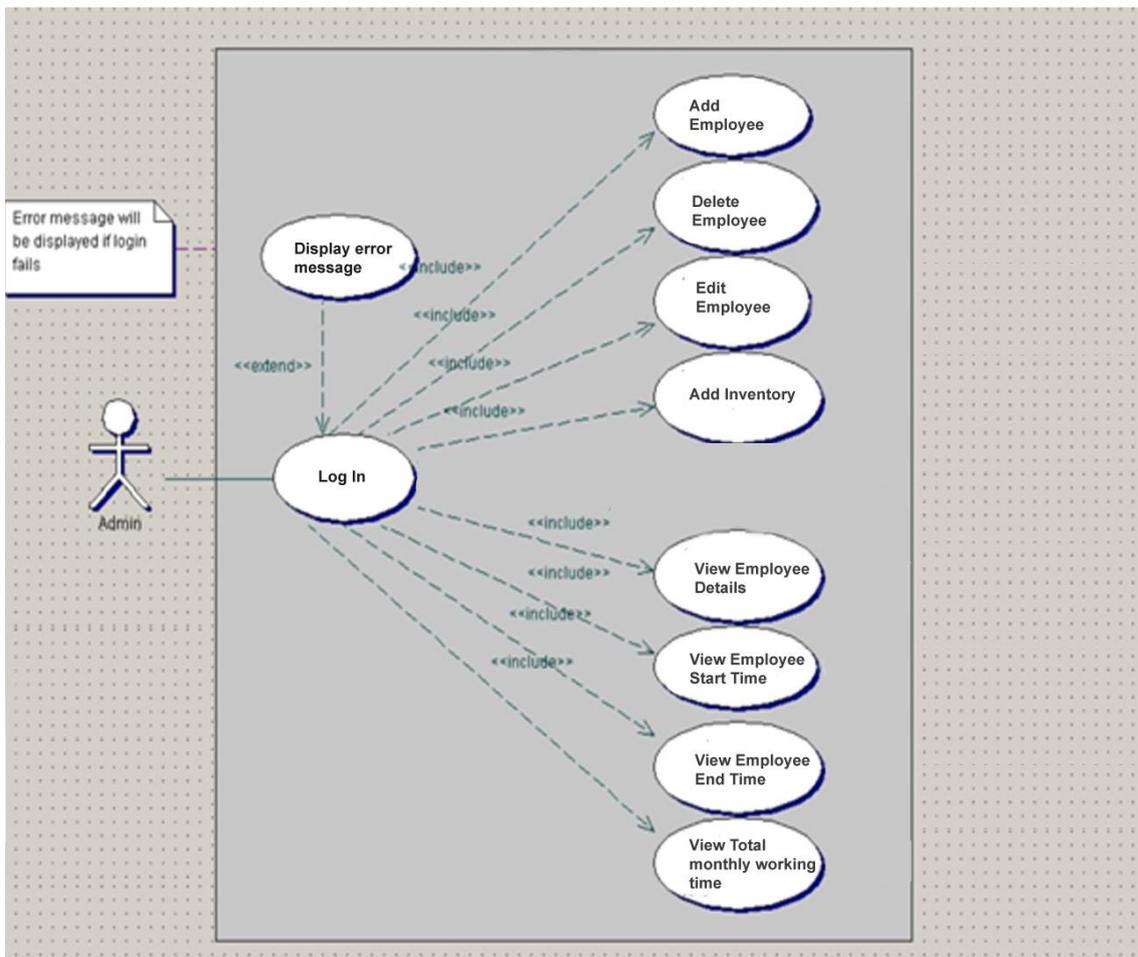
There are different models for viewing and explaining how and what this software does. Generally, Unified Modeling Language (UML) is used as basis for modeling tools. It offers a standard way to write a systems blueprint including conceptual processes such as business process and the system functions as well as concrete things such as programming language, database schemas and reusable software components. It is a standard language for specifying, visualizing, constructing, and documenting all the artifacts of software system. Commonly used UML are:

- Use Case Diagram
- E-R Diagram
- Schema Diagram
- Deployment Diagram
- Sequence Diagram
- Class Diagram

### 4.1 Use Case Diagram

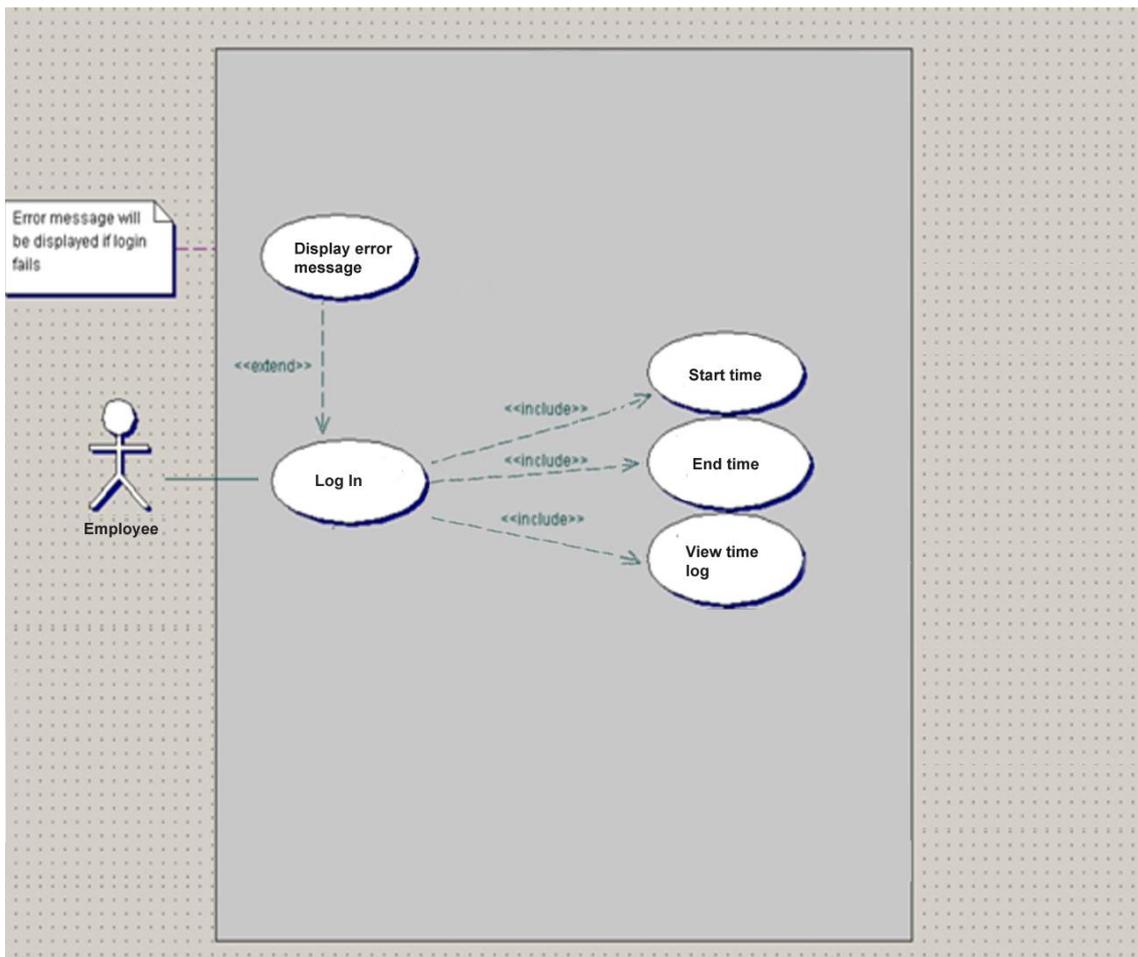
A use case illustrates a unit of functionality provided by the system. The main purpose of the use-case diagram is to help development teams visualize the functional requirements of a system, including the relationship of "actors" (human beings who will interact with the system) to essential processes, as well as the relationships among different use cases. Use-case diagrams generally show groups of use cases — either all use cases for the complete system, or a breakout of a particular group of use cases with related functionality (e.g., all security administration-related use cases).

To show a use case on a use-case diagram, an oval is drawn in the middle of the diagram and the name of the use case is put in the center of, or below, the oval. To draw an actor (indicating a system user) on a use-case diagram a stick person is drawn to the left or right of the diagram. /4/



**Figure 3:Admin use case diagram**

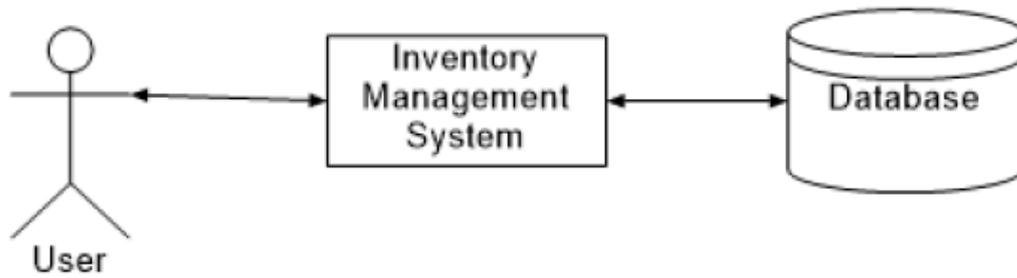
An inventory management is a closed system. The actor is a user who uses the system. A user's responsibility is to insert the records, delete records, view record details, update the records if necessary and search the records for planning, analyzing and accurate reporting with the help of the software. The user is also able to print the required information from the system. To perform all these tasks the user has to pass through a simple password check by login.



**Figure 4: User use case diagram**

## 4.2 Deployment Diagram

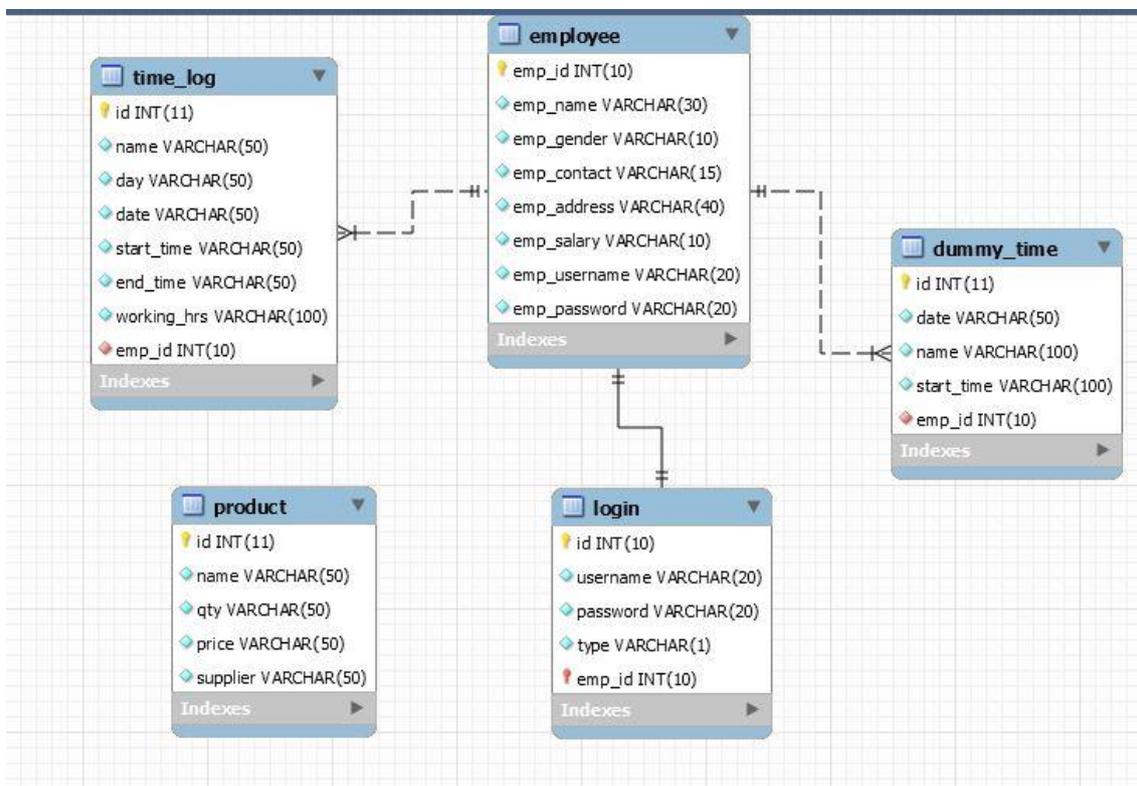
The deployment diagram shows how a system will be physically deployed in the hardware environment. Its purpose is to show where the different components of the system will physically run and how they will communicate with each other. Since the diagram models the physical runtime, a system's production staff will make considerable use of this diagram. /5/. The Deployment Diagram for the IMS is shown below in Figure 5.



**Figure 5:Deployment diagram**

### 4.3 ER Diagram

Entity – Relationship model (ER model) is an abstract and conceptual representation of data. Entity–relationship modeling is a database modeling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion. Diagrams created by this process are called entity–relationship diagrams or ER diagrams. It consists of *entity* which is anything that can be uniquely identified and a *relationship*, which is the association among the entities. /6/



**Figure 6:E-R diagram**

The system has different entities with their respective attributes. Each entity has its own unique key to identify each record uniquely. Some entities also include a *foreign key* to

link with another table. There are three different types of relationships in the system: one-to-one, one-to-many, many-to-one and many-to-many relationship. In this system all the users-the admin and the employees can login, and this is one to one relationship. There is a one to many relationships between the employee and the time log tables since foreign key emp\_id in time\_log table could be used as references to employee table. Besides, employees use this system to log in the start and end times as well as view their own time log.

#### **4.4 Class Diagram**

A class diagram is an illustration of the relationship and source code dependencies among the classes in the Unified Modeling Language (UML). In this context, a class defines the methods and variables in an object, which is specific entity in a program or the unit code representing the entity. /7/ The class diagram is shown below in Figure 7.

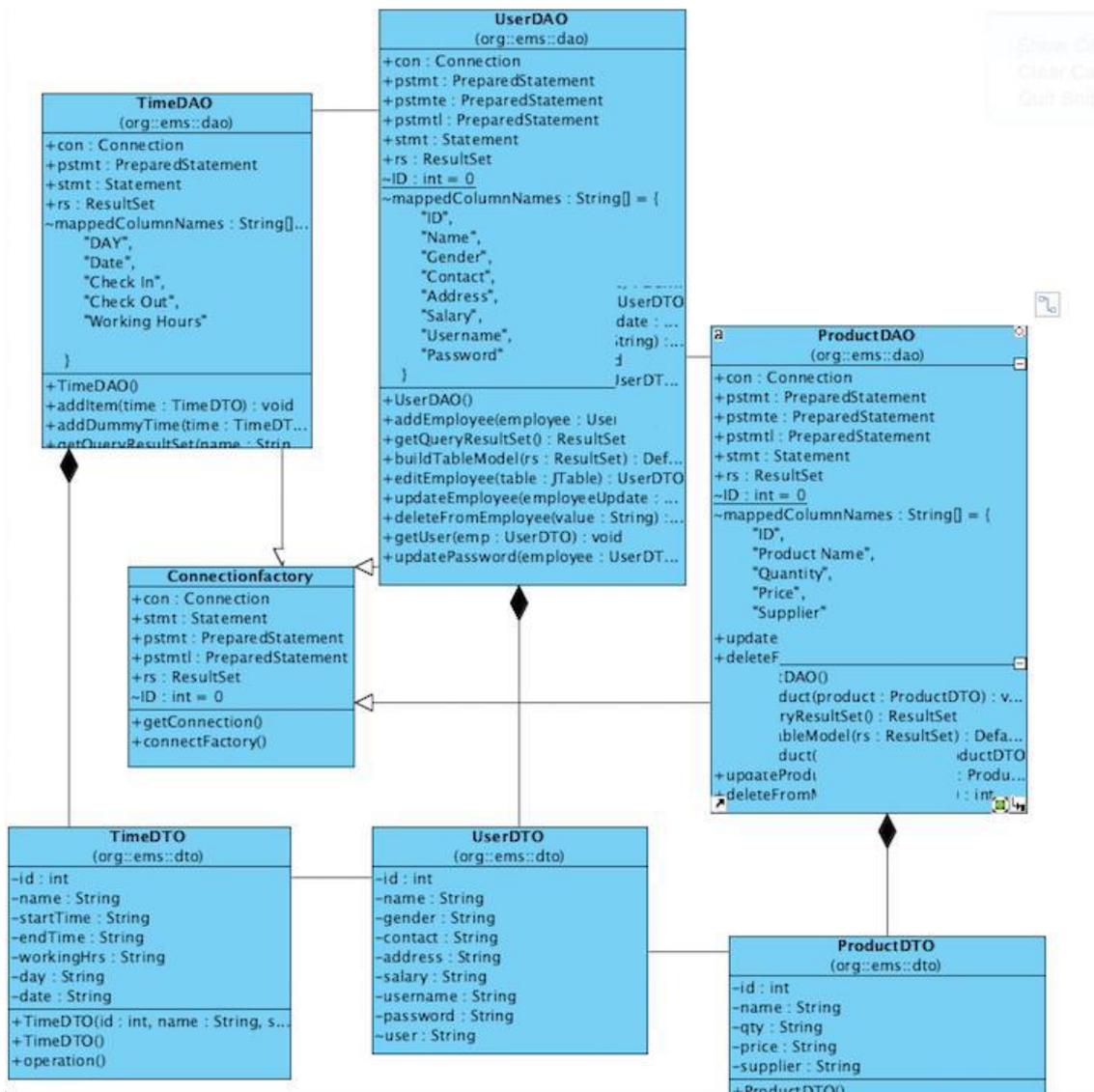


Figure 7: Class diagram

#### 4.5 Sequence Diagram

A sequence diagram (also called Interaction diagram) is a UML construct of a message sequence chart. It shows how the system operates within each entity and their respective order. Sequence diagram describes interactions among classes in terms of an exchange over time. It shows a detailed flow for a specific use or even just part of a specific use case. The sequence diagram of the IMS is shown in Figure 8 /8/

### 4.5.1 Admin login

This sequence diagram shows how the admin can login to the system. The IMS users simply run the application and the login form will activate. The user, then, inputs the username and password for the login and thereafter the entered information is verified with the stored user information in the database. After verifying the provided information, the home page will be activated and the login form will disappear.

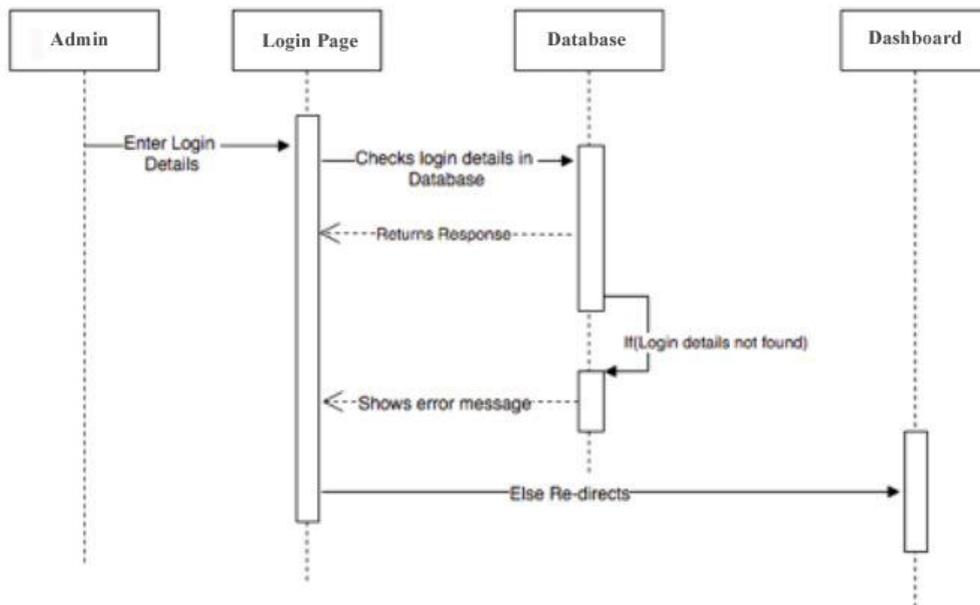


Figure 8: Sequence diagram for login

### 4.5.2 Adding User

To add new user the admin simply clicks on the add user option which opens an add new user page. And, the admin can enter necessary details like name, gender, contact, address, salary, username and password and click the add employee button. Then, the action is sent to the database where the entered parameters are checked. If all the entered data are correct, thus, new user is created in the database otherwise an error message is displayed.

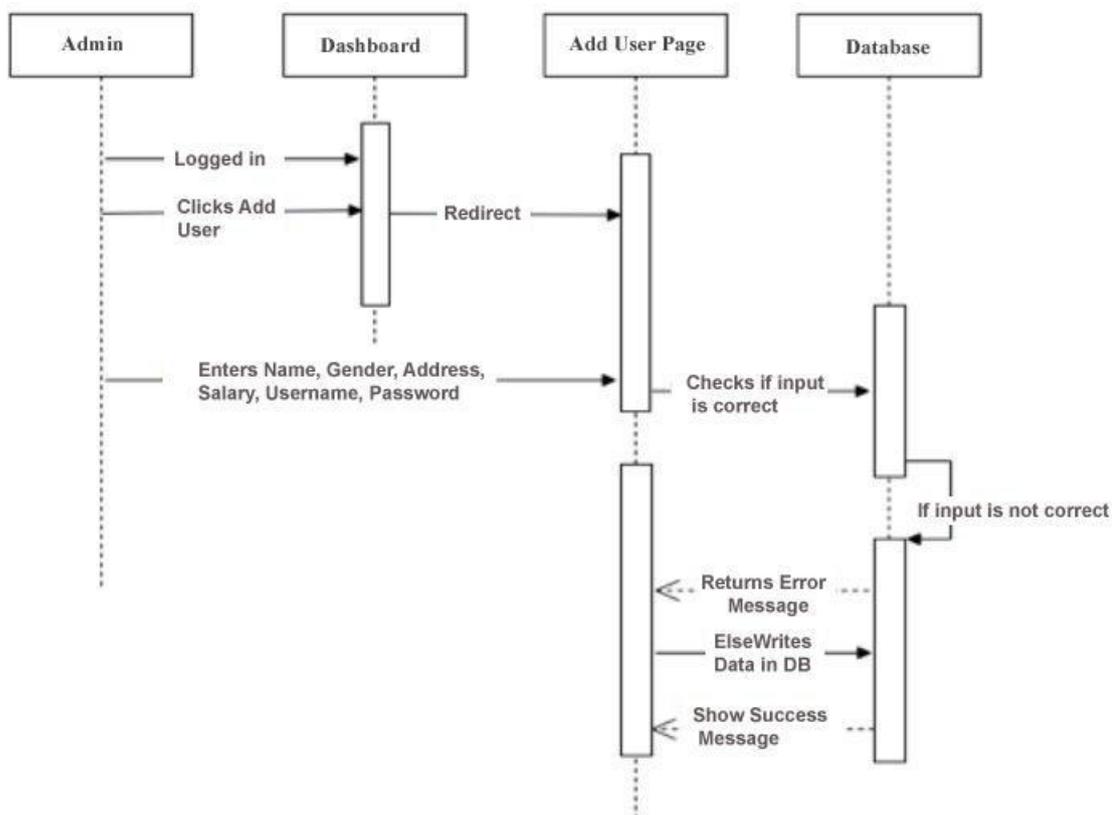


Figure 9: Adding new user

### 4.5.3 Deleting User

To delete the user the admin clicks on the staff option on the home page, which opens a delete user page. Subsequently, the admin clicks on the name of the employee to be deleted and clicks on the delete employee button. Then, the action is sent to the database and a success message is return.

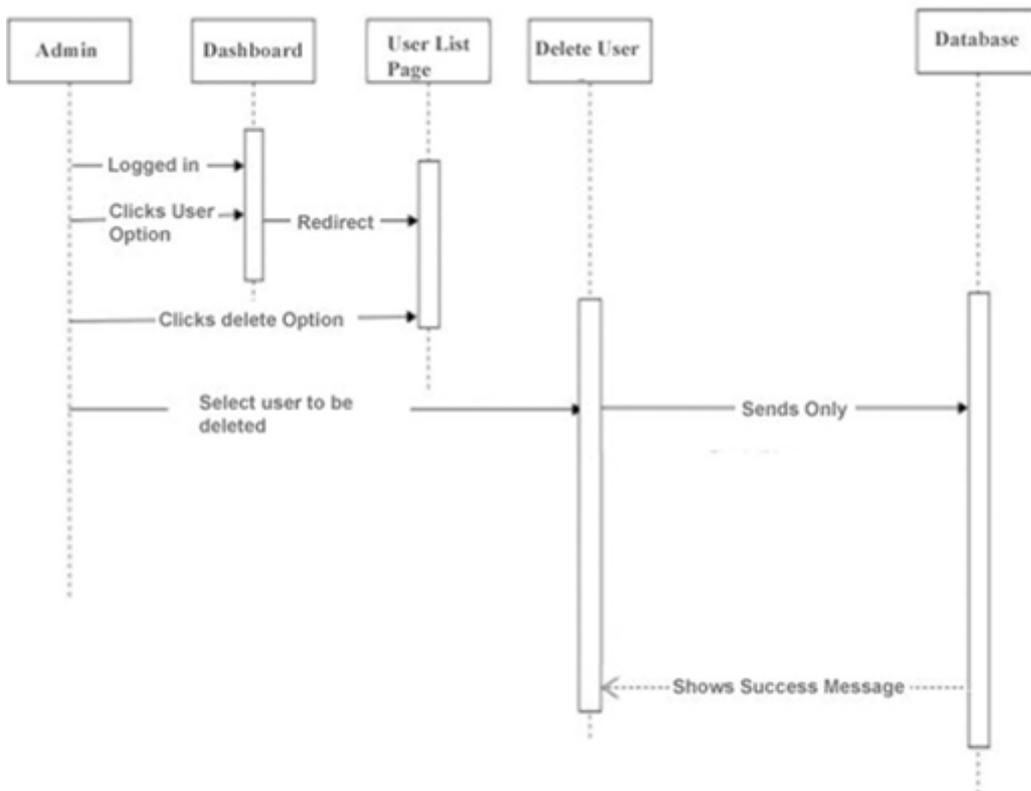
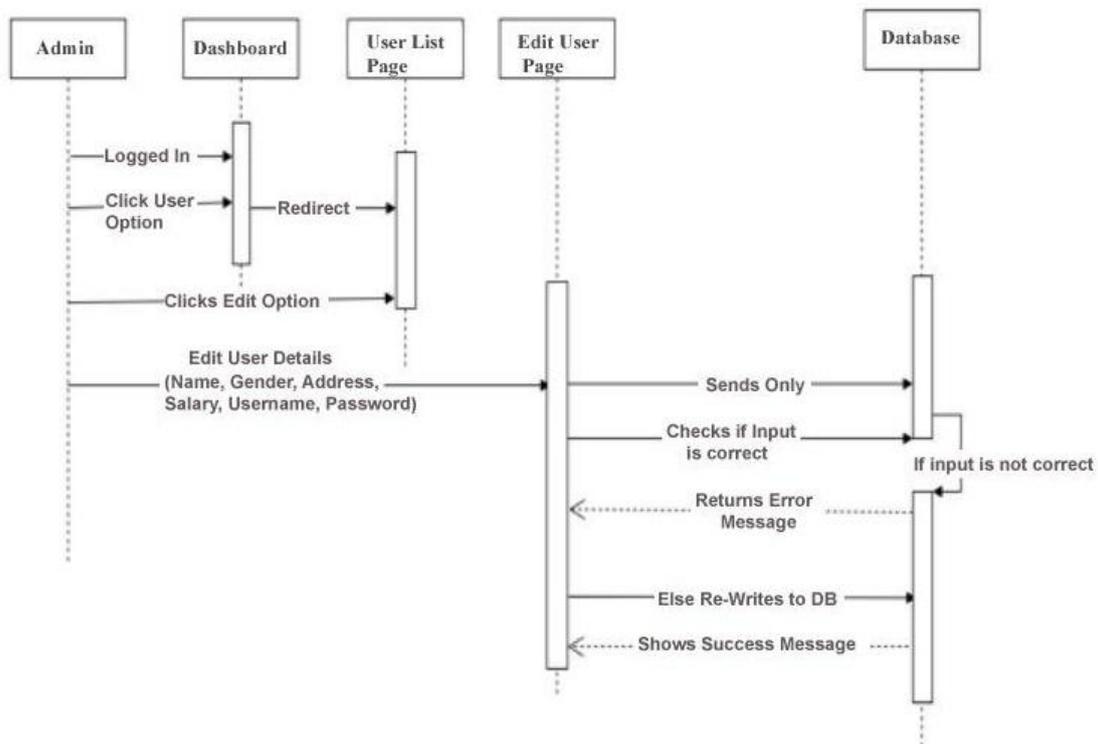


Figure 10: Deleting user

#### 4.5.4 Editing User

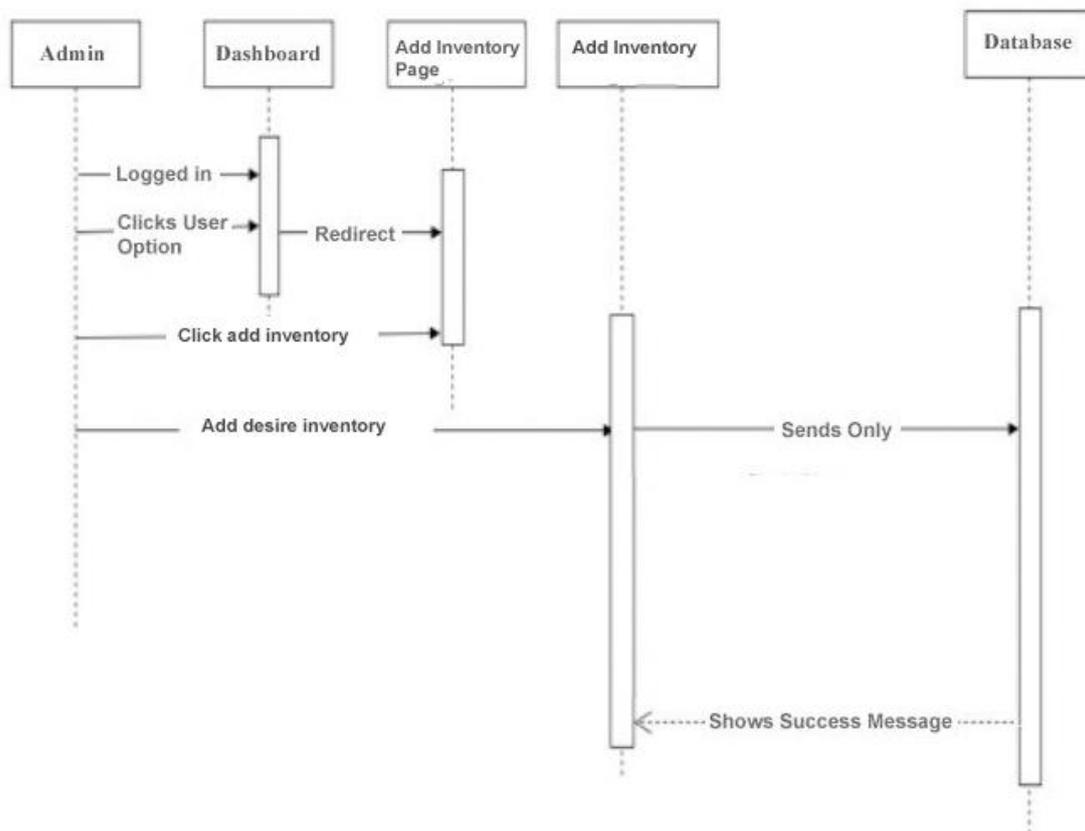
Once admin login to the application one can edit the user account by choosing the edit user option from the home page and make all the necessary changes and click on edit employee button. Secondly, the action is sent to the database, all the entered parameters are checked in the database and if all the entered data are correct, finally, user information is updated. Otherwise error message is displayed.



**Figure 11: Editing user**

#### 4.5.5 Add Inventory

Here, admin can add or remove inventory. Once admin is logged in one can click on add inventory button which leads to add inventory page. Similarly, the admin can enter all the necessary details about the inventory and click on add inventory. Then the action is sent to database and if all the entered parameters are correct the new inventory will be added in the database.



**Figure 12: Adding inventory**

### 4.5.6 User login

In the login page users can enter their username and password provided by the admin for the first log in and then selects the user option and click on log in. The entered values are checked in the database, and if they exist then the log in is granted to the users. Otherwise an error message will be displayed.

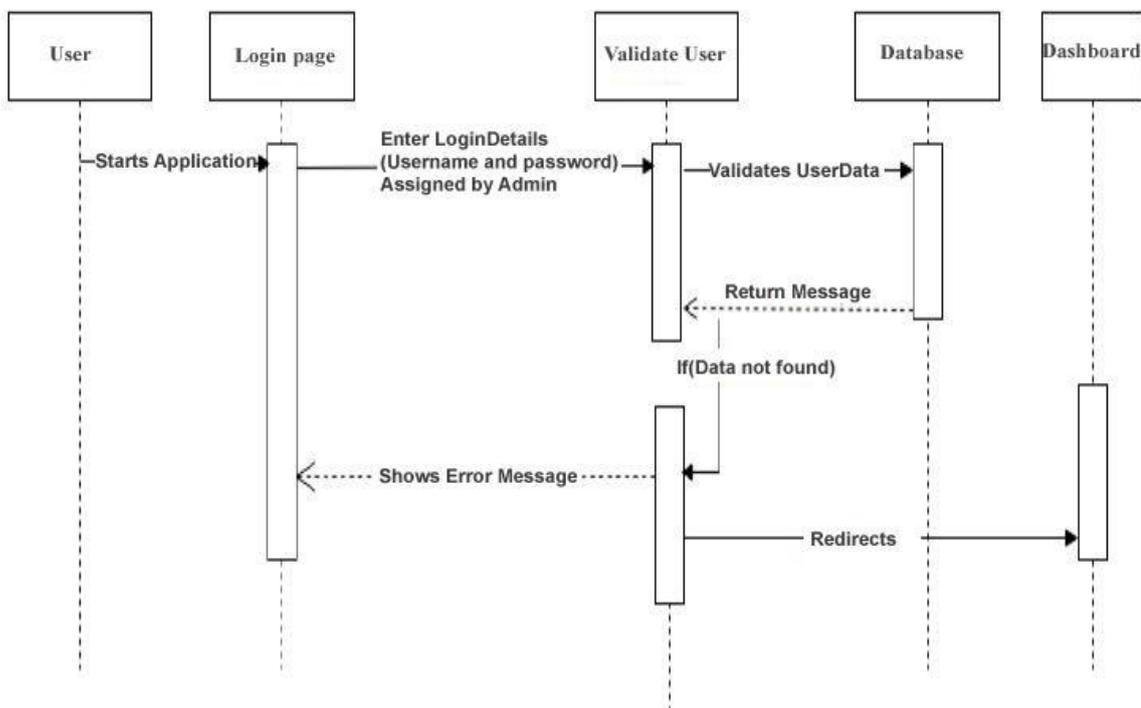


Figure 13: User log in

## 5 SYSTEM DEVELOPMENT

This is the programming phase where a particular programming language is selected in order to code the designed system. Modular and subsystem programming code will be accomplished during this stage. The description of the programming language, databases language and tools that are used are examined below.

### 5.1 Java Programming Language

Java is a simple and yet a powerful object oriented programming language. Java was developed with the concept of “write once and run everywhere” concept. The JVM is the environment in which the Java program executes. The Java programming language has the following features:

- Reusability of code
- Emphasis on data rather than procedure
- It is platform independent
- Objects can communicate with each other through functions
- New data and functions can easily be added

Java Programming was done using Netbeans IDE 8.0.1. This IDE can be used to code in various programming languages like for example PHP, HTML.

### 5.2 MySQL

The Structured Query Language (SQL) is a very popular database language, and its standardization makes it quite easy to store, update and access data. One of the most powerful SQL servers is called MySQL and surprisingly enough, it's free. Some of the features of MySQL are:

- Handles large databases, in the area of 50,000,000+ records
- Tested with a commercial memory leakage detector (purify) /9/
- A privilege and password system is provided which is very flexible, secure and allows host-based verification

- Passwords are secure since all password traffic when connecting to a server is encrypted.

MySQL server is used as database at backend in this study provided by VAMK.

The ER diagram describes the relationship between the different tables in the database.

Figure 14 shows that the application database contains five main database tables which are:

**Admin:** In this table admin user name and password are stored.

**Employee:** This table consists of details of all the employees like name, address, contact info, salary.

**Product:** This table consists of all the different type of products or inventories available in the organizations.

**Dummy\_time:** Once the users starts their work in the morning, the starting work time in first stored in this dummy\_time table.

**Time\_log:** Once the users starts their work in the morning, the starting work time in first stored in this dummy\_time table and when the user finishes his or her work and sign out the time then the starting and ending time are stored in this table along with the total working hour of the day.

Table	Action	Rows	Type	Collation	Size	Overhead
admin	Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	16 KiB	-
dummy_time	Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	16 KiB	-
employee	Browse Structure Search Insert Empty Drop	4	InnoDB	latin1_swedish_ci	16 KiB	-
login	Browse Structure Search Insert Empty Drop	17	InnoDB	latin1_swedish_ci	16 KiB	-
product	Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	16 KiB	-
time_log	Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	16 KiB	-

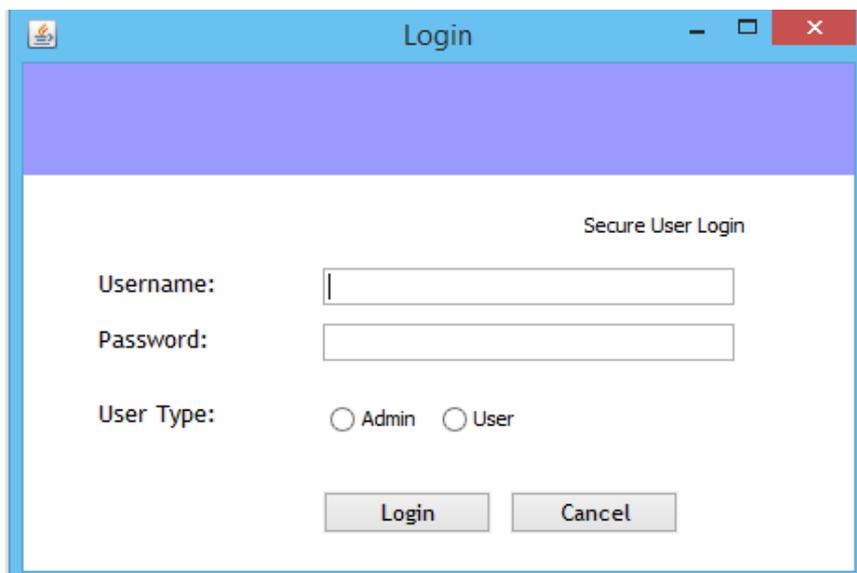
**Figure 14: Application's Database**

## 6 USER INTERFACE

This section provides the detailed description of the user interface for the application. There are two user interfaces. Admin's Interface and User's Interface

### 6.1 Login Page

Admin can log in using the user name and password created while developing this application. Hence, the admin can create the username and password for other users or employees. The log in page contains two input text fields username and password, two radio fields to specify either the admin or a local user and still the login button and the cancel button. Figure 15 shows the login interface.



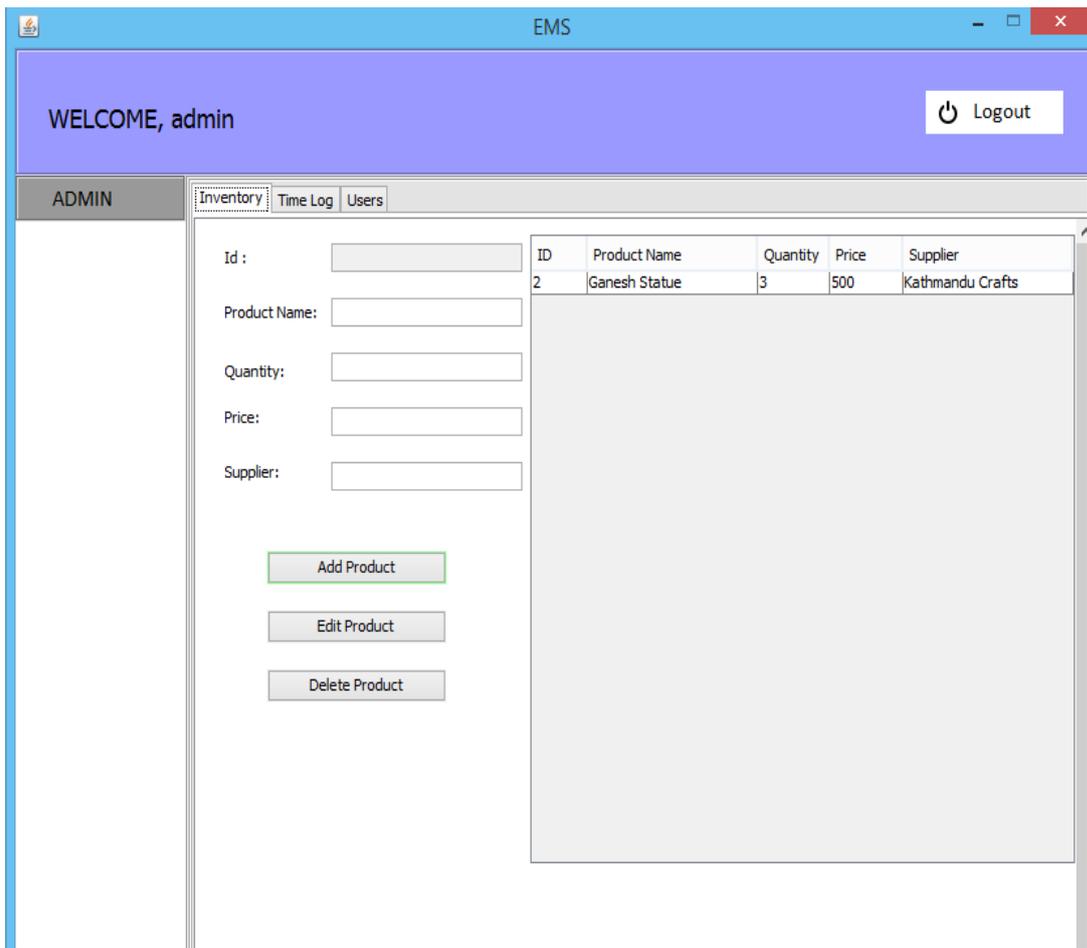
The screenshot shows a window titled "Login" with a blue header bar. Below the header is a purple bar. The main content area is white and contains the following elements:

- Text "Secure User Login" in the top right corner.
- Label "Username:" followed by a text input field.
- Label "Password:" followed by a text input field.
- Label "User Type:" followed by two radio buttons: "Admin" and "User".
- Two buttons at the bottom: "Login" and "Cancel".

**Figure 15: Log in**

## 6.2 Home page

Once the admin is logged in, the admin's name appears on the top left side of the home page. There are various clickable options on the home page which can be used to perform various tasks. The admin can log out of the application by clicking on the logout button which is on the top right side of the home page. The figure 16 shows dashboard of the admin homepage.



**Figure 16: Home page**

### 6.3 Add or remove user

After a successful log in, the admin can create or remove users. To do that the admin can simply click on the user option on the home page. Once the option is clicked it will redirect the user to the add remove or edit user page. Here the admin can add, remove or edit user information according to the requirements as shown in Figure 17, Figure 18 and Figure 19 respectively.

The screenshot shows the 'Add Employee' form in the EMS application. The form has the following fields and values:

- Name:** Harry
- Contact:** 123456789
- Salary:** 100000
- Gender:** Male (selected)
- Address:** (empty)
- Username:** (empty)

A 'Message' dialog box is displayed in the center of the form, indicating that a record has been successfully inserted. The message reads: "One Record Inserted." with an 'OK' button.

Below the form, there is a table listing existing employees:

ID	Name	Salary	Username	Password			
9	Anil Maharja	329432	anil	anil			
12	Ram	34343	ram	ram			
20	sujita	3465346	sujita	sujita			
21	Tina	Female	3466	Chhetrapati	32423	tina	tina
29	manju	123456	konepajakatu	11	1111111	manju	manju

**Figure 17: Add employee**

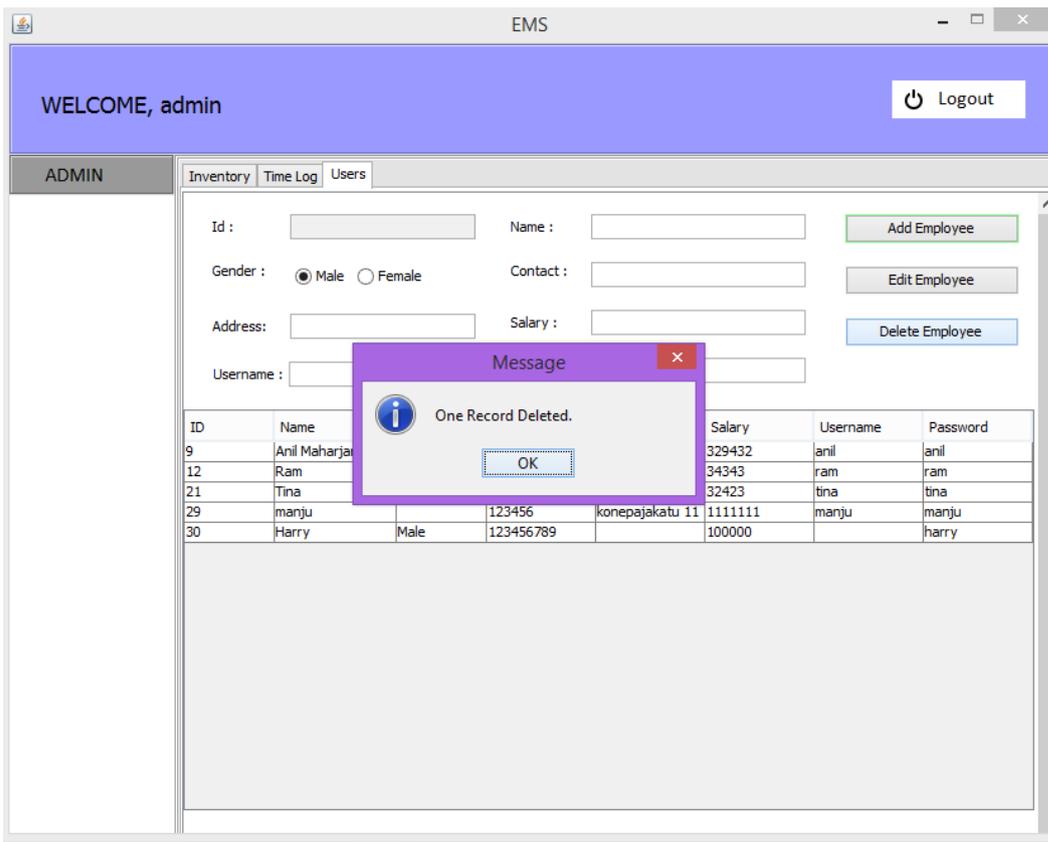


Figure 18: Delete Employee

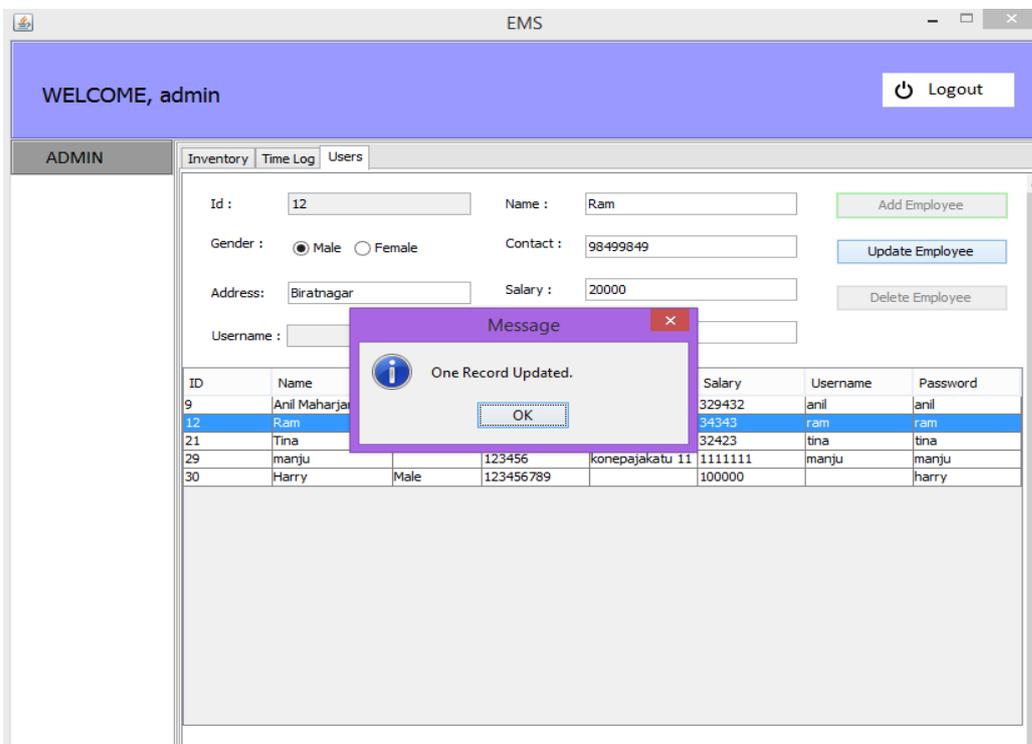
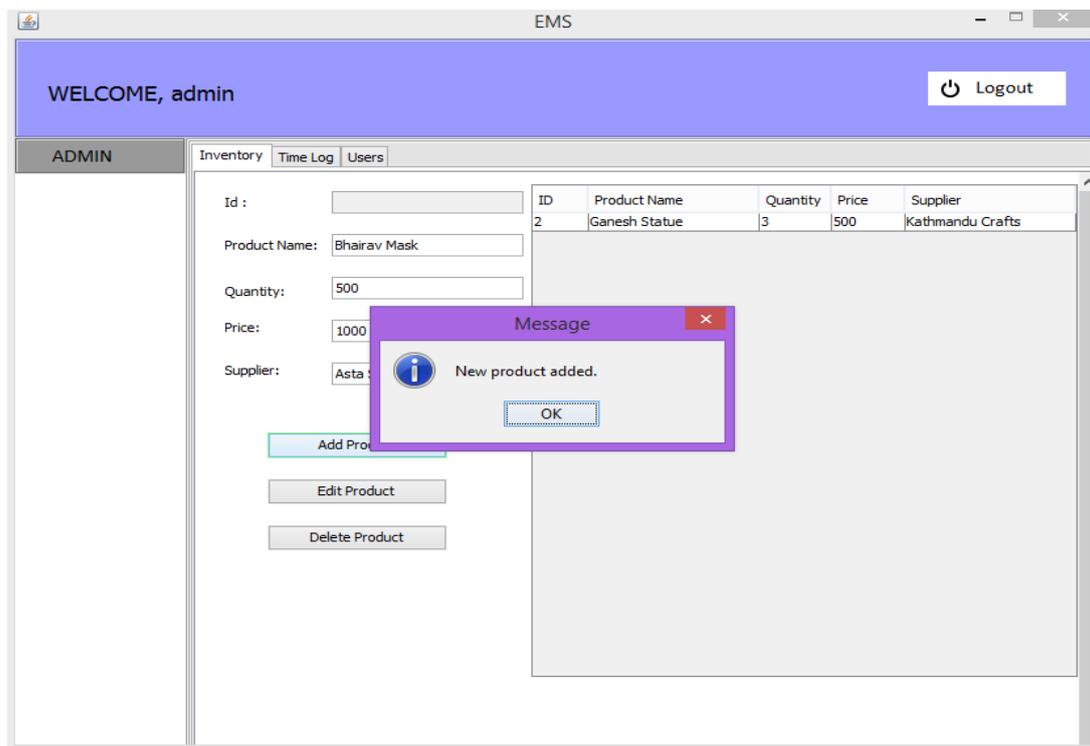


Figure 19: Update Employee info

## 6.4 Inventory

The admin can add remove or edit the inventory by inserting all the correct information in the field. Figure 20, Figure 21 and Figure 22 shows add inventory, remove inventory and update inventory interface respectively.



**Figure 20: Add Inventory**

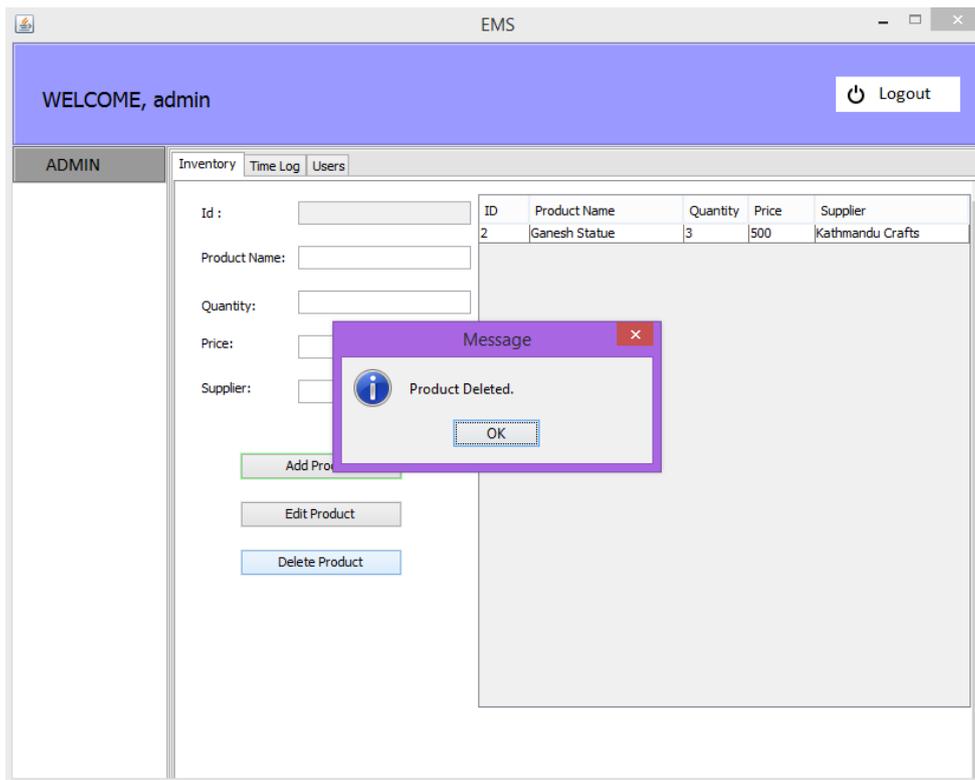


Figure 21: Remove Inventory

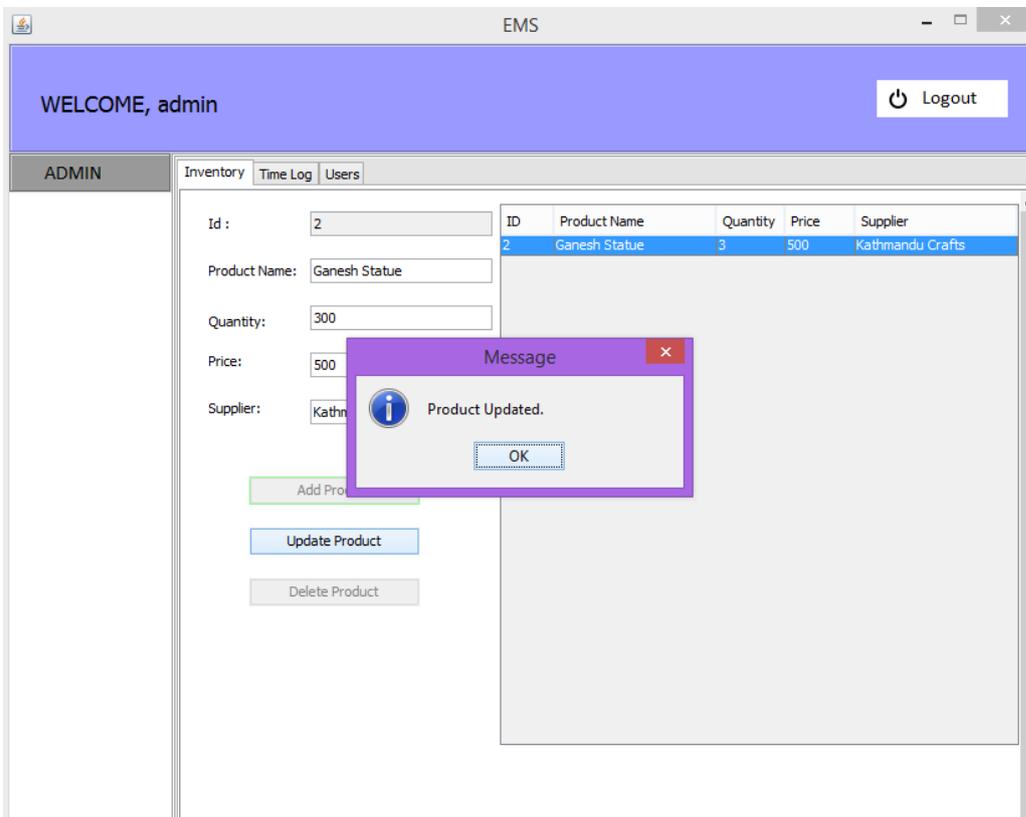
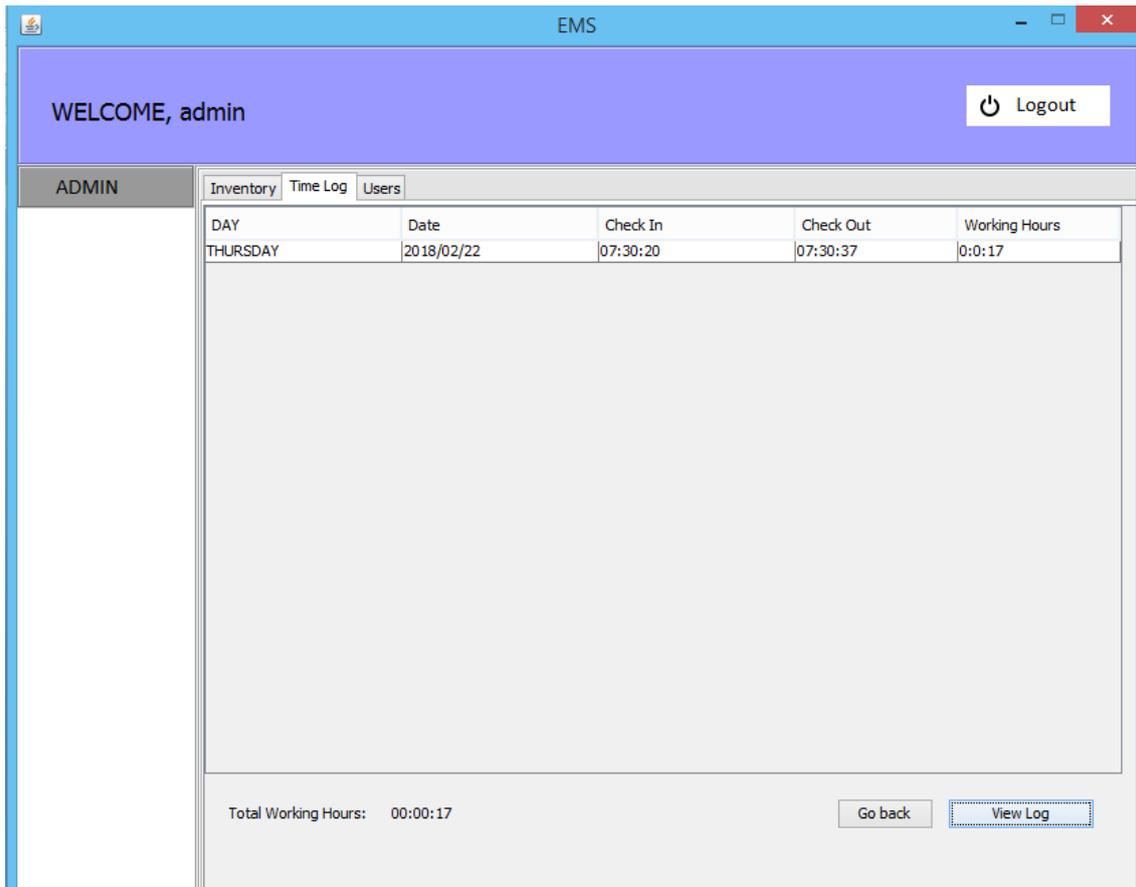


Figure 22: Update Inventory Info

## 6.5 Time log

The admin can view the working time histories of all the employees. For this the admin must click time log tab on the home page, which will redirect to the page with all the employees. Here the admin can click on the name of the employee to view his or her working time history as shown in Figure 23.



The screenshot displays the EMS application interface. At the top, a blue header bar contains the text "WELCOME, admin" and a "Logout" button. Below this is a navigation menu with three tabs: "Inventory", "Time Log", and "Users". The "Time Log" tab is selected, and the main content area shows a table with the following data:

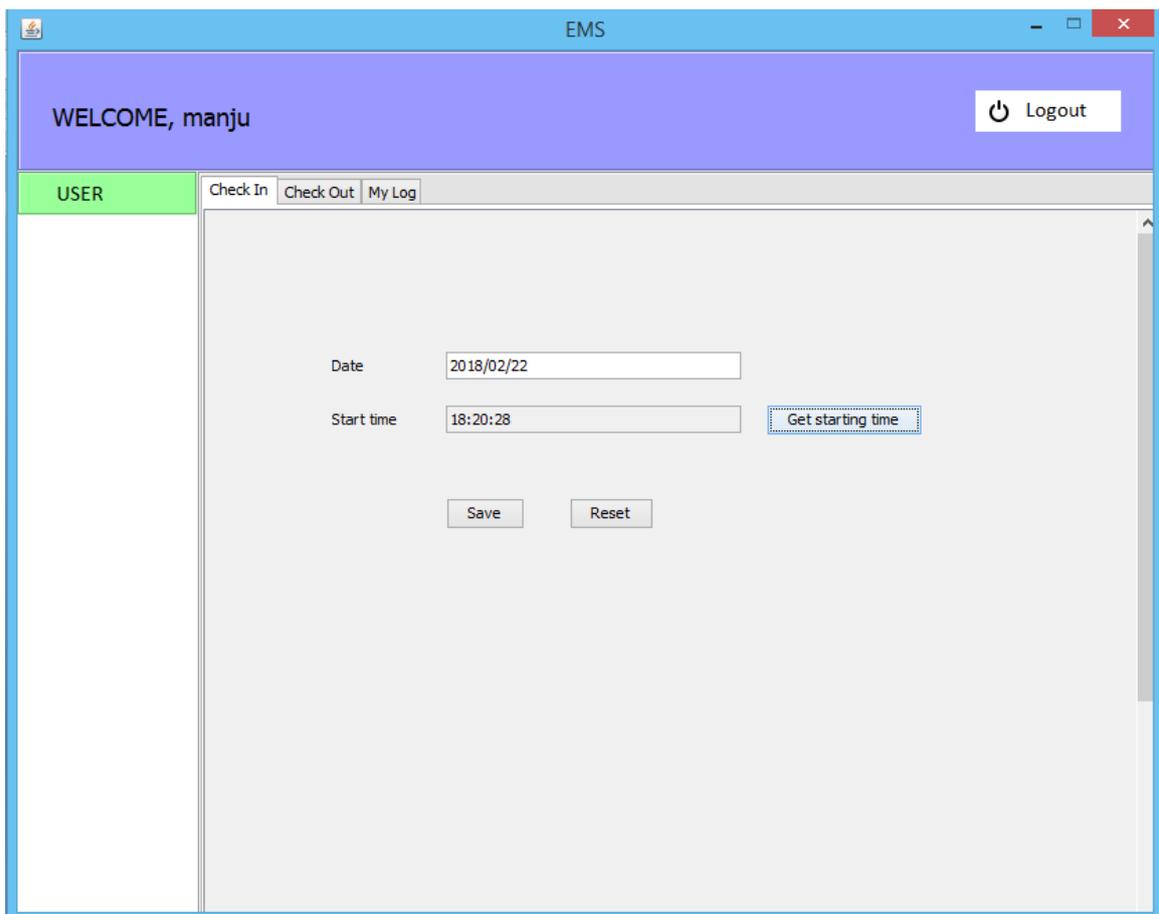
DAY	Date	Check In	Check Out	Working Hours
THURSDAY	2018/02/22	07:30:20	07:30:37	0:0:17

At the bottom of the table, there is a summary row: "Total Working Hours: 00:00:17". To the right of this summary are two buttons: "Go back" and "View Log".

**Figure 23: Time log showing employee daily work hour**

## 6.6 User Home Page

A user can log in to the application by using the username and password provided by the admin. Once the user inputs correct log in information in the log in page he or she will be redirected to the user home page. There are several clickable options on the user home page where user can start the work by clicking on "Get starting time" button on the homepage. Users' home page is shown below in Figure 24.



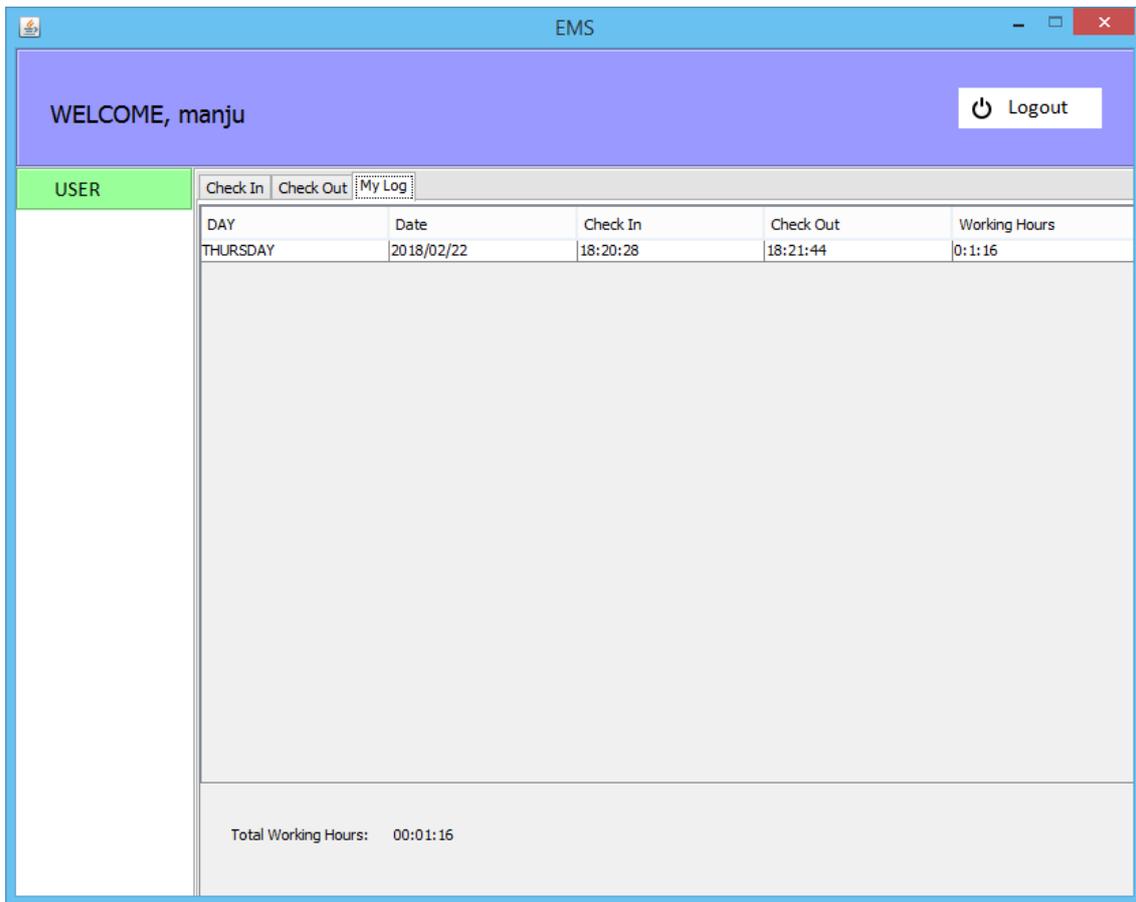
The screenshot shows a web application window titled "EMS". The main header is purple and contains the text "WELCOME, manju" on the left and a "Logout" button on the right. Below the header is a navigation bar with a green "USER" tab and three sub-tabs: "Check In", "Check Out", and "My Log". The main content area is light gray and contains a form with the following elements:

- A "Date" input field with the value "2018/02/22".
- A "Start time" input field with the value "18:20:28".
- A "Get starting time" button with a dotted border.
- "Save" and "Reset" buttons.

**Figure 24: User home page**

## 6.7 User time log

The user can view the working time history. On the homepage of the user, the user can click on the log button and view working time history. User time log is shown below in Figure 25.



DAY	Date	Check In	Check Out	Working Hours
THURSDAY	2018/02/22	18:20:28	18:21:44	0:1:16

Total Working Hours: 00:01:16

**Figure 25: User time log**

## 7 IMPLEMENTATION

This section provides a description of the implementation of the graphical user interfaces which are built for the application. The detailed process of how the codes are implemented in the application is explained here.

Java is used for the frontend of the application whereas MySQL handles the database in the server which is the so-called backend. The following code snippets show the implementation of the applications

### 7.1 Database Connection

The MySQL database server is used to build the application. The connection is checked and the username and password provided are verified and the connection to the database is successfully made or otherwise an error message is displayed.

```
public Connection getConnection() throws SQLException {
    try {
        Class.forName("com.mysql.jdbc.Driver");
        con = DriverManager.getConnection("jdbc:xxxx://xxxx.xx.xxxxxx/xxxxxxxxxxx_xxx", "x xxxxxxxx ", "
xxxxxxxxxxxx ");
    } catch (Exception e) {
        e.printStackTrace();
    }
    return con;
}
```

#### Code Snippet 1: Database Connection

## 7.2 Admin Login

The database connection is checked and a method checkLogin is called passing three arguments, namely username, password (pwd) and type. If type equals to "e" and user checks the admin radio button and then the admin dashboard is displayed. Connection-Factory.java class is used here.

```
else if (cf.checkLogin(username, password, type)) {  
    try {  
        USERNAME = username;  
        dispose();  
        Dashboard db = new Dashboard();  
        db.setResizable(false);  
        db.setVisible(true);  
        db.setLocationRelativeTo(null);  
    } catch (SQLException ex) {  
        Logger.getLogger(LoginFrame.class.getName()).log(Level.SEVERE, null, ex);  
    }  
}
```

### Code Snippet 2: Admin login Code

### 7.3 User Login

The database connection is checked and a method checkLogin is called passing three arguments, i.e. username, password (pwd) and type. If the type equals to "e" and user checks the user radio button and then the user dashboard is displayed.

```
if (cf.checkLogin(username, password, type) && (type.equalsIgnoreCase("e"))) {  
    USERNAME = username;  
    dispose();  
    Dashboard db = null;  
    try {  
        db = new Dashboard();  
    } catch (SQLException ex) {  
        Logger.getLogger(LoginFrame.class.getName()).log(Level.SEVERE, null, ex);  
    }  
    db.setVisible();  
    db.setResizable(false);  
    db.setVisible(true);  
    db.setLocationRelativeTo(null);  
}
```

#### Code Snippet 3: User Login Code

#### **7.4 Add user**

UserDAO.java class method is used to create a new user. After entering all the necessary data in the text field, the admin clicks add user button and then the query is sent to the database. If all the entered data are correct, then a new user is created otherwise an error message is displayed.

```

public void addEmployee(UserDTO employee) {
    try {
        String sql1 = "INSERT INTO employee
(emp_name,emp_gender,emp_contact,emp_address,emp_salary,emp_username,emp_password)VALU
ES (?, ?, ?, ?, ?, ?, ?)";

        String sql2 = "INSERT INTO login (username,password,type)VALUES (?, ?, ?)";

        pstmte = (PreparedStatement) con.prepareStatement(sql1);

        pstmte.setString(1, employee.getName());

        pstmte.setString(2, employee.getGender());

        pstmte.setString(3, employee.getContact());

        pstmte.setString(4, employee.getAddress());

        pstmte.setString(5, employee.getSalary());

        pstmte.setString(6, employee.getUsername());

        pstmte.setString(7, employee.getPassword());

        pstmtl = (PreparedStatement) con.prepareStatement(sql2);

        pstmtl.setString(1, employee.getUsername());

        pstmtl.setString(2, employee.getPassword());

        pstmtl.setString(3, "e");

        if (pstmte.executeUpdate() == 1 && pstmtl.executeUpdate() == 1) {
            JOptionPane.showMessageDialog(null, "One Record Inserted.");
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

#### Code Snippet 4: Add User Code

## **7.5 Delete User**

UserDAO.java class has been used to delete the users who are no longer working in the organization by selecting the name of the employee in the dashboard and clicking the delete employee button. Once the delete employee button is clicked, the request is sent to database and the selected employee is deleted from the database.

```
public int deleteFromEmployee(String value) {  
    int result = 0;  
    try {  
        String query = "DELETE FROM employee WHERE emp_id=?";  
        pstmt = con.prepareStatement(query);  
        pstmt.setString(1, value);  
        int clicked = JOptionPane.showConfirmDialog(null, "Are you sure?", "Delete Confirmation",  
JOptionPane.YES_NO_CANCEL_OPTION);  
        if (clicked == 0) {  
            result = pstmt.executeUpdate();  
            String query1 = "DELETE FROM login WHERE id = ?";  
            pstmt = con.prepareStatement(query1);  
            int a = Integer.parseInt(value);  
            System.out.println(a);  
            a++;  
            pstmt.setInt(1, a);  
            a = pstmt.executeUpdate();  
        }  
    } catch (SQLException e) {  
        e.printStackTrace();  
    }  
    return result;  
}
```

### Code Snippet 5: Delete User Code

## **7.6 Update User Info**

User DTO class has been used to update user information. To update user information the admin selects the user whose information is to be updated and clicks the edit employee button and then a query is sent to the database. If all the edited data is correct then the information is updated.

```
public void updateEmployee(UserDTO employeeUpdate) {  
    try {  
        String sql = "UPDATE employee SET  
emp_name=?,emp_gender=?,emp_contact=?,emp_address=?,emp_salary=?,emp_password=? WHERE  
emp_id=?";  
  
        pstmt = (PreparedStatement) con.prepareStatement(sql);  
  
        pstmt.setString(1, employeeUpdate.getName());  
  
        pstmt.setString(2,employeeUpdate.getGender());  
  
        pstmt.setString(3, employeeUpdate.getContact());  
  
        pstmt.setString(4, employeeUpdate.getAddress());  
  
        pstmt.setString(5, employeeUpdate.getSalary());  
  
        pstmt.setString(6, employeeUpdate.getPassword());  
  
        pstmt.setInt(7, employeeUpdate.getId());  
  
        if (pstmt.executeUpdate() == 1) {  
            JOptionPane.showMessageDialog(null, "One Record Updated.");  
        }  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

### **Code Snippet 6: User Info Modify**

## 7.7 Add Inventory

Product DTO class has been used to add inventory. The admin can enter all the necessary data in the admin's homepage under inventory menu and click the add button. Then the query is sent to the database and if, all the entered data are correct a new inventory is added otherwise an error message is displayed.

```
public void addProduct(ProductDTO product) {  
    try {  
        String sql1 = "INSERT INTO product (name,qty,price,supplier)VALUES (?,?,?,?)";  
        pstmte = (PreparedStatement) con.prepareStatement(sql1);  
        pstmte.setString(1, product.getName());  
        pstmte.setString(2, product.getQty());  
        pstmte.setString(3, product.getPrice());  
        pstmte.setString(4, product.getSupplier());  
        if (pstmte.executeUpdate() == 1) {  
            JOptionPane.showMessageDialog(null, "New product added.");  
        }  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

### Code Snippet 7: Add new Inventory

## 7.8 Remove Inventory

To remove the inventory, the admin can simply choose the inventory to be deleted from the list and click on the remove button.

```
public int deleteFromMenu(String value) {  
    int result = 0;  
    try {  
        String query = "DELETE FROM product WHERE id=?";  
        pstmt = con.prepareStatement(query);  
        pstmt.setString(1, value);  
        int clicked = JOptionPane.showConfirmDialog(null, "Are you sure?", "Delete Conformation",  
JOptionPane.YES_NO_CANCEL_OPTION);  
        if (clicked == 0) {  
            result = pstmt.executeUpdate();  
        }  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
    return result;  
}
```

### Code Snippet 8: Delete Inventory

## 7.9 Update Inventory

To update the inventory information, the admin can simply click on edit button and enter new data for the inventory.

```
public void updateProduct(ProductDTO productUpdate) {  
    try {  
        String sql = "UPDATE product SET name=?,qty=?,price=?,supplier=? WHERE id=?";  
        pstmt = (PreparedStatement) con.prepareStatement(sql);  
        pstmt.setString(1, productUpdate.getName());  
        pstmt.setString(2, productUpdate.getQty());  
        pstmt.setString(3, productUpdate.getPrice());  
        pstmt.setString(4, productUpdate.getSupplier());  
        pstmt.setInt(5, productUpdate.getId());  
        if (pstmt.executeUpdate() == 1) {  
            JOptionPane.showMessageDialog(null, "Product Updated.");  
        }  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

### Code Snippet 9: Update Inventory

## 7.10 TimeLog check

TimeDAO.java class is used to check time log. The system auto fetches the system data and time. When the user clicks *get starting time* button, it will display the current time. It checks with the database whether the user has already checked in today or not and checks the time that will be saved in the database.

```
public void addDummyTime(TimeDTO time) {  
    try {  
        String sql = "INSERT INTO dummy_time(name,start_time,date)VALUES(?,?,?)";  
  
        pstmt = (PreparedStatement) con.prepareStatement(sql);  
        pstmt.setString(1, time.getName());  
        pstmt.setString(2, time.getStartTime());  
        pstmt.setString(3, time.getDate());  
  
        if (pstmt.executeUpdate() == 1) {  
            JOptionPane.showMessageDialog(null, "Checked In");  
        }  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

### Code Snippet 10: Starting Time

## 7.11 TimeLog checkout

The check in time is fetched from the database and the current time is fetched from the system, if user has already checked out then an error message will be shown, JOptionPane.showMessageDialog has been used to display this error message, otherwise a method addTimefunction is called to save the checkout time into the database.

```
public void addItem(TimeDTO time) {
    try {
        String sql = "INSERT INTO
time_log(name,start_time,end_time,working_hrs,day,date)VALUES(?,?,?,?,?)";
        PreparedStatement pstmt = con.prepareStatement(sql);
        pstmt.setString(1,time.getName());
        pstmt.setString(2,time.getStartTime());
        pstmt.setString(3,time.getEndTime());
        pstmt.setString(4,time.getWorkingHrs());
        pstmt.setString(5,time.getDay());
        pstmt.setString(6,time.getDate());
        if(pstmt.executeUpdate() != 1) {
            JOptionPane.showMessageDialog(null,"Checked Out");
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}
```

### Code Snippet 11: End Time

## 8 TESTING

Testing is done in order to check the performance of the system before its actual implementation. The testing phase requires organizations to complete various tests to ensure the accuracy of programmed code. Some of the tests done for this system are as below:

- Unit testing
- System testing

**Unit testing**, or program testing, consists of testing each program separately in the system. Unit testing was done to guarantee that programs are error free, but this goal is realistically impossible.

**System testing** was done to test the functioning of the information system as a whole. It tries to determine whether discrete modules will function together as planned and whether discrepancies exist between the way the system actually works and the way it was conceived.

In order to start testing, application file (.exe) was created first with the help of launch4j software.

**Table 3.**Desktopapplicationtesting

No	Test Performed	Description	Result	Corrections and Result
1	Application file	.exe file was launched	Login page opens	OK
2	Login Window	Text field and radio button	Contains text field for username and password, radio field for selecting either admin or user and login and cancel button.	OK
3	Login using wrong infor-	Entered wrong user	Displays error message.	OK

	mation's	name and password.		
4	Login using correct information's	Entered correct user-name and password	Login successful	OK
5	Adding user with empty data	On the staff page an attempt to add staff without entering any data	An error message was displayed.	OK
6	Adding user	On the userpage all the data was entered and click add employee.	New employee added.	OK
7	Removing user	On the employee page an employee was selected and remove staff button was clicked.	"Are you sure?" message was displayed along with yes, No and cancel button.	OK
8	Removing user	On the staff page a staff member was selected and remove staff button was clicked.	Click Yes. Staff deleted.	OK
9	Adding Inventory without data	On the inventory page a new inventory without entering proper data was entered.	An error message was displayed	OK
10	Adding Inventory	On the inventory page all the necessary details about new inven-	New Inventory added	OK

		tory were entered.		
11	Start time	On the user home page clicked get starting time	Time started	OK
12	End time	On the user home page get end times were clicked.	Time ended	OK
13	Start time twice for same user	On the user home page get started time for second time was clicked	Already Started	OK
14	End work without starting	On the user home page an attempt to click end work button.	The button can't be clicked.	OK
15	My Log	On the user home page my log was clicked.	Working time is displayed, if there is any	OK

## 9 SUMMARY AND CONCLUSION

The main objective for developing this application was to make everyday operations of the Heritage Art and Crafts easy and time saving. With the help of this application, the manager can know who is present at the work place and at what time the employee has started to work. Also, the manager can know at what time an employee left the work place and the total working hours for the day or the month. With the help of this application the manager can add or remove the employee from the list, can view the starting and ending times of the employee and can also view the total working hours of employees for a day and the whole month. In addition, manager can add or remove inventory of the store with the application and the program can also be used to check the inventory level of particular item.

There are two sides of this application the admin side and the user / employee side. An employee can login with his/her login credentials and start their working time at the beginning of the day and end the working time at the end of the day. Apart from that, employees can also view their total working hour for the day and also for the whole month.

In order to make the day to day operation more effortless and less time consuming this project was carried out. The main goal of this project was to develop a desktop based application for the organization

This application has been tested in the Windows environment and it runs smoothly and gives desired results, so we can say that the desired objective has been achieved with this application.

## REFERENCES

/1/Feasibility study: Accessed on 17.03.2018

<https://www.simplilearn.com/feasibility-study-article>

/2/ Economic Feasibility: Accessed on 17.03.2018

<http://www.businessdictionary.com/definition/economic-feasibility.html>

/3/Schedule feasibility: Accessed on 17.03.2018

<https://iamglyka.wordpress.com/2013/02/06/schedule-feasibility/>

/4/Use Case Diagram: Accessed on 19.03.2018

<https://staff.emu.edu.tr/sensevpayanilkan/Documents/courses/ITEC403/seminars/ITEC%20403%20Seminar.ppt>

/5/Deployment Diagram: Accessed on 20.03.2018

[https://ois2.ttu.ee/uusois!/uus\\_ois2.ois\\_public.page?\\_page=DF9175AB402E2383E097925418BCFB6818116956F635EC3A90AECEB2532C8A79&p\\_id=0AD146C4295036D6](https://ois2.ttu.ee/uusois!/uus_ois2.ois_public.page?_page=DF9175AB402E2383E097925418BCFB6818116956F635EC3A90AECEB2532C8A79&p_id=0AD146C4295036D6)

/6/ER Diagram: Accessed on 20.03.2018

[https://commons.wikimedia.org/wiki/Category:Entity-relationship\\_models](https://commons.wikimedia.org/wiki/Category:Entity-relationship_models)

/7/ Class Diagram: Accessed on: 22.03.2018

<http://searchmicroservices.techtarget.com/definition/class-diagram>

/8/Sequence Diagram: Accessed on 22.03.2018

[https://en.wikipedia.org/wiki/Sequence\\_diagram](https://en.wikipedia.org/wiki/Sequence_diagram)

/9/ MySQL: Accessed on 22.03.2018

<http://www.anoneh.com/039.php>