PATNA COLLEGE



A Project Report

Submitted to: Department of Computer Applications

For the partial fulfilment of Bachelor in Computer Applications

# Submitted By:-

Vidya surbhi ROLL–22

Srijan Lal ROLL–05

Rahul Mehta ROLL–09

Session: 2017–2020

A Project Report On

“CANTEEN MANAGEMENT SYSTEM” Sponsored By:

# “ST.MICHAEL’S SCHOOL”

LALGANJ , Vaishali, Bihar 844121

Submitted to:

Department of Computer Applications

PATNA COLLEGE

## CERTIFICATE OF APPROVAL

This is to certify that, The Project Entitled “CANTEEN

MANAGEMENT SYSTEM” Embodies the original work done by

VIDYA SURBHI (ROLL -22), SRIJAN LAL (ROLL -05), and

RAHUL MEHTA (ROLL -09) Department Of Computer science

(BSc. in Computer Applications) Session 2017-2020 of Patna College is

worthy of consideration , For the partial fulfilment of Project, for the

award of degree in Computer Applications from Patna University.

…………………………………………….. .………..…………………………………

INTERNAL EXTERNAL

……………………………………………..

HEAD OF THE DEPARTMENT

## ACKNOWLEDGEMENT

Acknowledgement is the most beautiful page in any project’s final pages. More than a formality, this appears to us the best opportunity to express our gratitude.

First of all we would humbly express our gratitude to Mrs

PRAMOD KUMAR, Pri.ST.MICHAEL’S SCHOOL Lalganj.

We would like to thank Miss Shyamali Rani for her Guidance, Supervision and posing the Confidence in us and assigning the works on this Project.

We are also grateful to Mr.Ahinash Kumar, HOD (BCA) and other faculty members for their consistent guidance and support throughout the Project.

### PREFACE

This Project training is the development of Software, under the demand of “Bachelor in Computer Applications” course to undergo INTERNSHIP in the final year. The objective of the training is that, we should experience the practical work and gain the knowledge to develop the qualities of a working environment.

## 

## INDEX

|  |  |  |
| --- | --- | --- |
| S.No | Topic | Page No. |
| 1 | Acknowledgement | 4 |
| 2 | Preface | 5 |
| 3 | Introduction of Project | 10 |
| 4 | Objective | 11 |
| 5 | Tools and Environment | 12-24 |
| 6 | Existing System | 25 |
| 7 | Proposed System | 26 |
| 8 | System Development Life Cycle | 27-35 |
| 9 | System Analysis   * Entity Relationships Diagrams * Data Flow Diagrams | 36-44 |
| 10 | Runtime Windows | 45-58 |
| 11 | Backend Coding | 59-79 |
| 12 | Frontend Coding | 80-276 |
| 13 | Conclusion | 277 |
| 14 | Bibiliography | 278 |
|  |  |  |

INTRODUCTION

TO

ST. MICHAEL’S SCHOOL

### St. Michael’s School is an CBSE based school,

### Established in 2003 and working successfully under the guidance of Prinilple Pramod Kumar .He is very qualified with high degree in maths MA and B.Ed.The school is situated at Budh Vihar, Mathurapurganj, Lalganj.The affiliation number is 330684.

### The school is running under the trust Parth Sarthi Balya Shilpi Education Trust. The school has proper library, music room, canteen and a big playground with basket ball court and cricket pitch, science lab and medical and health dispensary etc. The school provide basic elementary education up to 10th standard.

INTRODUCTION TO CANTEEN

The main objective of the project on Canteen Management System is to manage the details of Stocks, Sales, Bills, Items, and Employees. It manages all the information about Employee, Items, stocks and spenditure. The project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work for managing the Employee information, Stock, Sales and Reports.

FUNCTIONALITIES PROVIDED BY CANTEEN ARE AS FOLLOWS:

* Provides the basic and proper count of all bills and items.

* It also manage Stock and Sales details of Items.

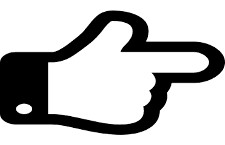
* Tracks all the information of Employees.

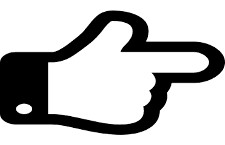
* Provides adding, updating and deleting records functionality.

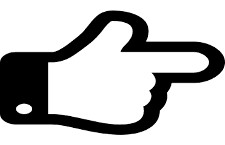
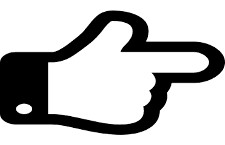
* It also generates bill report on all bill generated.

### OBJECTIVE

 To make the interface user friendly.

 To make the system secure by adding separate login details and password for each user.

 To make the system self-sustainable such that the physical work can be avoided.

To acquire knowledge about billing procedures which will be beneficial for us in future.

To enhance the facilities of existing system. Customer satisfaction and implementation of new technology.

On the whole the system should be profitable than the previous non computerized system.

TOOLS AND

ENVIRONMENTS

* OPERATING SYSTEM

WINDOWS 10

* FRONT END

JAVA

* BACK END

ORACLE 10g

#### WINDOWS 10



**Windows 10** is a series of personal computer operating systems produced by Microsoft as part of its Windows NT family of operating systems. It is the successor to Windows 8.1, and was released to manufacturing on July 15, 2015, and broadly released for retail sale on July 29, 2015.

Some of the new features introduced in Windows 10 include the following:

* User Interface Desktop
* System Security
* Command Line
* Storage Requirements
* Online Service and functionality
* A better overall performance compared to previous versions of Windows.

Windows 10 is available in five main editions for personal computing devices, of which the Home and Pro versions are sold at retail in most countries, and as pre-loaded software on new computers. Home is aimed at home users, while Pro is aimed at small businesses. Each edition of Windows 10 includes all of the capabilities and features of the edition below it, and add additional features oriented towards their market segments; for example, Pro adds additional networking and security features such as BitLocker, Device Guard, Windows Update for Business, and the ability to join a domain.

Ultimate contained a superset of the features from Home Premium and Professional.



#### WINDOWS 10

* Intel Core i5-7200U Processor
* 15.6” HD LCD
* Intel GMA 4500M
* 1 GB DDR3 Memory
* 250 GB HDD
* DVD-Super Multi DL drive
* 802.11b/g/n
* 6-cell Li-ion battery

# JAVA



Java is a set of several computer software and

specifications developed by Sun Microsystems, later acquired by Oracle Corporation that provides a system for developing application software and deploying it in a crossplatform computing environment. Java is used in a wide variety of computing platforms from embedded devices and mobile phones to enterprise servers and supercomputers.

While less common, Java applets run in

secure, sandboxed environments to provide many features of native applications and can be embedded in HTML pages.

## General Purpose

Java capabilities are not limited to any specific application domain rather it can be used in various application domain and hence it is called General Purpose Programming Language.

## Class based

Java is a class based/oriented programming language which means Java supports inheritance feature of object-oriented Programming Language.

## Object oriented

Java is object-oriented means software developed in Java are combination of different types of object.

## Platform Independent

A java code will run on any JVM (JAVA

VIRTUAL MACHINE).Literally we can run same Java Code on Windows JVM LINUX JVM,Mac JVM or any otherJVM practically and get same result every time.

## Distributed

Java Supports distributed System which means we can access files over Internet just by calling the methods. Portable

A Java program when compiled produce bytecodes. Bytecodes are magic. These bytecodes can be transferred via network and can be executed by any JVM, hence came the concept of „Write once, Run Anywhere(WORA)‟.



## ORACLE



Oracle Corporation is the largest software company whose

primary business is [database](http://www.webopedia.com/TERM/D/database.html) products.

Historically, Oracle has targeted high-end [workstations](http://www.webopedia.com/TERM/W/workstation.html) and [minicomputers](http://www.webopedia.com/TERM/M/minicomputer.html) as the [server](http://www.webopedia.com/TERM/S/server.html) [platforms](http://www.webopedia.com/TERM/P/platform.html) to run its database systems. Its [relational database](http://www.webopedia.com/TERM/R/relational_database.html) was the first to support the [SQL language,](http://www.webopedia.com/TERM/S/SQL.html) which has since become the industry standard.

Along with Sun Microsystems, Oracle has been one of the leading champions of [network computers.](http://www.webopedia.com/TERM/N/network_computer.html)Over the last 27 years, Oracle has made tremendous improvements in its core database product. Now, that product is not only the world‟s most reliable and performing database, but also part of a complete software infrastructure for enterprise computing. With each new release comes a sometimes dizzying display of new capabilities and features, sometimes leaving developers, IT managers, and even seasoned DBAs wondering which new features will benefit them most.

Oracle Database 10g offers many new tools that help

DBAs work more efficiently (and perhaps more enjoyably), freeing them for more strategic, creative endeavors not to mention their nights and weekends. Oracle Database

10g really is that big of a deal for DBAs.

### In Accelerates Oracle 10g SQL

Programming training we learn how to create, retrieve, and manipulate objects in Oracle 10g Structured Query Language (SQL).We will also be introduced to Oracle 10g database features and tools. We also develop deeper insight into relational database design and RDBMS operation; learn concepts and specific SQL syntax for extended Oracle data types learn analysis and tuning techniques to increase SQL performance, and master advanced features of Oracle® SQL for large data sets and data warehouses.



NetBeans is a [software](https://en.wikipedia.org/wiki/Software_development)

[development](https://en.wikipedia.org/wiki/Software_development) platform written in [Java.](https://en.wikipedia.org/wiki/Java_(programming_language)) The

NetBeans [Platform](https://en.wikipedia.org/wiki/Platform_(computing)) allows applications to be developed from a set of modular[software components](https://en.wikipedia.org/wiki/Software_component) called modules. Applications based on the NetBeans Platform, including the NetBeans [integrated development environment(](https://en.wikipedia.org/wiki/Integrated_development_environment)IDE), can be extended by [third party developers.](https://en.wikipedia.org/wiki/Third_party_developer)

The NetBeans IDE is primarily intended for development in

Java, but also supports other languages, in particular [PHP,](https://en.wikipedia.org/wiki/PHP) [C](https://en.wikipedia.org/wiki/C_(programming_language))[/C++](https://en.wikipedia.org/wiki/C%2B%2B) and [HTML5](https://en.wikipedia.org/wiki/HTML5)

### VERSIONS OF NETBEANS

* NetBeans IDE 7.0 was released in April 2011. On

August 1, 2011, the NetBeans Team released NetBeans IDE 7.0.1, which has full support for the official release of the [Java SE 7](https://en.wikipedia.org/wiki/Java_version_history#Java_SE_7_.28July_28.2C_2011.29) platform.[[10]](https://en.wikipedia.org/wiki/NetBeans#cite_note-10)

* NetBeans IDE 7.3 was released in February 2013 which added support for HTML5 and web technologies.[[11]](https://en.wikipedia.org/wiki/NetBeans#cite_note-11)

* NetBeans IDE 7.4 was released on October 15, 2013.
* NetBeans IDE 8.0 was released on March 18, 2014.
* NetBeans IDE 8.1 was released on November 4, 2015.

Documentation, and Tooling Support

The NetBeans IDE, which is the software development kit (SDK) of the NetBeans Platform, provides many templates and tools, such as the award winning Matisse GUI Builder that enables you to very easily design your application's layout.

### EXSISTING SYSTEM

#### LIMITATIONS

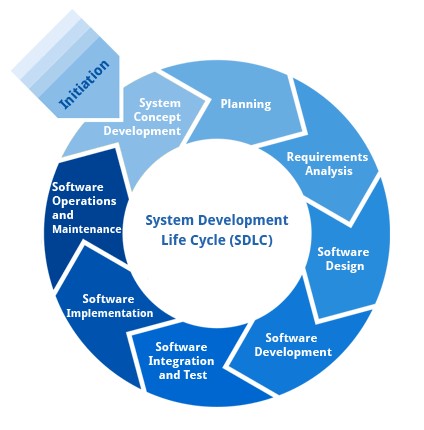
* Time consuming
* Fraud at each step
* Inappropriate procedural flow
* Limited report
* Less security
* Inefficient method of record keeping
* More man power
* Consume large volume of paper work
* Needs manual calculations.

### PROPOSED SYSTEM

The aim of proposed system is to develop a system of improved facilities. The proposed system can overcome all the limitations of the existing system. The system provides proper security and reduces the manual work.

* Security of data.
* Ensure data accuracy.
* Minimize manual data entry.
* Minimum time needed for the various processing.
* Greater efficiency.
* Better service.
* User friendly.

## S D L C



System development life cycle is a process for planning, creating, testing and deploying system. A system begins (is born) with decision to analyse user requirement. There are various software development approaches defined and designed which are used/employed during development process of software, these approaches are also referred as “Software Development Process Models”.

[Waterfall model,](http://istqbexamcertification.com/what-is-waterfall-model-advantages-disadvantages-and-when-to-use-it/) [incremental model,](http://istqbexamcertification.com/what-is-incremental-model-advantages-disadvantages-and-when-to-use-it/) [V-model,](http://istqbexamcertification.com/what-is-v-model-advantages-disadvantages-and-when-to-use-it/) [iterative model,](http://istqbexamcertification.com/what-is-iterative-model-advantages-disadvantages-and-when-to-use-it/) [RAD model,](http://istqbexamcertification.com/what-is-rad-model-advantages-disadvantages-and-when-to-use-it/) [Agile model,](http://istqbexamcertification.com/what-is-agile-model-advantages-disadvantages-and-when-to-use-it/) [Spiral model,](http://istqbexamcertification.com/what-is-spiral-model-advantages-disadvantages-and-when-to-use-it/) [Prototype model](http://istqbexamcertification.com/what-is-prototype-model-advantages-disadvantages-and-when-to-use-it/) etc. Each process model follows a particular life cycle in order to ensure success in process of software development.

Software life cycle models describe phases of the software cycle and the order in which those phases are executed. Each phase produces deliverables required by the next phase in the life cycle. Requirements are translated into design. Code is produced according to the design which is called development phase. After coding and development

the testing verifies the deliverable of the implementation phase against requirements.

PHASES OF SDLC

1. Preliminary investigation
2. Feasibility study
3. System analysis
4. System Design
5. Software coding
6. System testing
7. Implementation
8. Evaluation/Maintainance





PRELIMINARY INVESTIGATION

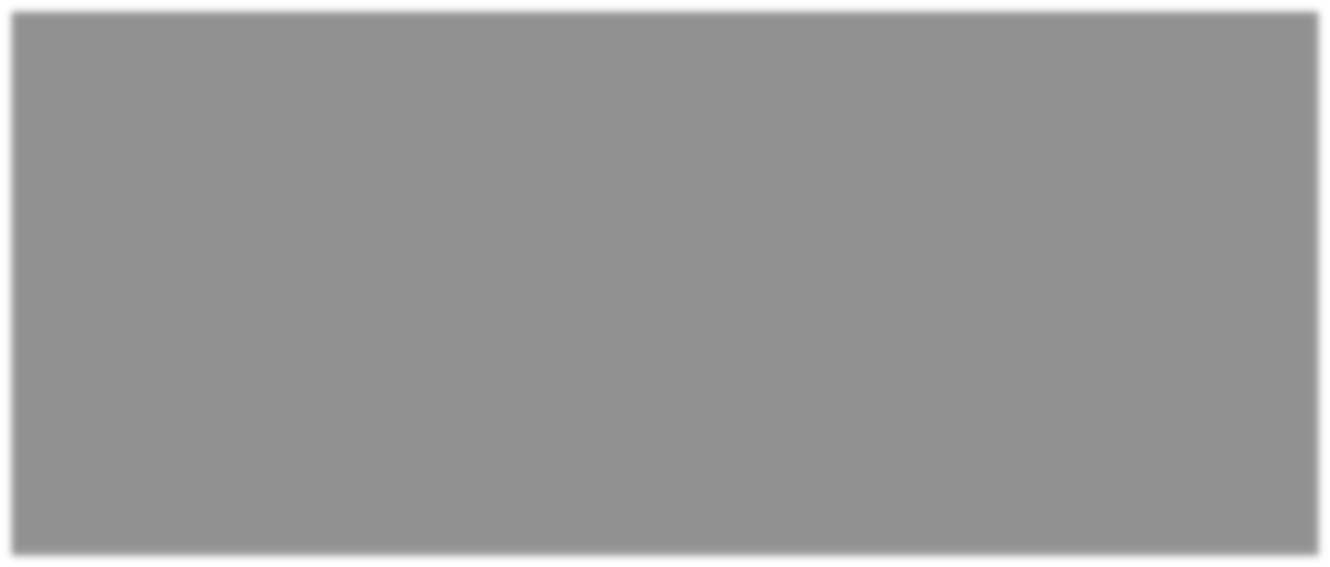
Preliminary Investigation is one of the activities which is done under system development life cycle. SDLC is the well defined process by which a system is conceived, developed and implemented.There is some problem which arises in the existing system due to which there is a need require to develop a new system and outline the functional requirement for the project.when the real problem of the existing system is identified by the analyst then only any work is done,unless it will lead to wastage of effort at a later stage. It determines the boundaries of the project by taking into consideration the limitation of the available resources in EBMS with the help of the feasibility study. The main objective is to design a system, so that it automates the functionalities of

EBMS.

In the existing system many works regarding equipments are done manually which is quite cumbersome to handle all management of such big organization. In this software we have applied our best effort to handle all jobs regarding the bill generation. In this software we have tried to handle the details of an users,connections,bill payment etc.. Thus, through this software we have tried to reduce the problem from the existing system. Various reports based in the data entered by the user at office are generated. These reports are helpful in Manpower management decisions.

The benefits of the new proposed system:-

* It will reduce the workload of every operator.
* Now organization is implied in structured in ordered manner.
* System is accurate,timely and comprehensive.
* According to the need of the future the system can be changed because it is flexible.
* It provides high level of security and user level access.
* There are no chance of duplicity to be ordered by updating at one place.
* It will avoid the occuring of errors.



In case the system proposal is acceptable to the management, the next phase is to examine the feasibility of the system. The feasibility study is basically the test of the proposed system in the light of its workability, meeting user‟s requirements, effective use of resources and of course, the cost effectiveness.

To present,as pointed out earlier, most of the work done manually.And there is no denying the fact that computers would quicken the work,if a solution is developed prudently.so, the first question that whether computerization is necessary is not so important. The main concern was then regarding the selection of software/hardware/mode of working/interface.

So, the types of feasibility studies we done in our software are as below:

ECONOMICAL FEASIBILITY

Major cost at this point of time would be incurred towards acquiring an RDBMS package only. The Bill management and the concerned authorities in order to get this much facility is ready to all the basic software according to the requirement. No extra software or hardware needs to be acquired other than explained in the heading “proposed system structure”.

Major Benefits Expected From this system are:

* Increased speed of work with many of the mundane task being automatically done.
* Less time consuming.
* More accurate. Certainly, computers are less prone to error than human beings.
* More flexible.
* As benefits expected far outweigh the cost to be incurred, the new system is very much economically feasible.

TECHNICAL FEASIBILITY

HARDWARE: - Hardware selected has been examined against the processing capacity, and memory requirement at both client and server ends and found satisfactory for current as well as new future workload.

SOFTWARE: - Cost benefits analysis for this part was challenging. Some option were there regarding the selection of DBMS package, front end tools, etc.

INTERFACE:-Hardware available can easily support the demand of graphical user Interface is a natural choice.

BACK END:-When some database handling is to be considered, RDBMS are the best bet these days. But the choice was between SQL server and Oracle. The detailed explanation about Oracle in next section is self-explanatory regarding this aspect.

FRONT END:- Again, as front end, after selection of Oracle as backend, the feasible option left were visual Basic, JAVA, V.B. , .Net since JAVA was better and easier solution for the software so as the front end tool JAVA was decided upon.

OPERATIONAL FEASIBILITY

As the old system was paper based, so if any type of calculation is needed to be performed such that preparing report of any particular year, calculation on bill, etc. it would be very cumbersome process to find the related records from the number of registers on the department but the new system is computerized desktop based system. So, all these work become so easy after the implementation of proposed system. So, the proposed system is operationally feasible.

TIME FEASIBILITY

The existing system was paper based so, the records of user,

Connections, calculation of electric bill, payments record, etc. all were maintained in registers which was a very time consuming and hectic process. But our proposed system is computerized and a desktop based application. So, every mentioned step becomes so easy that each can be done only by just clicking. Thus the proposed system is very less time consuming as compared to old existing system. Hence, our proposed system is also time feasible.

SYSTEM

ANALYSIS

System analysis is a process of studying the

System, its components and it’s inter relatedness for the purpose of finding problems areas and finding the remedial the remedial solution and to suggest to the management with the intent of improving the quality of the product.

ENTITY

RELATIONSHIP DIAGRAM

An entity relationship model, also called an entity-

relationship(ER) diagram, is a graphical representation of entities and their relationships to each other, typically used in computing in regard to the organization of [data](http://www.webopedia.com/TERM/D/data.html) within [databases](http://www.webopedia.com/TERM/D/database.html) or information systems. An entity is a piece of data-an [object o](http://www.webopedia.com/TERM/O/object.html)r concept about which data is stored.

An entity–relationship model is the result of using a systematic process to describe and define a subject area of business data. It does not define business process; only visualize business data.

The data is represented as components (entities) that are linked with each other by relationships that express the dependencies and requirements between them, such as: one

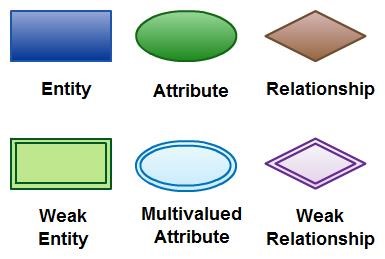
building may be divided into zero or more apartments, but one apartment can only be located in one building.

Entities may have various properties (attributes) that characterize them.

Diagrams created to entities, attributes, and relationships graphically are called entity–relationship diagrams.

SYMBOLS USED IN

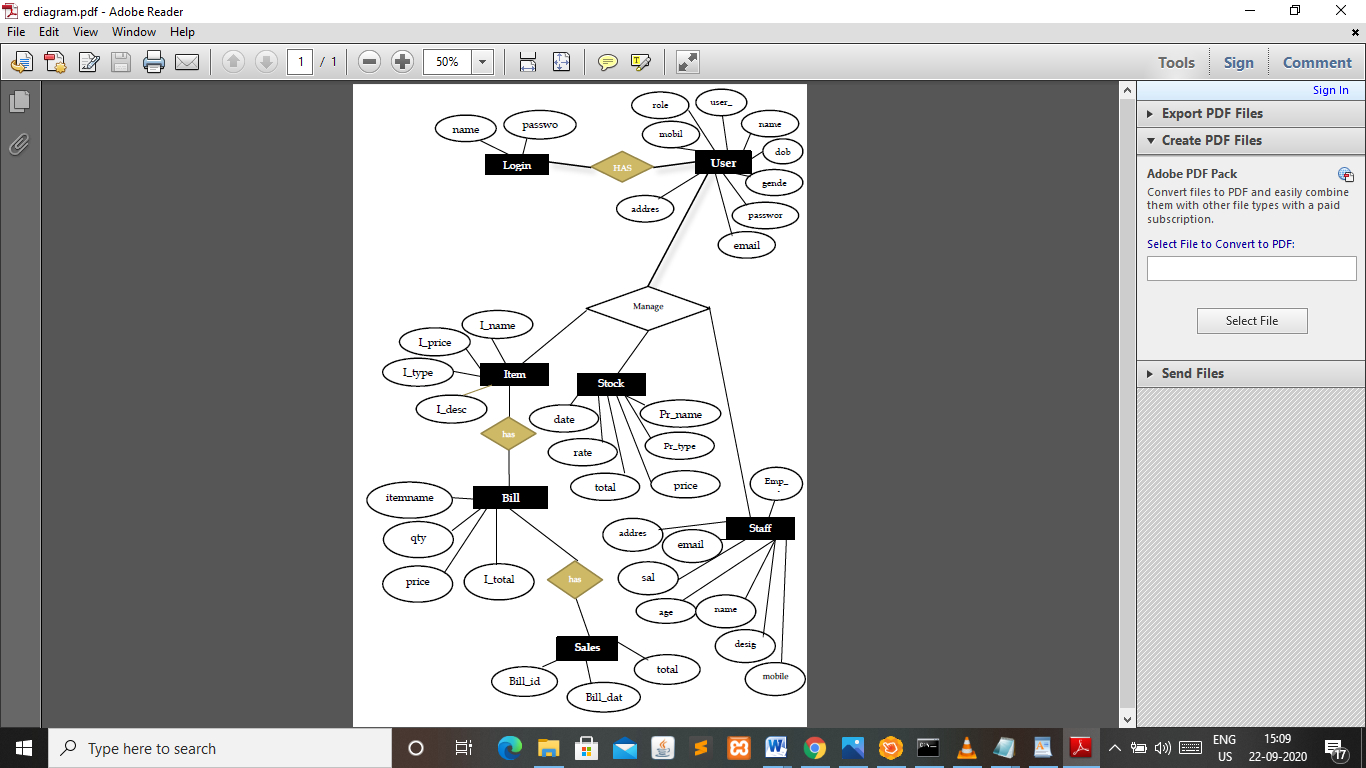
ERD.



#### ENTITY RELATIONSHIP DIAGRAM

**User**

**User**



DATA FLOW DIAGRAM

A Data Flow Diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated.

DFDs can also be used for the visualization of data processing (structured design).

A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or in parallel (which is shown on a flowchart).

##### HISTORY

Data flow diagrams were proposed by Larry Constantine, the original developer of structured design, based on Martin and Estrin's "Data Flow Graph" model of computation. Starting in the 1970s, data flow diagrams (DFD) became a popular way to visualize the major steps and data that involved in software system processes. DFDs were usually used to show data flows in a computer system, although they could in theory be applied to business process modelling.

DFD were useful to document the major data flows or to explore a new high-level design in terms of data flow.

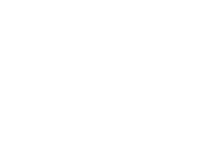
##### SYMBOLS REPRESENT

ENTITY

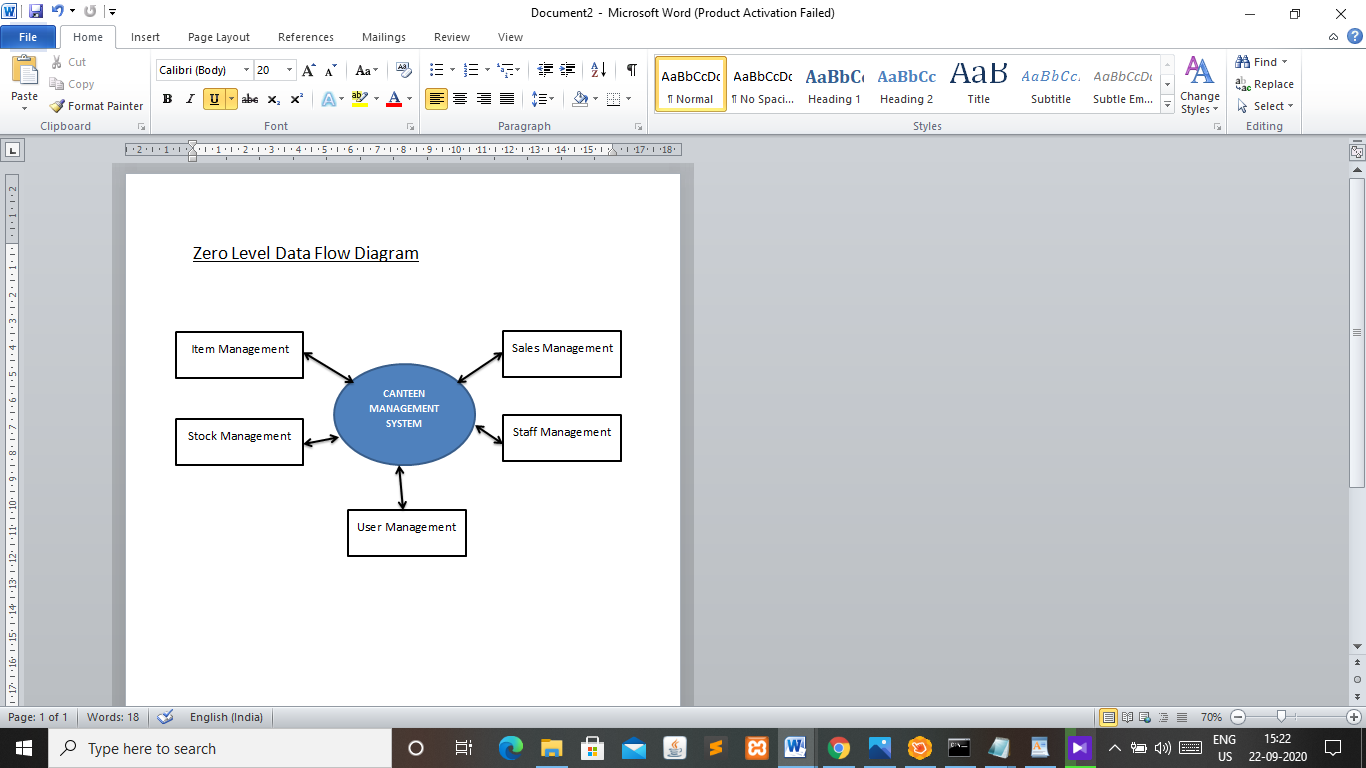
FLOW OF DATA

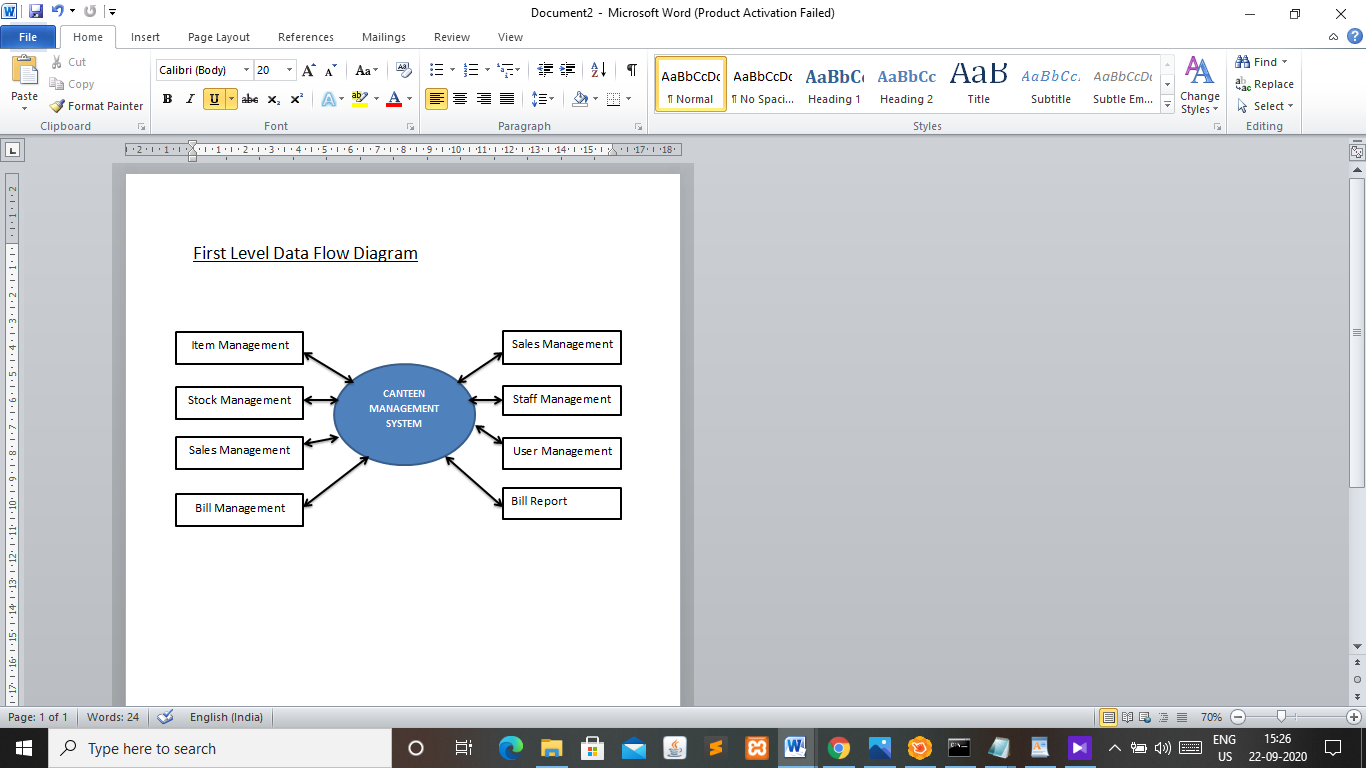
PROCESS

DATA STORE

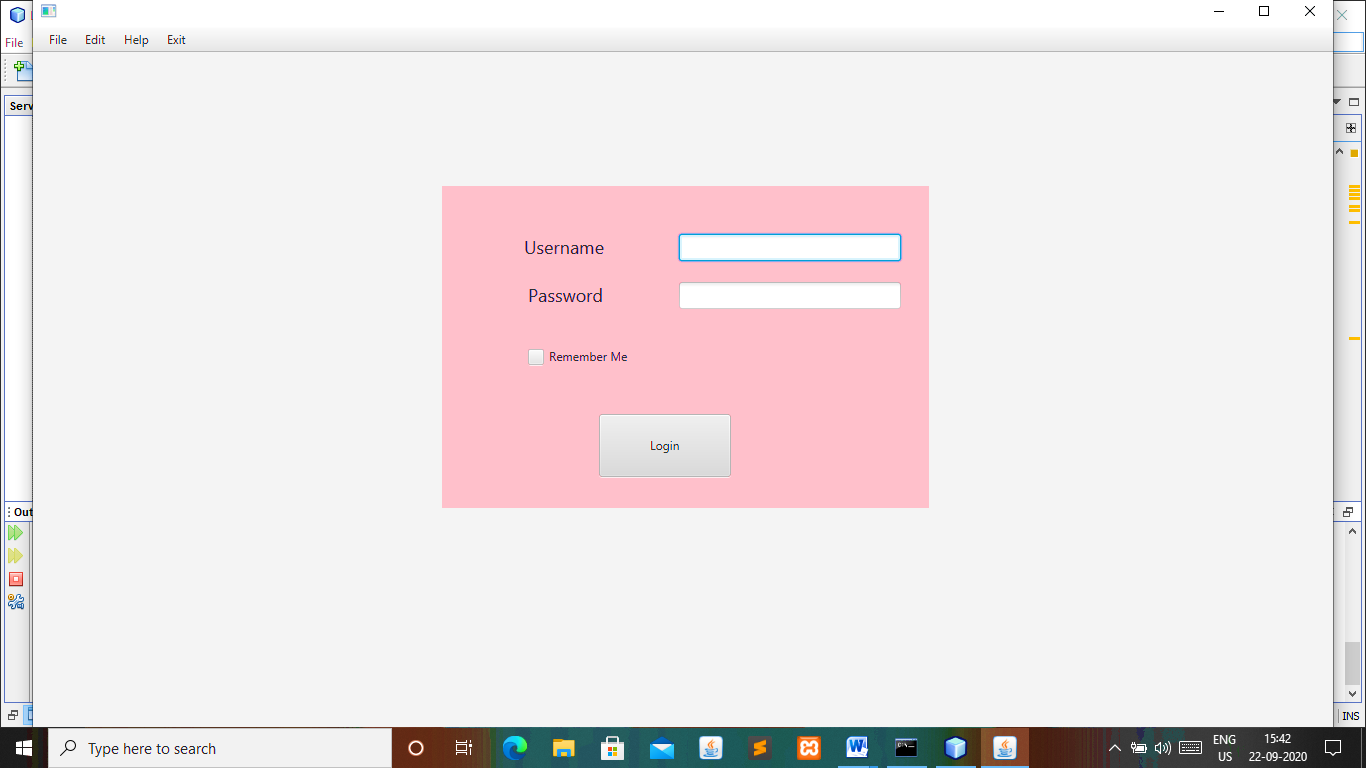


Data Flow Diagram





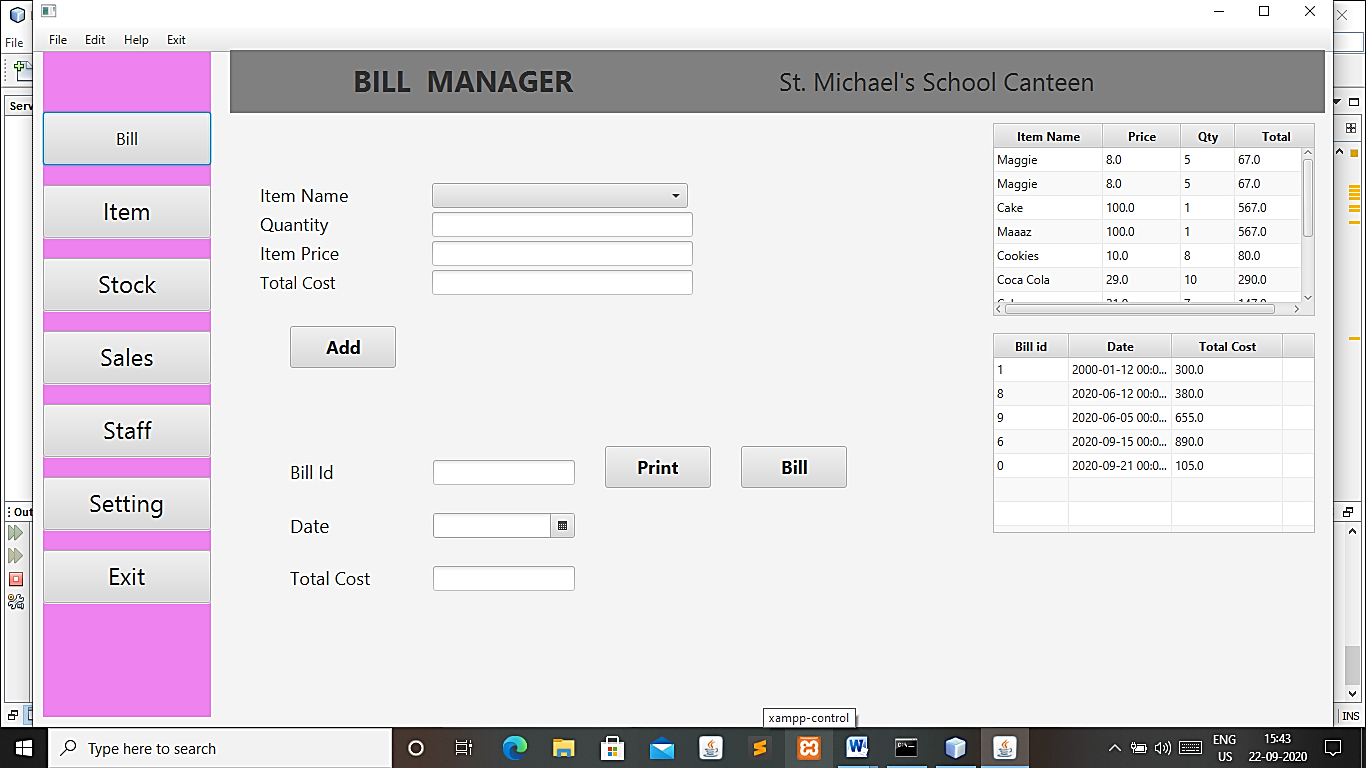
###### LOGIN WINDOW



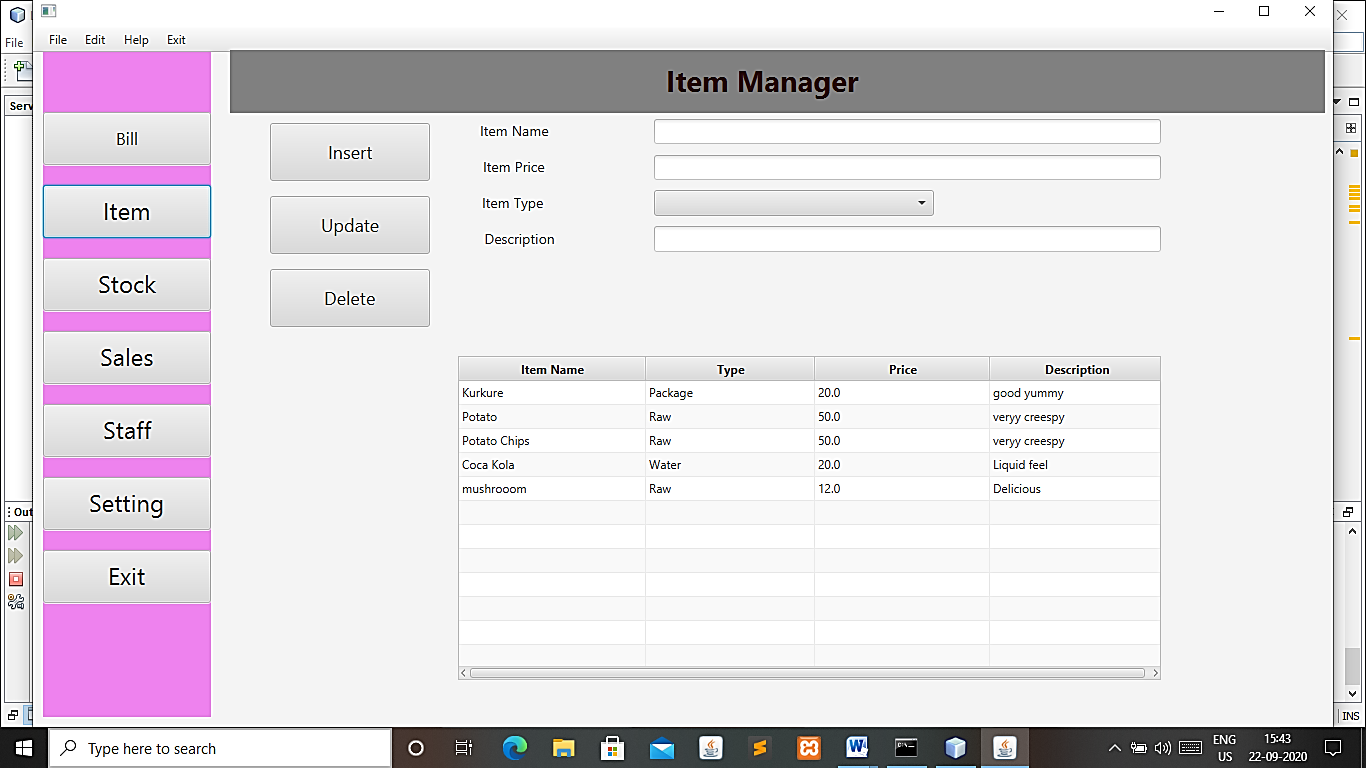
AFTER LOGIN WINDOW



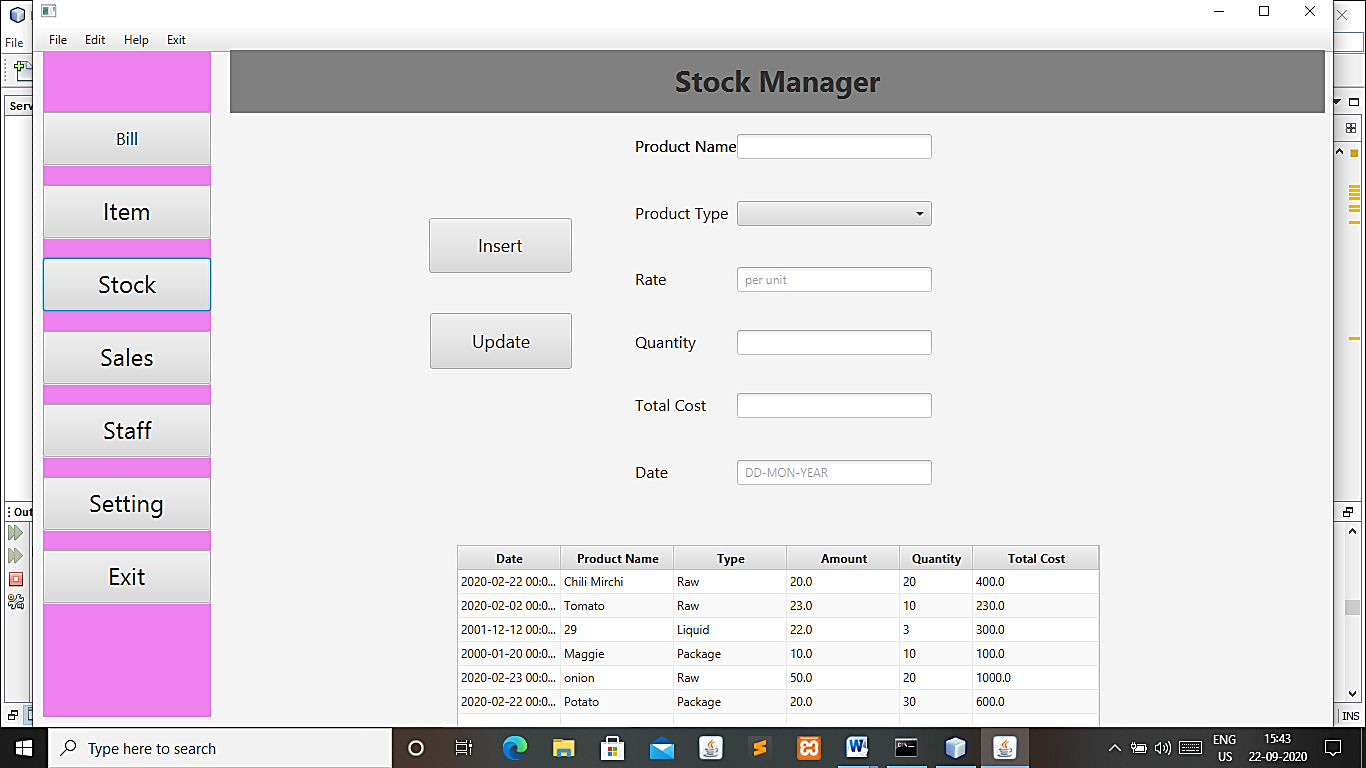
BILL WINDOW



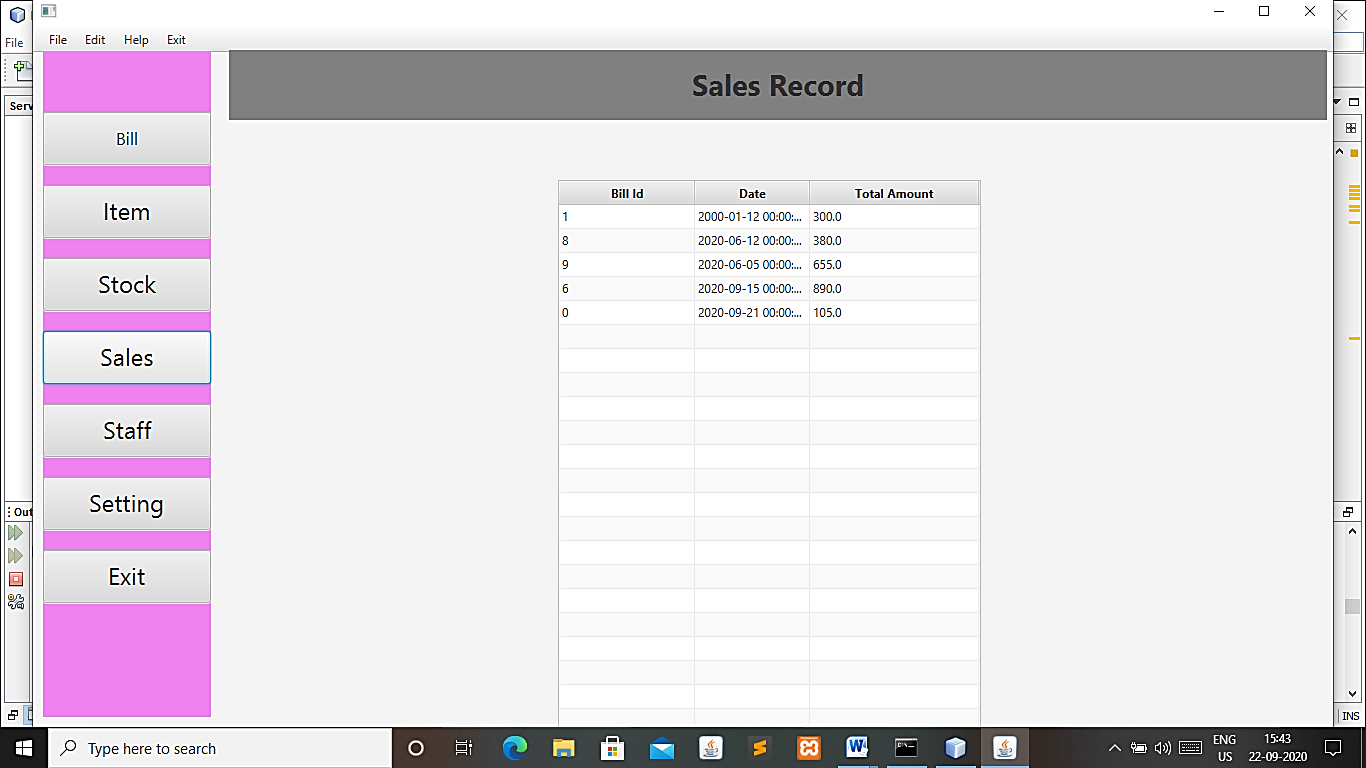
ITEM WINDOW



STOCK WINDOW

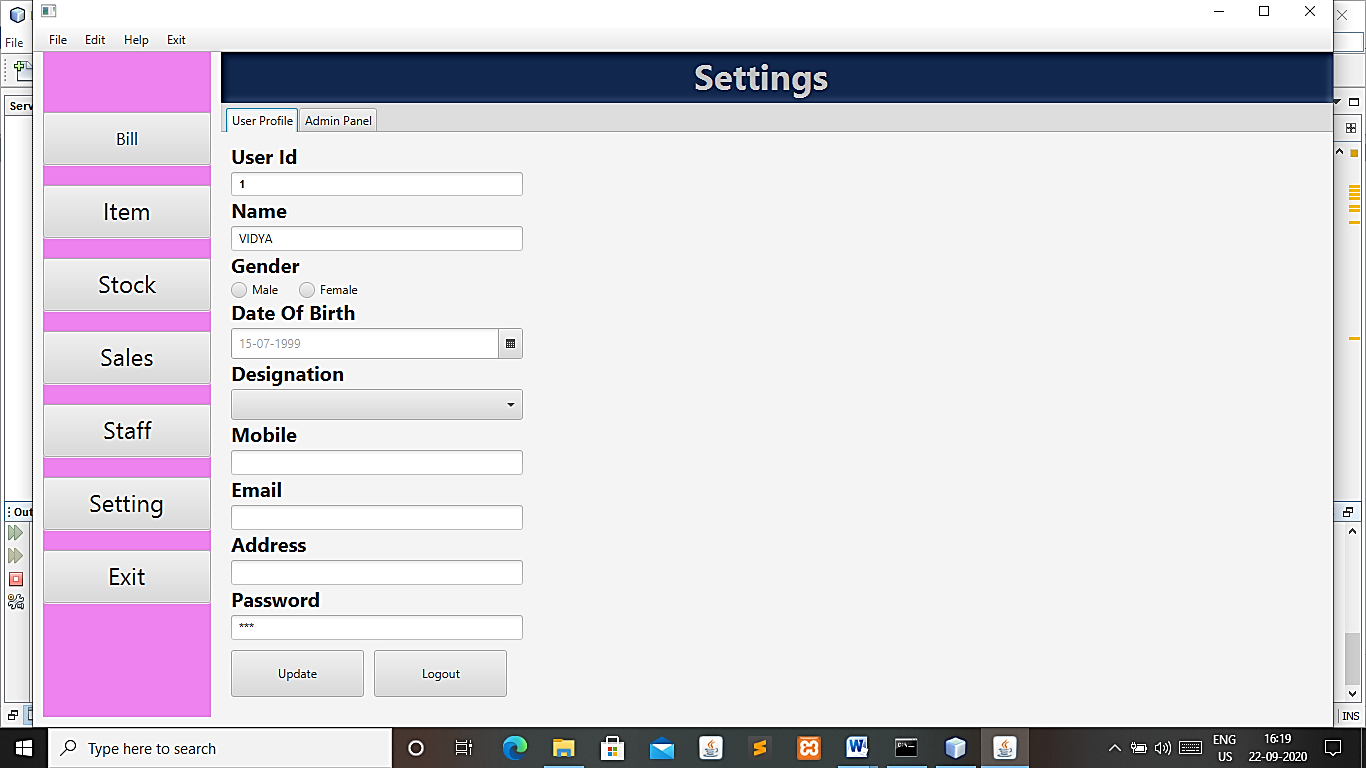


SALES WINDOW

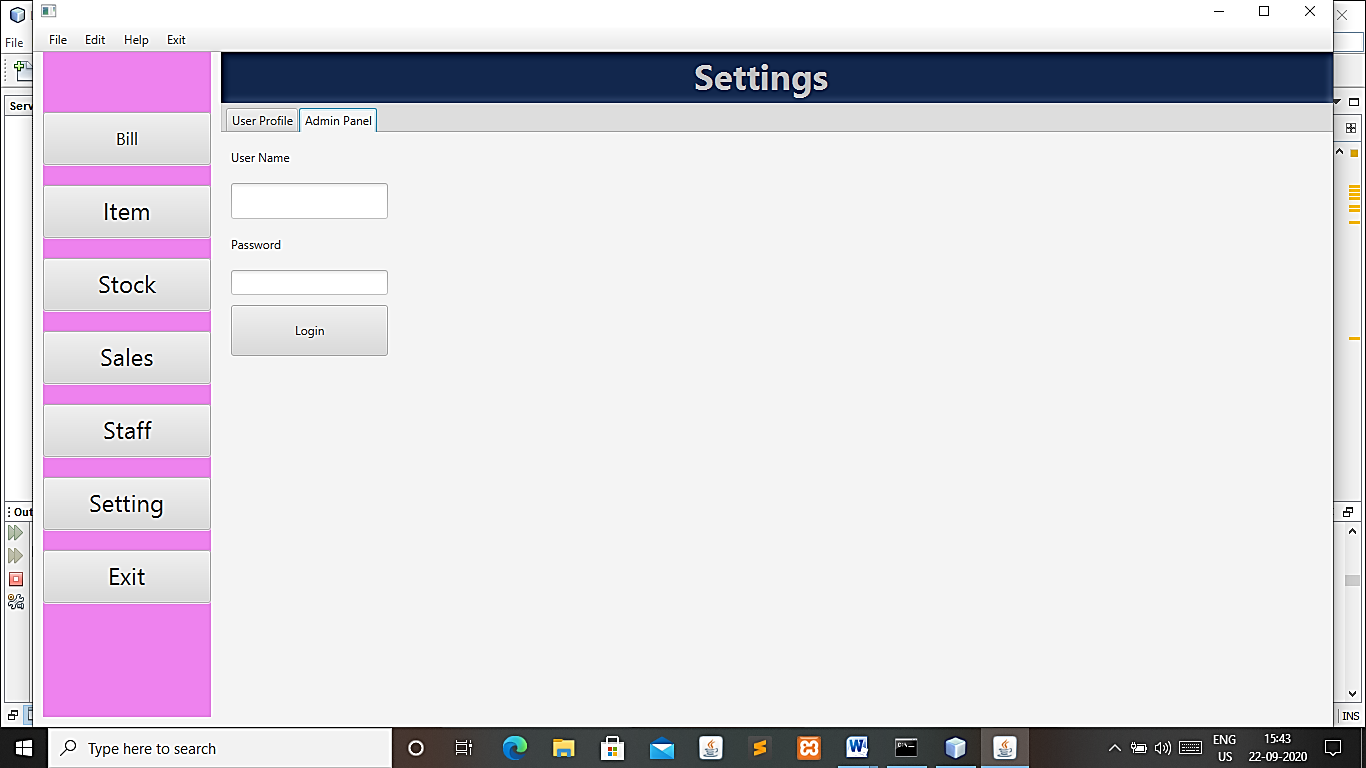


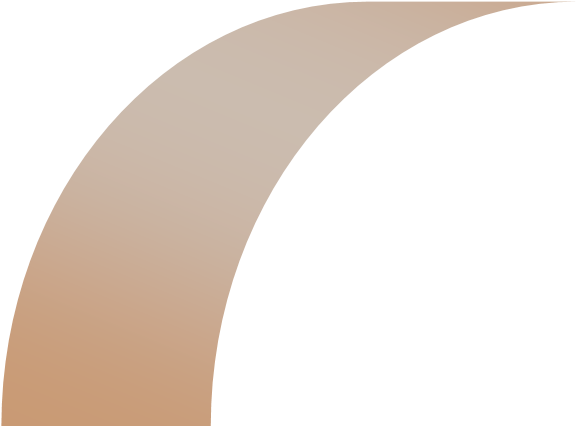
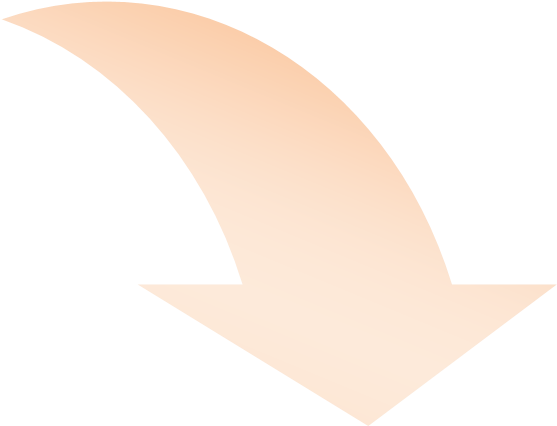
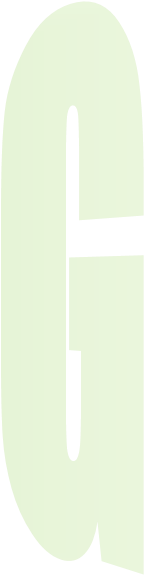
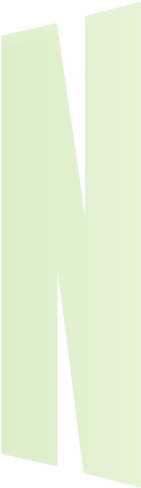
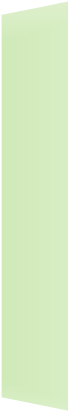
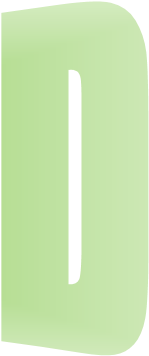
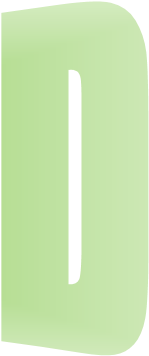
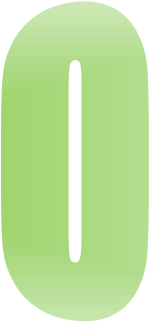
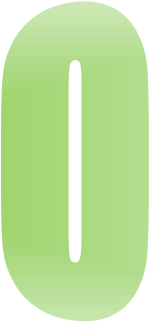
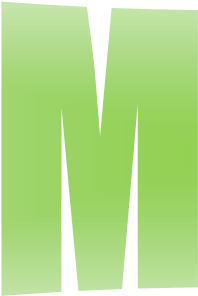
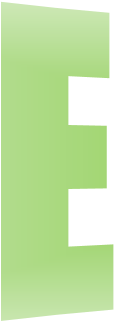
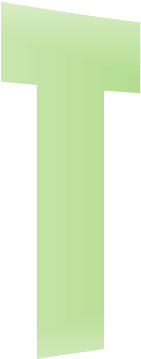
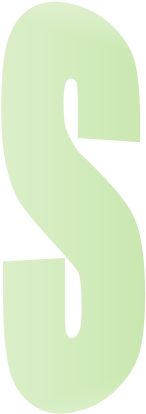
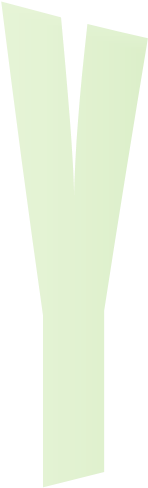
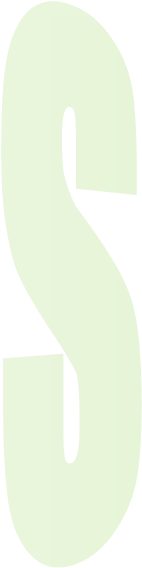
SETTING WINDOW

USER PROFILE:-



ADMIN PANNEL:-





BACKEND DATABASE (table)

CREATE DATABASE

create user canteen identified by canteen

/

grant dba to canteen

/

connect canteen/canteen

/

USERS TABLE

create table users (

user\_id number(10) primary key,

name varchar2(10),

gender varchar2(10),

dob date,

desig varchar2(20),

mobile varchar2(15),

email varchar2(50),

address varchar2(100),

role varchar2(10),

password varchar2(15))

/

Temp users table

create table temp\_users

as

select \* from users

where 1=2;

alter table temp\_users

drop column user\_id;

Trigger

create or replace trigger ai\_temp\_users

after insert on temp\_users

for each row

declare

user\_id users.user\_id%type;

begin

select num into user\_id from autoid where table\_name='users';

user\_id:=user\_id+1;

update autoid set num=user\_id where table\_name='users';

insert into users values(

user\_id,

:NEW.NAME,

:NEW.GENDER,

:NEW.DOB,

: NEW.DESIG,

:NEW.MOBILE,

:NEW.EMAIL,

:NEW.ADDRESS,

:NEW.ROLE,

: NEW.PASSWORD

);

end;

/

AUTO ID TABLE

create table autoid (

table\_name varchar2(100) ,

num number(15),

pattern varchar2(10)

)

/

insert into autoid values('bill\_gen',0,null)

/

insert into autoid values('users',0,null)

/

ITEM TABLE

create table item\_mstr

(item\_name

varchar2(20)Primary key,

item\_price number(10,2),

item\_descr varchar(40), type varchar(10))

/

BILL DETAILS TABLE

create table bill\_item( bi\_name varchar(20) constraint item\_name references item\_mstr(item\_name),

bqty number(10),

bitem\_name number(10,2) ,

btotal\_price number(10,2))

/

STOCK TABLE

create table stock\_mstr(pr\_name varchar(20),pr\_type varchar(20),qty number(4),pr\_amt number(10,2),total\_cost number(10,2),pr\_date date)

/

STAFF TABLE

create table staff\_table (emp\_id number(3),

emp\_name varchar(20),

age number(3),

design varchar(20),

salr number(10),

mob number(10),

email varchar(20),

addre varchar(25))

/

SALES TABLE

create table bill\_gen(b\_id number(10) primary key,

bill\_date date,

bill\_total number(10,2))

/

FRONT END

CANTEEN

Controller

ITEM CONTROLLER

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package loginpage;

import beans.UserBeans;

import java.io.IOException;

import java.net.URL;

import java.util.Optional;

import java.util.ResourceBundle;

import java.util.logging.Level;

import java.util.logging.Logger;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Parent;

import javafx.scene.control.Alert;

import javafx.scene.control.Alert.AlertType;

import javafx.scene.control.Button;

import javafx.scene.control.ButtonType;

import javafx.scene.control.ComboBox;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.TextField;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.input.MouseEvent;

import model.Item;

/\*\*

\* FXML Controller class

\*

\* @author DELL

\*/

public class ItemController implements Initializable {

@FXML

private TextField itemname;

@FXML

private ComboBox itemtype;

@FXML

private TextField itemdesc;

private Item itemobj;

@FXML

private TextField itemprice;

@FXML

private TableColumn<UserBeans,String> colname;

@FXML

private TableColumn<UserBeans,String> coltype;

@FXML

private TableColumn<UserBeans,String> colprice;

@FXML

private TableView<UserBeans> tableItem;

@FXML

private Button updateitem;

@FXML

private Button deleteitem;

@FXML

private TableColumn<UserBeans,String> coldesc;

@FXML

private Button itemreport;

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

itemobj = new Item();

itemtype.getItems().addAll("Water","Raw","Package");

colname.setCellValueFactory(new PropertyValueFactory<>("Iname"));

colprice.setCellValueFactory(new PropertyValueFactory<> ("Iprice"));

coldesc.setCellValueFactory(new PropertyValueFactory<> ("Idesc"));

coltype.setCellValueFactory(new PropertyValueFactory<> ("Itype"));

tableItem.setItems(itemobj.showTableRecords());

// TODO

}

@FXML

private void insertitem(ActionEvent event) {

UserBeans e = new UserBeans();

e.setIname(itemname.getText());

e.setItype(itemtype.getValue().toString());

e.setIdesc(itemdesc.getText());

e.setIprice(Double.parseDouble(itemprice.getText()));

boolean flag = itemobj.insertitem(e);

if(!flag){

Alert alert = new Alert(Alert.AlertType.INFORMATION);

alert.setHeaderText("Item Information Successfully Inserted!");

alert.show();

//clearRecord();

tableItem.setItems(itemobj.showTableRecords());

}

else{

Alert alert = new Alert(Alert.AlertType.WARNING);

alert.setHeaderText("Somethig went wrong, try again!");

alert.show();

}

}

private void clearRecord(){

itemname.setText("");

itemtype.setValue("");

itemdesc.setText("");

itemprice.setText("");

}

@FXML

private void update(ActionEvent event) {

UserBeans u = new UserBeans();

//u.setName(itemname.getText());

u.setIprice(Double.parseDouble(itemprice.getText()));

u.setIdesc(itemdesc.getText());

u.setItype(itemtype.getValue().toString());

boolean flag=itemobj.updateItem(u);

if(flag){

Alert alert=new Alert(Alert.AlertType.WARNING);

alert.setHeaderText("Item Information not updated");

alert.show();

}

else{

Alert alert=new Alert(Alert.AlertType.INFORMATION);

alert.setHeaderText("Item Information successfully updated!");

alert.show();

tableItem.setItems(itemobj.showTableRecords());

}

}

@FXML

private void delete(ActionEvent event) {

String name = itemname.getText();

boolean flag = itemobj.deleteItem(name);

if(flag){

Alert alert=new Alert(Alert.AlertType.INFORMATION);

alert.setHeaderText("Item Information successfully Deleted!");

alert.show();

tableItem.setItems(itemobj.showTableRecords());

clearRecord();

}

else{

Alert alert=new Alert(Alert.AlertType.WARNING);

alert.setHeaderText("Something went wrong try again!");

alert.show();

}

}

@FXML

private void setdata(MouseEvent event) {

try{

String name = tableItem.getSelectionModel().getSelectedItem().getIname();

printItem(itemobj.searchItem(name));

}

catch(NullPointerException ex){

Alert alert = new Alert(AlertType.WARNING);

alert.setHeaderText("Empty Record Selected!");

alert.show();

}

}

public void printItem(UserBeans e){

if(e!=null){

itemname.setText(e.getIname());

itemprice.setText(e.getIprice()+"");

itemtype.setValue(e.getItype()+"");

itemdesc.setText(e.getIdesc());

}

}

}

BILL CONTROLLER

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package loginpage;

import beans.UserBeans;

import java.net.URL;

import java.time.LocalDate;

import java.time.format.DateTimeFormatter;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.Initializable;

import javafx.scene.control.Alert;

import javafx.scene.control.Alert.AlertType;

import javafx.scene.control.Button;

import javafx.scene.control.ComboBox;

import javafx.scene.control.DatePicker;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.TextField;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.input.MouseEvent;

import javafx.scene.layout.BorderPane;

import static jdk.nashorn.internal.objects.Global.getDate;

import model.Bill;

/\*\*

\* FXML Controller class

\*

\* @author DELL

\*/

public class BillController implements Initializable {

@FXML

private BorderPane borderpanebill;

@FXML

private TextField subtotal;

@FXML

private ComboBox billitemname;

@FXML

private TextField totalcost;

@FXML

private TextField qty;

@FXML

private Button additembtn;

@FXML

private TextField billid;

@FXML

private DatePicker billdate;

@FXML

private TextField billtotal;

@FXML

private Button printbillbtn;

@FXML

private Button billgenbtn;

@FXML

private TableView<UserBeans> tablebill1;

@FXML

private TableColumn<UserBeans, String> ciname;

@FXML

private TableColumn<UserBeans, String> cprice;

@FXML

private TableColumn<UserBeans, String> cqty;

@FXML

private TableColumn<UserBeans, String> ctotal1;

@FXML

private TableView<UserBeans> tablebill2;

@FXML

private TableColumn<UserBeans, String> cbid2;

@FXML

private TableColumn<UserBeans, String> cdate;

@FXML

private TableColumn<UserBeans, String> ctotal2;

private Bill billobj ;

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

billobj=new Bill();

billitemname.getItems().addAll("Maggie","Cake","Biscuit","Cookies","Coca Cola","Maaaz","Water","Lays","Chips","Others");

cbid2.setCellValueFactory(new PropertyValueFactory<>("bid"));

cdate.setCellValueFactory(new PropertyValueFactory<>("bdate"));

ctotal2.setCellValueFactory(new PropertyValueFactory<>("btotal"));

ciname.setCellValueFactory(new PropertyValueFactory<> ("biname"));

cqty.setCellValueFactory(new PropertyValueFactory<> ("billqty"));

cprice.setCellValueFactory(new PropertyValueFactory<> ("bprice"));

ctotal1.setCellValueFactory(new PropertyValueFactory<> ("itemtotal"));

tablebill1.setItems(billobj.showTableRecords());

tablebill2.setItems(billobj.showTableRecords1());

}

private void setTableData(MouseEvent event) {

try{

String name = tablebill1.getSelectionModel().getSelectedItem().getBiname();

printBill(billobj.searchItem(name));

}

catch(NullPointerException ex){

Alert alert = new Alert(Alert.AlertType.WARNING);

alert.setHeaderText("Empty Record Selected!");

alert.show();

}

}

@FXML

private void addaction(ActionEvent event) {

UserBeans e = new UserBeans();

e.setBiname(billitemname.getValue().toString());

e.setBillqty(Integer.parseInt(qty.getText()));

e.setBprice(Double.parseDouble(subtotal.getText()));

e.setItemtotal(Double.parseDouble(totalcost.getText()));

/\* double tot = Double.parseDouble(totalcost.getText());

int total = (int)tot;

int quant = Integer.parseInt(qty.getText());

double price = Double.parseDouble(subtotal.getText());

tot = quant \* price;

e.setItemtotal(Double.parseDouble(totalcost.getText()));\*/

boolean flag = billobj.insertitem(e);

if(!flag){

Alert alert = new Alert(Alert.AlertType.INFORMATION);

alert.setHeaderText("Item Information Successfully Inserted!");

alert.show();

//totalcost.setText(totalcost+"");

//clearRecord();

tablebill1.setItems(billobj.showTableRecords());

}

else{

Alert alert = new Alert(Alert.AlertType.WARNING);

alert.setHeaderText("Somethig went wrong, try again!");

alert.show();

}

}

public void printBill(UserBeans e){

if(e!=null){

billitemname.setValue(e.getBiname());

qty.setText(e.getBillqty()+"");

subtotal.setText(e.getBprice()+"");

totalcost.setText(e.getItemtotal()+"");

}

}

@FXML

private void printaction(ActionEvent event) {

}

@FXML

private void billaction(ActionEvent event) {

UserBeans e = new UserBeans();

//e.setBid(Integer.parseInt(billid.getText()));

e.setBdate((String)getDate(billdate.getValue()));

e.setBtotal(Double.parseDouble(billtotal.getText()));

boolean flag = billobj.billrun(e);

if(!flag){

Alert alert = new Alert(Alert.AlertType.INFORMATION);

alert.setHeaderText("Bill generated successfully!");

alert.show();

clearRecord();

tablebill2.setItems(billobj.showTableRecords1());

}

else{

Alert alert = new Alert(Alert.AlertType.WARNING);

alert.setHeaderText("Somethig went wrong, try again!");

alert.show();

}

}

public String getDate(LocalDate date){

DateTimeFormatter formatter = DateTimeFormatter.ofPattern("dd-MMM-yyyy");

String dt = date.format(formatter);

return dt;

}

private void clearRecord() {

billitemname.setValue("");

qty.setText("");

subtotal.setText("");

totalcost.setText("");

}

}

STOCK CONTROLLER

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package loginpage;

import beans.UserBeans;

import java.net.URL;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.Initializable;

import javafx.scene.control.Alert;

import javafx.scene.control.Alert.AlertType;

import javafx.scene.control.Button;

import javafx.scene.control.ComboBox;

import javafx.scene.control.DatePicker;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.TextField;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.input.MouseEvent;

import javafx.scene.layout.BorderPane;

import model.Stock;

/\*\*

\* FXML Controller class

\*

\* @author DELL

\*/

public class StockController implements Initializable {

@FXML

private Button insertbtn;

@FXML

private BorderPane borderpanestock;

@FXML

private TextField productname;

@FXML

private TextField amount;

@FXML

private TextField quantity;

@FXML

private TextField totalcost;

@FXML

private TextField date;

@FXML

private ComboBox producttype;

private Stock stockobj;

@FXML

private TableView<UserBeans> tablestock;

@FXML

private TableColumn<UserBeans, String> coldate;

@FXML

private TableColumn<UserBeans, String> colprdname;

@FXML

private TableColumn<UserBeans, String> coltype;

@FXML

private TableColumn<UserBeans, String> colamt;

@FXML

private TableColumn<UserBeans, String> colqty;

@FXML

private TableColumn<UserBeans, String> coltotal;

@FXML

private Button printbtn;

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

stockobj = new Stock();

producttype.getItems().addAll("Water","Juice","Liquid","Raw","Package");

colprdname.setCellValueFactory(new PropertyValueFactory<>("prdname"));

coltype.setCellValueFactory(new PropertyValueFactory<>("prdtype"));

colqty.setCellValueFactory(new PropertyValueFactory<>("qty"));

colamt.setCellValueFactory(new PropertyValueFactory<>("prdamt"));

coltotal.setCellValueFactory(new PropertyValueFactory<>("prdcost"));

coldate.setCellValueFactory(new PropertyValueFactory<>("prddate"));

tablestock.setItems(stockobj.showTableRecords());

}

@FXML

private void insertstock(ActionEvent event) {

UserBeans e = new UserBeans();

e.setPrdname(productname.getText());

e.setPrdtype(producttype.getValue().toString());

e.setPrdamt(Double.parseDouble(amount.getText()));

e.setQty(Integer.parseInt(quantity.getText()));

e.setPrdcost(Double.parseDouble(totalcost.getText()));

e.setPrddate(date.getText());

boolean flag = stockobj.insertstock(e);

if(!flag){

Alert alert = new Alert(AlertType.INFORMATION);

alert.setHeaderText("Stock Information Successfully Inserted!");

alert.show();

clearRecord();

tablestock.setItems(stockobj.showTableRecords());

}

else{

Alert alert = new Alert(AlertType.WARNING);

alert.setHeaderText("Somethig went wrong, try again!");

alert.show();

}

}

private void clearRecord(){

productname.setText("");

producttype.setValue("");

amount.setText("");

quantity.setText("");

totalcost.setText("");

date.setText("");

// txtStatus.setText("");

}

private void update(ActionEvent event) {

UserBeans e = new UserBeans();

// e.setPrdname(productname.getText());

e.setPrdtype(producttype.getValue().toString());

e.setQty(Integer.parseInt(quantity.getText()));

e.setPrdamt(Double.parseDouble(amount.getText()));

e.setPrdcost(Double.parseDouble(totalcost.getText()));

e.setPrddate(date.getText());

boolean flag=stockobj.updateStock(e);

if(flag){

Alert alert=new Alert(Alert.AlertType.WARNING);

alert.setHeaderText("Stock Information not updated");

alert.show();

}

else{

Alert alert=new Alert(Alert.AlertType.INFORMATION);

alert.setHeaderText("Stock Information successfully updated!");

alert.show();

tablestock.setItems(stockobj.showTableRecords());

}

}

private void setdata(MouseEvent event) {

try{

String name = tablestock.getSelectionModel().getSelectedItem().getIname();

printItem(stockobj.searchStock(name));

}

catch(NullPointerException ex){

Alert alert = new Alert(AlertType.WARNING);

alert.setHeaderText("Empty Record Selected!");

alert.show();

}

}

@FXML

private void tableaction(MouseEvent event) {

try{

String name = tablestock.getSelectionModel().getSelectedItem().getPrdname();

printItem(stockobj.searchStock(name));

}

catch(NullPointerException ex){

Alert alert = new Alert(AlertType.WARNING);

alert.setHeaderText("Empty Record Selected!");

alert.show();

}

}

public void printItem(UserBeans e){

if(e!=null){

productname.setText(e.getPrdname()+"");

producttype.setValue(e.getPrdtype());

amount.setText(e.getPrdamt()+"");

quantity.setText(e.getQty()+"");

totalcost.setText(e.getPrdcost()+"");

date.setText(e.getPrddate());

}

}

STAFF CONTROLLER

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package loginpage;

import beans.UserBeans;

import java.net.URL;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.Initializable;

import javafx.scene.control.Alert;

import javafx.scene.control.Alert.AlertType;

import javafx.scene.control.Button;

import javafx.scene.control.ComboBox;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.TextField;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.input.MouseEvent;

import javafx.scene.layout.BorderPane;

import model.Staff;

/\*\*

\* FXML Controller class

\*

\* @author DELL

\*/

public class StaffController implements Initializable {

@FXML

private BorderPane borderstaff;

@FXML

private Button btninsert;

@FXML

private TextField empname;

@FXML

private TextField empid;

@FXML

private ComboBox desigcb;

@FXML

private TextField salary;

@FXML

private TextField age;

@FXML

private TextField email;

@FXML

private TextField mobile;

@FXML

private TextField address;

private Staff staffobj;

@FXML

private TableColumn<UserBeans, String> colempname;

@FXML

private TableColumn<UserBeans, String> colage;

@FXML

private TableView<UserBeans> tableemp;

@FXML

private TableColumn<UserBeans, String> colempid;

@FXML

private TableColumn<UserBeans, String> colmob;

@FXML

private TableColumn<UserBeans, String> coldes;

@FXML

private TableColumn<UserBeans, String> colsal;

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

staffobj=new Staff();

desigcb.getItems().addAll("Manager","Worker","Volunteer","Cashier");

colempid.setCellValueFactory(new PropertyValueFactory<>("eid"));

colempname.setCellValueFactory(new PropertyValueFactory<>("ename"));

colage.setCellValueFactory(new PropertyValueFactory<>("eage"));

colsal.setCellValueFactory(new PropertyValueFactory<>("esal"));

colmob.setCellValueFactory(new PropertyValueFactory<>("emob"));

coldes.setCellValueFactory(new PropertyValueFactory<>("edesig"));

tableemp.setItems(staffobj.showTableRecords());

}

private void setTableData(MouseEvent event) {

try{

String name = tableemp.getSelectionModel().getSelectedItem().getEname();

//printEmployeeInfo(empModel.searchEmployee(id));

}

catch(NullPointerException ex){

Alert alert = new Alert(AlertType.WARNING);

alert.setHeaderText("Empty Record Selected!");

alert.show();

}

}

@FXML

private void insertaction(ActionEvent event) {

UserBeans e = new UserBeans();

e.setEname(empname.getText());

e.setEid(Integer.parseInt(empid.getText()));

e.setEdesig(desigcb.getValue().toString());

e.setEmob(Integer.parseInt(mobile.getText()));

e.setEemail(email.getText());

e.setEage(Integer.parseInt(age.getText()));

e.setEadd(address.getText());

e.setEsal(Integer.parseInt(salary.getText()));

boolean flag = staffobj.insertaction(e);

if(!flag){

Alert alert = new Alert(AlertType.INFORMATION);

alert.setHeaderText("Employee Information Successfully Inserted!");

alert.show();

clearRecord();

tableemp.setItems(staffobj.showTableRecords());

}

else{

Alert alert = new Alert(AlertType.WARNING);

alert.setHeaderText("Somethig went wrong, try again!");

alert.show();

}

}

private void clearRecord(){

empid.setText("");

empname.setText("");

desigcb.setValue("");

address.setText("");

mobile.setText("");

email.setText("");

salary.setText("");

age.setText("");

// txtStatus.setText("");

}

@FXML

private void updateStaff(ActionEvent event) {

UserBeans e=new UserBeans();

// e.setEname(empname.getText());

e.setEid(Integer.parseInt(empid.getText()));

e.setEdesig(desigcb.getValue().toString());

e.setEmob(Integer.parseInt(mobile.getText()));

e.setEemail(email.getText());

e.setEage(Integer.parseInt(age.getText()));

e.setEadd(address.getText());

e.setEsal(Integer.parseInt(salary.getText()));

boolean flag=staffobj.updateStaffInfo(e);

if(flag){

Alert alert=new Alert(Alert.AlertType.WARNING);

alert.setHeaderText("Employee Information not updated");

alert.show();

}

else{

Alert alert=new Alert(Alert.AlertType.INFORMATION);

alert.setHeaderText("Employee Information successfully updated!");

alert.show();

tableemp.setItems(staffobj.showTableRecords());

}

}

@FXML

private void delete(ActionEvent event) {

String name = empname.getText();

boolean flag = staffobj.deleteItem(name);

if(flag){

Alert alert=new Alert(Alert.AlertType.INFORMATION);

alert.setHeaderText("Item Information successfully Deleted!");

alert.show();

tableemp.setItems(staffobj.showTableRecords());

clearRecord();

}

else{

Alert alert=new Alert(Alert.AlertType.WARNING);

alert.setHeaderText("Something went wrong try again!");

alert.show();

}

}

private void setdata(MouseEvent event) {

try{

String name = tableemp.getSelectionModel().getSelectedItem().getEname();

printItem(staffobj.searchStaff(name));

}

catch(NullPointerException ex){

Alert alert = new Alert(AlertType.WARNING);

alert.setHeaderText("Empty Record Selected!");

alert.show();

}

}

public void printItem(UserBeans e){

if(e!=null){

empid.setText(e.getEid()+"");

empname.setText(e.getEname());

desigcb.setValue(e.getEdesig()+"");

mobile.setText(e.getEmob()+"");

email.setText(e.getEemail());

age.setText(e.getEage()+"");

address.setText(e.getEadd());

salary.setText(e.getEsal()+"");

}

}

@FXML

private void tableaction(MouseEvent event) {

try{

String name = tableemp.getSelectionModel().getSelectedItem().getEname();

printItem(staffobj.searchStaff(name));

}

catch(NullPointerException ex){

Alert alert = new Alert(AlertType.WARNING);

alert.setHeaderText("Empty Record Selected!");

alert.show();

}

}

SALES CONTROLLER

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package loginpage;

import beans.UserBeans;

import java.net.URL;

import java.util.ResourceBundle;

import javafx.fxml.FXML;

import javafx.fxml.Initializable;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.cell.PropertyValueFactory;

import model.Sales;

/\*\*

\* FXML Controller class

\*

\* @author DELL

\*/

public class SalesController implements Initializable {

@FXML

private TableView<UserBeans> tablesales;

@FXML

private TableColumn<UserBeans, String> csbid;

@FXML

private TableColumn<UserBeans, String> csbdate;

@FXML

private TableColumn<UserBeans, String> csbtotal;

private Sales saleobj;

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

saleobj = new Sales();

csbtotal.setCellValueFactory(new PropertyValueFactory<>("btotal"));

csbid.setCellValueFactory(new PropertyValueFactory<>("bid"));

csbdate.setCellValueFactory(new PropertyValueFactory<>("bdate"));

tablesales.setItems(saleobj.showTableRecords());

// TODO

}

}

LOGINPAGE CONTROLLER

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package loginpage;

import beans.UserBeans;

import java.io.IOException;

import java.net.URL;

import java.util.ResourceBundle;

import java.util.logging.Level;

import java.util.logging.Logger;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Parent;

import javafx.scene.control.Alert;

import javafx.scene.control.Alert.AlertType;

import javafx.scene.control.Button;

import javafx.scene.control.TextField;

import javafx.scene.image.ImageView;

import javafx.scene.input.MouseEvent;

import javafx.scene.layout.AnchorPane;

import javafx.scene.layout.BorderPane;

import javafx.scene.layout.VBox;

import javafx.stage.Stage;

import model.User;

/\*\*

\* FXML Controller class

\*

\* @author DELL

\*/

public class LoginPageController implements Initializable {

@FXML

private BorderPane borderpane;

@FXML

private TextField txtUser;

@FXML

private TextField txtpass;

@FXML

private Button btnLogin;

private User userModel;

@FXML

private VBox sideMenu;

@FXML

private ImageView cimage;

@Override

public void initialize(URL url, ResourceBundle rb) {

userModel=new User();

borderpane.setLeft(null);

}

private void loadUI(String ui){

Parent root = null;

try {

root = FXMLLoader.load(getClass().getResource(ui+".fxml"));

} catch (IOException ex) {

Logger.getLogger(LoginPageController.class.getName()).log(Level.SEVERE, null, ex);

}

borderpane.setCenter(root);

// cimage.setCenter(root);

// -fx-background-image: url("../images/login1.jpg");

//-fx-background-size: cover;

}

@FXML

private void billaction(MouseEvent event) {

loadUI("Bill");

}

@FXML

private void itemaction(MouseEvent event) {

loadUI("Item");

}

@FXML

private void stockaction(MouseEvent event) {

loadUI("Stock");

}

@FXML

private void salesaction(MouseEvent event) {

loadUI("Sales");

}

@FXML

private void staffaction(MouseEvent event) {

loadUI("Staff");

}

@FXML

private void setting(MouseEvent event) {

loadUI("SettingK");

}

@FXML

private void exitaction(MouseEvent event) {

System.exit(0);

}

@FXML

private void loginUser(ActionEvent event) {

String username=txtUser.getText();

String password=txtpass.getText();

UserBeans ub=new UserBeans();

ub.setName(username);

ub.setPassword(password);

boolean flag=userModel.userLogin(ub);

if(flag){

Alert alert=new Alert(AlertType.INFORMATION);

borderpane.setLeft(sideMenu);

loadUI("home");

}

else{

Alert alert=new Alert(AlertType.WARNING);

alert.setHeaderText("Login Failed!");

alert.show();

System.out.println(flag);

}

}

}

MAINPAGE CONTROLLER

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package loginpage;

import java.io.IOException;

import java.net.URL;

import java.util.ResourceBundle;

import java.util.logging.Level;

import java.util.logging.Logger;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Parent;

import javafx.scene.control.Button;

import javafx.scene.control.ComboBox;

import javafx.scene.control.DatePicker;

import javafx.scene.control.RadioButton;

import javafx.scene.control.Tab;

import javafx.scene.control.TextField;

import javafx.scene.input.MouseEvent;

import javafx.scene.layout.BorderPane;

import javafx.stage.Stage;

/\*\*

\* FXML Controller class

\*

\* @author DELL

\*/

public class MainPageController implements Initializable {

@FXML

private BorderPane borderpane;

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

}

private void close(MouseEvent event) {

Stage stage = (Stage) borderpane.getScene().getWindow();

stage.close();

}

private void loadUI(String ui){

Parent root = null;

try {

root = FXMLLoader.load(getClass().getResource(ui+".fxml"));

} catch (IOException ex) {

Logger.getLogger(MainPageController.class.getName()).log(Level.SEVERE, null, ex);

}

borderpane.setCenter(root);

}

private void okaction(MouseEvent event) {

loadUI("Bill");

}

}

SETTING CONTROLLER

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package loginpage;

import beans.UserBeans;

import java.net.URL;

import java.time.LocalDate;

import java.time.format.DateTimeFormatter;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.Initializable;

import javafx.scene.control.Alert;

import javafx.scene.control.Alert.AlertType;

import javafx.scene.control.Button;

import javafx.scene.control.ComboBox;

import javafx.scene.control.DatePicker;

import javafx.scene.control.PasswordField;

import javafx.scene.control.RadioButton;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.TextField;

import javafx.scene.control.ToggleGroup;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.input.MouseEvent;

import javafx.scene.layout.BorderPane;

import javafx.scene.layout.HBox;

import javafx.scene.layout.VBox;

import static jdk.nashorn.internal.objects.Global.getDate;

import model.User;

/\*\*

\* FXML Controller class

\*

\* @author DELL

\*/

public class SettingKController implements Initializable {

@FXML

private TextField txtId;

@FXML

private TextField txtName;

@FXML

private RadioButton rbMale;

@FXML

private RadioButton rbFemale;

@FXML

private DatePicker dpDob;

@FXML

private TextField txtMobile;

@FXML

private TextField txtEmail;

@FXML

private TextField txtAddress;

@FXML

private TextField txtPassword;

@FXML

private Button btnUpdateUser;

@FXML

private Button btnUserLogout;

@FXML

private BorderPane adminPane;

@FXML

private VBox adminLoginPane;

@FXML

private TextField atxtAdminName;

@FXML

private PasswordField adminPassword;

@FXML

private Button btnAdminLogin;

@FXML

private HBox adminContentPane;

@FXML

private Button btnNewUser;

@FXML

private Button btnDelete;

@FXML

private Button btnUpdate;

@FXML

private Button btnPrintUser;

@FXML

private Button btnResetUser;

@FXML

private Button btnAdminLogout;

@FXML

private TextField atxtId;

@FXML

private TextField atxtName;

@FXML

private RadioButton arbMale;

@FXML

private RadioButton arbFemale;

private ToggleGroup atgGender;

private ToggleGroup tgGender;

@FXML

private DatePicker adpDob;

@FXML

private ComboBox acbDesignation;

@FXML

private TextField atxtMobile;

@FXML

private TextField atxtEmail;

@FXML

private TextField atxtAddress;

@FXML

private ComboBox acbRole;

private ComboBox acbActive;

@FXML

private TextField atxtPassword;

@FXML

private TableView<UserBeans> tableUser;

@FXML

private TableColumn<UserBeans, String> colId;

@FXML

private TableColumn<UserBeans, String> colName;

@FXML

private TableColumn<UserBeans, String> colGender;

@FXML

private TableColumn<UserBeans, String> colDesignation;

@FXML

private TableColumn<UserBeans, String> colMobile;

@FXML

private TableColumn<UserBeans, String> colAddress;

private User userModel;

@FXML

private ComboBox cbDesignation;

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

userModel=new User();

adminPane.setCenter(null);

ToggleGroup atgGender = new ToggleGroup();

arbMale.setToggleGroup(atgGender);

arbFemale.setToggleGroup(atgGender);

ToggleGroup tgGender = new ToggleGroup();

rbMale.setToggleGroup(tgGender);

rbFemale.setToggleGroup(tgGender);

//setUserData(userModel.searchUserInfo(userModel.getLoginUser().getId()));

acbDesignation.getItems().addAll("Chef","Worker","Helper","Cashier");

acbRole.getItems().addAll("Admin","User");

//acbActive.getItems().addAll("1","0");

colId.setCellValueFactory(new PropertyValueFactory<>("userId"));

colName.setCellValueFactory(new PropertyValueFactory<>("name"));

colGender.setCellValueFactory(new PropertyValueFactory<>("gender"));

colDesignation.setCellValueFactory(new PropertyValueFactory<>("qualification"));

colMobile.setCellValueFactory(new PropertyValueFactory<>("mobile"));

colAddress.setCellValueFactory(new PropertyValueFactory<>("address"));

tableUser.setItems(userModel.addAllUser());

userModel.addAllUser();

}

@FXML

private void updateUser(ActionEvent event) {

boolean flag=userModel.updateUser(getUserInfo());

if(flag){

Alert alert = new Alert(AlertType.WARNING);

alert.setHeaderText("Something went wrong, Please try again!");

alert.show();

}

else{

Alert alert = new Alert(AlertType.INFORMATION);

alert.setHeaderText("User Successfully updated!");

alert.show();

}

}

@FXML

private void userLogout(ActionEvent event) {

}

@FXML

private void adminLogin(ActionEvent event) {

UserBeans u = new UserBeans();

u.setName(atxtAdminName.getText());

u.setPassword(adminPassword.getText());

boolean flag = userModel.adminLogin(event);

if(flag){

adminPane.setLeft(null);

adminPane.setCenter(adminContentPane);

}

else{

Alert alert = new Alert(AlertType.WARNING);

alert.setHeaderText("Wrong admin name and password, Please try again!");

alert.show();

System.out.print("vidya");

}

}

@FXML

private void adminLogout(ActionEvent event) {

adminPane.setLeft(adminLoginPane);

adminPane.setCenter(null);

atxtAdminName.setText(null);

adminPassword.setText(null);

atxtAdminName.requestFocus();

}

public void setUserInfo(UserBeans u){

if(u!=null){

atxtId.setText(u.getUserId()+"");

atxtName.setText(u.getName());

atxtMobile.setText(u.getMobile());

atxtEmail.setText(u.getEmail());

atxtAddress.setText(u.getAddress());

atxtPassword.setText(u.getPassword());

cbDesignation.setValue(u.getDesig());

acbRole.setValue(u.getRole());

// acbActive.setValue(u.getActive());

String gender = u.getGender();

if(gender!=null){

if(gender.equalsIgnoreCase("Male")){

arbMale.setSelected(true);

}

else{

arbFemale.setSelected(true);

}

}

else{

arbMale.setSelected(false);

arbFemale.setSelected(false);

}

System.out.println(u.getDob());

if(u.getDob().equals("null")==false)

adpDob.setValue(LocalDate.parse(u.getDob()));

}

}

public void setUserData(UserBeans u){

if(u!=null){

txtId.setText(u.getUserId()+"");

txtName.setText(u.getName());

txtMobile.setText(u.getMobile());

txtEmail.setText(u.getEmail());

txtAddress.setText(u.getAddress());

txtPassword.setText(u.getPassword());

acbDesignation.setValue(u.getDesig());

String gender = u.getGender();

if(gender!=null){

if(gender.equalsIgnoreCase("Male")){

rbMale.setSelected(true);

}

else{

rbFemale.setSelected(true);

}

}

else{

rbMale.setSelected(false);

rbFemale.setSelected(false);

}

if(u.getDob().equals("null")==false)

dpDob.setValue(LocalDate.parse(u.getDob()));

}

}

public UserBeans getUserInfo(){

UserBeans u = new UserBeans();

u.setName(atxtName.getText());

u.setMobile(atxtMobile.getText());

u.setEmail(atxtEmail.getText());

u.setAddress(atxtAddress.getText());

u.setPassword(atxtPassword.getText());

u.setDesig(acbDesignation.getValue()+"");

u.setRole(acbRole.getValue()+"");

// u.setActive(Integer.parseInt(acbActive.getValue()+""));

u.setDob((String) getDate(adpDob.getValue()));

RadioButton rbGender=(RadioButton)atgGender.getSelectedToggle();

if(rbGender!=null){

String gender = rbGender.getText();

u.setGender(gender);

}

return u;

}

public UserBeans getUserData(){

UserBeans u = new UserBeans();

u.setUserId(Long.parseLong(txtId.getText()));

u.setName(txtName.getText());

u.setMobile(txtMobile.getText());

u.setEmail(txtEmail.getText());

u.setAddress(txtAddress.getText());

u.setPassword(txtPassword.getText());

u.setDesig(cbDesignation.getValue()+"");

u.setDob((String) getDate(dpDob.getValue()));

RadioButton rbGender=(RadioButton)tgGender.getSelectedToggle();

if(rbGender!=null){

String gender = rbGender.getText();

u.setGender(gender);

}

return u;

}

@FXML

private void showUserInfo(MouseEvent event) {

try{

long id = tableUser.getSelectionModel().getSelectedItem().getUserId();

setUserInfo(userModel.searchUserInfo(id));

}

catch(NullPointerException ex){

Alert alert = new Alert(AlertType.WARNING);

alert.setHeaderText("Empty Record Selected!");

alert.show();

}

}

public String getDate(LocalDate date){

DateTimeFormatter formatter = DateTimeFormatter.ofPattern("dd-MMM-yyyy");

String dt = date.format(formatter);

return dt;

}

@FXML

private void createNewUser(ActionEvent event) {

boolean flag = userModel.createNewUser(getUserInfo());

if(flag){

Alert alert = new Alert(AlertType.WARNING);

alert.setHeaderText("Something went wrong, Please try again!");

alert.show();

}

else{

Alert alert = new Alert(AlertType.INFORMATION);

alert.setHeaderText("User Successfully created!");

alert.show();

}

}

@FXML

private void deleteAdmin(ActionEvent event) {

String id=atxtId.getText();

boolean flag = userModel.deleteAdminUser(id);

if(flag){

Alert alert=new Alert(Alert.AlertType.INFORMATION);

alert.setHeaderText("User Information successfully Deleted!");

alert.show();

}

else{

Alert alert=new Alert(Alert.AlertType.WARNING);

alert.setHeaderText("Something went wrong try again!");

alert.show();

}

}

@FXML

private void updateAdmin(ActionEvent event) {

boolean flag=userModel.updateAdminUser(getUserInfo());

if(flag){

Alert alert = new Alert(AlertType.WARNING);

alert.setHeaderText("Something went wrong, Please try again!");

alert.show();

}

else{

Alert alert = new Alert(AlertType.INFORMATION);

alert.setHeaderText("User Successfully updated!");

alert.show();

}

}

@FXML

private void printUser(ActionEvent event) {

UserBeans u = new UserBeans();

u.setUserId(Long.parseLong(atxtId.getText()));

u.setName(atxtName.getText());

RadioButton rbGender = (RadioButton)atgGender.getSelectedToggle();

u.setGender(rbGender.getText());

u.setDob(getDate(adpDob.getValue()));

u.setMobile(atxtMobile.getText());

u.setEmail(atxtEmail.getText());

u.setAddress(atxtAddress.getText());

u.setPassword(atxtPassword.getText());

u.setDesig(acbDesignation.getValue()+"");

u.setRole(acbRole.getValue()+"");

// u.setActive(Integer.parseInt(acbActive.getValue()+""));

userModel.printRecord(u);

}

@FXML

private void resetUser(ActionEvent event) {

atxtId.setText("");

atxtName.setText("");

arbMale.setSelected(false);

arbFemale.setSelected(false);

adpDob.setValue(null);

atxtMobile.setText("");

atxtEmail.setText("");

atxtAddress.setText("");

atxtPassword.setText("");

acbDesignation.setValue("");

acbRole.setValue("");

}

}

BEANS

Login Beans

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package beans;

/\*\*

\*

\* @author DELL

\*/

public class Loginbeans {

private long id;

private String name;

public long getId() {

return id;

}

public void setId(long id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

}

User Beans

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package beans;

/\*\*

\*

\* @author DELL

\*/

public class UserBeans {

private long userId;

private String name,gender,dob,desig,mobile,email,address,role,password;

private String ename,edesig,eemail,eadd;

private int eid,esal,emob,eage,active;

private String prdname,prdtype,prddate;

private String iname,itype,idesc;

private double iprice;

private int bid,billqty;

private double itemtotal,btotal,bprice;

private String biname,bdate;

public int getBid() {

return bid;

}

public void setBid(int bid) {

this.bid = bid;

}

public int getBillqty() {

return billqty;

}

public void setBillqty(int billqty) {

this.billqty = billqty;

}

public double getItemtotal() {

return itemtotal;

}

public void setItemtotal(double itemtotal) {

this.itemtotal = itemtotal;

}

public double getBtotal() {

return btotal;

}

public void setBtotal(double btotal) {

this.btotal = btotal;

}

public double getBprice() {

return bprice;

}

public void setBprice(double bprice) {

this.bprice = bprice;

}

public String getBiname() {

return biname;

}

public void setBiname(String biname) {

this.biname = biname;

}

public String getBdate() {

return bdate;

}

public void setBdate(String bdate) {

this.bdate = bdate;

}

public String getIname() {

return iname;

}

public void setIname(String iname) {

this.iname = iname;

}

public String getItype() {

return itype;

}

public void setItype(String itype) {

this.itype = itype;

}

public String getIdesc() {

return idesc;

}

public void setIdesc(String idesc) {

this.idesc = idesc;

}

public double getIprice() {

return iprice;

}

public void setIprice(double iprice) {

this.iprice = iprice;

}

public String getPrdname() {

return prdname;

}

public void setPrdname(String prdname) {

this.prdname = prdname;

}

public String getPrdtype() {

return prdtype;

}

public void setPrdtype(String prdtype) {

this.prdtype = prdtype;

}

public String getPrddate() {

return prddate;

}

public void setPrddate(String prddate) {

this.prddate = prddate;

}

public int getQty() {

return qty;

}

public void setQty(int qty) {

this.qty = qty;

}

public double getPrdamt() {

return prdamt;

}

public void setPrdamt(double prdamt) {

this.prdamt = prdamt;

}

public double getPrdcost() {

return prdcost;

}

public void setPrdcost(double prdcost) {

//prdcost=prdamt\*qty;

this.prdcost = prdcost;

}

private int qty;

private double prdamt,prdcost;

public String getEname() {

return ename;

}

public void setEname(String ename) {

this.ename = ename;

}

public String getEdesig() {

return edesig;

}

public void setEdesig(String edesig) {

this.edesig = edesig;

}

public String getEemail() {

return eemail;

}

public void setEemail(String eemail) {

this.eemail = eemail;

}

public String getEadd() {

return eadd;

}

public void setEadd(String eadd) {

this.eadd = eadd;

}

public int getEid() {

return eid;

}

public void setEid(int eid) {

this.eid = eid;

}

public int getEsal() {

return esal;

}

public void setEsal(int esal) {

this.esal = esal;

}

public int getEmob() {

return emob;

}

public void setEmob(int emob) {

this.emob = emob;

}

public int getEage() {

return eage;

}

public void setEage(int eage) {

this.eage = eage;

}

public UserBeans(){}

public UserBeans(long userId,int active,String name,String gender, String dob, String desig, String mobile, String email, String address, String role, String password) {

this.userId = userId;

this.name = name;

this.gender = gender;

this.dob = dob;

this.desig = desig;

this.mobile = mobile;

this.email = email;

this.address = address;

this.role = role;

this.password = password;

// this.active = active;

}

public long getUserId() {

return userId;

}

public void setUserId(long userId) {

this.userId = userId;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getGender() {

return gender;

}

public void setGender(String gender) {

this.gender = gender;

}

public String getDob() {

return dob;

}

public void setDob(String dob) {

this.dob = dob;

}

public String getDesig() {

return desig;

}

public void setDesig(String desig) {

this.desig = desig;

}

public String getMobile() {

return mobile;

}

public void setMobile(String mobile) {

this.mobile = mobile;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getAddress() {

return address;

}

public void setAddress(String address) {

this.address = address;

}

public String getRole() {

return role;

}

public void setRole(String role) {

this.role = role;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

}

MODALS

User

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package model;

import beans.Loginbeans;

import beans.UserBeans;

import java.io.File;

import static java.lang.System.exit;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.event.ActionEvent;

import loginpage.LoginPage.LoginBeans;

import net.sf.jasperreports.engine.JasperCompileManager;

import net.sf.jasperreports.engine.JasperFillManager;

import net.sf.jasperreports.engine.JasperPrint;

import net.sf.jasperreports.engine.JasperReport;

import net.sf.jasperreports.engine.design.JasperDesign;

import net.sf.jasperreports.engine.xml.JRXmlLoader;

import net.sf.jasperreports.view.JasperViewer;

/\*\*

\*

\* @author DELL

\*/

public class User {

private Connection conn;

private Statement stmt;

private ResultSet rs;

private DbConnection db;

public User(){

db=new DbConnection();

conn=db.getConnection();

stmt=db.getStatement();

}

public boolean userLogin(UserBeans ub){

boolean flag=false;

try{

String sql="select \* from temp\_users where name='"+ub.getName()+"' and password='"+ub.getPassword()+"'";

rs=stmt.executeQuery(sql);

if(rs.next()){

flag=true;

}

}catch(SQLException e){

db.setWarningMessage("UserModel.userLogin",e.toString());

}

return flag;

}

public boolean adminlogin(UserBeans ub){

boolean flag=false;

try{

String sql="select \* from users where name='"+ub.getName()+"' and password='"+ub.getPassword()+"' and role='Admin'";

rs=stmt.executeQuery(sql);

if(rs.next()){

flag=true;

}

}catch(SQLException e){

db.setWarningMessage("UserModel.adminLogin",e.toString());

}

return (flag);

}

public Loginbeans getLoginUser(){

Loginbeans l = null;

try{

String sql = "select \* from temp\_users";

rs = stmt.executeQuery(sql);

if(rs.next()){

l = new Loginbeans();

l.setId(rs.getLong("id"));

l.setName(rs.getString("name"));

}

}catch(SQLException e){

db.setWarningMessage("UserModel.getLoginUser",e.toString());

}

return l;

}

public ObservableList<UserBeans>addAllUser(){

ObservableList<UserBeans> list=null;

try{

list=FXCollections.observableArrayList();

String sql="select \* from users";

rs=stmt.executeQuery(sql);

while(rs.next()){

list.add(getUserInfo(rs));

}

refreshDatabase();

}

catch(SQLException e){

db.setWarningMessage("UserModel.addAllUser",e.toString());

}

return list;

}

public UserBeans getUserInfo(ResultSet rs){

UserBeans u=null;

if(rs!=null){

try{

u=new UserBeans();

u.setUserId(rs.getInt("user\_id"));

u.setName(rs.getString("name"));

u.setGender(rs.getString("gender"));

u.setDob(rs.getDate("dob")+"");

u.setDesig(rs.getString("desig"));

u.setMobile(rs.getString("mobile"));

u.setAddress(rs.getString("address"));

u.setEmail(rs.getString("email"));

u.setRole(rs.getString("role"));

u.setPassword(rs.getString("password"));

}

catch(SQLException e){

db.setWarningMessage("UserModel.getUserInfo",e.toString());

}

}

return u;

}

public boolean createNewUser(UserBeans u){

boolean flag = true;

try{

String sql = "insert into temp\_users values(?,?,?,?,?,?,?,?,?)";

PreparedStatement p = conn.prepareCall(sql);

p.setString(1, u.getName());

p.setString(2, u.getGender());

p.setString(3, u.getDob());

p.setString(4, u.getDesig());

p.setString(5, u.getMobile());

p.setString(6, u.getEmail());

p.setString(7, u.getAddress());

p.setString(8, u.getRole());

p.setString(9, u.getPassword());

flag = p.execute();

}

catch(SQLException e){

db.setWarningMessage("UserModel.createNewUser",e.toString());

}

return flag;

}

/\* public Object getLoginUser() {

throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.

}

\*/

public boolean adminLogin(ActionEvent event)

{

boolean flg=true;

exit(0);

return flg;

}

public UserBeans searchUserInfo(long id) {

UserBeans b = null;

try{

String sql = "select \* from users where user\_id="+id;

rs = stmt.executeQuery(sql);

if(rs.next()){

b = getUserInfo(rs);

}

}

catch(SQLException e){

db.setWarningMessage("UserModel.searchUserInfo",e.toString());

}

return b;

//throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.

}

public boolean updateUser(UserBeans u) {

boolean flag=true;

try{

String sql="update users set name=?,gender=?,dob=?,desig=?,mobile=?,email=?,address=?,role=?,password=? where user\_id=?";

PreparedStatement p=conn.prepareStatement(sql);

p.setString(1,u.getName());

p.setString(2,u.getGender());

p.setString(3,u.getDob());

p.setString(4,u.getDesig());

p.setString(5,u.getMobile());

p.setString(6,u.getEmail());

p.setString(7,u.getAddress());

p.setString(8,u.getRole());

p.setString(9,u.getPassword());

p.setString(10,rs.getString("user\_id"));

flag=p.execute();

}

catch(SQLException ex){

db.setWarningMessage("UserModel.UpdateUser",ex.toString());

}

return flag;

// throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.

}

public void printRecord(UserBeans u) {

try{

/\* HashMap hp=new HashMap();

hp.put("id",u.getUserId());

hp.put("name",u.getName());

hp.put("gender",u.getGender());

hp.put("desig",u.getDesig());

hp.put("mobile",u.getMobile());

hp.put("role",u.getRole());

JasperDesign jasperDesign=JRXmlLoader.load(new File("").getAbsolutePath()+"/src/hrms/jasperReport/UserReport.jrxml");

JasperReport jasperReport=JasperCompileManager.compileReport(jasperDesign);

JasperPrint jasperPrint;

jasperPrint = JasperFillManager.fillReport(jasperReport,hp, conn);

JasperViewer jasperViewer = new JasperViewer(jasperPrint, false);

jasperViewer.setVisible(true);\*/

}

catch(Exception ex){

db.setWarningMessage("UserModel.printRecord",ex.toString());

}

//throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.

}

public boolean deleteAdminUser(String id) {

boolean flag = true;

try{

rs = stmt.executeQuery("delete from users where user\_id='"+id+"'");

rs.next();

}

catch(SQLException ex){

//db.setWarningMessage("LeaveModel.deleteLeaveInfo", ex.toString());

flag = false;

}

refreshDatabase();

return flag;

}

private void refreshDatabase(){

try{

rs = stmt.executeQuery("select \* from users");

}

catch(SQLException e){

db.setWarningMessage("UserModel.refreshDatabase",e.toString());

}

//throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.

}

public boolean updateAdminUser(UserBeans u) {

boolean flag=true;

try{

String sql="update users set name=?,gender=?,dob=?,qualification=?,mobile=?,email=?,address=?,role=?,active=?,password=? where user\_id=?";

PreparedStatement p=conn.prepareStatement(sql);

p.setString(1,u.getName());

p.setString(2,u.getGender());

p.setString(3,u.getDob());

p.setString(4,u.getDesig());

p.setString(5,u.getMobile());

p.setString(6,u.getEmail());

p.setString(7,u.getAddress());

p.setString(8,u.getRole());

p.setString(9,u.getPassword());

p.setString(10,rs.getString("user\_id"));

flag=p.execute();

}

catch(SQLException ex){

db.setWarningMessage("UserModel.UpdateUser",ex.toString());

}

return flag;

}

}

Bill

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package model;

import beans.UserBeans;

import java.io.File;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.logging.Level;

import java.util.logging.Logger;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import net.sf.jasperreports.engine.JasperCompileManager;

import net.sf.jasperreports.engine.JasperFillManager;

import net.sf.jasperreports.engine.JasperPrint;

import net.sf.jasperreports.engine.JasperReport;

import net.sf.jasperreports.engine.design.JRDesignQuery;

import net.sf.jasperreports.engine.design.JasperDesign;

import net.sf.jasperreports.engine.xml.JRXmlLoader;

import net.sf.jasperreports.view.JasperViewer;

/\*\*

\*

\* @author DELL

\*/

public class Bill {

private DbConnection db;

private Connection conn;

private Statement stmt;

private ResultSet rs;

private int total;

public Bill(){

try{

db = new DbConnection();

conn = db.getConnection();

stmt = db.getStatement();

rs = stmt.executeQuery("Select \* from bill\_item");

if(rs.next()){

//total = rs.getInt(1);

}

} catch (SQLException e) {

Logger.getLogger(Bill.class.getName()).log(Level.SEVERE, null, e);

db.setWarningMessage("Bill", e.toString());

}

}

public boolean insertitem(UserBeans e) {

boolean flag = true;

try

{

String sql = "insert into bill\_item values(?,?,?,?)";

PreparedStatement p = conn.prepareStatement(sql);

p.setString(1, e.getBiname());

p.setInt(2, e.getBillqty());

p.setDouble(3, e.getBprice());

p.setDouble(4, e.getItemtotal());

flag = p.execute();

System.out.println("flag = "+flag);

}catch(SQLException ex){

db.setWarningMessage("Bill.insertaction", ex.toString());

}

return flag;

}

public UserBeans setBill(ResultSet rs){

UserBeans e = null;

if(rs!=null){

try

{

e = new UserBeans();

e.setBiname(rs.getString("bi\_name"));

e.setBillqty(rs.getInt("bqty"));

e.setBprice(rs.getDouble("bitem\_name"));

e.setItemtotal(rs.getDouble("btotal\_price"));

}

catch(SQLException ex){

db.setWarningMessage("Bill\_item.setBill", ex.toString());

}

}

return e;

}

public ObservableList<UserBeans> showTableRecords() {

ObservableList list = null;

try{

list = FXCollections.observableArrayList();

rs = stmt.executeQuery("select \* from bill\_item");

while(rs.next()){

list.add(setBill(rs));

}

// updateRecord();

}

catch(SQLException ex){

db.setWarningMessage("Bill\_item Model.showTableRecords", ex.toString());

}

return list;

}

public UserBeans searchItem(String Iname){

UserBeans e=null;

try{

String sql="select \* from bill\_item where bi\_name='"+Iname+"'";

rs=stmt.executeQuery(sql);

if(rs.next()){

e=setBill(rs);

}

else{

db.setWarningMessage(" Item Info does not Exist!","");

}

}

catch(SQLException ex){

db.setWarningMessage("ItemModel.searchLeaveId",ex.toString());

}

return e;

}

public boolean billrun(UserBeans e) {

try {

rs = stmt.executeQuery("Select \* from bill\_gen");

} catch (SQLException ex) {

Logger.getLogger(Bill.class.getName()).log(Level.SEVERE, null, ex);

}

boolean flag = true;

try

{

String sql = "insert into bill\_gen values(?,?,?)";

PreparedStatement p = conn.prepareStatement(sql);

p.setInt(1, e.getBid());

p.setString(2, e.getBdate());

p.setDouble(3, e.getBtotal());

flag = p.execute();

System.out.println("flag = "+flag);

}catch(SQLException ex){

db.setWarningMessage("Bill.billrun", ex.toString());

}

return flag;

}

public UserBeans setBill1(ResultSet rs){

UserBeans e = null;

if(rs!=null){

try

{

e = new UserBeans();

e.setBid(rs.getInt("b\_id"));

e.setBdate(rs.getString("bill\_date"));

e.setBtotal(rs.getDouble("bill\_total"));

}

catch(SQLException ex){

db.setWarningMessage("Bill\_gen.setBill1", ex.toString());

}

}

return e;

}

public ObservableList<UserBeans> showTableRecords1()

{

ObservableList list = null;

try{

list = FXCollections.observableArrayList();

rs = stmt.executeQuery("select \* from bill\_gen");

while(rs.next()){

list.add(setBill1(rs));

}

// updateRecord();

}

catch(SQLException ex){

db.setWarningMessage("Bill\_gen Model.showTableRecord1", ex.toString());

}

return list;

}

/\*public UserBeans searchItem1(Integer name) {

throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.

}

\*/

public void printRecord(UserBeans e){

try{

java.util.HashMap hp=new java.util.HashMap();

hp.put("bid",e.getBid());

hp.put("bdate",e.getBdate());

hp.put("btotal",e.getBtotal());

JasperDesign jasperDesign=JRXmlLoader.load(new File("").getAbsolutePath()+"src/LoginPage/report/itemreport.jrxml");

JasperReport jasperReport=JasperCompileManager.compileReport(jasperDesign);

JasperPrint jasperPrint=JasperFillManager.fillReport(jasperReport,hp, conn);

JasperViewer jasperViewer = new JasperViewer(jasperPrint, false);

jasperViewer.setVisible(true);

}

catch(Exception ex){

db.setWarningMessage("BillModel.printRecord",ex.toString());

}

}

public void printItemInfo(String sql){

try{

JasperDesign jasperDesign=JRXmlLoader.load(new File("").getAbsolutePath()+"/src/LoginPage/report/Sales.jrxml");

JRDesignQuery jrQuery=new JRDesignQuery();

jrQuery.setText(sql);

jasperDesign.setQuery(jrQuery);

JasperReport jasperReport=JasperCompileManager.compileReport(jasperDesign);

JasperPrint jasperPrint=JasperFillManager.fillReport(jasperReport,null, conn);

JasperViewer jasperViewer = new JasperViewer(jasperPrint, false);

jasperViewer.setVisible(true);

}

catch(Exception ex){

db.setWarningMessage("ItemModel.printEmpInfo",ex.toString());

}

}

}

Item

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package model;

import beans.UserBeans;

import java.io.File;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.logging.Level;

import java.util.logging.Logger;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.scene.control.Alert;

import java.util.HashMap;

import javafx.scene.control.Alert.AlertType;

import net.sf.jasperreports.engine.JasperCompileManager;

import net.sf.jasperreports.engine.JasperFillManager;

import net.sf.jasperreports.engine.JasperPrint;

import net.sf.jasperreports.engine.JasperReport;

import net.sf.jasperreports.engine.design.JRDesignQuery;

import net.sf.jasperreports.engine.design.JasperDesign;

import net.sf.jasperreports.engine.xml.JRXmlLoader;

import net.sf.jasperreports.view.JasperViewer;

/\*\*

\*

\* @author DELL

\*/

public class Item {

private DbConnection db;

private Connection conn;

private Statement stmt;

private ResultSet rs;

private int total, current;

public Item(){

try

{

db = new DbConnection();

conn = db.getConnection();

stmt = db.getStatement();

rs = stmt.executeQuery("select (\*) from item\_mstr");

}

catch (SQLException ex) {

//Logger.getLogger(Item.class.getName()).log(Level.SEVERE, null, ex);

}

}

public boolean insertitem(UserBeans e){

boolean flag = true;

try

{

String sql = "insert into item\_mstr values(?,?,?,?)";

PreparedStatement p = conn.prepareStatement(sql);

p.setString(1, e.getIname());

p.setDouble(2, e.getIprice());

p.setString(3, e.getIdesc());

p.setString(4, e.getItype());

flag = p.execute();

System.out.println("flag = "+flag);

}catch(SQLException ex){

db.setWarningMessage("Item.insertitem", ex.toString());

}

return flag;

}

public UserBeans setItem(ResultSet rs){

UserBeans e = null;

if(rs!=null){

try

{

e = new UserBeans();

e.setIname(rs.getString("item\_name"));

e.setIprice(rs.getDouble("item\_price"));

e.setIdesc(rs.getString("item\_descr"));

e.setItype(rs.getString("type"));

}

catch(SQLException ex){

db.setWarningMessage("Item.setItem", ex.toString());

}

}

return e;

}

private void refreshDatabase(){

try{

rs = stmt.executeQuery("select \* from item\_mstr");

}

catch(SQLException e){

System.out.println(e.toString());

Alert alert = new Alert(Alert.AlertType.WARNING);

alert.setHeaderText("Exception Raised in method Item.refreshDatabase");

alert.setContentText(e.toString());

alert.show();

}

}

public ObservableList<UserBeans> showTableRecords(){

ObservableList list = null;

try{

list = FXCollections.observableArrayList();

rs = stmt.executeQuery("select \* from item\_mstr");

while(rs.next()){

list.add(setItem(rs));

}

updateRecord();

}

catch(SQLException ex){

db.setWarningMessage("Itemmodel.showTableRecords", ex.toString());

}

return list;

}

public UserBeans updateRecord(){

UserBeans e = null;

try{

rs = stmt.executeQuery("select \* from item\_mstr");

rs.next();

e = setItem(rs);

}

catch(SQLException ex){

db.setWarningMessage("ItemModel.updateRecord", ex.toString());

}

return e;

}

public boolean updateItem(UserBeans e){

boolean flag = true;

try{

String sql = "update item\_mstr set item\_price=?,item\_descr=?,type=? where item\_name=?";

PreparedStatement p = conn.prepareStatement(sql);

p.setDouble(1,e.getIprice());

p.setString(2,e.getIdesc());

p.setString(3,e.getItype());

p.setString(4,rs.getString("item\_name"));

// ps.setString(4, e.getIname());

flag = p.execute();

}

catch(SQLException ex){

db.setWarningMessage("ItemModel.updateItem", ex.toString());

}

return flag;

}

public boolean deleteItem(String iname){

boolean flag = true;

try{

rs = stmt.executeQuery("delete from item\_mstr where item\_name='"+iname+"'");

rs.next();

}

catch(SQLException ex){

db.setWarningMessage("LeaveModel.deleteLeaveInfo", ex.toString());

flag = false;

}

return flag;

}

public UserBeans searchItem(String Iname){

UserBeans e=null;

try{

String sql="select \* from item\_mstr where item\_name='"+Iname+"'";

rs=stmt.executeQuery(sql);

if(rs.next()){

e=setItem(rs);

}

else{

db.setWarningMessage(" Item Info does not Exist!","");

}

}

catch(SQLException ex){

db.setWarningMessage("ItemModel.searchLeaveId",ex.toString());

}

return e;

}

}

Stock

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package model;

import beans.UserBeans;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.logging.Level;

import java.util.logging.Logger;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.scene.control.Alert;

/\*\*

\*

\* @author DELL

\*/

public class Stock {

private DbConnection db;

private Connection conn;

private Statement stmt;

private ResultSet rs;

private int total, current;

public Stock(){

try

{

db = new DbConnection();

conn = db.getConnection();

stmt = db.getStatement();

rs = stmt.executeQuery("select \* from stock\_mstr");

if(rs.next()){

total = rs.getInt(1);

}

}

catch (SQLException ex) {

Logger.getLogger(Stock.class.getName()).log(Level.SEVERE, null, ex);

db.setWarningMessage("StockModel", ex.toString());

}

}

public boolean insertstock(UserBeans e){

boolean flag = true;

try

{

String sql = "insert into stock\_mstr values(?,?,?,?,?,?)";

PreparedStatement p = conn.prepareStatement(sql);

p.setString(1, e.getPrdname());

p.setString(2, e.getPrdtype());

p.setInt(3, e.getQty());

p.setDouble(4, e.getPrdamt());

p.setDouble(5, e.getPrdcost());

p.setString(6, e.getPrddate());

flag = p.execute();

System.out.println("flag = "+flag);

}catch(SQLException ex){

db.setWarningMessage("Stock.insertstock", ex.toString());

}

return flag;

}

public UserBeans setStock(ResultSet rs){

UserBeans e = null;

if(rs!=null){

try

{

e = new UserBeans();

e.setPrdname(rs.getString("pr\_name"));

e.setPrdtype(rs.getString("pr\_type"));

e.setQty(rs.getInt("qty"));

e.setPrdamt(rs.getDouble("pr\_amt"));

e.setPrdcost(rs.getDouble("total\_cost"));

e.setPrddate(rs.getString("pr\_date"));

}

catch(SQLException ex){

db.setWarningMessage("Stock.setStock", ex.toString());

}

}

return e;

}

private void refreshDatabase(){

try{

rs = stmt.executeQuery("select \* from stock\_mstr");

}

catch(SQLException e){

System.out.println(e.toString());

Alert alert = new Alert(Alert.AlertType.WARNING);

alert.setHeaderText("Exception Raised in method StockModel.refreshDatabase");

alert.setContentText(e.toString());

alert.show();

}

}

public ObservableList<UserBeans> showTableRecords(){

ObservableList list = null;

try{

list = FXCollections.observableArrayList();

rs = stmt.executeQuery("select \* from stock\_mstr");

while(rs.next()){

list.add(setStock(rs));

}

updateRecord();

}

catch(SQLException ex){

db.setWarningMessage("StockModel.showTableRecords", ex.toString());

}

return list;

}

public UserBeans updateRecord(){

UserBeans e = null;

try{

rs = stmt.executeQuery("select \* from stock\_mstr");

rs.next();

e = setStock(rs);

}

catch(SQLException ex){

db.setWarningMessage("StockModel.updateRecord", ex.toString());

}

return e;

}

public boolean updateStock(UserBeans e){

boolean flag = true;

try{

String sql = "update stock\_mstr set pr\_type=?,qty=?,pr\_amt=?,total\_cost=?,pr\_date=? where pr\_name=?";

PreparedStatement p = conn.prepareStatement(sql);

p.setString(1,e.getPrdtype());

p.setInt(2,e.getQty());

p.setDouble(3,e.getPrdamt());

p.setDouble(4,e.getPrdcost());

p.setString(5,e.getPrddate());

p.setString(6,rs.getString("pr\_name"));

// ps.setString(4, e.getIname());

flag = p.execute();

}

catch(SQLException ex){

db.setWarningMessage("StockModel.updateItem", ex.toString());

}

return flag;

}

public UserBeans searchStock(String Prdname){

UserBeans e=null;

try{

String sql="select \* from stock\_mstr where pr\_name='"+Prdname+"'";

rs=stmt.executeQuery(sql);

if(rs.next()){

e=setStock(rs);

}

else{

db.setWarningMessage(" Stock Info does not Exist!","");

}

}

catch(SQLException ex){

db.setWarningMessage("StockModel.searchLeaveId",ex.toString());

}

return e;

}

}

Staff

package model;

import beans.UserBeans;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.logging.Level;

import java.util.logging.Logger;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.scene.control.Alert;

import javafx.scene.control.Alert.AlertType;

public class Staff {

private DbConnection db;

private Connection conn;

private Statement stmt;

private ResultSet rs;

private int total, current;

public Staff(){

try{

db = new DbConnection();

conn = db.getConnection();

stmt = db.getStatement();

rs = stmt.executeQuery("select (\*) from staff\_table");

if(rs.next()){

total = rs.getInt(1);

}

} catch (SQLException e) {

Logger.getLogger(Staff.class.getName()).log(Level.SEVERE, null, e);

db.setWarningMessage("Staff", e.toString());

}

}

public boolean insertaction(UserBeans e){

boolean flag = true;

try

{

String sql = "insert into staff\_table values(?,?,?,?,?,?,?,?)";

PreparedStatement p = conn.prepareStatement(sql);

p.setInt(1, e.getEid());

p.setString(2, e.getEname());

p.setInt(3, e.getEage());

p.setString(4, e.getEdesig());

p.setInt(5, e.getEsal());

p.setInt(6, e.getEmob());

p.setString(7, e.getEemail());

p.setString(8, e.getEadd());

flag = p.execute();

System.out.println("flag = "+flag);

}catch(SQLException ex){

db.setWarningMessage("Staff.insertaction", ex.toString());

}

return flag;

}

public UserBeans updateRecord(){

UserBeans e = null;

if(current>1){

try{

rs = stmt.executeQuery("select \* from staff\_table");

rs.next();

e = setStaff(rs);

}

catch(SQLException ex){

db.setWarningMessage("Staff.updateRecord", ex.toString());

}

}

return e;

}

public boolean updateStaffInfo(UserBeans e){

boolean flag=true;

try{

String sql="update staff\_table set emp\_id=?,age=?,design=?,salr=?,mob=?,email=?,addre=? where emp\_name=?";

PreparedStatement p=conn.prepareStatement(sql);

p.setInt(1, e.getEid());

p.setInt(2, e.getEage());

p.setString(3, e.getEdesig());

p.setInt(4, e.getEsal());

p.setInt(5, e.getEmob());

p.setString(6, e.getEemail());

p.setString(7, e.getEadd());

p.setString(8,rs.getString("emp\_name"));

flag=p.execute();

}

catch(SQLException ex){

db.setWarningMessage("StaffModel.updateStaff", ex.toString());

Alert alert = new Alert(AlertType.WARNING);

alert.setHeaderText("Exception Raised in method EmployeeModel.updateEmployeeInfo");

alert.setContentText(ex.toString());

alert.show();

}

return flag;

}

public UserBeans setStaff(ResultSet rs){

UserBeans e = null;

if(rs!=null){

try

{

e = new UserBeans();

e.setEid(rs.getInt("emp\_id"));

e.setEname(rs.getString("emp\_name"));

e.setEdesig(rs.getString("design"));

e.setEage(rs.getInt("age"));

e.setEsal(rs.getInt("salr"));

e.setEmob(rs.getInt("mob"));

e.setEadd(rs.getString("addre"));

e.setEemail(rs.getString("email"));

}

catch(SQLException ex){

db.setWarningMessage("Staff.setStaff", ex.toString());

}

}

return e;

}

private void refreshDatabase(){

try{

rs = stmt.executeQuery("select \* from staff\_table");

//int cur=1;

//for(rs.next(); cur<current; cur++, rs.next());

}

catch(SQLException e){

System.out.println(e.toString());

Alert alert = new Alert(AlertType.WARNING);

alert.setHeaderText("Exception Raised in method EmployeeModel.refreshDatabase");

alert.setContentText(e.toString());

alert.show();

}

}

public ObservableList<UserBeans> showTableRecords(){

ObservableList list = null;

try{

list = FXCollections.observableArrayList();

rs = stmt.executeQuery("select \* from staff\_table");

while(rs.next()){

list.add(setStaff(rs));

}

updateRecord();

}

catch(SQLException ex){

db.setWarningMessage("StaffModel.showTableRecords", ex.toString());

}

return list;

}

public UserBeans searchStaff(String Ename){

UserBeans e=null;

try{

String sql="select \* from staff\_table where emp\_name='"+Ename+"'";

rs=stmt.executeQuery(sql);

if(rs.next()){

e=setStaff(rs);

}

else{

db.setWarningMessage(" Staff Info does not Exist!","");

}

}

catch(SQLException ex){

db.setWarningMessage("StaffModel.searchLeaveId",ex.toString());

}

return e;

}

public boolean deleteItem(String Ename){

boolean flag = true;

try{

rs = stmt.executeQuery("delete from staff\_table where emp\_name='"+Ename+"'");

rs.next();

}

catch(SQLException ex){

db.setWarningMessage("StaffModel.deleteStaffInfo", ex.toString());

//flag = false;

}

return flag;

}

}

Sales

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package model;

import beans.UserBeans;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.logging.Level;

import java.util.logging.Logger;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

/\*\*

\*

\* @author DELL

\*/

public class Sales {

private DbConnection db;

private Connection conn;

private Statement stmt;

private ResultSet rs;

private int total;

public Sales(){

try{

db = new DbConnection();

conn = db.getConnection();

stmt = db.getStatement();

rs = stmt.executeQuery("Select \* from bill\_gen");

if(rs.next()){

total = rs.getInt(1);

}

} catch (SQLException e) {

Logger.getLogger(Bill.class.getName()).log(Level.SEVERE, null, e);

db.setWarningMessage("Bill", e.toString());

}

}

public UserBeans setSales(ResultSet rs){

UserBeans e = null;

if(rs!=null){

try

{

e = new UserBeans();

e.setBid(rs.getInt("b\_id"));

e.setBdate(rs.getString("bill\_date"));

e.setBtotal(rs.getDouble("bill\_total"));

}

catch(SQLException ex){

db.setWarningMessage("Bill\_gen.setSales", ex.toString());

}

}

return e;

}

public ObservableList<UserBeans> showTableRecords() {

ObservableList list = null;

try{

list = FXCollections.observableArrayList();

rs = stmt.executeQuery("select \* from bill\_gen");

while(rs.next()){

list.add(setSales(rs));

}

// updateRecord();

}

catch(SQLException ex){

db.setWarningMessage("Bill\_gen Model.showTableRecord1", ex.toString());

}

return list;

}

}

Db

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package model;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.sql.Statement;

import javafx.scene.control.Alert;

import javafx.scene.control.Alert.AlertType;

/\*\*

\*

\* @author DELL

\*/

public class DbConnection {

private Connection conn;

private Statement stmt;

public DbConnection(){

try{

Class.forName("oracle.jdbc.driver.OracleDriver");

conn=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","canteen","canteen");

}

catch(ClassNotFoundException e){

setWarningMessage("Dbconnection",e.toString());

}

catch(SQLException e){

setWarningMessage("Dbconnection",e.toString());

}

}

public Connection getConnection() {

return conn;

}

public Statement getStatement() {

try{

if(conn!=null){

stmt=conn.createStatement();

}

}

catch(SQLException e){

setWarningMessage("DbConnection.getStatement",e.toString());

}

return stmt;

}

public void setWarningMessage(String header,String content){

Alert alert=new Alert(AlertType.WARNING);

alert.setHeaderText("Exception :"+header);

alert.setContentText(content);

alert.showAndWait();

}

}

CONCLUSION

Working on the project “CANTEEN MANAGEMENT

SYSTEM” was a great learning experience.

It helped us by giving technical as well as interpersonal skills and we can definitely say that this period will always act as a stepping stone in our career as software professionals.

This type of project helps us a lot to enhance our skills in coding, designing the interfaces, learning how to operate databases and much more.

This project is user-friendly so that almost every end user can operate this software smoothly.It also helps to eliminate paper-based system in which there is a possibility to loss of information of the user and employees.Moreover,this type of paper-based system occupies more spaces to maintain different types of files.

Due to theses type of drawbacks there should be such types of digitalized software .Hence CANTEEN MANAGEMENT SYSTEM helps in digitizing all the previous paper works

This software is quite user-friendly i.e all interfaces or dialog windows in it are quite catchy. This project is so easy to use that every user can understand it easily. This helps to overcome the drawbacks of existing system.

It was a great learning experience for our team. It helped me to gain a technical as well as corporate world of organization.

BIBLIOGRAPHY

BOOKS REFFERED SITES VISITED

Java primer…E.Balaguruswami

System Analysis and Design by Elias Awad

Relational Database Management System by B.S. Desai

Oracle 10g sql Jason price

[www.google.com](http://www.google.com/) [www.wikipedia.com](http://www.wikipedia.com/)

[www.tutorialspoint.com](http://www.tutorialspoint.com/) [www.geeksforgeeks.com](http://www.geeksforgeeks.com/) [www.stackoverflow.com](http://www.stackoverflow.com/)

