OIL DELIVERY SYSTEM

Software Document

XOCODE Team

W.Suhada Prasanna Perera

# Documentation About

The purpose of this document is to describe about Oil Delivery System , how it was developed and how it works. In this document will describe What is this software about?, How it works?, What are the component have used?, Description about user interfaces and description about how it works and also Who developed this software.

# About Software

This is Oil Delivery system that can manage oil delivery process. Using this system user can manage products, delivery logs, money control, customer managing, supplier managing and bank deals managing.

# Client Requirement

Client’s requirement was that client needed to manage his business process and get business status reports on specific date. Also this client has oil bottle production department and his own oil selling shop.

There are two types of employees. They are permanent and temporary employees. Employees payment will be decide by the owner of the company.

# Components

While developing this system we have used some predefine components and frame works,

* Java 0.8
* Java Swing Library for design user interface
* MySql Database
* Hibernate Frame work for manage database
* JDate component
* Jasper Report for generate reports

# Process & User Interfaces

When starting the software, it will load database(Hibernate) and after that will redirect to the login interface.

On start system

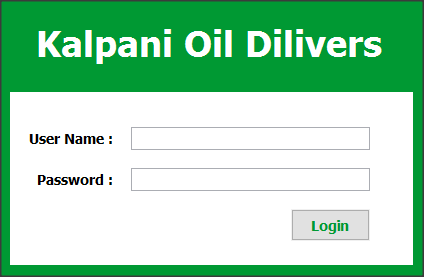
Session ses = conn.NewHibernateUtil.getSessionFactory().openSession();

ses.beginTransaction().commit();

ses.close();

## login

When start the software it will go to login window. After enter the user’s username and password user can login to the system.



After entering correct username and password user will redirect to the main page of system.

on login button click

new m.Login().login(txt\_uname.getText(), txt\_pword.getText(), this);

public void login(String uname, String pass, JFrame frame) {

Session ses = conn.NewHibernateUtil.getSessionFactory().openSession();

try {

Criteria cry = ses.createCriteria(pojo.Login.class);

cry.add(Restrictions.eq("username", uname));

cry.add(Restrictions.eq("password", pass));

pojo.Login log = (pojo.Login) cry.uniqueResult();

if (log != null) {

System.out.println("Login Sucsees");

v.Home.getInstance().setVisible(true);

frame.dispose();

} else {

System.out.println("Try Again");

JOptionPane.showMessageDialog(frame, "Username or Password is wrong");

}

} catch (Exception e) {

e.printStackTrace();

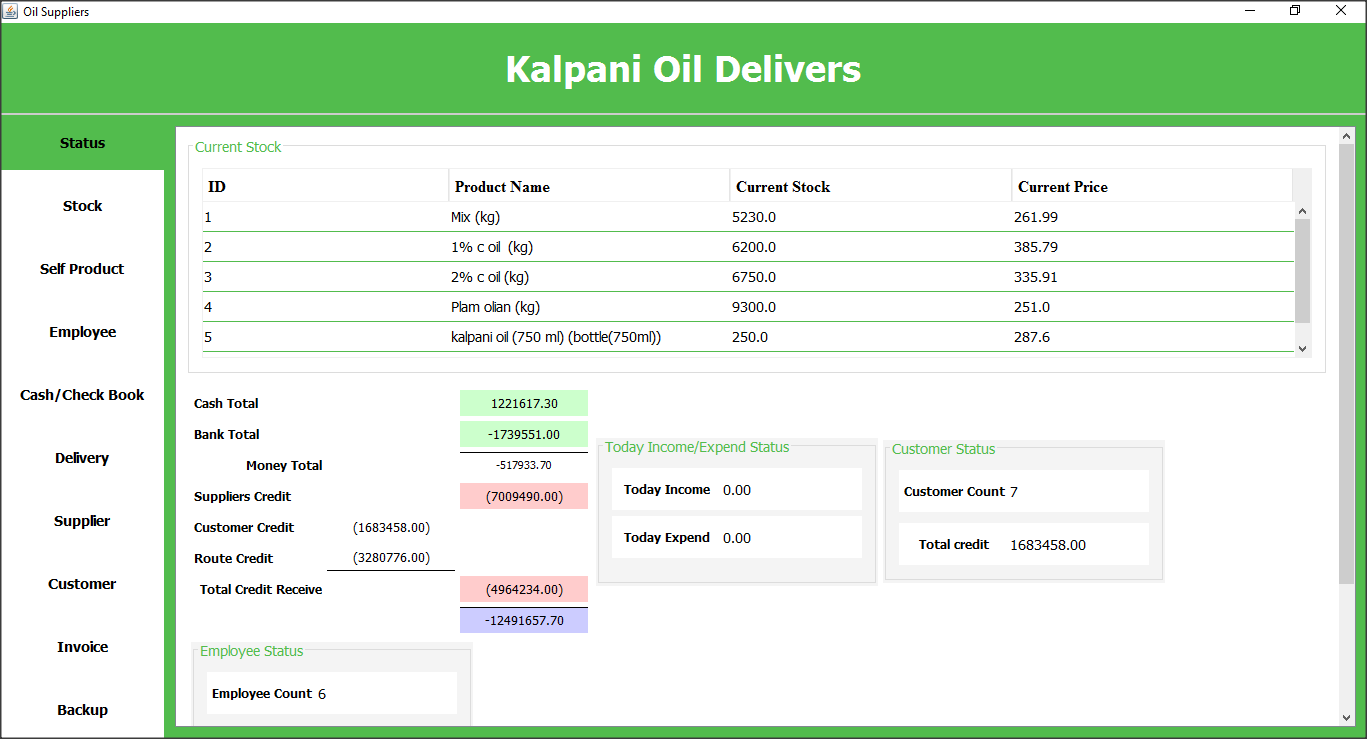
} finally {

ses.close();

}

}

## Main Page (Status)



this main page(status) will display summary details about business status. It will show product status (Current product and current price), business cash status, daily income and expend, customer status (how many customers have registered and total credit they have got).

On load Interface

setProductSummoryToTable(this.status.tbl\_currentstock);

MoneyBook moneyBook = new m.MoneyBook();

ValueValidation validation = m.ValueValidation.getInstance();

this.status.lbl\_income\_today.setText(validation.toDeciaml(moneyBook.getTotalIncomeBy(new Date()), 2));

this.status.lbl\_expend\_today.setText(validation.toDeciaml(moneyBook.getTotalExpendBy(new Date()), 2));

this.status.lbl\_customer\_count.setText(new m.Customer().getCustomerCount()+"");

customerCredit=new m.CusCredit().getTotalCredit();

this.status.lbl\_customer\_total\_credit.setText(validation.toDeciaml(customerCredit, 2));

this.status.lbl\_employee\_count.setText(new m.Employee().getEmployeeCount()+"");

cash=new m.AssetStatus().getById(1).getAmount();

bank=new m.Bank().getTotal();

supplierCredit=new m.Supplier().getTotalCredit();

routeCredit=new m.RouteReg().getTotalCredit();

this.status.lbl\_cash\_total.setText(validation.toDeciaml(cash, 2));

this.status.lbl\_bank\_total.setText(validation.toDeciaml(bank, 2));

this.status.lbl\_total\_cash.setText(validation.toDeciaml(bank+cash, 2));

this.status.lbl\_supplier\_credit.setText("("+validation.toDeciaml(supplierCredit, 2)+")");

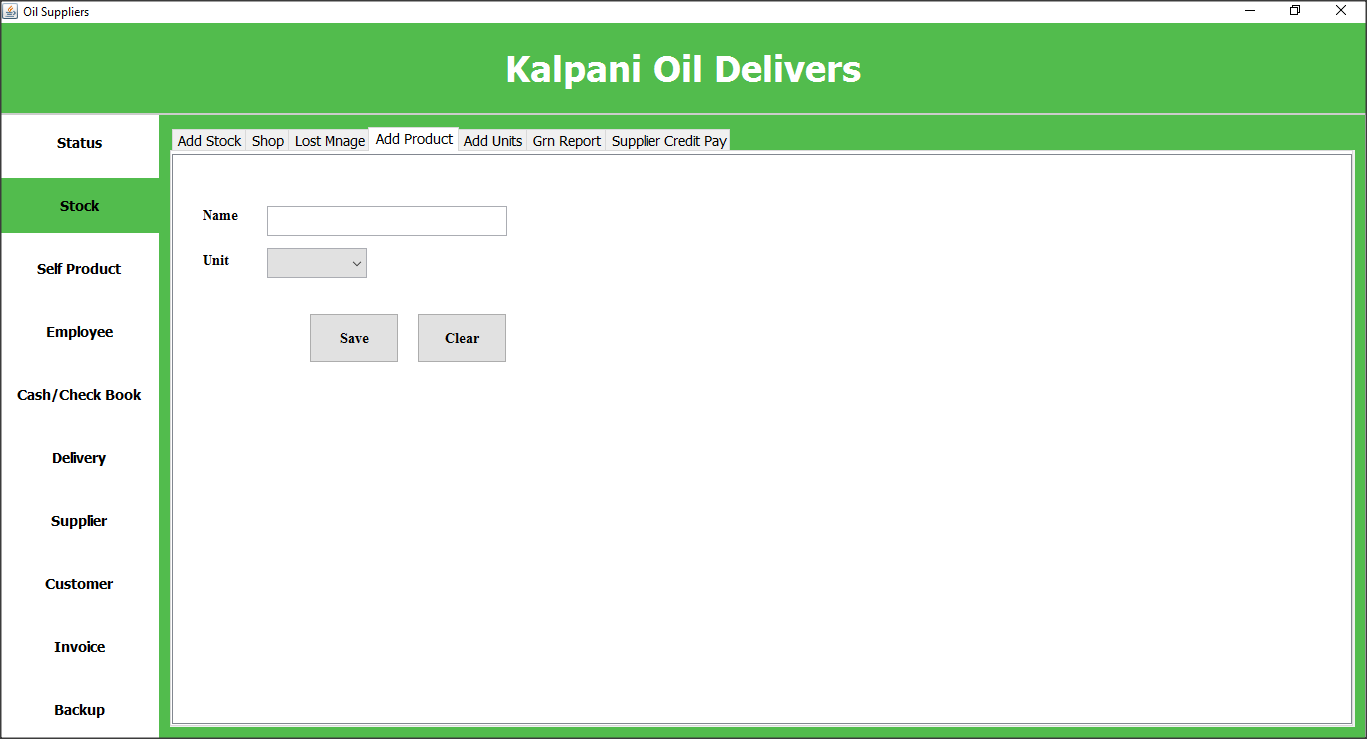
this.status.lbl\_customer\_credit.setText("("+validation.toDeciaml(customerCredit, 2)+")");

this.status.lbl\_route\_credit.setText("("+validation.toDeciaml(routeCredit, 2)+")");

this.status.lbl\_total\_credit\_receive.setText("("+validation.toDeciaml(routeCredit+customerCredit, 2)+")");

this.status.lbl\_balance.setText(validation.toDeciaml((cash+bank)-(routeCredit+customerCredit+supplierCredit), 2));

## Add New Product



Using this user interface client can register new products to the system. First enter product name and then choose product measure type and press save. Then it will save as new product.

On load Interface

loadProductToCombo();

clearShop();

On Save Button Click

st = new StockControler();

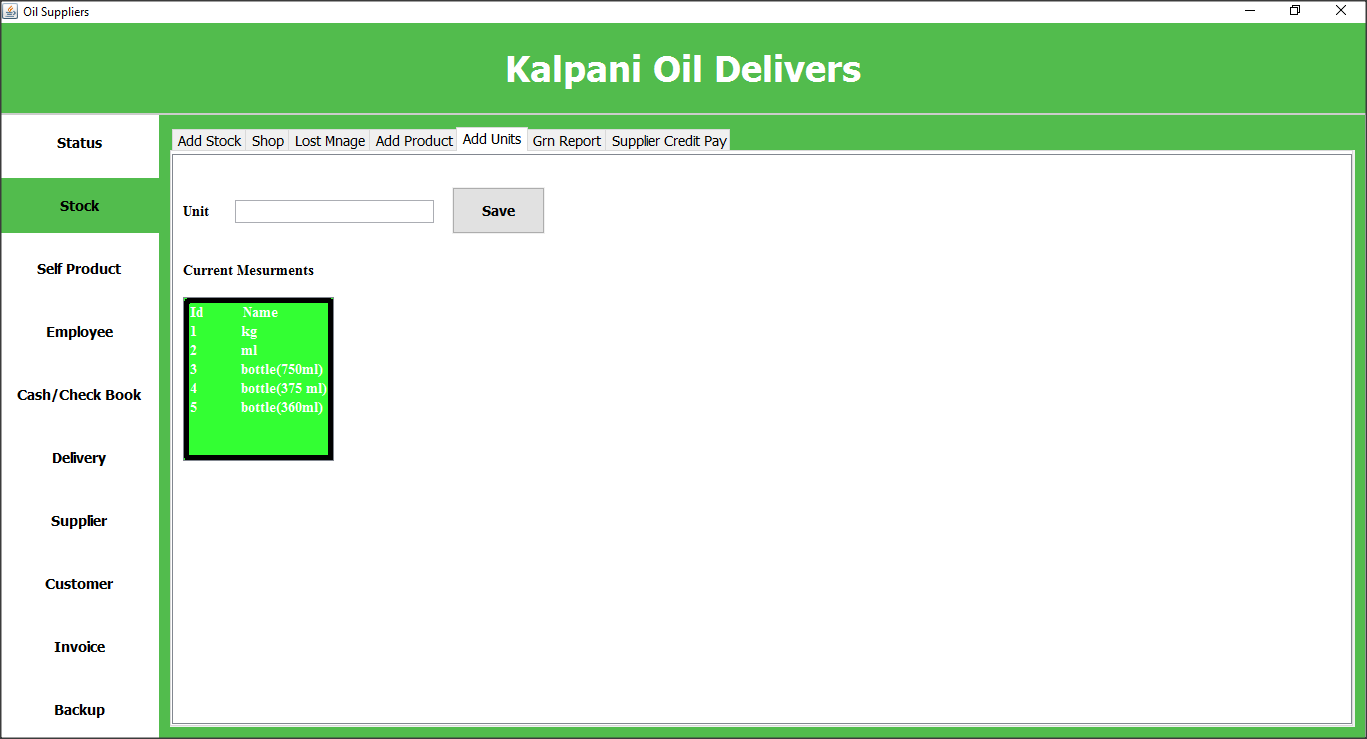
Units id = st.setunitid((String) combounitpr.getSelectedItem());

st.productSave(txtproductname.getText(), id);

txtproductname.setText("");

combounitpr.setSelectedItem(null);

## Add New Unit (Measure type)



Client can add product measure types using this interface.

On Load Interface

loadProductToCombo();

clearShop();

On Save Button Click

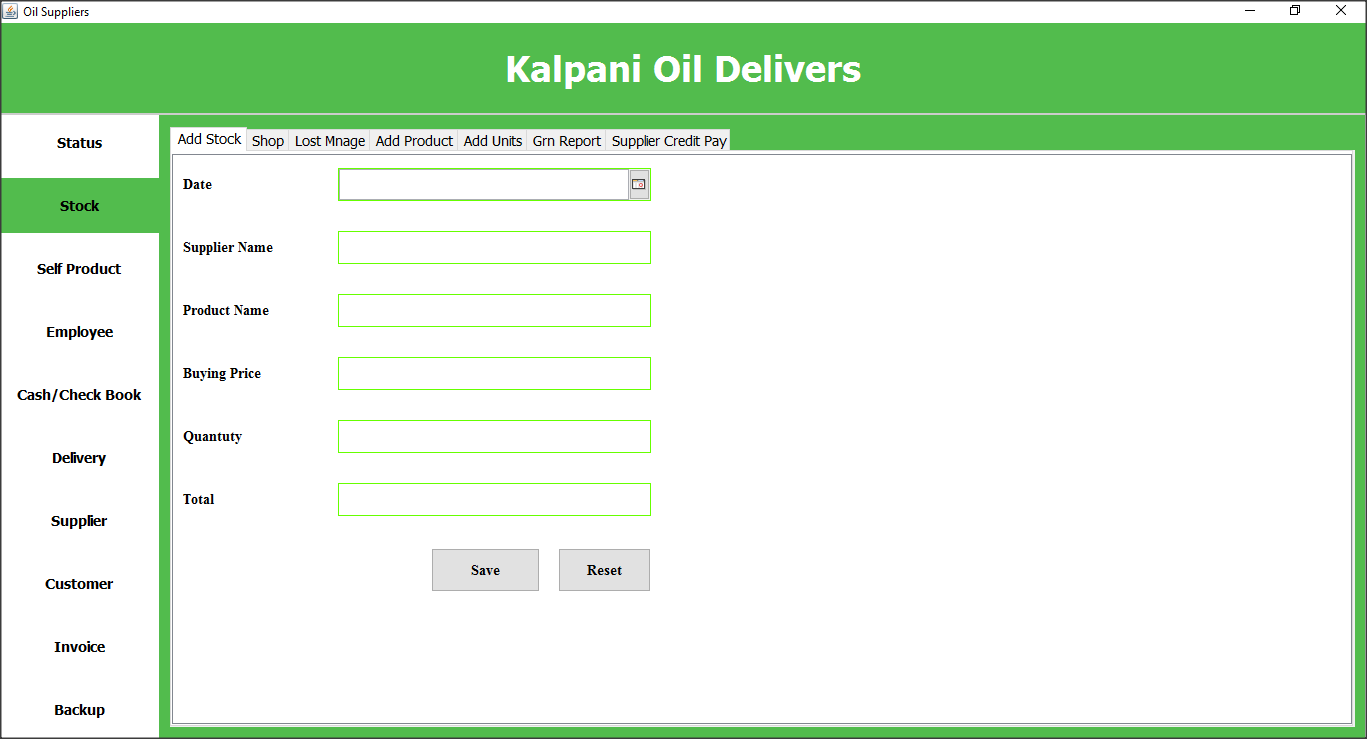
st = new StockControler();

st.unitSave(txtunitname.getText());

listView.setTableData(listunit);

txtunitname.setText("");

## Add New Stock

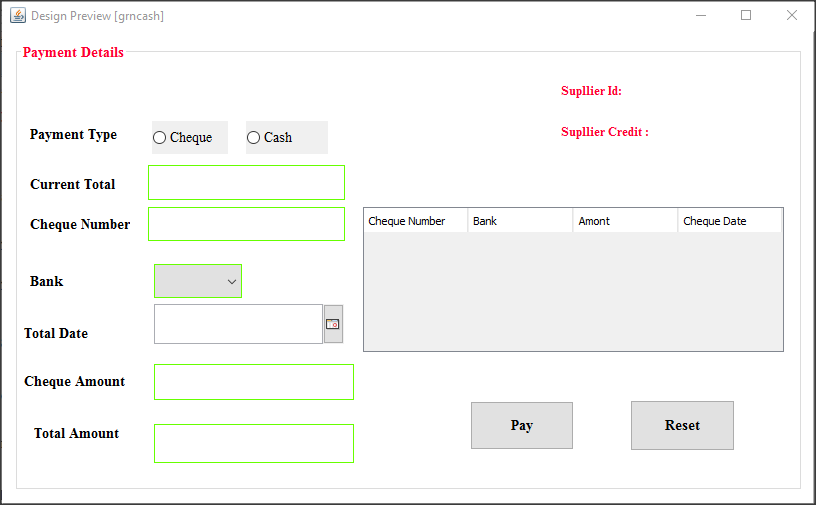


Client can add new stock using this interface.

On Click Save Button

viewgrncash();

## GRN Generator



Using this interface can generate the grn.

On Pay Button Click

if (radiocheque.isSelected() || radiocash.isSelected()) {

if (!txttototalamount.getText().trim().isEmpty()) {

double totammount = Double.parseDouble(txttototalamount.getText());

double tot1 = Double.parseDouble(lblsupcredit.getText());

double tot2 = Double.parseDouble(lbltotal.getText());

if (tot1 >= 0) {

if (tot2 >= 0) {

if (totammount >= 0) {

if (tot1 + tot2 >= totammount) {

int done = saveGrnNloadGrnCash();

if (done != 0) {

pay();

} else {

Errormzg.displayerrorMessage("Unable to save stock");

}

} else {

Errormzg.displayerrorMessage("please enter less value");

}

}

}

}

} else {

Errormzg.displayerrorMessage("Empty Fields");

}

} else {

Errormzg.displayerrorMessage("please select pay method");

}

On Reset Button Click

public void clearcash() {

lbltotal.setText("");

txtchequeno.setText("");

cmbbank.setSelectedItem(null);

txtchequeamount.setText("");

txttototalamount.setText("");

lblsupid.setText("");

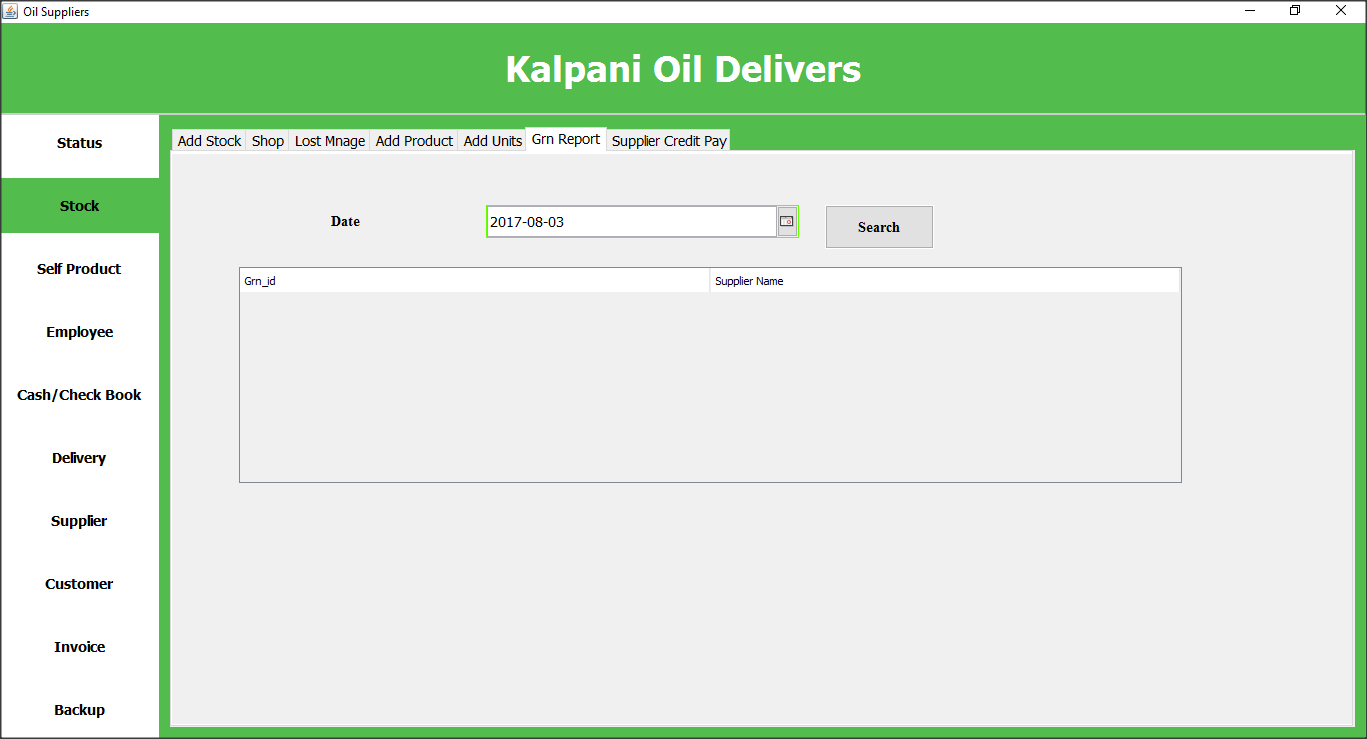
lblsupcredit.setText("");

DefaultTableModel model = (DefaultTableModel) this.tblchequedetails.getModel();

model.setRowCount(0);

}

## View Grn



Client can view GRN Reports using this interface. By selecting date, can get grn that issues on that day.

On Search Button Click

try {

Product products = new Product();

ArrayList<pojo.Grn> grnlist = new m.Grn().getBy(grnreportdate.getDate());

DefaultTableModel defaultTableModel = (DefaultTableModel) tblgrnreport.getModel();

defaultTableModel.setRowCount(0);

m.Supplier s = new m.Supplier();

for (pojo.Grn g : grnlist) {

pojo.Supplier supplier = s.getBy(g.getSupplier().getId());

Vector v = new Vector();

v.add(g.getId().toString());

v.add(supplier.getName());

defaultTableModel.addRow(v);

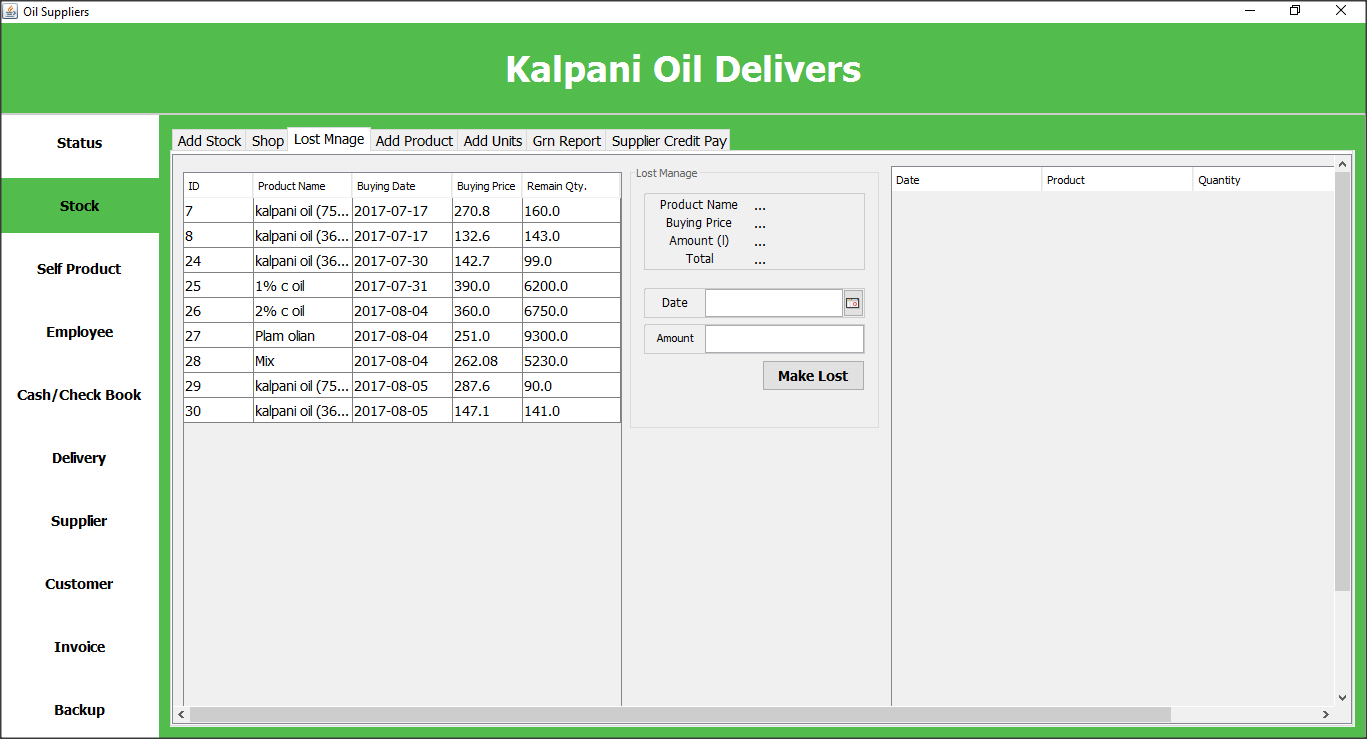
}

} catch (Exception e) {

e.printStackTrace();

}

## Lost Manage



user can manage product lost using this user interface.

On Interface Load

this.setMsg(v.Stock.getInstance().getLabelMsgLost());

this.setLost\_productname(v.Stock.getInstance().getLabelProductnameLost());

this.setLost\_buyingprice(v.Stock.getInstance().getLabelBuyingPriceLost());

this.setLost\_amount\_lbl(v.Stock.getInstance().getLabelAmountLost());

this.setLost\_total(v.Stock.getInstance().getLabelTotalLost());

if(setTbl(v.Stock.getInstance().getTblLostGrn())){

this.loadAvailabelStock();

this.reset();

}

if(setTbl(v.Stock.getInstance().getTblLost())){

this.loadLost();

}

On Make Lost Button Click

int id=(int)this.getTbl().getValueAt(this.getTbl().getSelectedRow(), 0);

m.GrnLog GRNLog=new m.GrnLog();

GrnLog grn = new m.GrnLog().getById(id);

Product Product = new m.Product();

pojo.Product product = Product.getByGRNLog(grn);

if(this.lost\_amount<=grn.getRemainingQuantity()){

Lost Lost = new pojo.Lost();

Lost.setAmount(this.lost\_amount);

Lost.setDate(new Date());

Lost.setGrnLog(grn);

String add = new m.Lost().add(Lost);

if(add.equalsIgnoreCase("done")){

product.setCurrentStock(product.getCurrentStock()-this.lost\_amount);

Product.update(product);

DefaultTableModel dtm=(DefaultTableModel) this.tbl.getModel();

double remain = grn.getRemainingQuantity()-this.lost\_amount;

dtm.setValueAt(remain, this.tbl.getSelectedRow(), 4);

grn.setRemainingQuantity(remain);

GRNLog.update(grn);

JOptionPane.showMessageDialog(null, "Added Successful.","Message",JOptionPane.INFORMATION\_MESSAGE);

setTbl(v.Stock.getInstance().getTblLost());

this.loadLost();

}else JOptionPane.showMessageDialog(null, "Adding failed.","Fail",JOptionPane.WARNING\_MESSAGE);

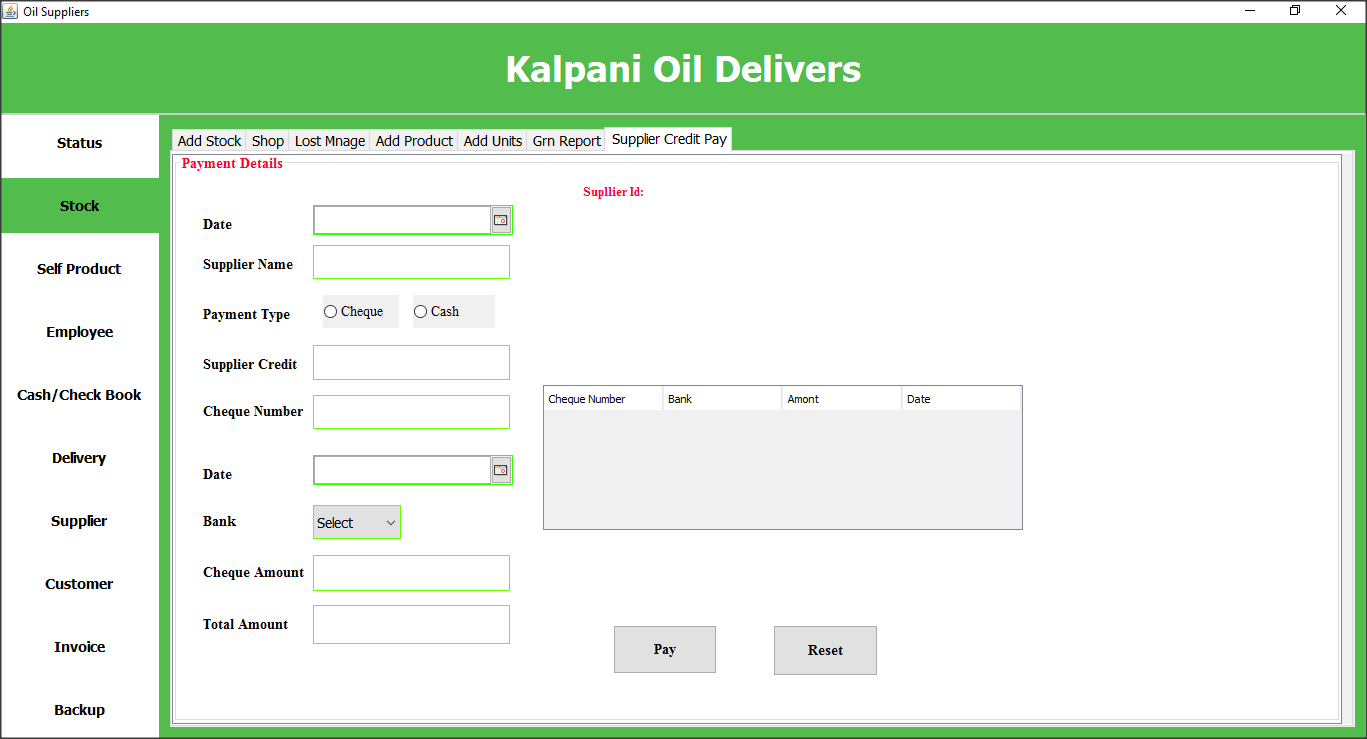
}else{

JOptionPane.showMessageDialog(stock, "Not enogh quantity.Enter quantity less than or equal \n according to the remain quantity \n reset reduce from another stock");

}

product.getCurrentPrice();

## Supplier Credit Manage

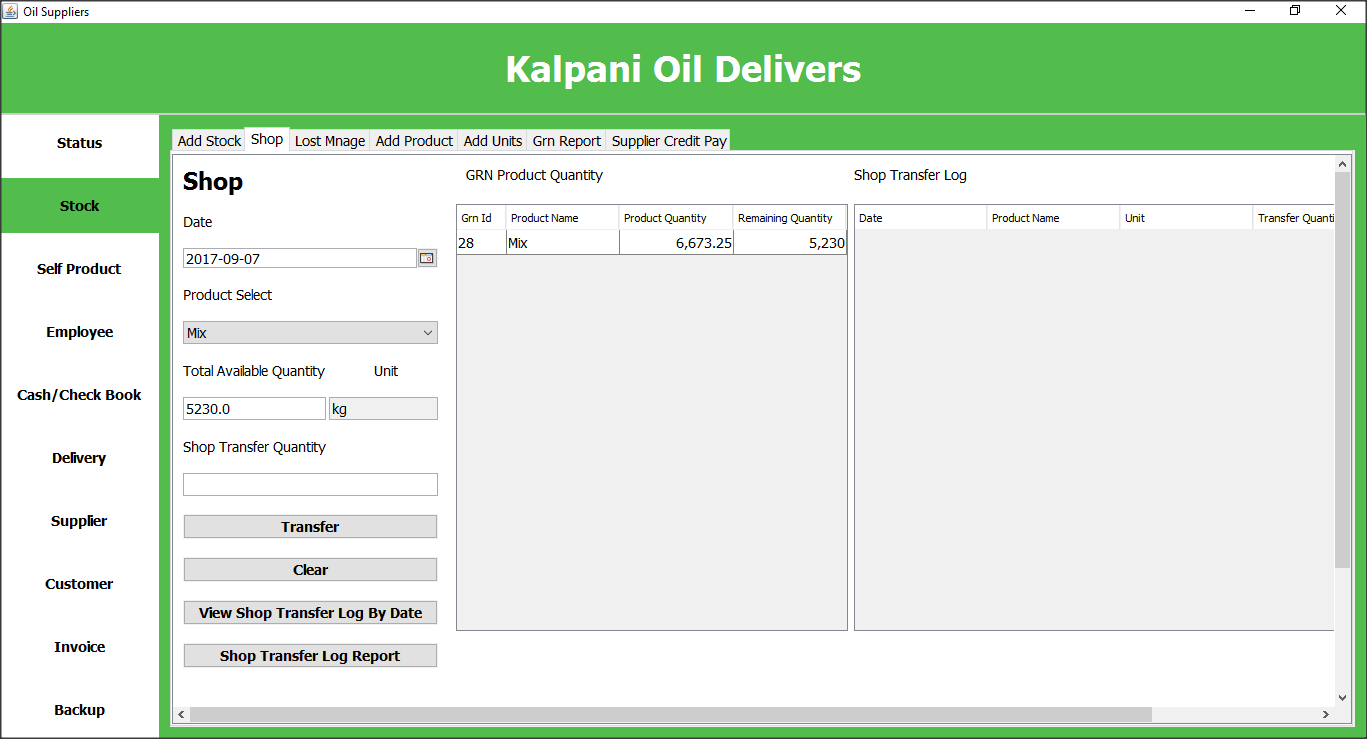


User can manage supplier credit using this user interface.

On Pay Button Click

paycredit();

## Transfer Item to Shop

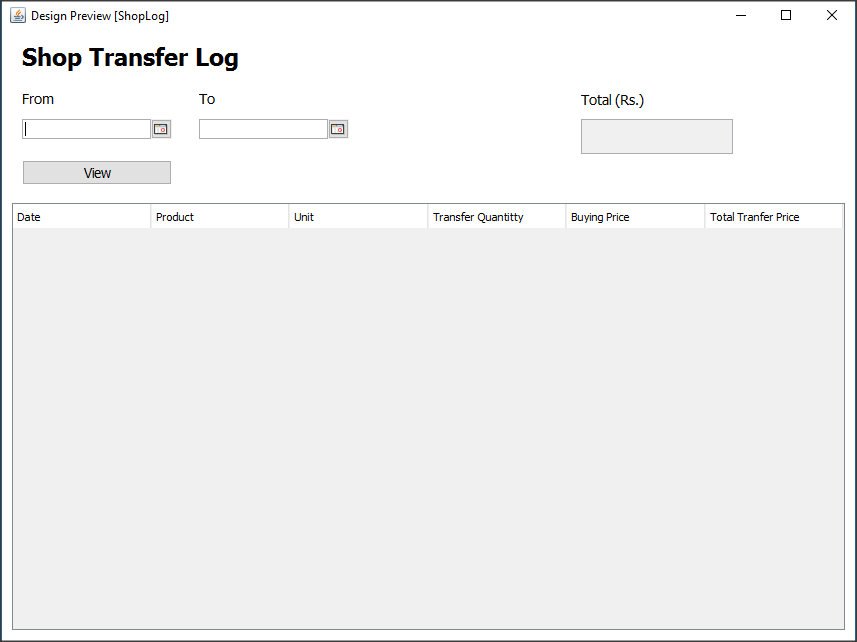


user can transfer their products to shop using this interface. Also can view transferred product details

On Transfer Button Click

new ShopTransferSave().saveShopTransfer(dc\_shopDate, tbl\_viewProduct, tf\_quantity);

## Shop TransferLog View



User can view shop transfer details using this interface

On View Button Click

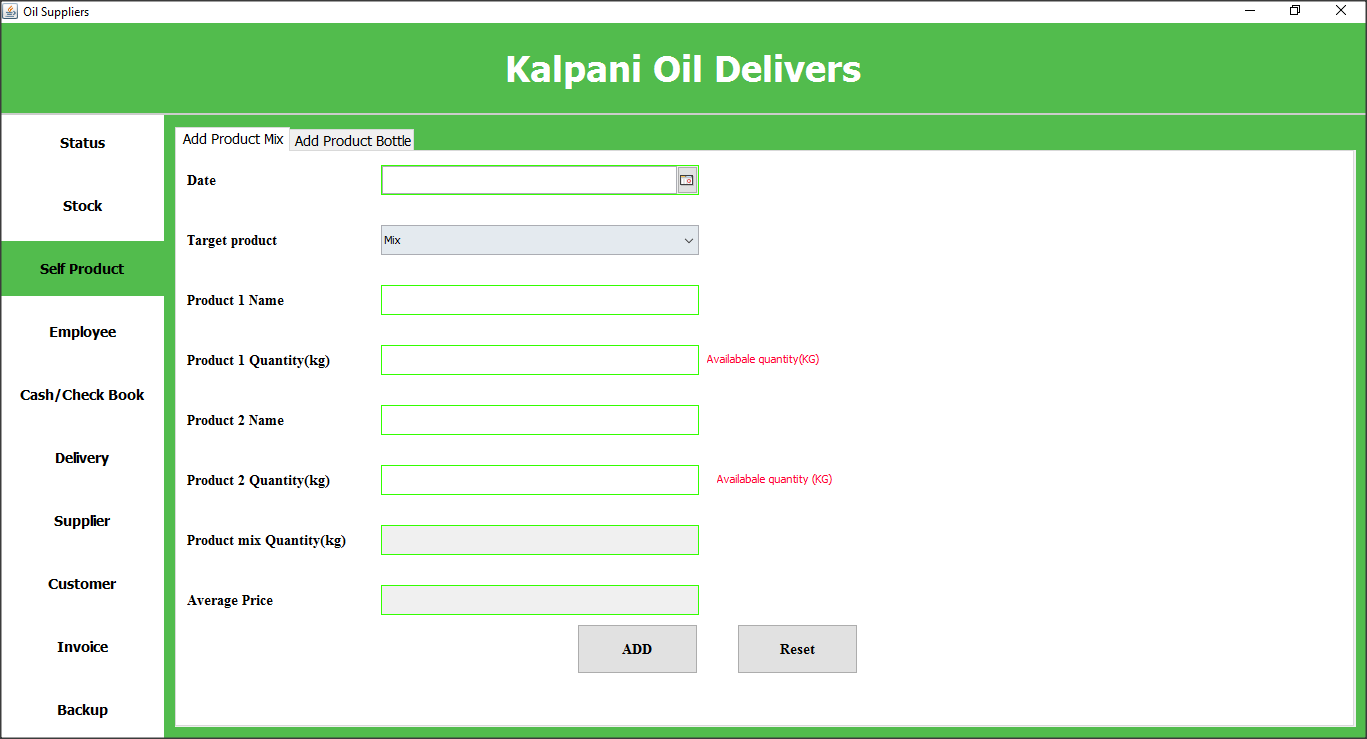
ShopLogControl shopLogControl = new c.ShopLogControl();

shopLogControl.viewShopLog(dc\_from, dc\_to, tbl\_shopLog);

Double tableTotal = shopLogControl.getTableTotal(tbl\_shopLog);

tf\_total.setText(tableTotal.toString());

## Add Product Mix



client produce some product his own by mixing other products. User can add details about that mixed products using this interface.

On Add Button Click

public int addMix() {

try {

if (pr1stock <= findPridByName1.getCurrentStock()) {

if (pr2stock <= findPridByName2.getCurrentStock()) {

int mixquantity = new c.selfproductcontroller().getStockToMix(findPridByName1, findPridByName2, pr1stock, pr2stock);

if (mixquantity != 0) {

GrnLog grnlogpojo = new c.selfproductcontroller().savegrnlog(mixdate.getDate(), mixstock, averageprice,cmbtargetproduct.getSelectedItem().toString());

if (grnlogpojo != null) {

new c.selfproductcontroller().savemix(findPridByName1, findPridByName2, pr1stock, pr2stock, mixdate.getDate(), mixstock, averageprice, grnlogpojo);

pojo.Product pr = new m.Product().getByName(cmbtargetproduct.getSelectedItem().toString());

int i = new c.StockControler().getaverageprice(pr);

if (i != 0) {

Errormzg.displayconfirmMessage("Stock added succssfully");

return i;

} else {

Errormzg.displayerrorMessage("Unable to Add");

return 0;

}

} else {

Errormzg.displayerrorMessage("unable to add grn");

return 0;

}

} else {

Errormzg.displayerrorMessage("Unable to add grn");

return 0;

}

} else {

Errormzg.displayerrorMessage("Available quantity is not enough");

return 0;

}

} else {

Errormzg.displayerrorMessage("Available quantity is not enough");

return 0;

}

} catch (Exception e) {

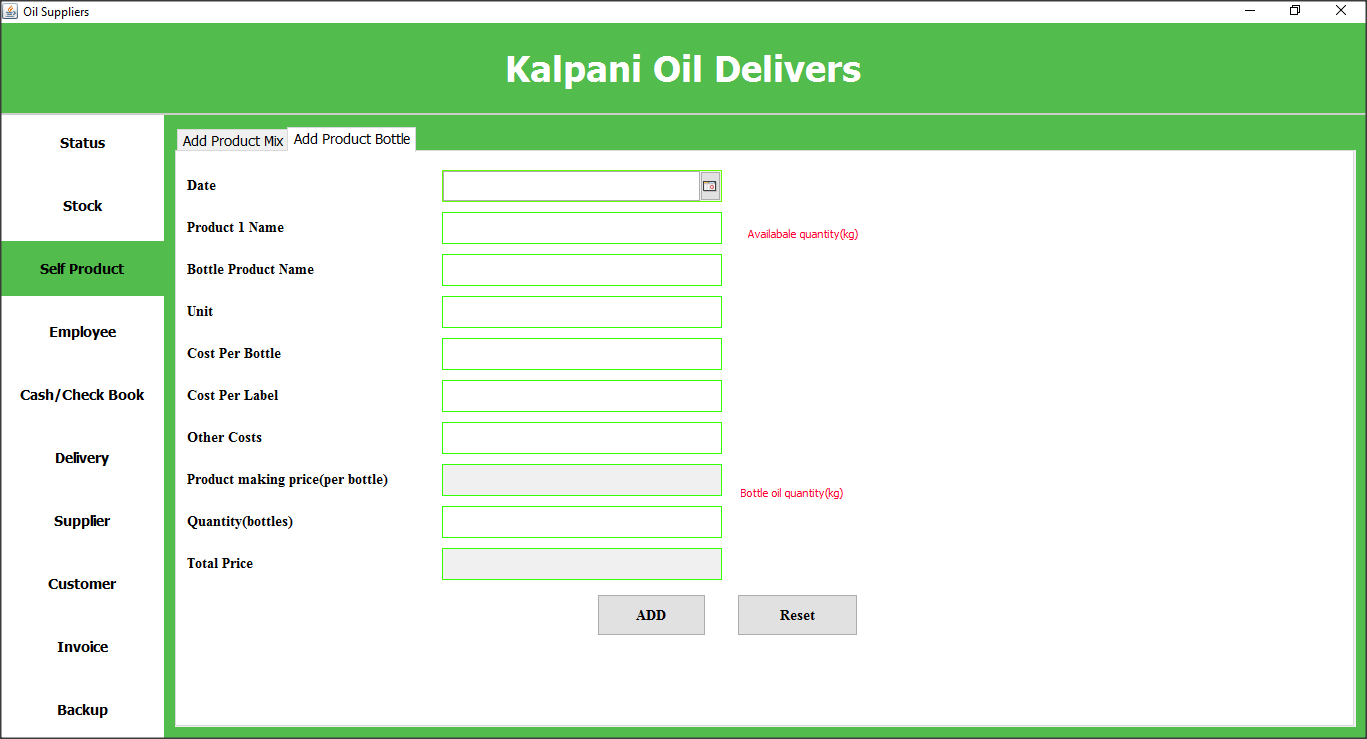
e.printStackTrace();

return 0;

}

}

## Bottle Production



This client also product oil bottles. Using this window user can add bottle production details.

On Add Button Click

int ok = bottleadd();

if (ok != 0) {

saveExtraCosts();

int ook = new c.StockControler().getaverageprice(findbottlepr1ByName1);

if (ook != 0) {

Errormzg.displayconfirmMessage("stock added");

clearbtl();

} else {

Errormzg.displayerrorMessage("unable to add");

}

} else {

Errormzg.displayerrorMessage("unable to add moneybook");

}

public int saveExtraCosts() {

try {

double extracost = morecost \* bottleqty;

extracost = Math.round(extracost \* 100.0) / 100.0;

int done = new c.selfproductcontroller().btlExtraCosts(extracost, bottledate.getDate());

return done;

} catch (Exception e) {

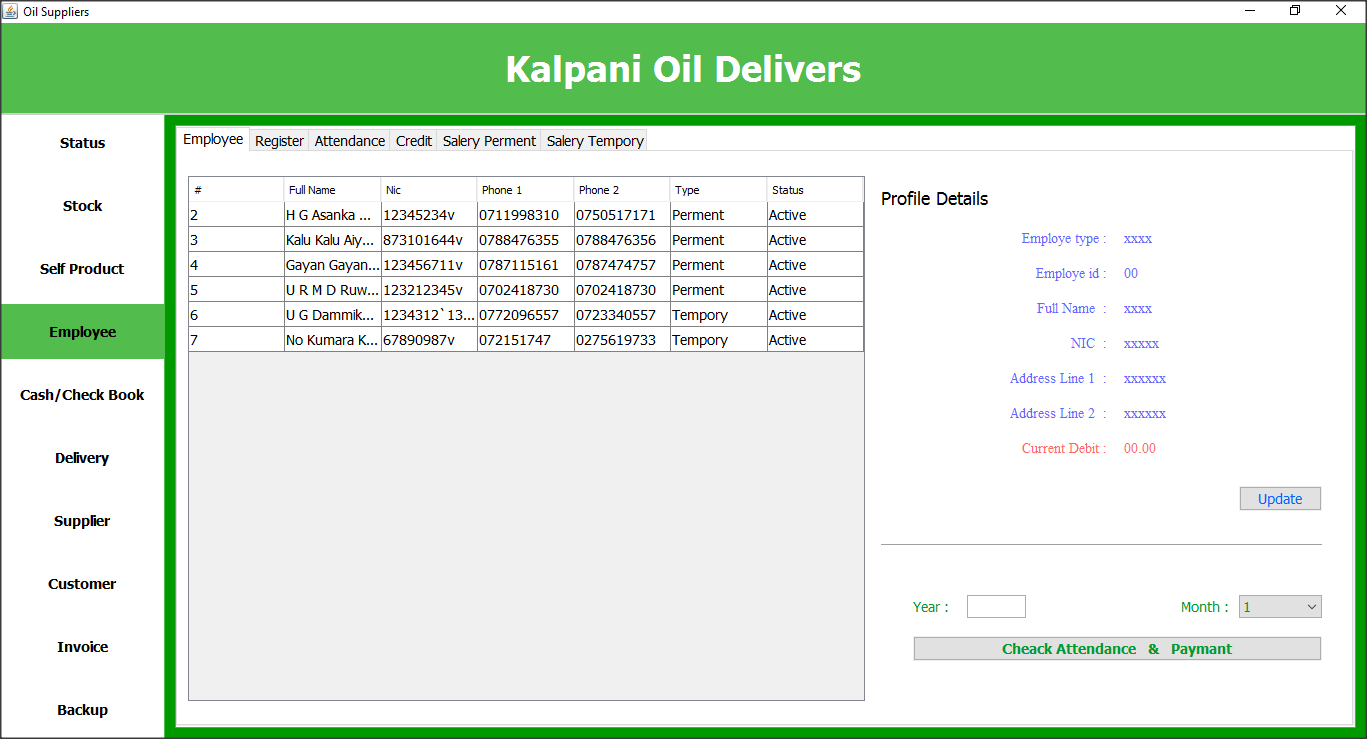
e.printStackTrace();

return 0;

}

}

## View Employee Details



Using this window user can view employees details that have register in system. Also can check their attendance and salary payment’s thru this window.

On Check attendance and payment button click

v.Attendance at = new Attendance(c\_month, lb\_eid, tx\_year);

at.setVisible(true);

On Click Table

public void LoadEmployee(JTable tbl\_emp, JLabel lb\_etype, JLabel lb\_eid, JLabel lb\_efullname, JLabel lb\_enic, JLabel lb\_eaddress1, JLabel lb\_eaddress2, JLabel lbl\_cucredit) {

Session ses = conn.NewHibernateUtil.getSessionFactory().openSession();

int id = Integer.parseInt(tbl\_emp.getValueAt(tbl\_emp.getSelectedRow(), 0).toString());

try {

pojo.Employee empo = (pojo.Employee) ses.createCriteria(pojo.Employee.class).add(Restrictions.eq("id", id)).uniqueResult();

setEmployee(empo);

lb\_etype.setText(empo.getEmpType().getType());

lb\_eid.setText(empo.getId().toString());

lb\_efullname.setText(empo.getSname() + " " + empo.getFname() + " " + empo.getMname() + " " + empo.getLname());

lb\_enic.setText(empo.getNic());

lb\_eaddress1.setText(empo.getAdressLine1());

lb\_eaddress2.setText(empo.getAdressLine2());

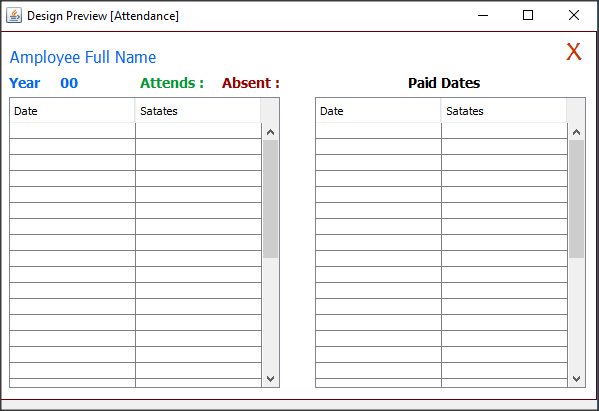
lbl\_cucredit.setText(v.Employee.getRound(empo.getCurrentCredit()));

} catch (Exception e) {

} finally {

ses.close();}}

## Employee Attendance View



User can view employee attendance and payments using this window.

On Load Window

this.month = month.getSelectedItem().toString();

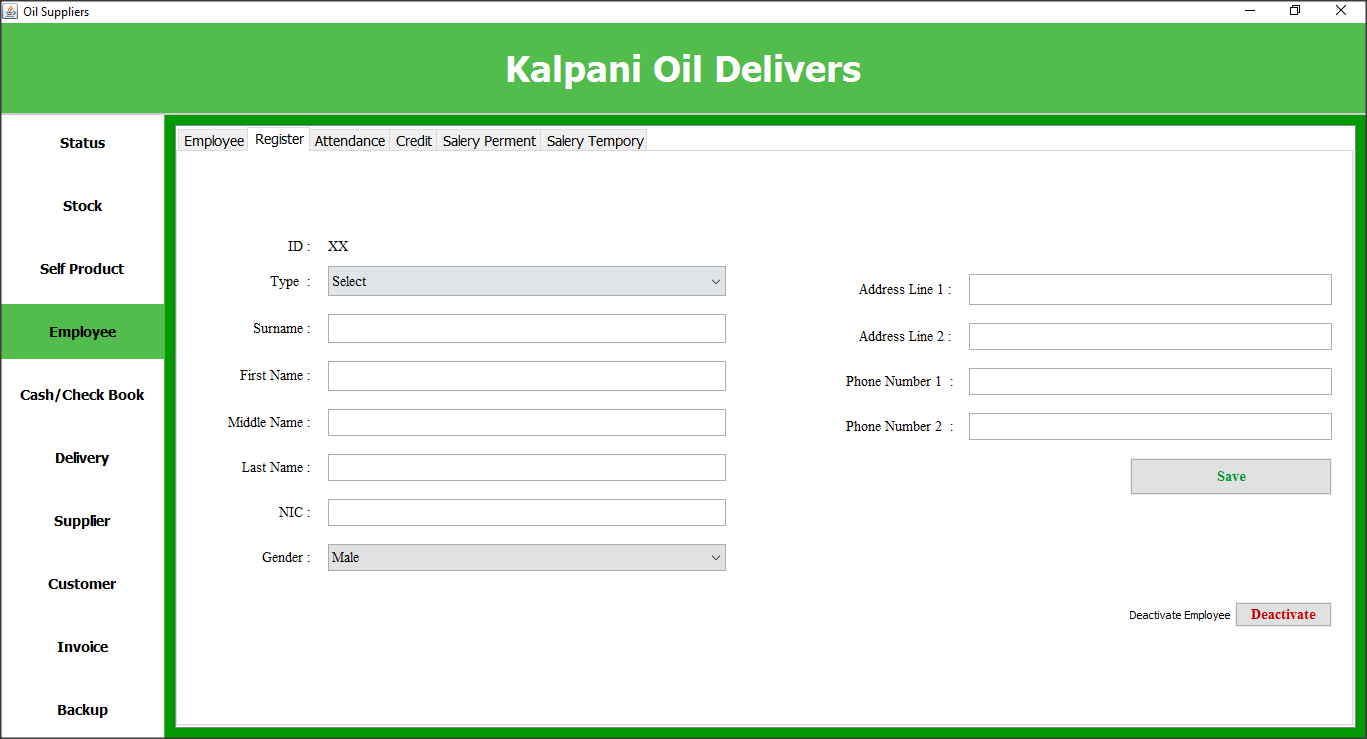
this.id = Integer.parseInt(id.getText());

lb\_mont.setText(this.month);

this.year = year.getText();

getAttendCountByEmpPerMont(this.month, this.id);

## Employee Register



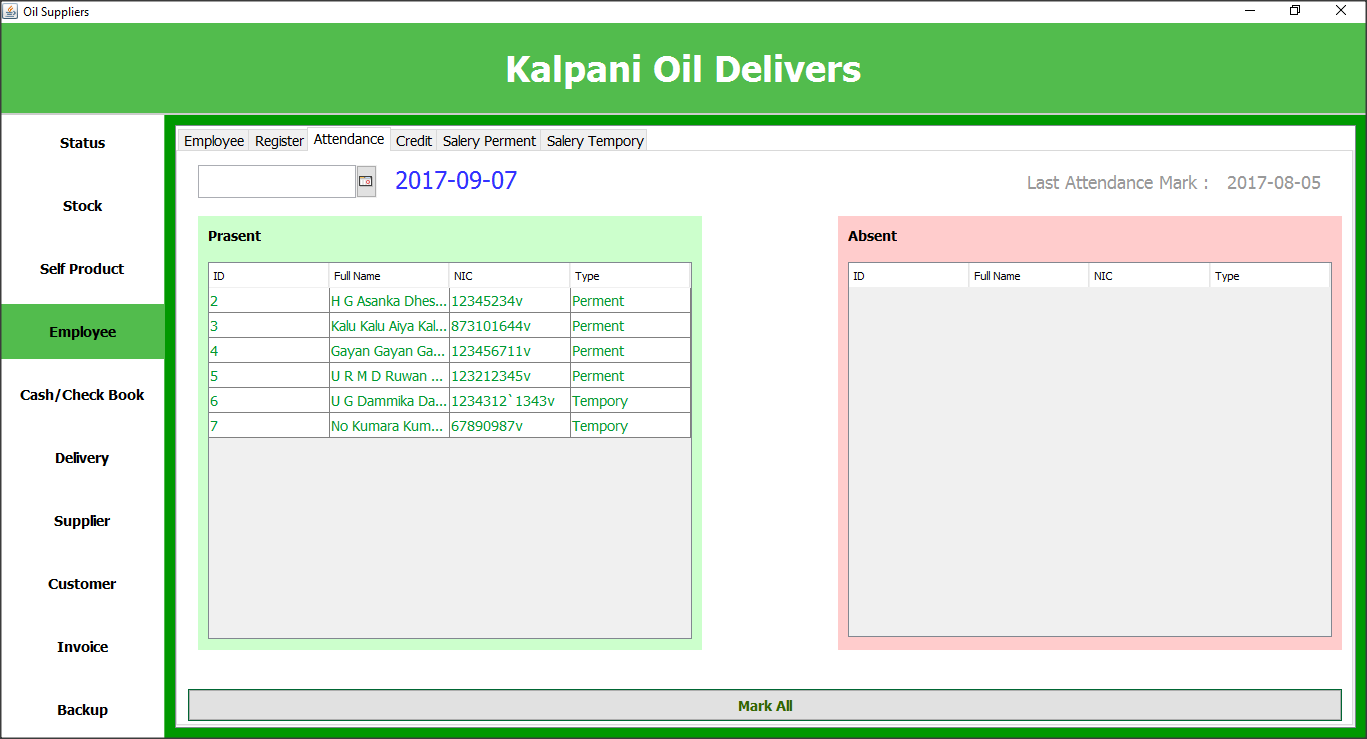
User can register new employee using this window.

On Save Button Click

employee.save(lbl\_empid, combo\_emtype, tf\_sname, tf\_fname, tf\_mname, tf\_lname, tf\_nic, combo\_gender, tf\_addressline1, tf\_addressline2, tf\_phone1, tf\_phone2);

loadTables();

## Employee Attendance Manage



User can manage employee’s attendance thru this window.

On Present Table Click

attendance.selectAbsent(tbl\_attend, tbl\_absent);

On Absent Table Click

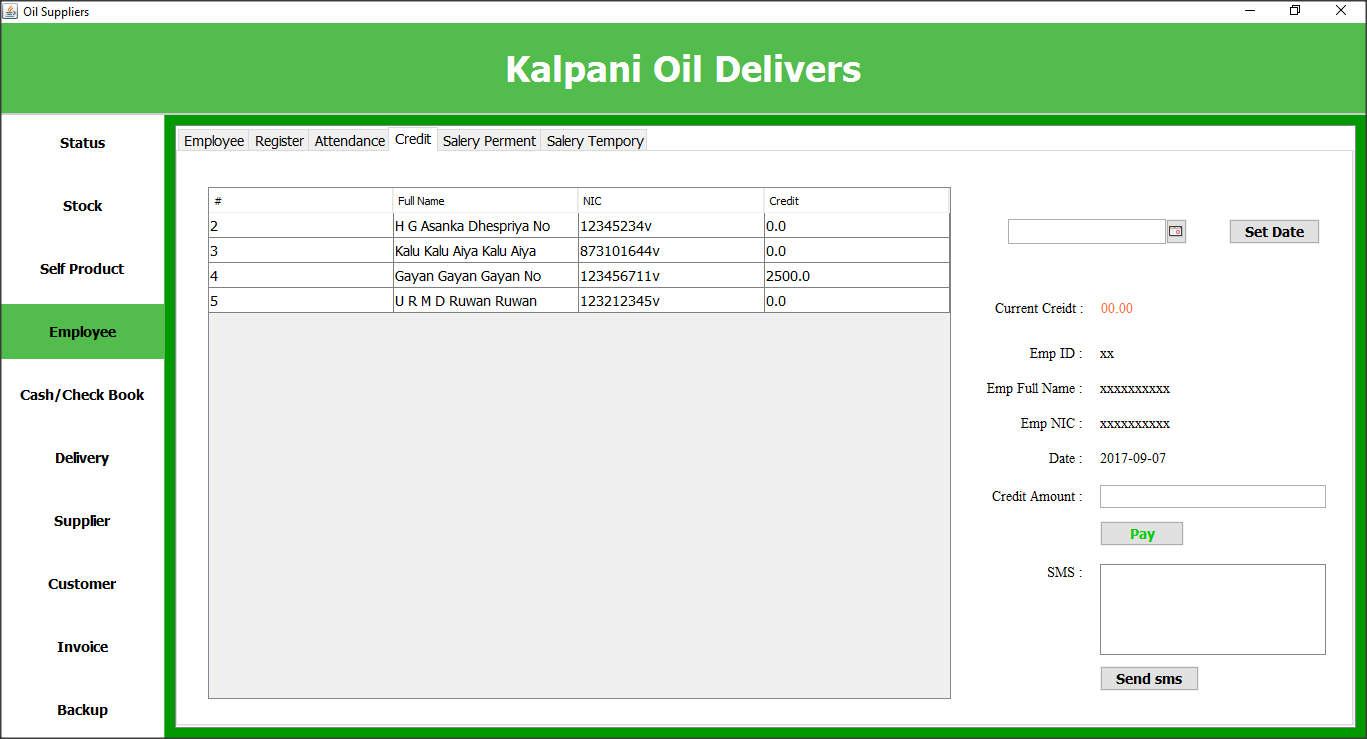
attendance.selectPresent(tbl\_absent, tbl\_attend);

On Mark All Button Click

attendance.marckAttendance(tbl\_attend, tbl\_absent, label\_today, dc\_attendance);

attendance.getLastAttendanceMarckDate(lastdate);

## Employee Credit Manage



User can manage employee’s credit using this window.

On Pay Button Click

empcredit.pay(lb\_empid, tf\_creditamount, tf\_sms, lb\_Credit\_Date);

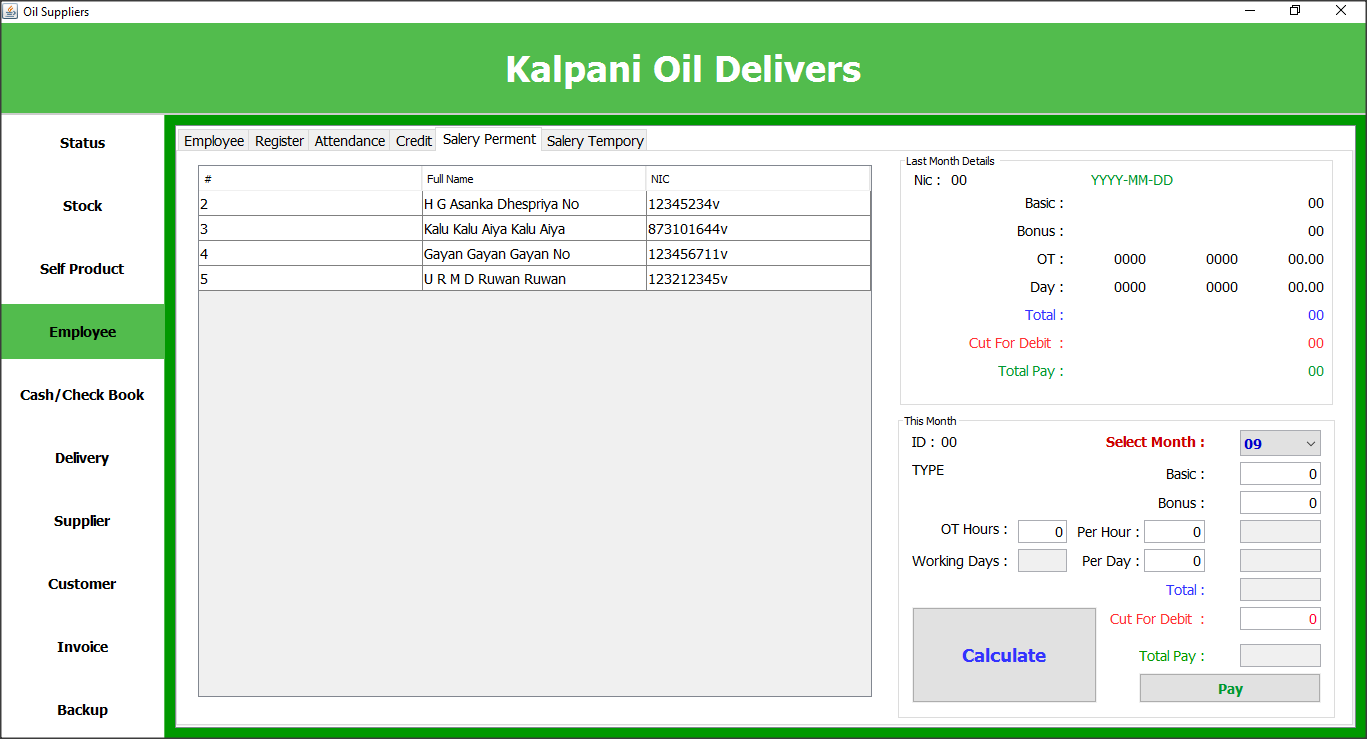
loadTables();

On Send sms Button Click

empcredit.sendSMS(lb\_empid, tf\_creditamount, tf\_sms); // SMS SEND

loadTables();

## Salary Payment for Permeant employees



Using this window can user can add record about salary payment for permanent employees.

On Calculate Button Click

salary.cheackRate(jcomb\_month, tf\_basic, tf\_bonas, tf\_ot\_hours, tf\_ot\_rate, tf\_ot\_tot, tf\_days, tf\_day\_rate, tf\_day\_tot, tf\_total, tf\_debit, tf\_pay\_tot);

On Pay Button Click

salary.paySalary(jcomb\_month, lb\_nic2, tf\_basic, tf\_bonas, tf\_ot\_hours, tf\_ot\_rate, tf\_ot\_tot, tf\_days, tf\_day\_rate, tf\_day\_tot, tf\_total, tf\_debit, tf\_pay\_tot);

loadTables();

On Load

public void loadTables() {

employee.loadEmpTable(tbl\_emp);

employee.loadEmcTable(tbl\_emc);

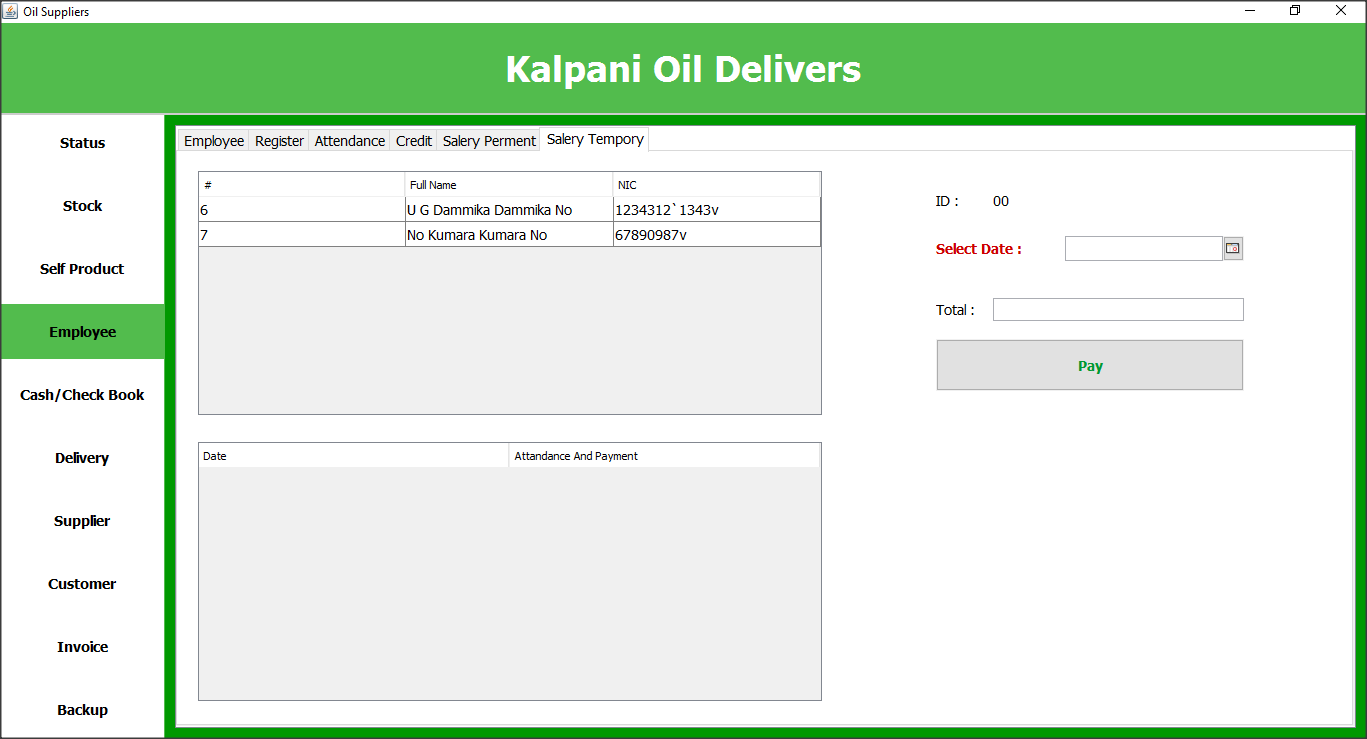
employee.loadSalTable(tbl\_salary);

employee.loadSalTableTemp(tbl\_temp);

attendance.loadAtendanceTable(tbl\_attend, tbl\_absent, label\_today);

}

## Salary Payment for temporary employees



Using this window user can add record about salary payment for temporary employees.

On Pay Button Click

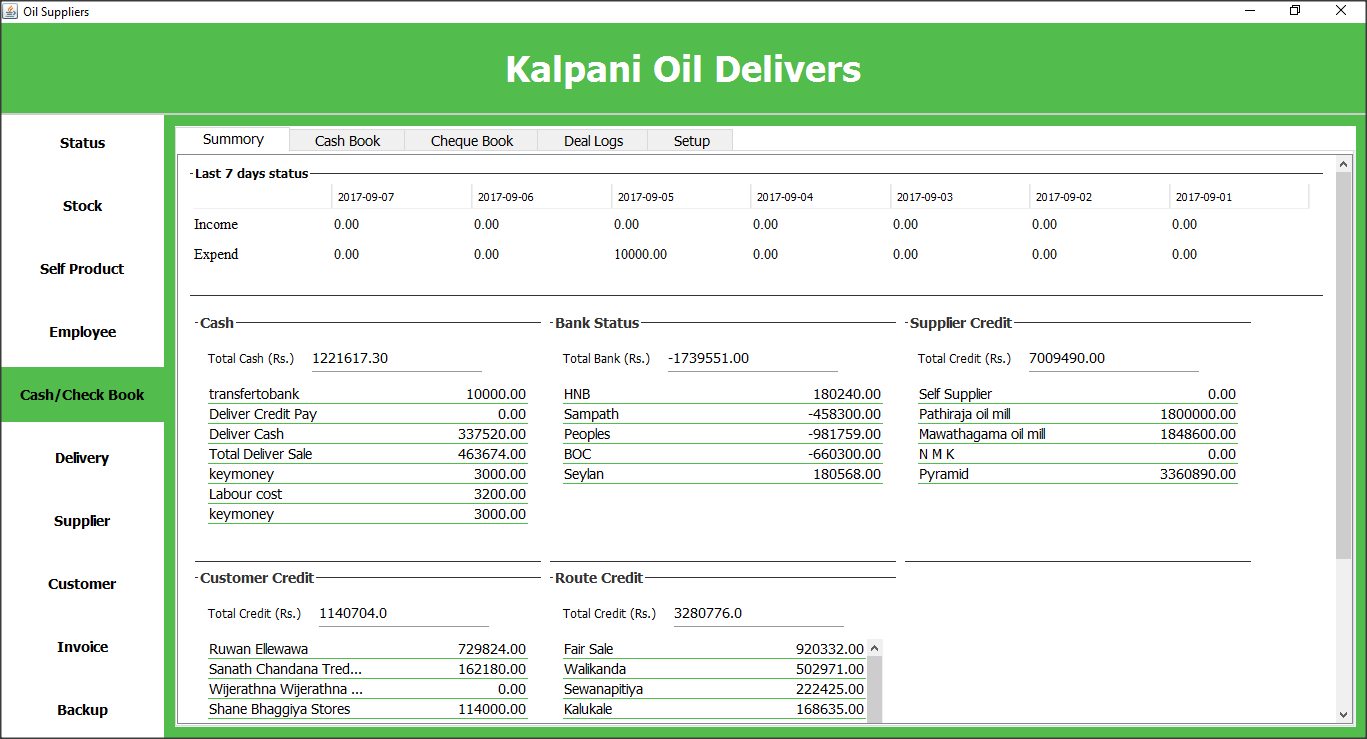
new m.Salary().temPay(lb\_tid, dc\_tmp\_date, tf\_temp\_tot);

loadTables();

On Table Click

employee.loadForTempSalery(tbl\_temp, lb\_tid, tbl\_ap);

## Cash/Bank Summary view



This window will display the summary about business cash status within seven days, bank status, supplier credit status, customer credit status and route credit status.

On Load Interface

loadCashSummory();

loadSummoryOfLast7();

loadSupplierSummory();

loadCustomerSummory();

loadRouteSummory();

loadBankSummory();

loadFinalizedStatus();

On Click Bank Table

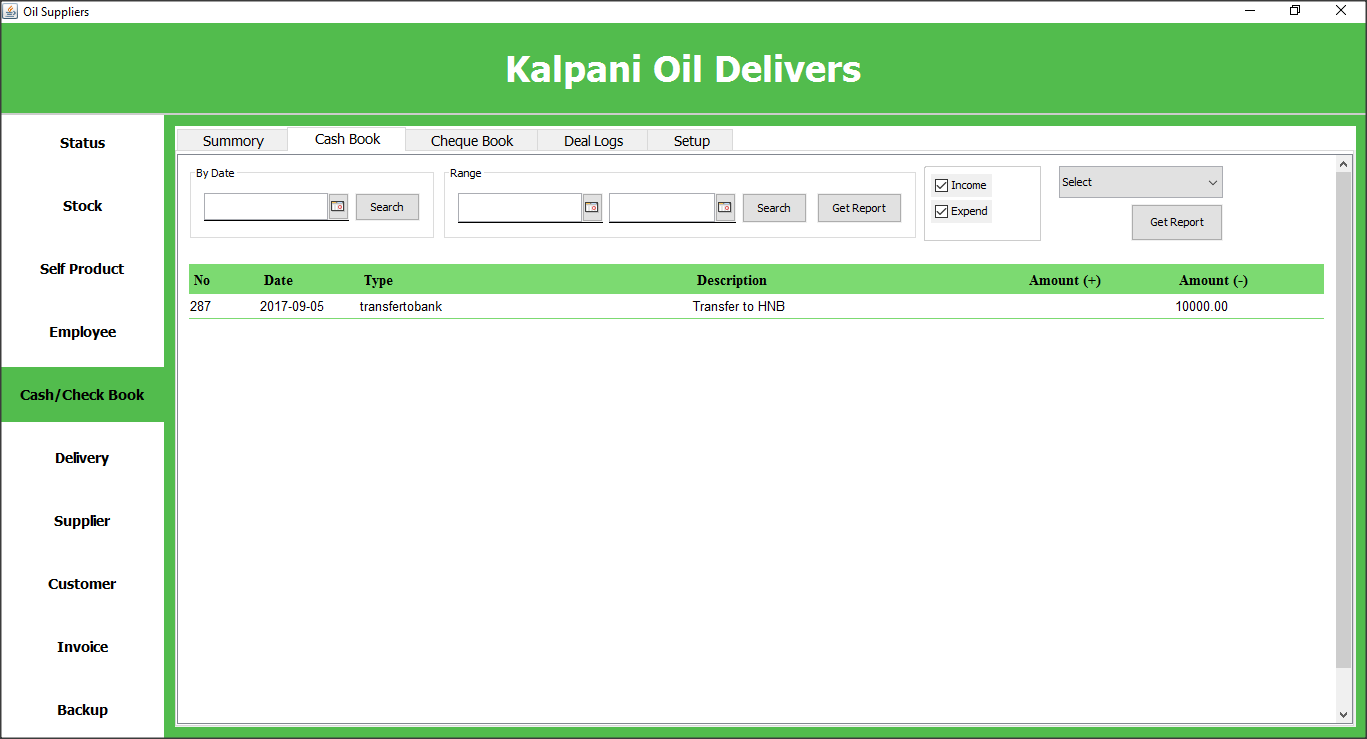
String bankName = this.Cash\_Check\_Book.tbl\_bank\_current\_status.getValueAt(selectedBank, 0).toString();

ShowBankDetails ShowBankDetails = v.ShowBankDetails.getInstance();

ShowBankDetails.bankName = bankName;

ShowBankDetails.setVisible(true);

## Cash Book



This window will display details about cash deals. Also can filter using date or date range.

On Search Button Click

Date dateFrom = this.Cash\_Check\_Book.jdc\_cash\_from.getDate();

Date dateTo = this.Cash\_Check\_Book.jdc\_cash\_to.getDate();

boolean cashincome = this.Cash\_Check\_Book.jcb\_income\_cash.isSelected();

boolean cashexpend = this.Cash\_Check\_Book.jcb\_expend\_cash.isSelected();

this.msg = this.Cash\_Check\_Book.lbl\_message\_cash;

this.tbl = this.Cash\_Check\_Book.tbl\_cash;

if (dateFrom != null) {

if (dateTo != null) {

if (cashincome && cashexpend) {

loadCashDealsByDateRange(dateFrom, dateTo);

} else if (cashincome) {

loadCashIncomeByDateRange(dateFrom, dateTo);

} else if (cashexpend) {

loadCashExpendByDateRange(dateFrom, dateTo);

}

} else {

JOptionPane.showMessageDialog(v.Cash\_Check\_Book.getInstance(), "Please Enter Date", "Warning", JOptionPane.WARNING\_MESSAGE);

}

} else {

JOptionPane.showMessageDialog(v.Cash\_Check\_Book.getInstance(), "Please Enter Date", "Warning", JOptionPane.WARNING\_MESSAGE);

}

On Get Report Button Click

Date df=this.Cash\_Check\_Book.jdc\_cash\_from.getDate();

Date dt=this.Cash\_Check\_Book.jdc\_cash\_to.getDate();

if(df!=null&&dt!=null){

int selectedIndex = this.Cash\_Check\_Book.cmb\_dealtype.getSelectedIndex();

if(selectedIndex>0){

ReportGenerator instance = report.ReportGenerator.getInstance();

HashMap<String, Object> para = new HashMap<String, Object>();

para.put("dealType", selectedIndex);

para.put("dateFrom", df);

para.put("dateTo", dt);

instance.generate("MoneyBookDealType.jrxml", para);

}else{

JOptionPane.showMessageDialog(v.Cash\_Check\_Book.getInstance(), "Select Deal Type", "Warning", JOptionPane.WARNING\_MESSAGE);

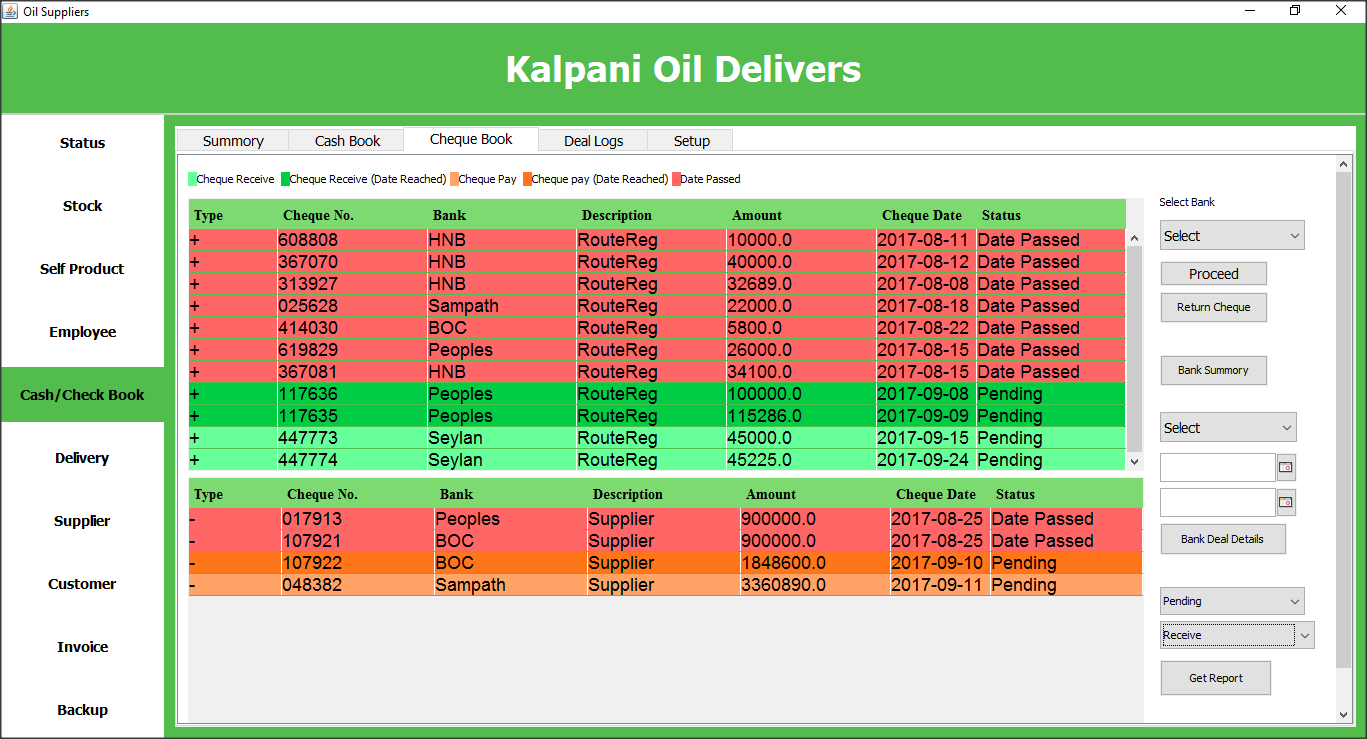
}

}else{

JOptionPane.showMessageDialog(v.Cash\_Check\_Book.getInstance(), "Select Date Range", "Warning", JOptionPane.WARNING\_MESSAGE);

}

## View Cheque Details



Using this window user can view all pending cheques details and can remove or transfer the cheques to relevant banks. Also can get reports about the cheques.

On Proceed Button Click

String[] split = cheque.getDiscription().split("/");

Bank Bank = new m.Bank().getBy(this.Cash\_Check\_Book.cmb\_chequebook\_bank.getSelectedItem().toString());

if (Bank != null) {

int val = 0;

if (split[0].equalsIgnoreCase("routereg")) {

val = new c.RouteDebitMoneyBook().routeCheckProceed(Integer.parseInt(split[1]), chequeNo, cheque.getChequeAmount(), Bank.getId());

} else if (split[0].equalsIgnoreCase("supplier")) {

val = new c.grncashcontroller().supliercheckproceed(Integer.parseInt(split[1]), chequeNo, cheque.getChequeAmount(), Bank.getId());

} else if (split[0].equalsIgnoreCase("cash")) {

val = c.CashChequeBookListner.getInstance().proceedCheque(chequeNo, Bank.getId());

} else if (split[0].equalsIgnoreCase("customer")) {

val = new c.invpayment().customercheckproceed(Integer.parseInt(split[1]), chequeNo, cheque.getChequeAmount(), Bank.getId());

}

if (val == 1) {

JOptionPane.showMessageDialog(v.Cash\_Check\_Book.getInstance(), "Cheque Proceed Success", "Message", JOptionPane.INFORMATION\_MESSAGE);

if (this.Cash\_Check\_Book.tbl\_cheque\_recive.getRowCount() == 1) {

((DefaultTableModel) this.Cash\_Check\_Book.tbl\_cheque\_recive.getModel()).setRowCount(0);

} else {

loadPendingCheques();

}

} else {

JOptionPane.showMessageDialog(v.Cash\_Check\_Book.getInstance(), "Cheque Proceed Fail", "Warning", JOptionPane.WARNING\_MESSAGE);

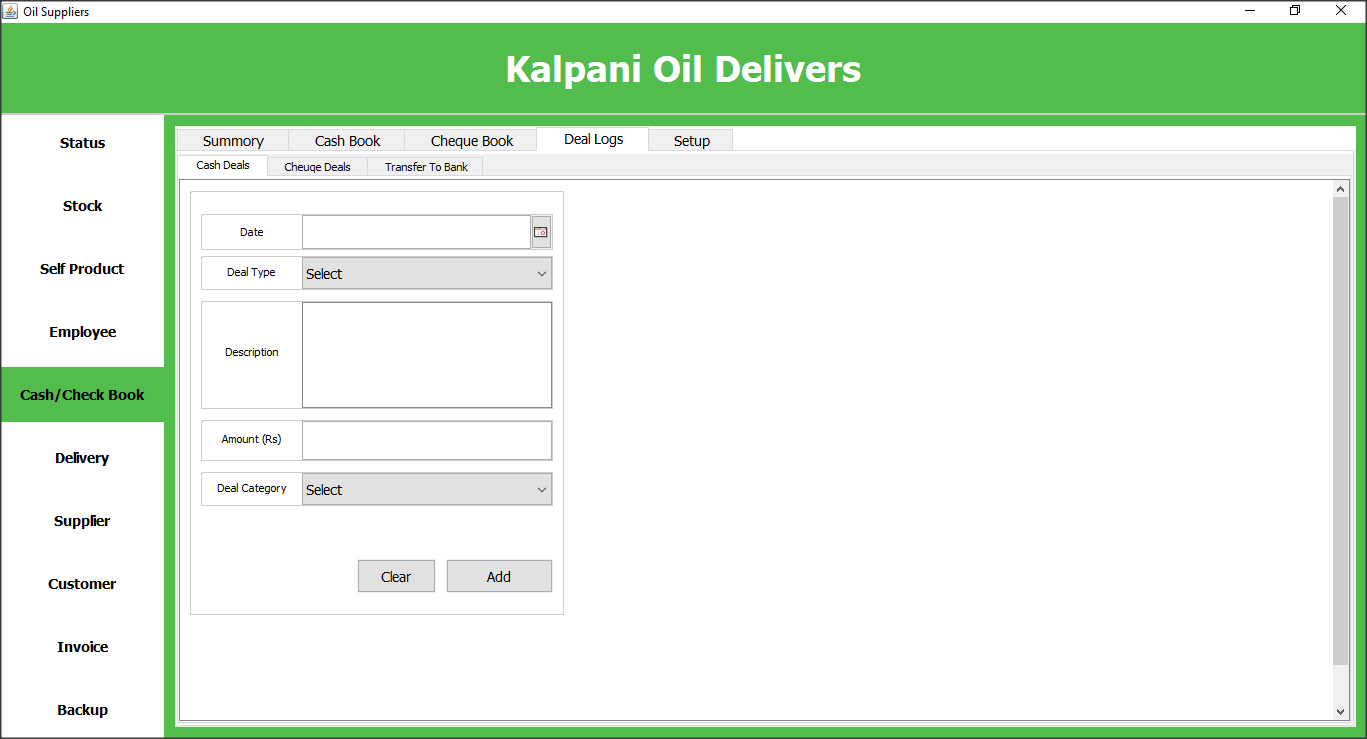
}

} else {

JOptionPane.showMessageDialog(v.Cash\_Check\_Book.getInstance(), "Please select bank", "Warning", JOptionPane.WARNING\_MESSAGE);

}

## Add Cash Deal

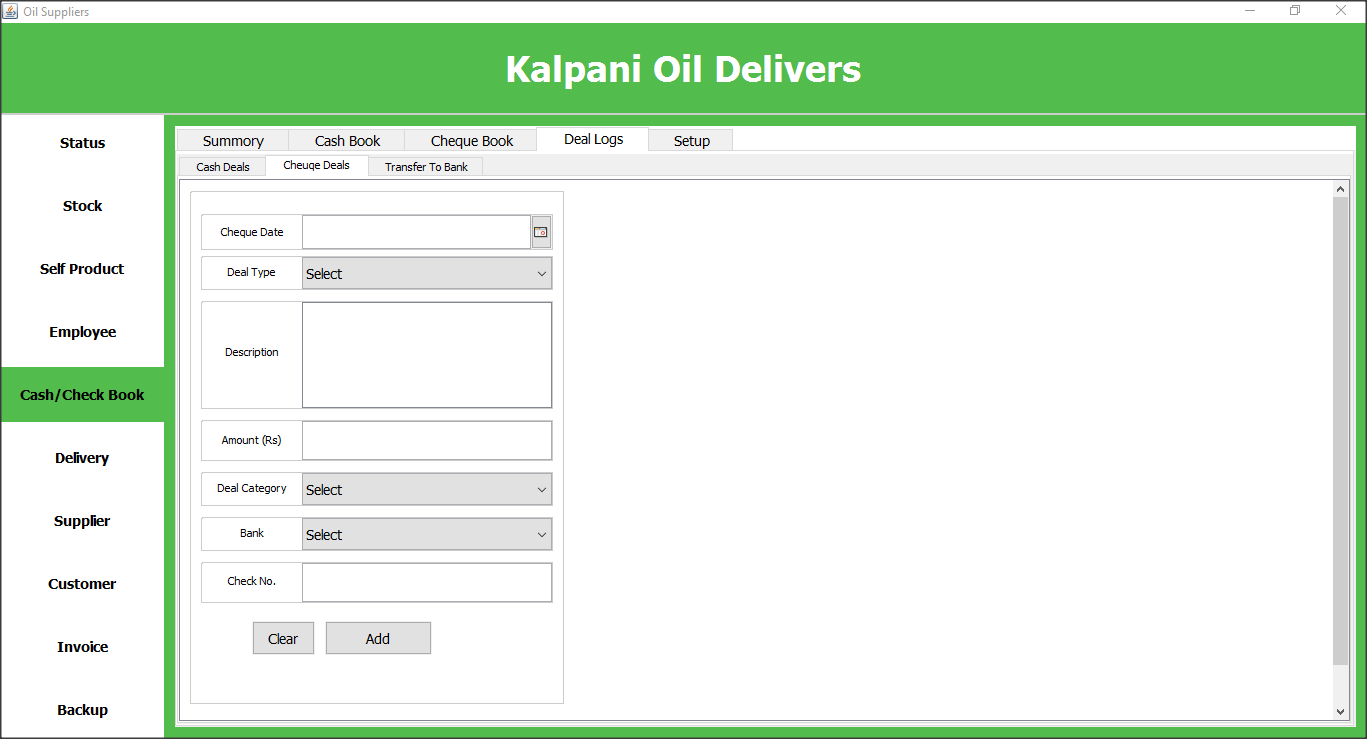


User can add cash deals using this window.

On Add Button Click

addCashDeal();

## Add Cheque Deals

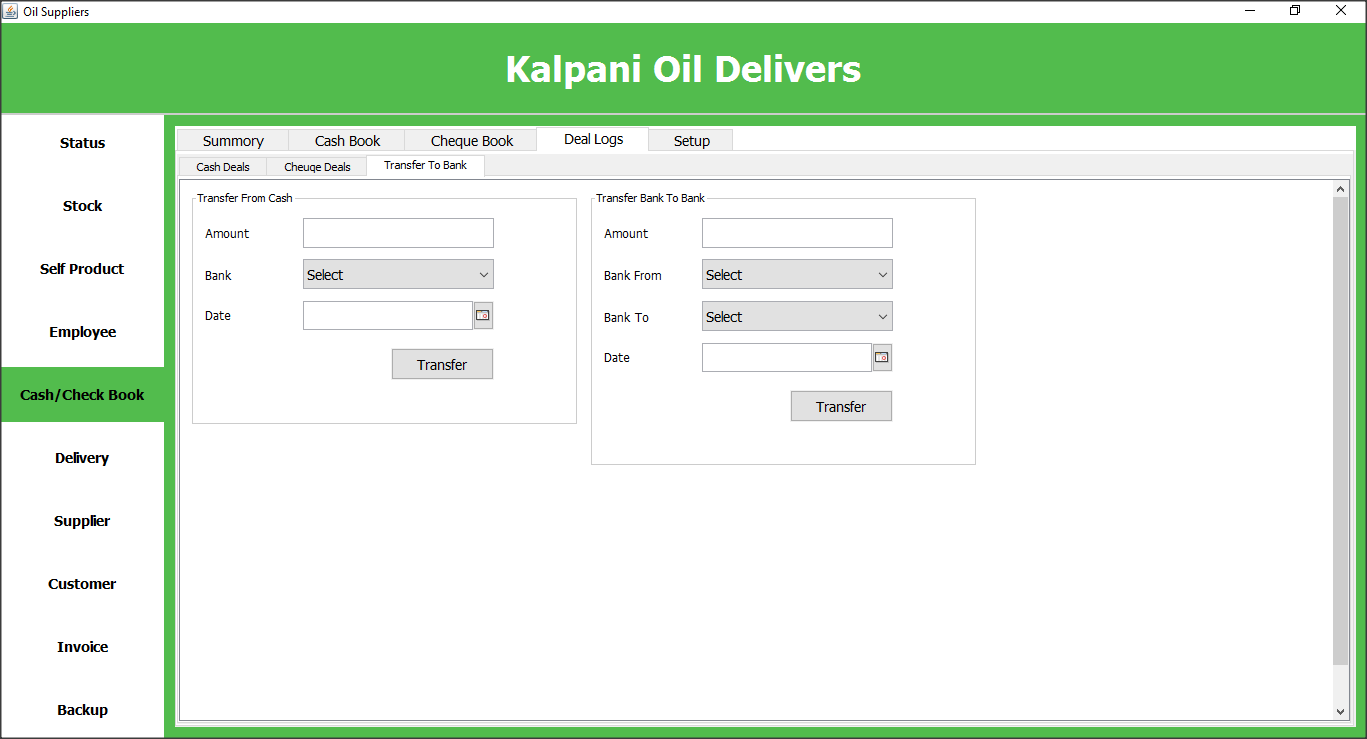


User can add cheque deals using this window.

On Add Button Click

addChequeDeal();

## Transfer money cash to bank/ bank to bank



User can transfer money from bank to bank or bank to cash using this window.

On Cash Transfer Click

try {

if (!this.Cash\_Check\_Book.txt\_trans\_cash\_amount.getText().equals("")) {

if (this.Cash\_Check\_Book.jdc\_trans\_cash\_date.getDate() != null) {

if (!this.Cash\_Check\_Book.cmb\_trans\_cash\_bank.getSelectedItem().toString().equalsIgnoreCase("select")) {

Date date = this.Cash\_Check\_Book.jdc\_trans\_cash\_date.getDate();

double amount = Double.parseDouble(this.Cash\_Check\_Book.txt\_trans\_cash\_amount.getText());

String bank = this.Cash\_Check\_Book.cmb\_trans\_cash\_bank.getSelectedItem().toString();

Bank Bank = new m.Bank().getBy(bank);

if (Bank != null) {

int showConfirmDialog = JOptionPane.showConfirmDialog(this.Cash\_Check\_Book, "Do you want to transfer money to " + Bank.getBank(), "Transfer", JOptionPane.YES\_NO\_OPTION);

if (showConfirmDialog == 0) {

MoneyBook moneyBook = new pojo.MoneyBook();

moneyBook.setDate(date);

moneyBook.setAmount(amount);

moneyBook.setDscription("Transfer to " + Bank.getBank());

moneyBook.setDealCategory(new m.DealCategory().getBy(2));

moneyBook.setDealType(new m.DealType().getBy(27));

String save = new m.MoneyBook().save(moneyBook);

if (save.equalsIgnoreCase("done")) {

c.AssetControl.getInstance().updateCashAsset(amount, '-');

BankDeals bankDeals = new pojo.BankDeals();

bankDeals.setDate(date);

bankDeals.setAmount(amount);

bankDeals.setDescription("Transfered from cash");

bankDeals.setBank(Bank);

String save2 = new m.BankDeals().save(bankDeals);

if (save2.equalsIgnoreCase("done")) {

String updateBankAsset = c.AssetControl.getInstance().updateBankAsset(bank, amount, '+');

if (updateBankAsset.equalsIgnoreCase("done")) {

JOptionPane.showMessageDialog(null, "Tranfer Successful", "Success", JOptionPane.INFORMATION\_MESSAGE);

this.Cash\_Check\_Book.txt\_trans\_cash\_amount.setText("");

this.Cash\_Check\_Book.cmb\_trans\_cash\_bank.setSelectedItem("Select");

} else {

JOptionPane.showMessageDialog(null, "Transfer Updating Failed", "Warning", JOptionPane.WARNING\_MESSAGE);

}

} else {

JOptionPane.showMessageDialog(null, "Transfer Saving Failed", "Warning", JOptionPane.WARNING\_MESSAGE);

}

} else {

JOptionPane.showMessageDialog(null, "Transfer Saving Failed", "Warning", JOptionPane.WARNING\_MESSAGE);

}

}

} else {

JOptionPane.showMessageDialog(null, "Not Found Bank", "Warning", JOptionPane.WARNING\_MESSAGE);

}

} else {

JOptionPane.showMessageDialog(null, "Please Select Bank", "Warning", JOptionPane.WARNING\_MESSAGE);

}

} else {

JOptionPane.showMessageDialog(null, "Please Select Date", "Warning", JOptionPane.WARNING\_MESSAGE);

}

} else {

JOptionPane.showMessageDialog(null, "Please Enter Amount", "Warning", JOptionPane.WARNING\_MESSAGE);

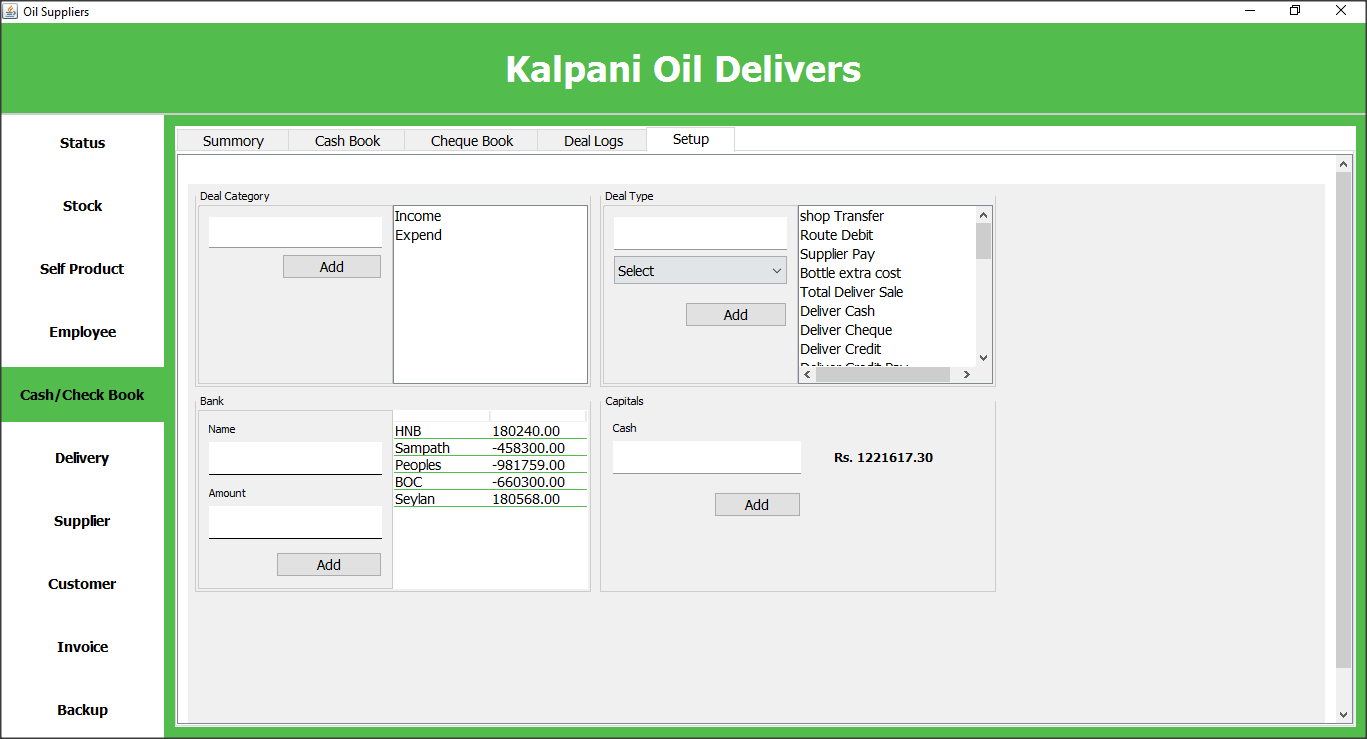
}

} catch (NumberFormatException ex) {

JOptionPane.showMessageDialog(null, "Please Enter Valide Number", "Warning", JOptionPane.WARNING\_MESSAGE);

}

## Config



Using this window user can register new banks, add new deal types, define cash status.

On Deal Category Add Button Click

addToDealCategory(type);

loadDealCategoryToList(this.Cash\_Check\_Book.list\_dealcategory);

loadDealCategoryToCombo(this.Cash\_Check\_Book.cmb\_dealcategory);

On Deal Type Add Button Click

String addToDealType = addToDealType(type, cate);

if (addToDealType.equalsIgnoreCase("done")) {

loadDealTypeToList(this.Cash\_Check\_Book.list\_dealtype);

clearSetup();

}

On Bank Add Button Click

String addBank = addBank(name, amount);

if (addBank.equalsIgnoreCase("done")) {

loadBankToTable(this.Cash\_Check\_Book.tbl\_setup\_bank\_list);

clearSetup();

}

On Cash Add Button Click

try {

if (!this.Cash\_Check\_Book.txt\_capital\_cash.getText().equals("")) {

double amount = Double.parseDouble(this.Cash\_Check\_Book.txt\_capital\_cash.getText());

AssetStatus assetStatus = new m.AssetStatus();

System.out.println(amount);

pojo.AssetStatus Cash = assetStatus.getById(1);

if (amount > 0) {

MoneyBook moneyBook = new pojo.MoneyBook();

moneyBook.setDate(new Date());

if (Cash.getAmount() > amount) {// reduce

moneyBook.setDealType(new m.DealType().getBy(14));

moneyBook.setDealCategory(new m.DealCategory().getBy(2));

moneyBook.setDscription("Reduce money(cash) asset");

moneyBook.setAmount(Cash.getAmount() - amount);

Cash.setAmount(amount);

assetStatus.updateAssetStatus(Cash);

new m.MoneyBook().save(moneyBook);

} else if (Cash.getAmount() < amount) {// increase

moneyBook.setDealType(new m.DealType().getBy(15));

moneyBook.setDealCategory(new m.DealCategory().getBy(1));

moneyBook.setDscription("Increase money(cash) asset");

moneyBook.setAmount(amount - Cash.getAmount());

Cash.setAmount(amount);

assetStatus.updateAssetStatus(Cash);

new m.MoneyBook().save(moneyBook);

} else if (Cash.getAmount() == amount) {

}

loadAssetCash(this.Cash\_Check\_Book.lbl\_cash\_capital);

this.Cash\_Check\_Book.txt\_capital\_cash.setText("");

}

} else {

JOptionPane.showMessageDialog(null, "Enter value", "Warning", JOptionPane.WARNING\_MESSAGE);

}

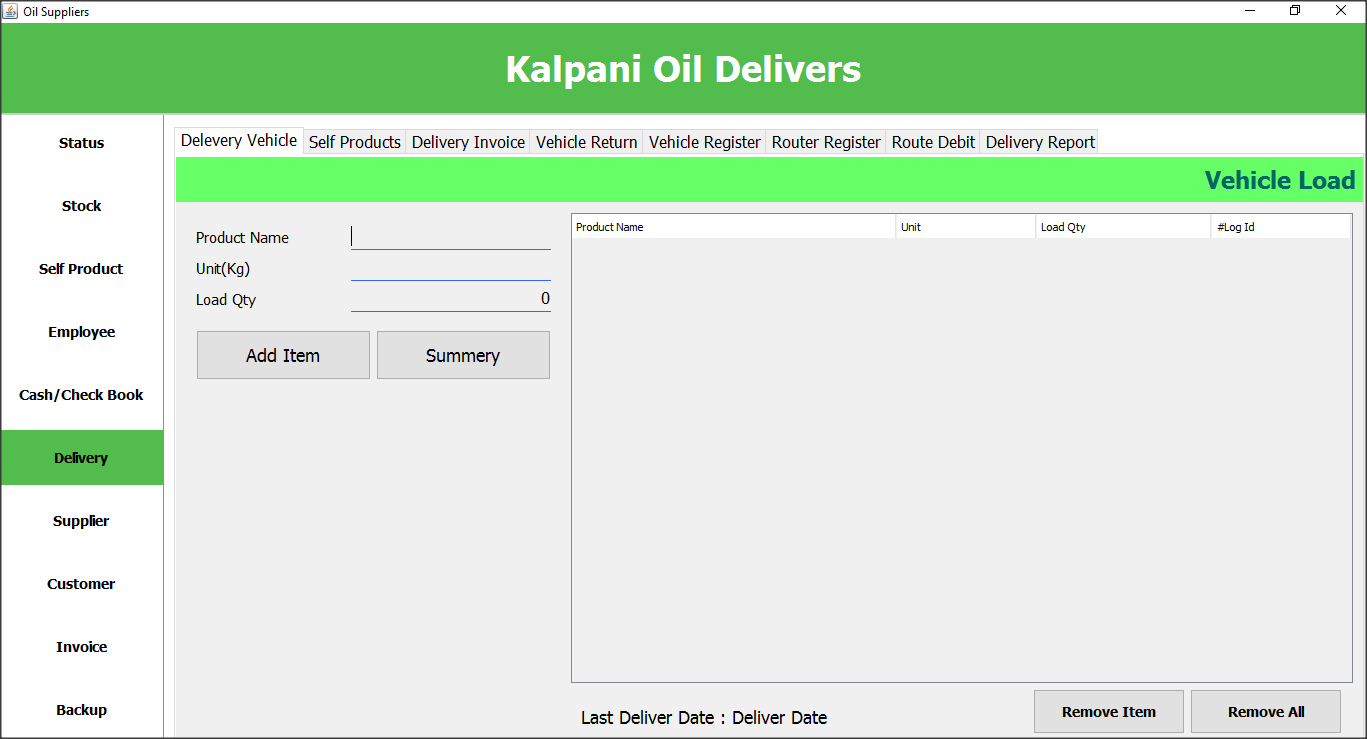
} catch (NumberFormatException ex) {

ex.printStackTrace();

JOptionPane.showMessageDialog(null, "Enter valide number", "Warning", JOptionPane.WARNING\_MESSAGE);

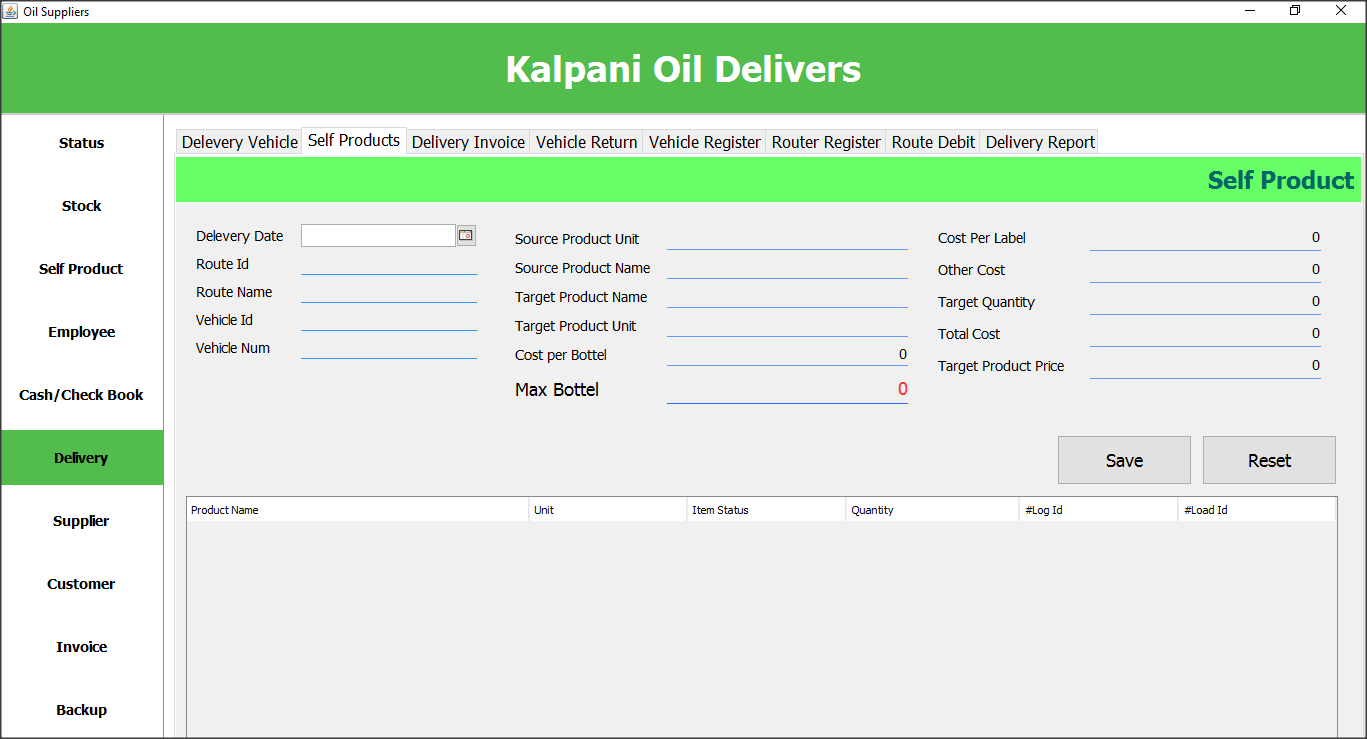
}

## Deliver Load

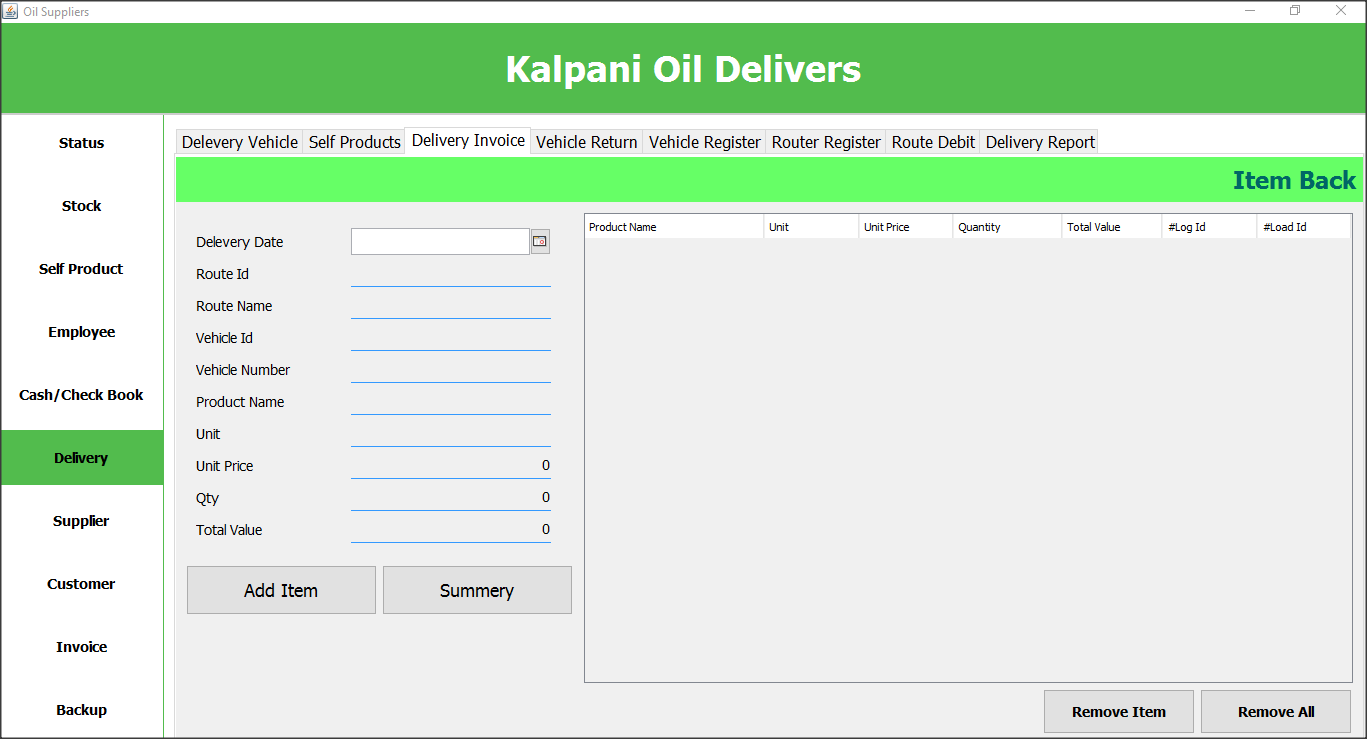


Using this interface user can define deliver load.

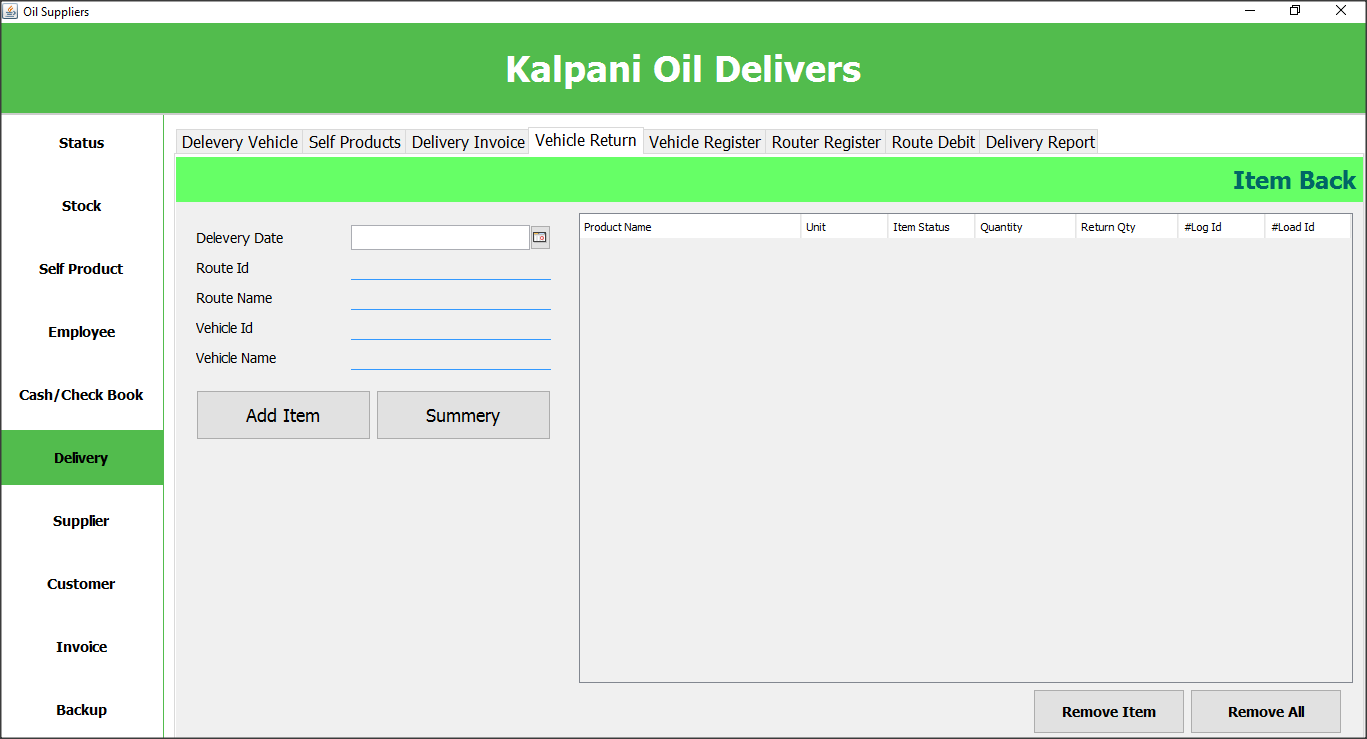
## Delivery Self Product



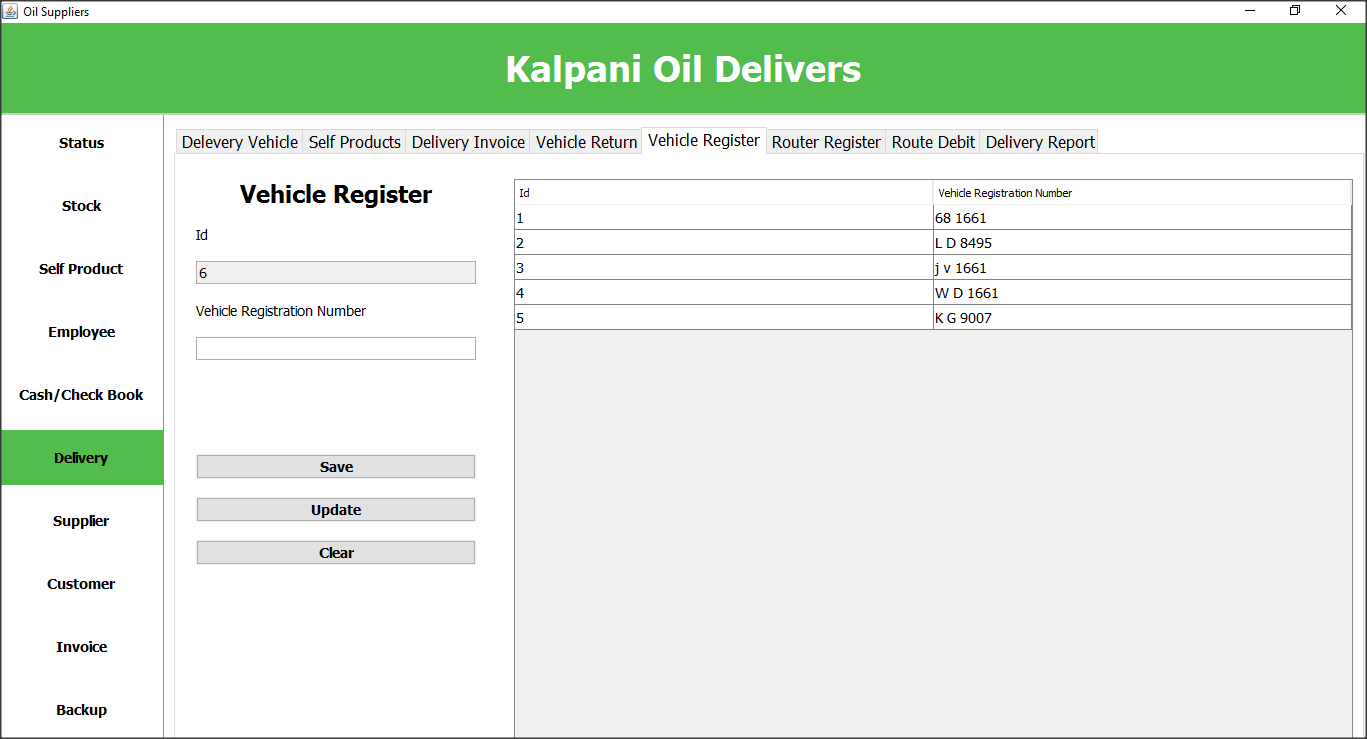
## Delivery Invoice



## Deliver Vehicle Return



## Vehicle Register



User can register their vehicles using this window.

On Save Button Click

VehicleRegister vehicleRegister = new VehicleRegister();

int setData = vehicleRegister.setData(tf\_regNumber1, tf\_id, vehicle);

if (setData == 3) {

int confirm = 0;

confirm = JOptionPane.showConfirmDialog(vehicle, "Do you want to update data", "Data Update", JOptionPane.OK\_CANCEL\_OPTION, JOptionPane.QUESTION\_MESSAGE);

if (confirm == 0) {

btn\_vUpdateMouseReleased(evt);

} else {

System.out.println("Cancel vehicle update");

}

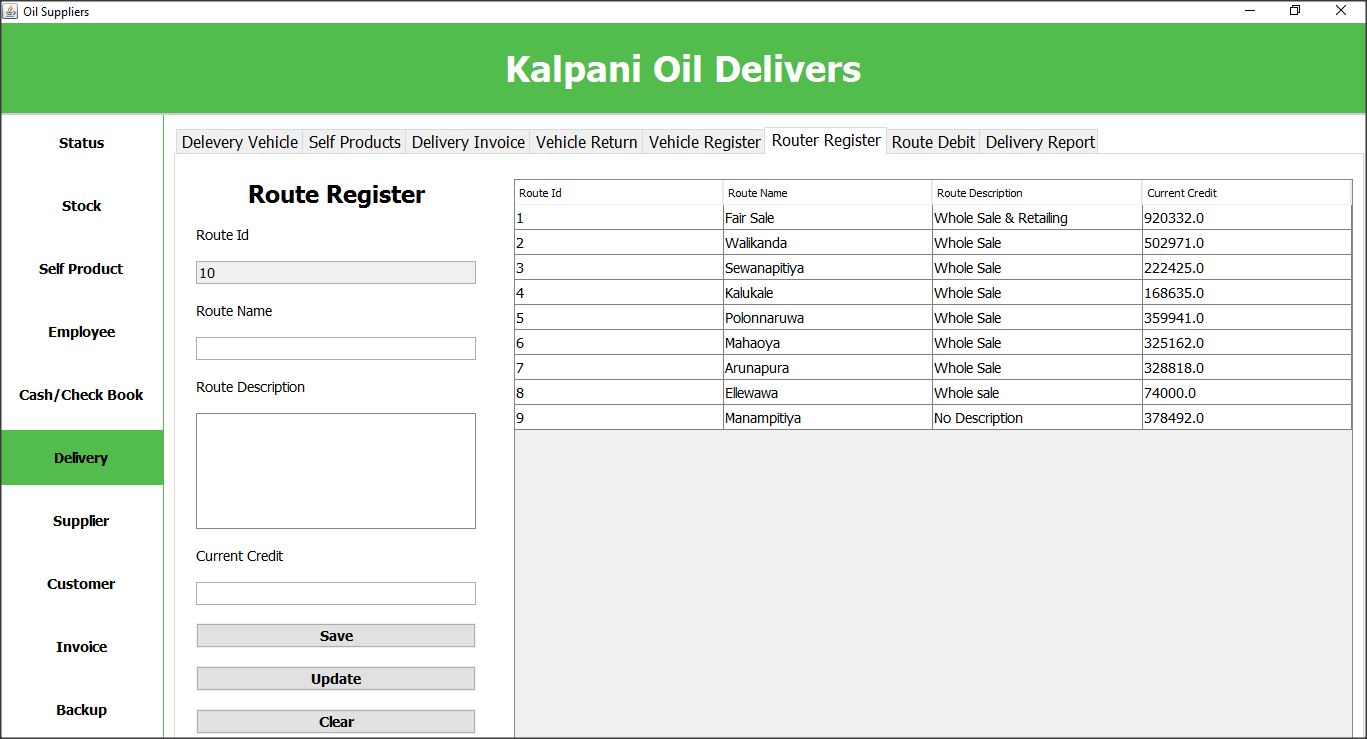
//btn\_vUpdateMouseReleased(evt);

}

vehicleViewTable.setTableData(tbl\_allVehicles);

clear();

## Route Register



User can register route details using this window

On Save Button Click

int saveRoute = new RouteRegister().saveRoute(tf\_routeId, tf\_routeName, ta\_routeDescription, tf\_currentCredit, vehicle);

if (saveRoute == 3) {

int showConfirmDialog = JOptionPane.showConfirmDialog(vehicle, "Do you want to update", "title", JOptionPane.OK\_CANCEL\_OPTION, JOptionPane.QUESTION\_MESSAGE);

if (showConfirmDialog == 0) {

btn\_routeUpdateMouseReleased(evt);

} else {

System.out.println("Cancel");

}

}

routeViewTable.setrouteTableData(tbl\_routeDetails);

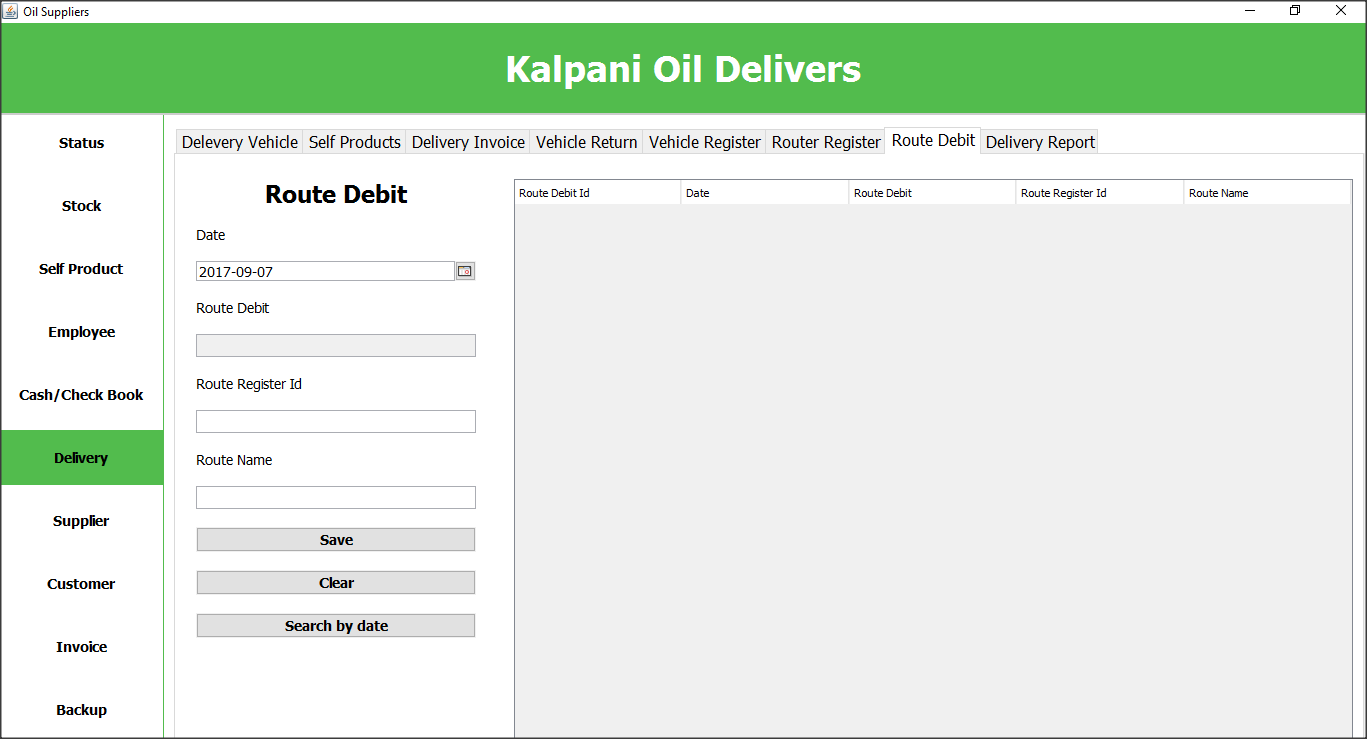
clearRoute();

On Update Button Click

new RouteUpdate().updateRoute(tf\_routeId, tf\_routeName, ta\_routeDescription, tf\_currentCredit);

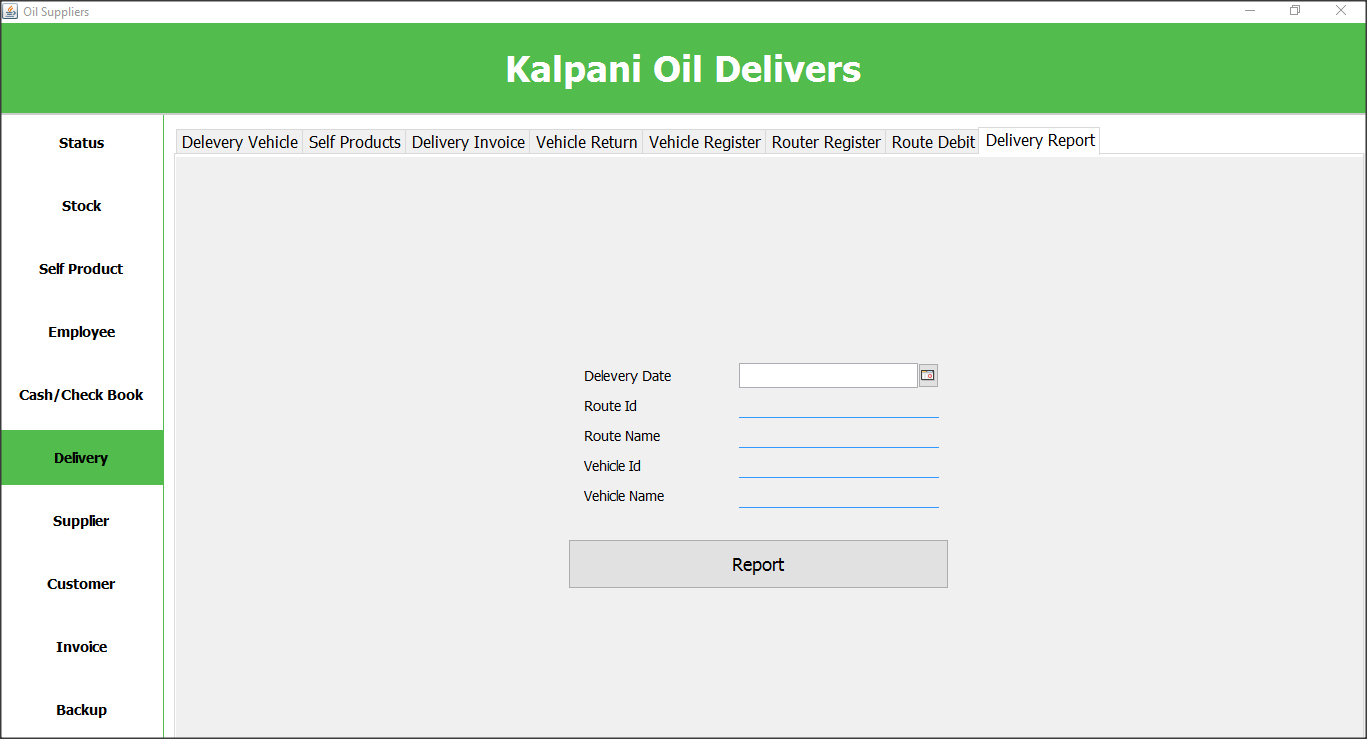
routeViewTable.setrouteTableData(tbl\_routeDetails);

## Route Debit



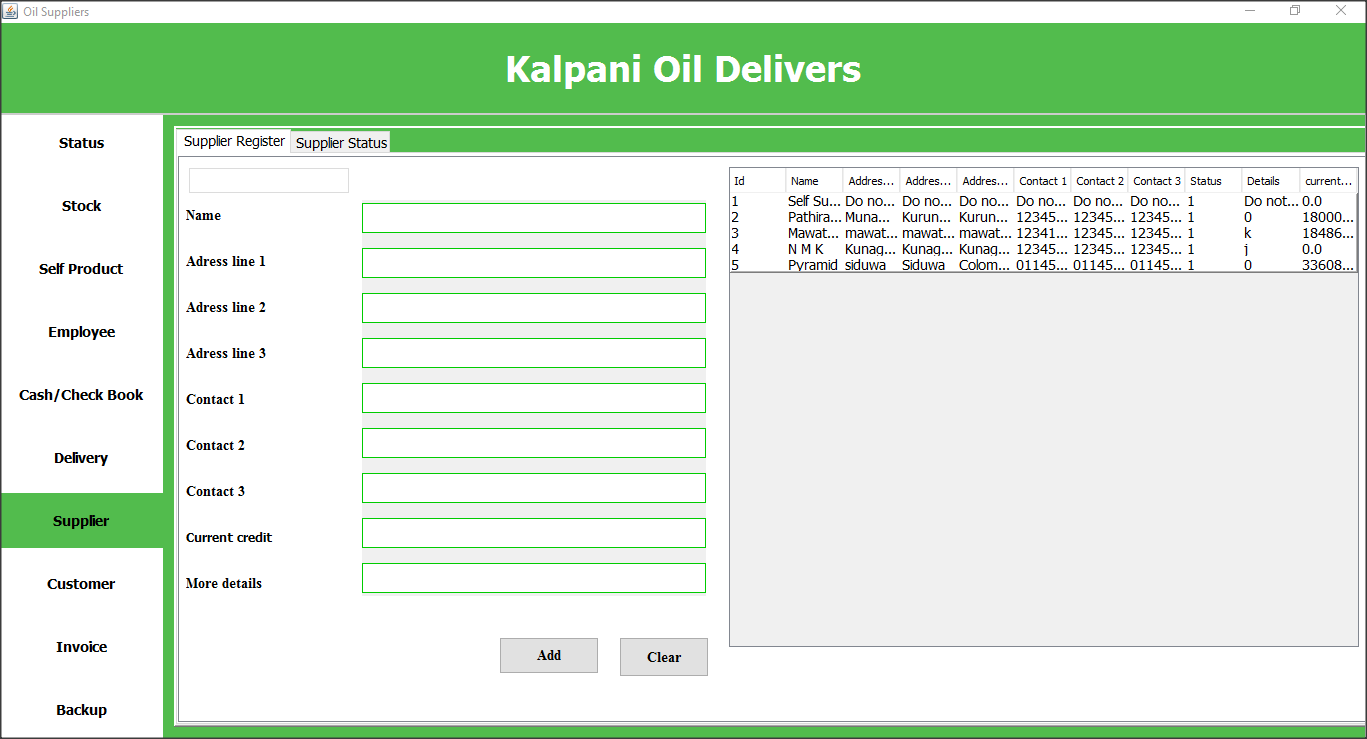
User can add records when route credit payed using this window.

## Deliver Report



User can get reports about delivers using this window.

## Supplier Register



User can register their supplier using this window.

On Load Window

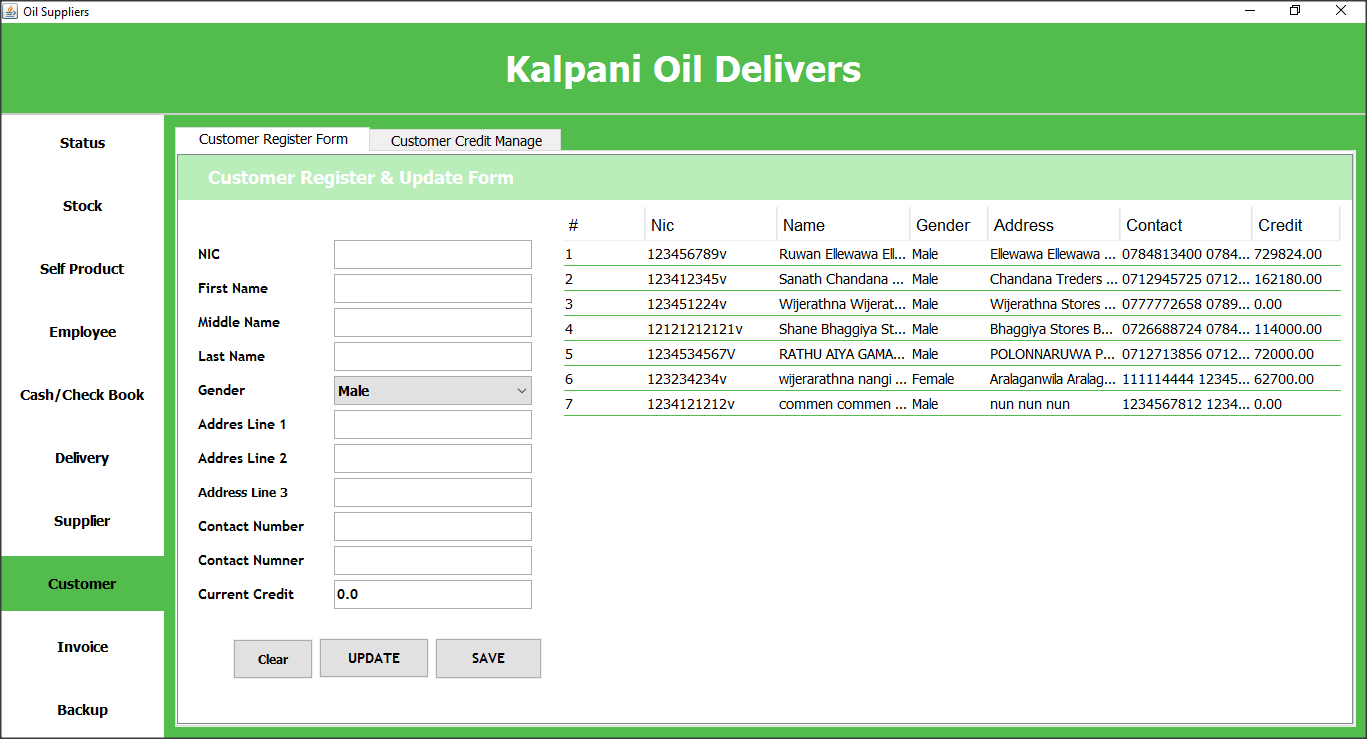
tableView = new SupplierTableView();

tableView.setTableData(jTable1);

On Add Button Click

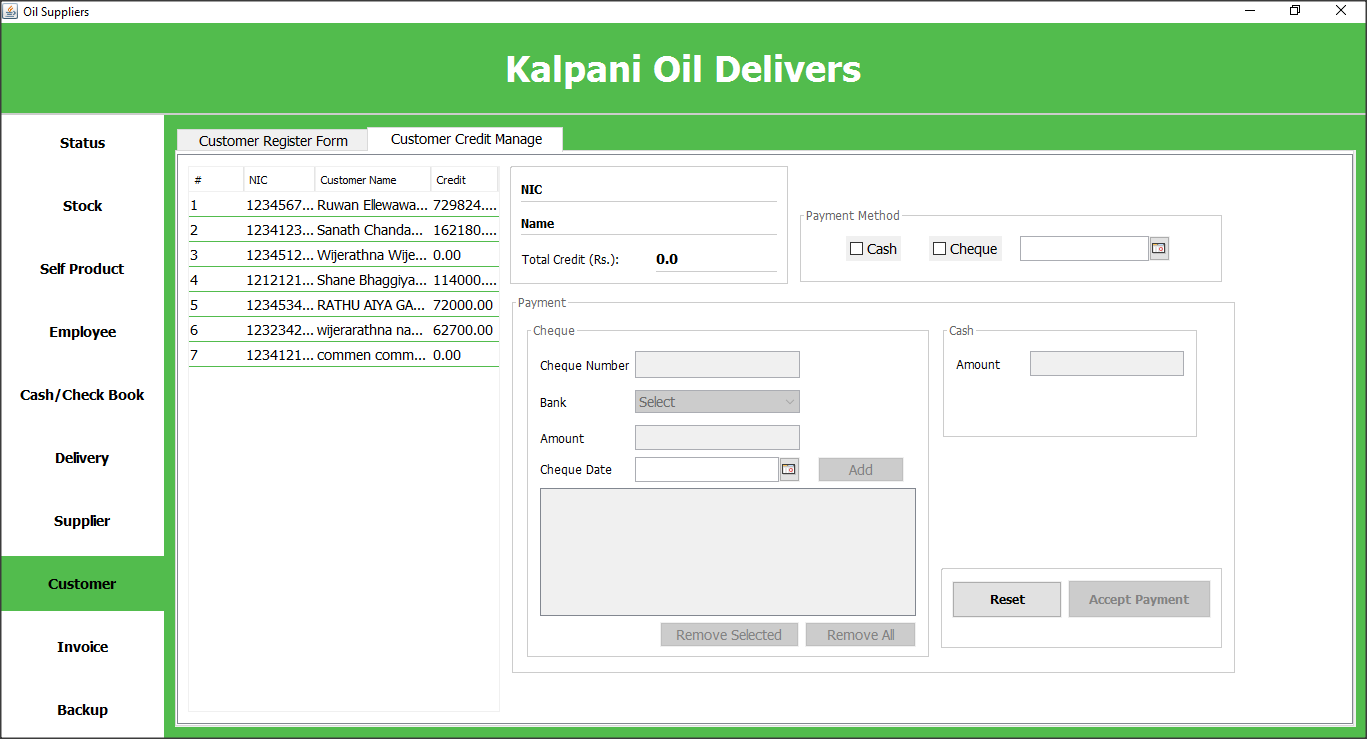
savesupp();

## Customer Register and View



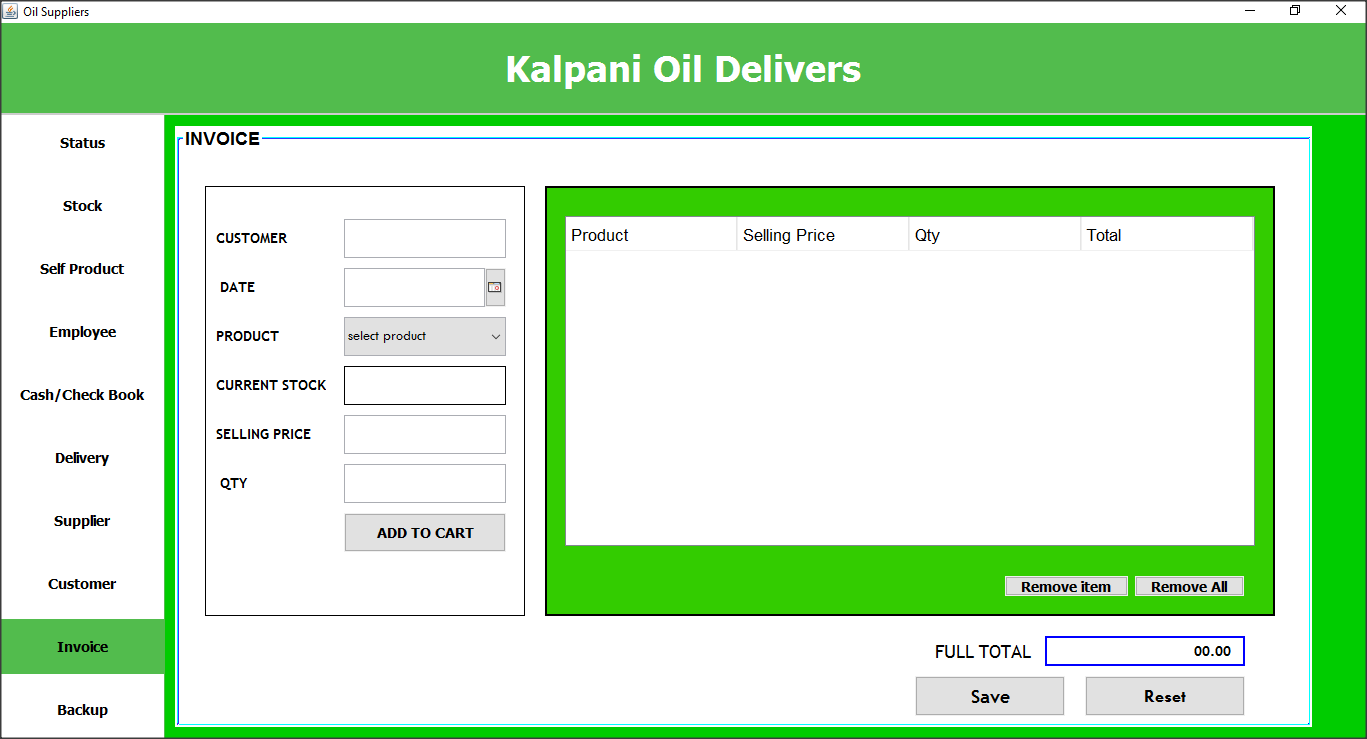
User can register their customers using this window Also can view registered customer.

## Customer Credit Manager



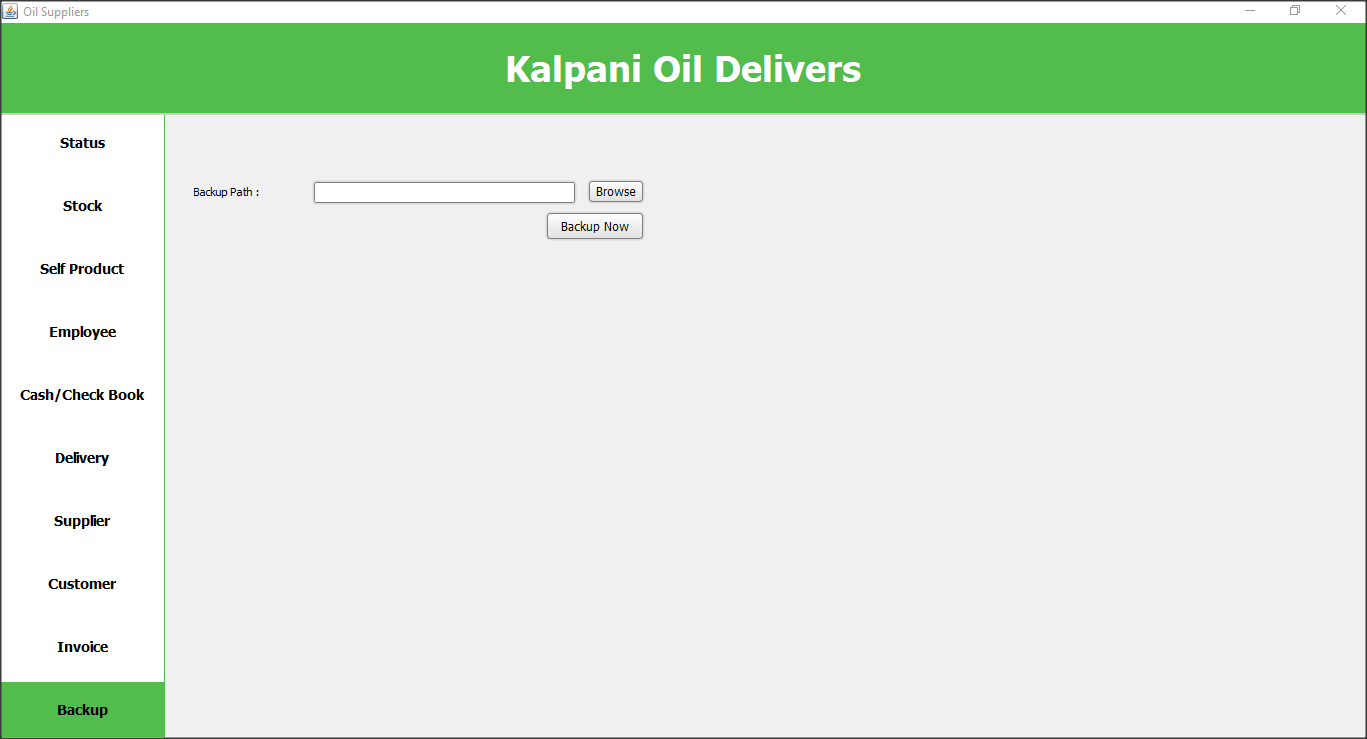
User can manage customer’s credit using this window. When load the window all customers will load to table. When selecting customers from the table all credit detail will load to panel.

## Invoice



User can generate invoice using this window.

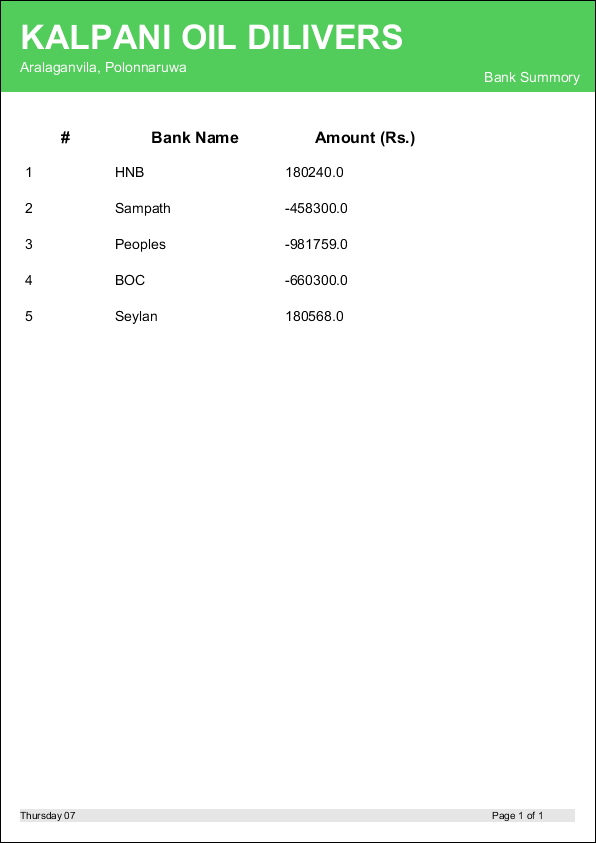
## Backup



User can backup database using this window. Backup destination can select for user. After select destination then press Backup Now button. Then it will backup the database.

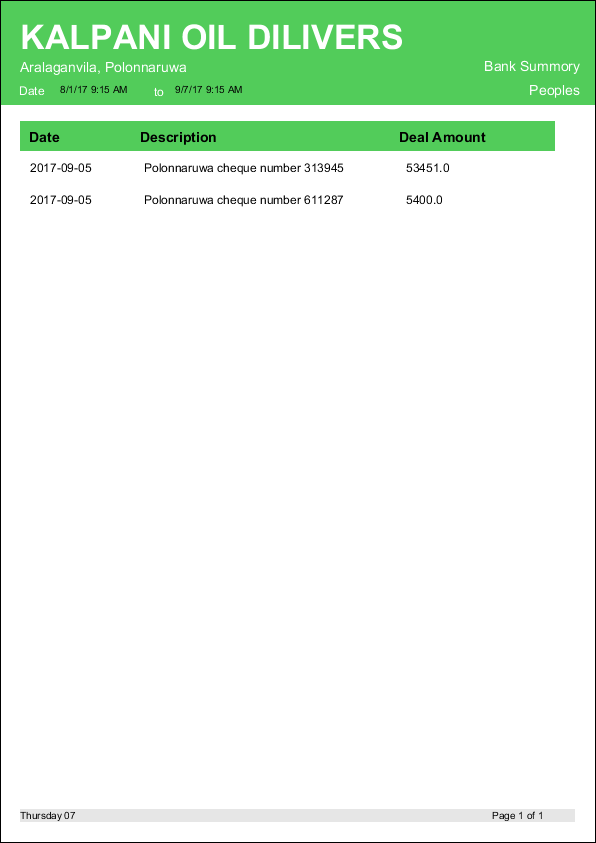
# Reports

## All Bank’s Summary



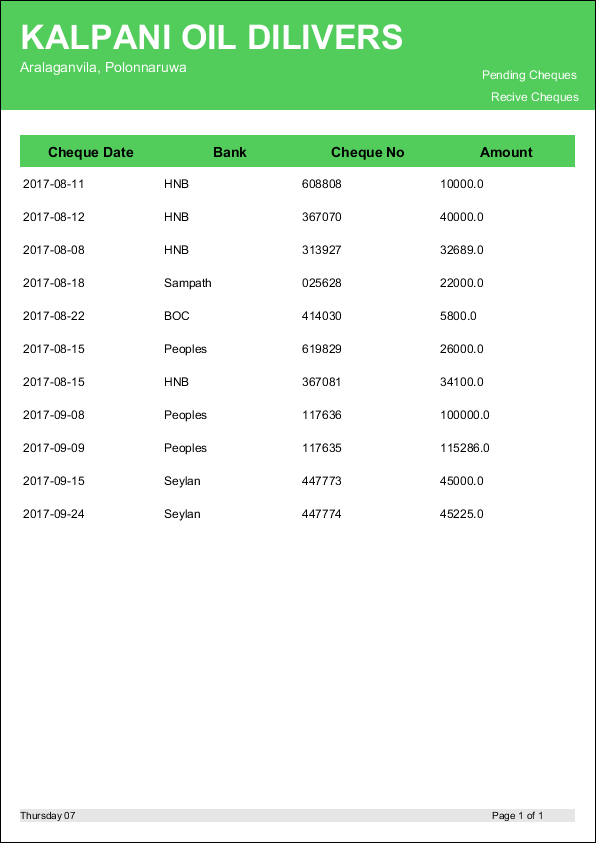
This report is about bank. This will get details about banks that how much they hold in that time.

## Bank Summary

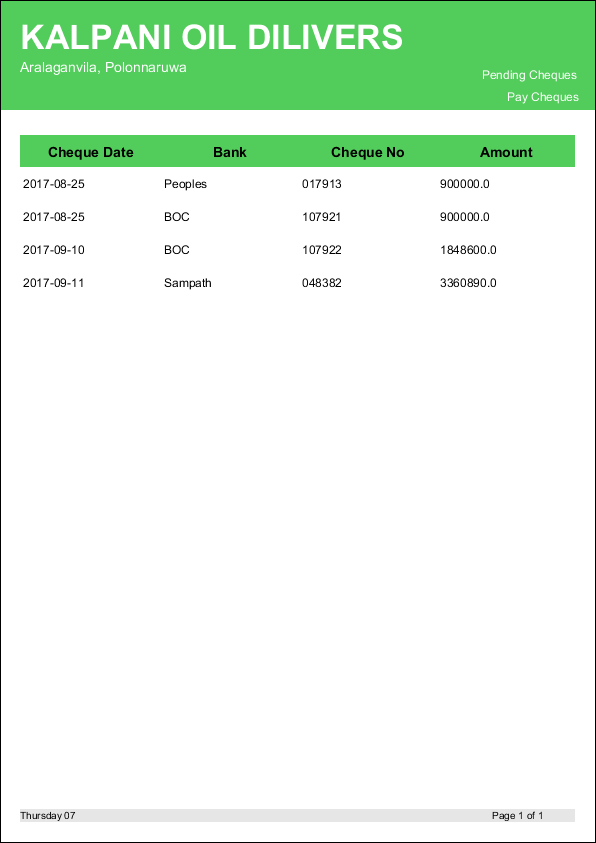


This report will show deal details of given bank.

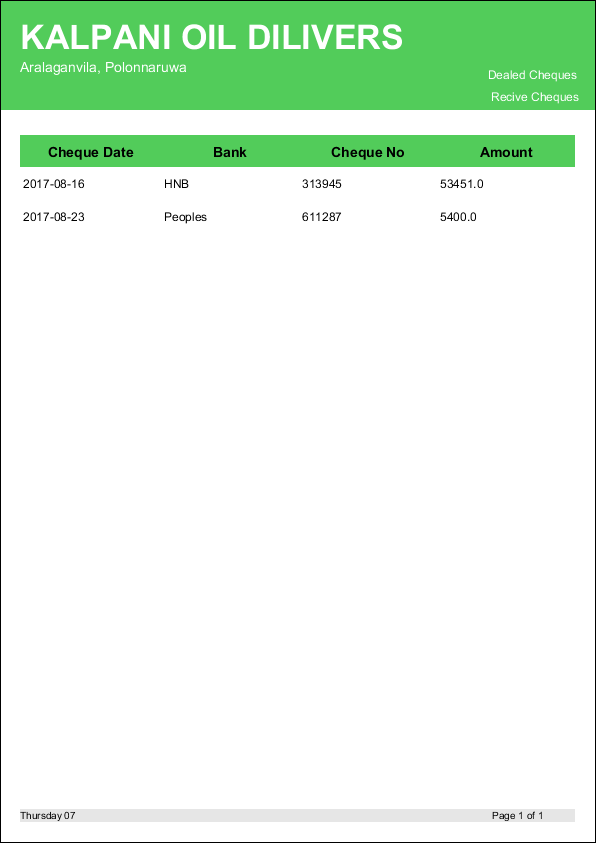
## Pending Cheques (Recive)



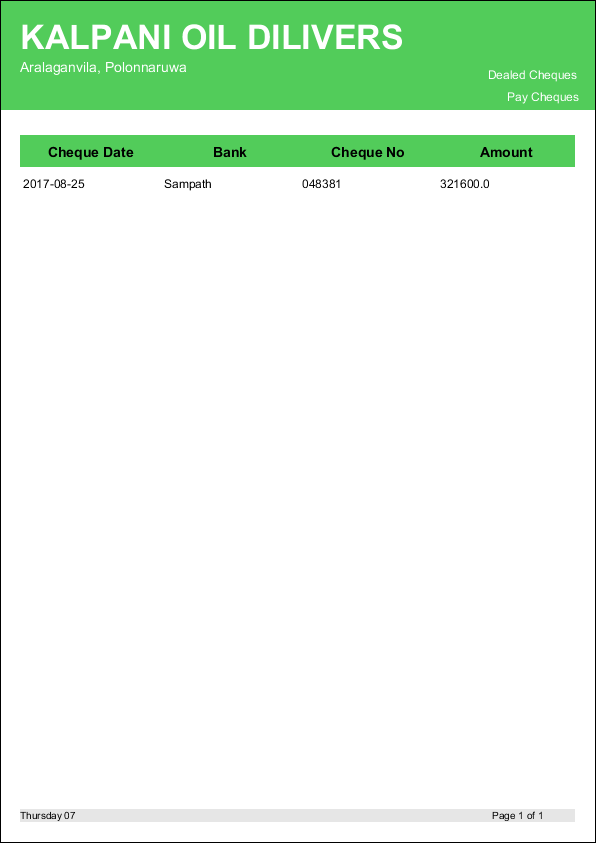
## Pending Cheques (pay)



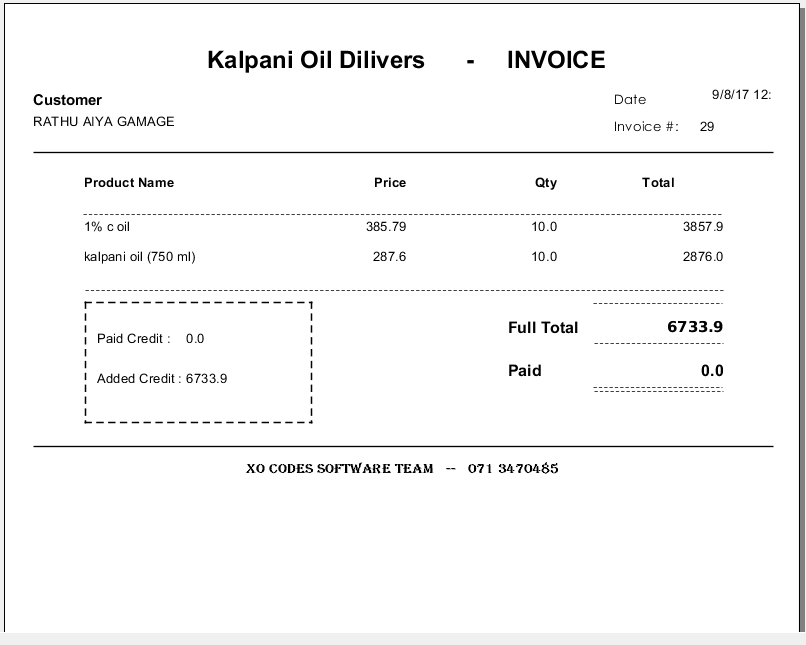
## Payed Cheque (Recive)



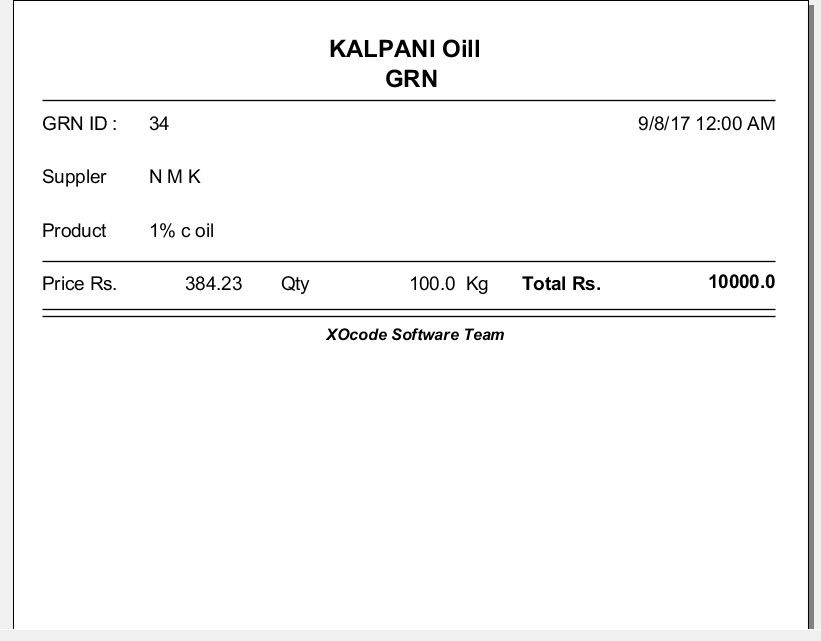
## Payed Cheque (Pay)



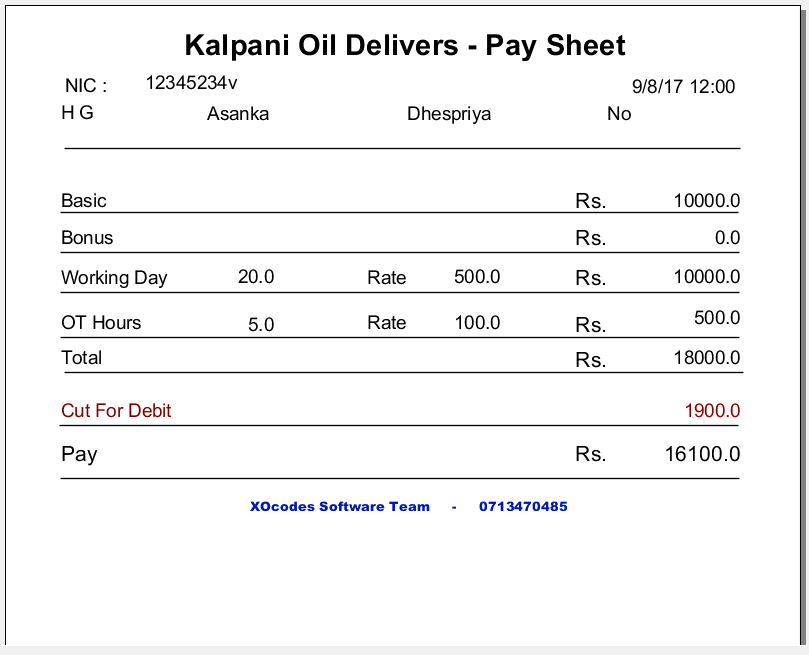
## Invoice



## GRN



## Salary



## Delivery

