Project ID2018\_ 06– Milco Outlet Handling System,Ragama.



Sri Lanka Institute of Information Technology

**Milco Outlet Handling System**

**Project Report**

### Information Systems Project - 2018

### Project ID: 06

Submitted by:

|  |  |
| --- | --- |
| Registration  Number | Student Name |
| IT17533942 | Jayana Gayathri D.M |
| IT17103978 | Liyanaarachchi L.A.A.S |
| IT17187060 | Perera M.N.H |
| IT17120494 | Wijesuriya.W.A.C.C |

Submitted to:

…………………………..

Ms.Chathurika Kahandawaarachchi

02.10.2018

Project ID2018\_ 06 -Milco Outlet Handling System

# Abstract

This project is proposing for a MILCO SALES OUTLET company which situated in RAGAMA. According to the company process they buying a different type of milco products from milco suppliers and selling them to outside buyers.

As the client’s requirement, they have two types of suppliers and they need a good database to handle their supplier’s details. And they need a good stock maintenance system too.as well as the company has a large number of working employees. So it has begun a huge problem in handling their day process. So that we proposing a project to implement a system which helps to solve those problems and a system that easily can handle in company day to day work process.

Our client name is Mrs. N.A.P Nirmala De Silva. She is the Agent of this Milco Outlet which situated in Ragama. She is a good experienced Business women and she hoping to spread her Milco outlets in Kadawathe, Colombo areas also.

Project ID2018\_06 – Milco Outlet Handling System,Ragama.

# Acknowledgement

# This project consumed huge amount of work, research and dedication. Still, implementation

# Would not have been possible if we did not have a support of many individuals and organizations.

# Therefore we would like to extend our sincere gratitude to all of them.

First of all we would like to give our sincere gratitude and appreciation to our project supervisor as well as our ISP lecturer in charge Ms. Chathurangika Kahandawaarachchi , whose immense help which guided us to improve our project in many ways, and provided wealthy information and for stimulating suggestions and encouragement, helped us to coordinate our project.

We would like to appreciate the guidance given by our senior batch mates as well as the panel members those who were in our project proposal presentation for their valuable feedback to improve on.

Lastly our very special thanks goes to the group members and their family, who always stood by us to complete our project successfully. Although they are unable to name the specific individuals, all their help is warmly appreciated and we would like to thank and show our gratitude to everyone who helped them in many ways to complete this project successfully.

Project ID2018\_06– Milco Outlet Handling System,Ragama.

# Declaration

We declare that the this project report or part of it was not a copy of a document done by any organization, university any other institute or a previous student project group at SLIIT and was not copied from the Internet or other sources.

Project Details

|  |  |
| --- | --- |
| Project Title | Milco Outlet Handling System,Ragama. |
| Project ID | 06 |

Group Members

|  |  |  |
| --- | --- | --- |
| **Reg. No** | **Name** | **Signature** |
| IT17533942 | Jayana Gayathri D.M |  |
| IT17103978 | Liyanaarachi L.A.A.S |  |
| IT17187060 | Perera M.N.H |  |
| IT17120494 | Wijesuriya W.A.C.C |  |
|  |  |  |

Project ID2018\_06– Milco Outlet Handling System,Ragama.

# Table of Contents

[Abstract ii](#_TOC_250022)

[Acknowledgement iiii](#_TOC_250021)

[Declaration iv](#_TOC_250020)

[Table of Contents v](#_TOC_250019)

[List of Figures vii](#_TOC_250018)

[List of Tables vii](#_TOC_250017)

[List of Acronyms and Abbreviations vii](#_TOC_250016)

1. [Introduction 1](#_TOC_250015)
   1. [Problem Statement 1](#_TOC_250014)
   2. [Product Scope 1](#_TOC_250013)
   3. [Project Report Structure 4](#_TOC_250012)
2. [Methodology 5](#_TOC_250011)
   1. [Requirements and Analysis 5](#_TOC_250010)
   2. [Design 10](#_TOC_250009)
   3. [Implementation 112](#_TOC_250008)
   4. [Testing 233](#_TOC_250007)
3. [Evaluation 25](#_TOC_250006)
   1. [Assessment of the Project results 25](#_TOC_250005)
   2. [Lessons Learned 26](#_TOC_250004)
   3. [Future Work 27](#_TOC_250003)
4. [Conclusion 28](#_TOC_250002)

[References 29](#_TOC_250001)

Appendix A: Design Diagrams 30

[Appendix C: Selected Code Listings 36](#_TOC_250000)

ProjectID2018\_06– Milco Outlet Handling System,Ragama.

# List of Figures

### Figure 2.1 Use Case Diagram

### Figure 2.2 Student Registration Activity Diagram

### Figure 2.3 Manage University Application Activity Diagram Figure 2.4 Manage Payment Details Activity Diagram Figure 2.5 Manage Visa Process Details Activity Diagram Figure 2.6 Class Diagram

### Figure 2.7 ER Diagram Figure A.1 Context Diagram

### Figure A.2 Data Flow Diagram Level 1

### Figure A.3 Data Flow Diagram Level 2-Student Registration

### Figure A.4 Data Flow Diagram Level 2-University Application Management Figure A.5 Data Flow Diagram Level 2-Payment Handling

### Figure A.6 Data Flow Diagram Level 2-Manage Visa Process Details

# List of Tables

Table 1-1

Table 1-2.1

Table 1-2.2

Table 1-2.3

Table 1-2.4

Table 1-2.5

Table 1-4 Test Cases

# List of Acronyms and Abbreviations

**Table 1-2**

|  |  |
| --- | --- |
| **Term** | **Description** |
| ER Diagram | Entity Relationship Diagram |
| UML Diagram | Unified Modeling Language Diagram |
| GUI | Graphical User Interface |
| IEEE | Institute of Electrical and Electronic Engineers |

# Introduction

## Problem Statement

Milco Outlet Handling System is designed and newly introduced for Milco Outlet, which situated in Ragama, as an automated system with the facilities to manage their All information systems effectively with an easy access to the system in order to overcome problems that have occurred due to the current manual system of handling large scale of physical file system.

This system gives the capability for user to insert, update, delete and search their all information about customers, employees, stocks and suppliers related details with well-organized user friendly functions in attractive user interfaces. The system keeps track on the updated documents, check them whether they are correct and up to date. Hence this automated system gives security to all the company related information that stored in this system throughout its life time.

## Product Scope

This software system will be an automated system for Milco Outlet Handling System which distributes milk items. Mainly the company focuses on selling, distributing and marketing milk products. So this system will help the client to do his work and handle employee details, stock details, and daily selling without any uncomfortable than manual data handle.

According to the client requirements, we decided to implement an automated standalone application which provides the tools that simplify task of managing data of the company and simplify day today work task in a good manner. The employee, customer and supplier registrations are to keep track on personal details of each employee, customer and supplier. The system will be help to calculate salary and print salary sheets as well as generate invoices too. As well increasing security by managing different user levels also done in here.

**Product Functions**

#### Employee Handling

#### Stock Handling

#### Sales Handling

#### Supplier handling

Table 1-2.1

|  |  |
| --- | --- |
| Function 1 | **Employee Handling** |
| Input | Employee details |
| Output | Employee details ,salary reports |
| Processing | Adding employee details.  Calculate salary.  Generate salary reports.  All validations. |

Table 1-2.2

|  |  |
| --- | --- |
| Function 2 | **Stock handling** |
| Input | Product details |
| Output | Product details , product reports |
| Processing | Adding product details.  Generate product reports.  All validations. |

Table 1-2.3

|  |  |
| --- | --- |
| Function 3 | **Sales handling** |
| Input | Customer details and sales details. |
| Output | Monthly sales report.  Customer details. |
| Processing | Add customer details.  Add sales details.  Calculate monthly sales.  Generate sales reports.  All validations. |

Table 1-2.4

|  |  |
| --- | --- |
| Function 4 | **Supplier handling** |
| Input | Supplier details. |
| Output | Supplier details report |
| Processing | Adding supplier details.  Generate the report.  All validations. |

## Project Report Structure

After the Introduction part of this document you will get to know more about the methodologies which we have implemented. The methodology part will comprise mainly of Requirement and Analysis containing detailed requirements and specifications of the project, Design, Implementation and Testing.

Next would come the Evaluation part where the Assessment of the Project results, Lessons we learned by doing this project and the future work we intend to do is described.

Finally to sum it all up, the conclusion is written.

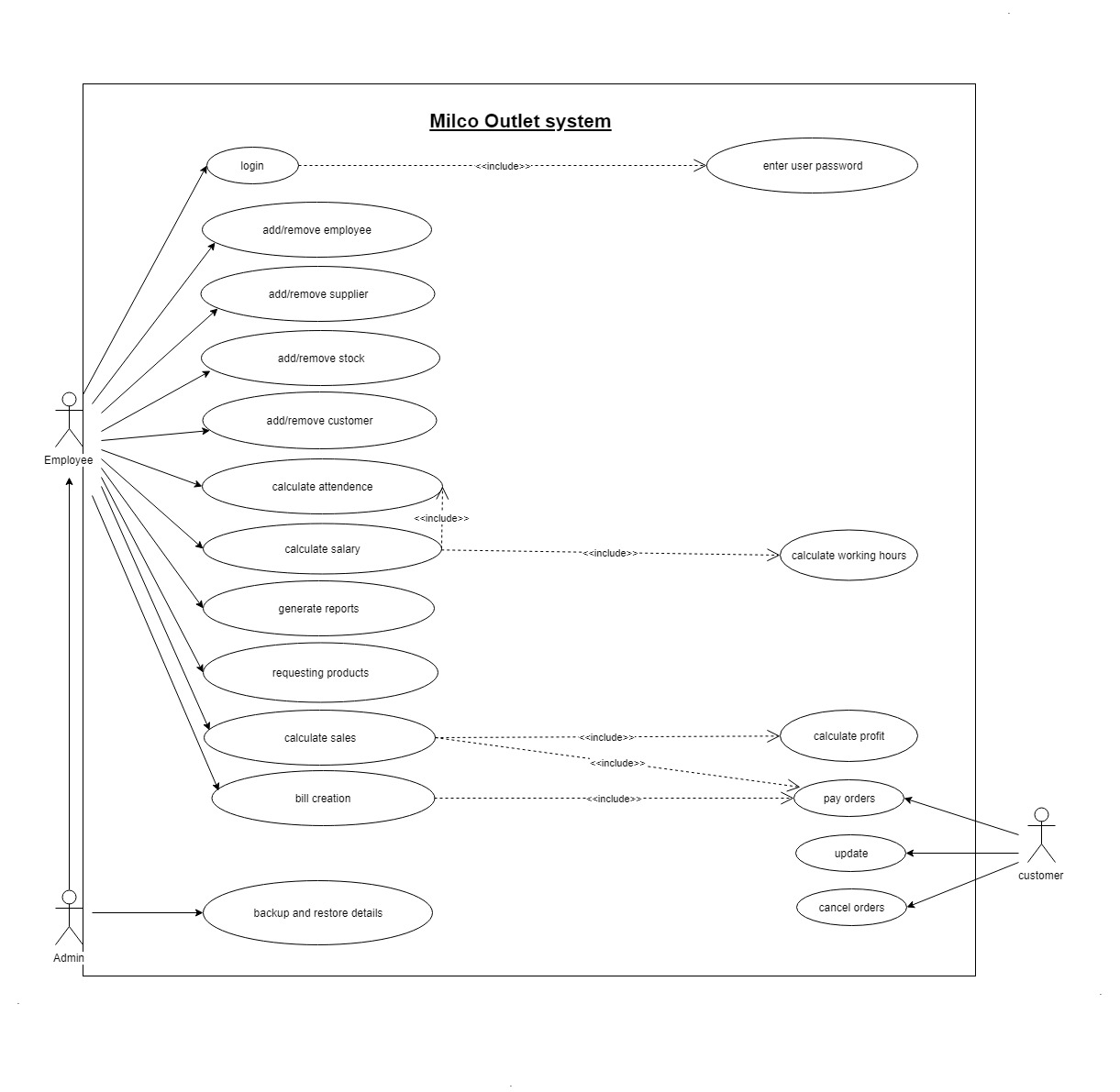
Also it is worthy of mentioning that after doing this project it was realized by us how to maintain the client-developer relationship in a professional manner and how to supply the client with his required needs in the most productive manner.

# Methodology

## Requirements and Analysis

### Use Case Diagram

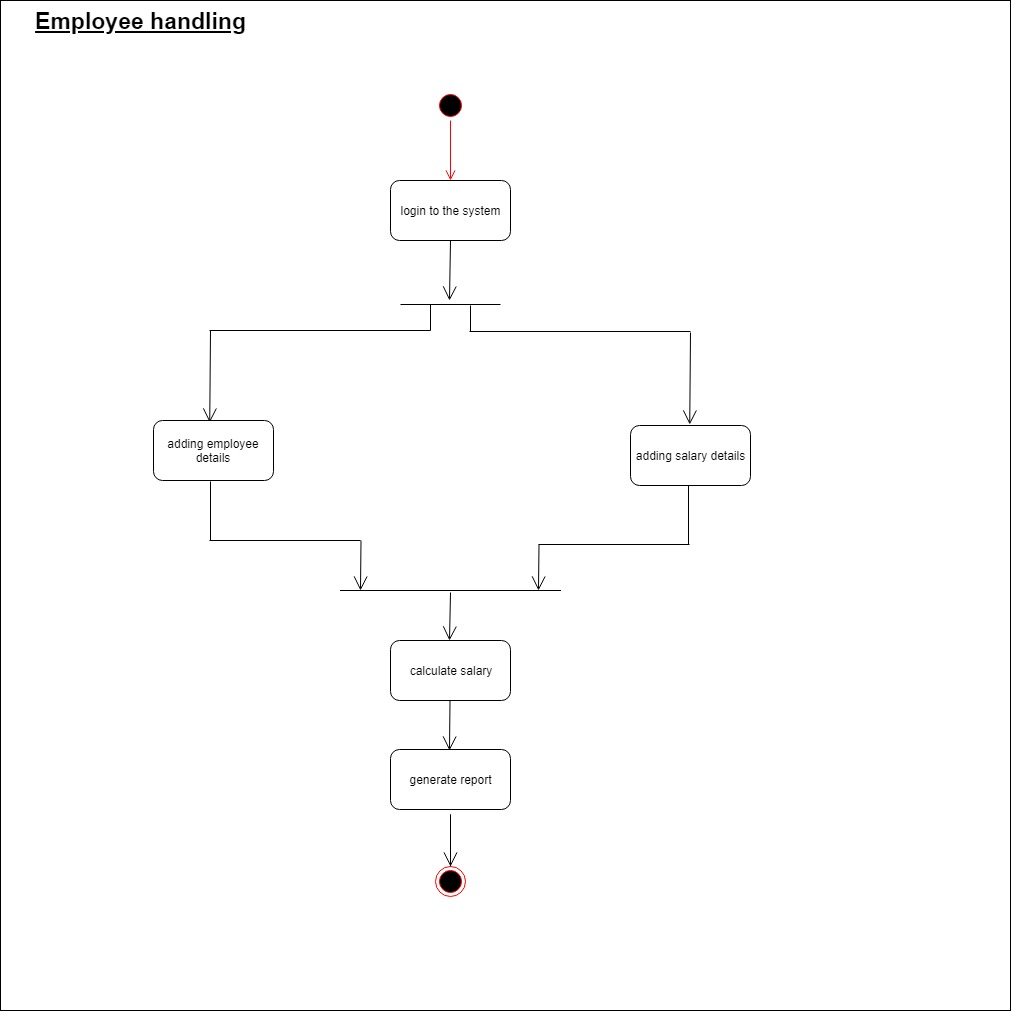
Figure 2.1 Use Case Diagram



* + 1. Activity Diagrams

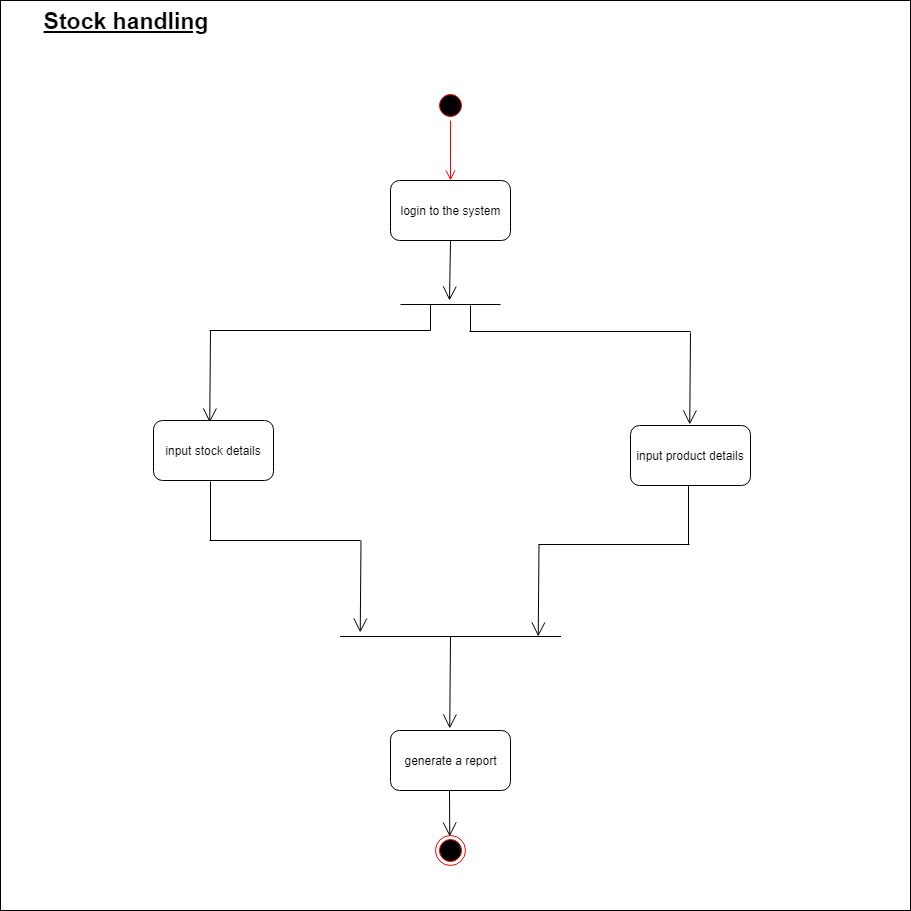
### Employee Handling

Figure 2.2 employee handling Activity Diagram



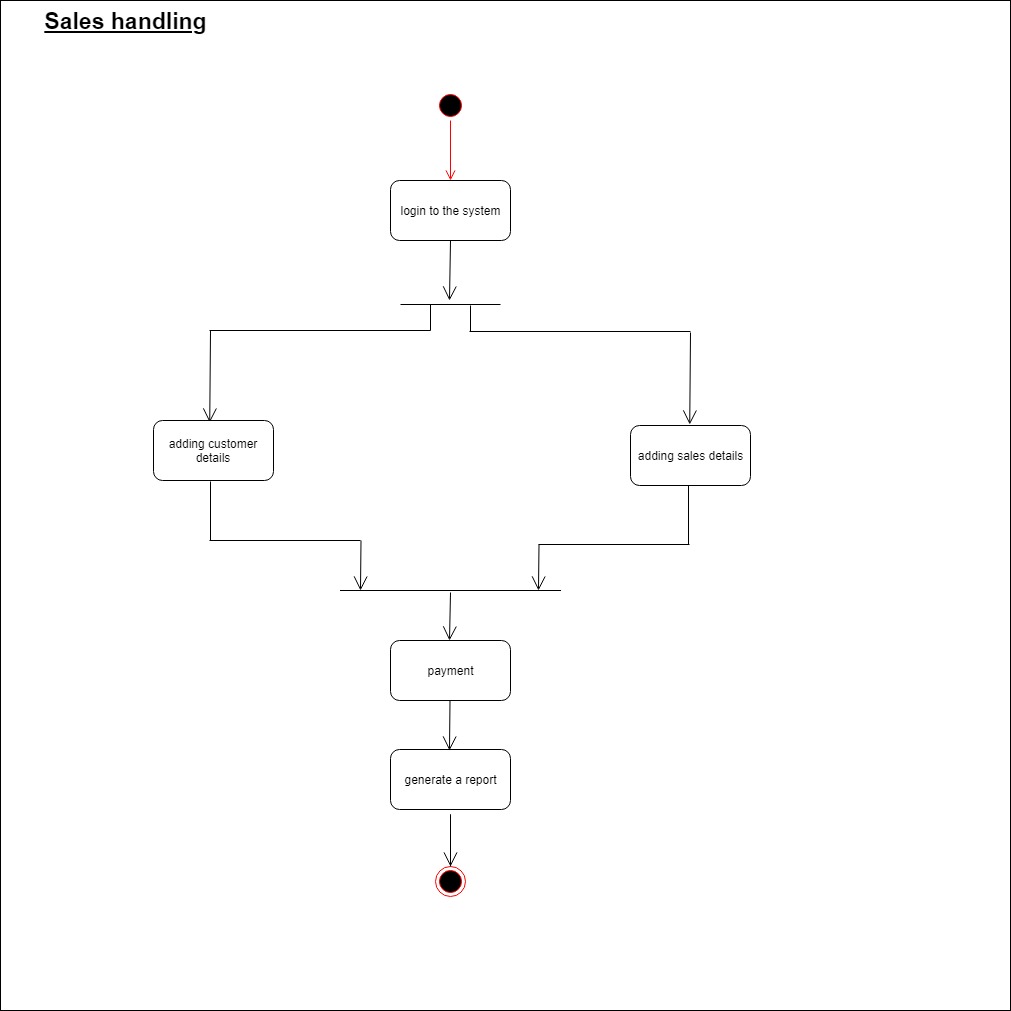
### Stock Handling

Figure 2.3 stock handling Activity Diagram



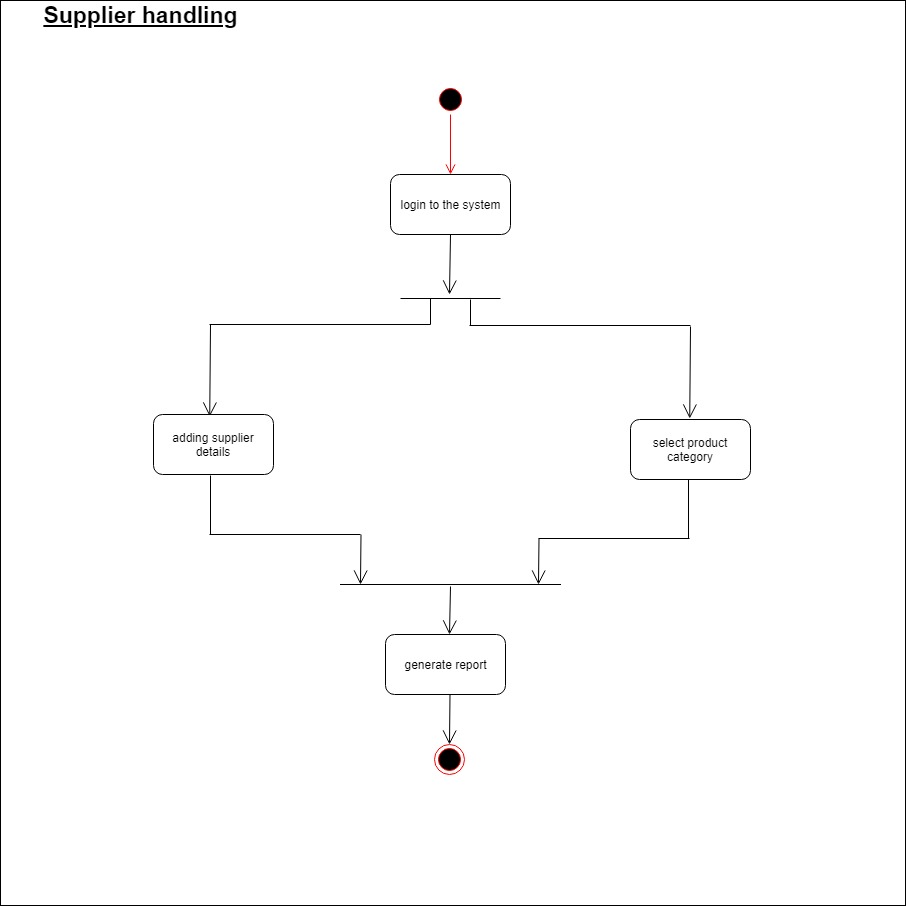
### Sales handling

Figure 2.4 Sales handling Activity Diagram



2.1.2.4 Supplier handling

Figure 2.5 supplier handling Activity Diagram



## Design

### Class Diagram

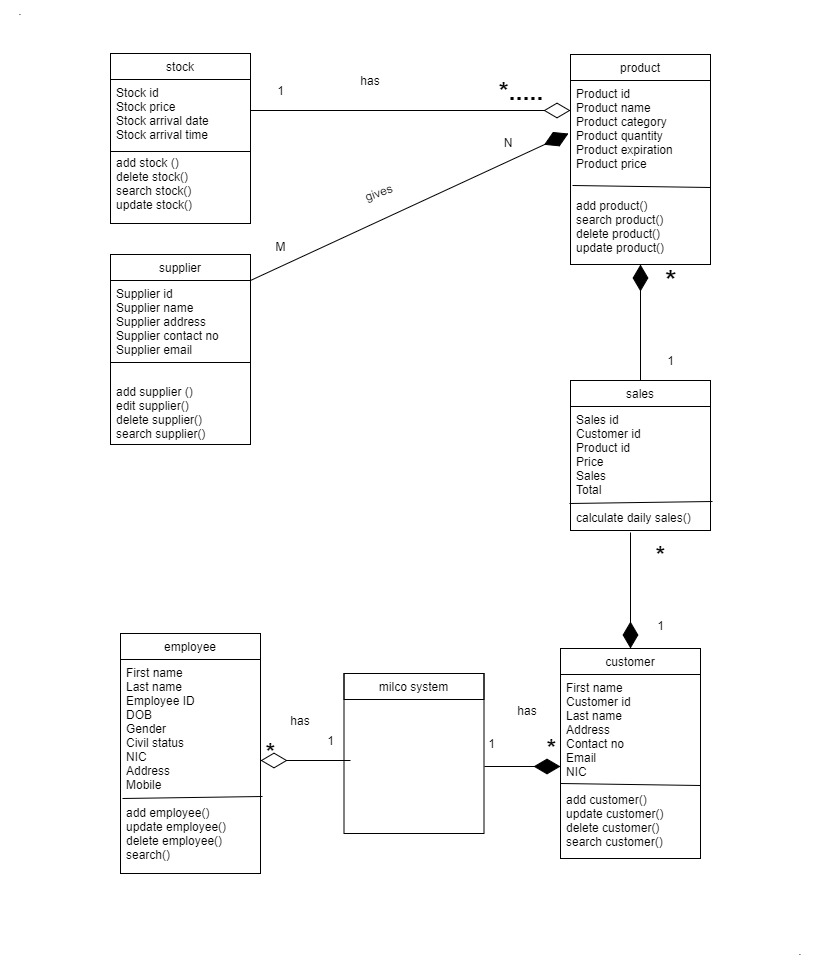


Figure 2.6 Class Diagram

### ER Diagram

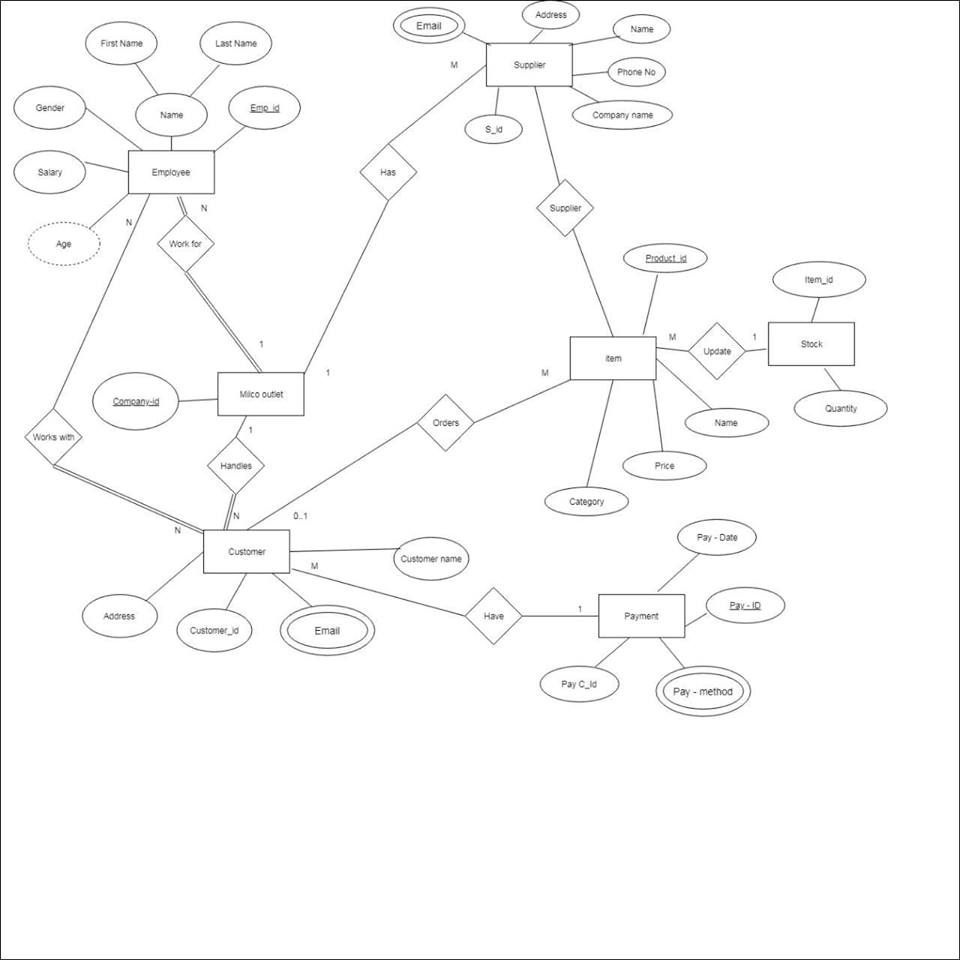
**

Figure 2.7 ER Diagram

## Implementation

Java was used as the developing language and the software was developed by each of us on our individual PCs using NetBeansIDE. Xampp Server with phpMyAdmin was used as the database server. Finally when all the individual components of the project were done, they were all combined together and integrated in one PC. Data is validated and valid data are sent to the database. Some of the special coding that were used is illustrated.

#### Add Employee

#### validation code

#### if (fname.getText().isEmpty()) {

#### JOptionPane.showMessageDialog(null, "First Name is empty");

#### } else if (lname.getText().isEmpty()) {

#### JOptionPane.showMessageDialog(null, "Last Name is empty");

#### } else if (dob.getText().isEmpty()) {

#### JOptionPane.showMessageDialog(null, "Date Of Birth is empty");

#### } else if (address.getText().isEmpty()) {

#### JOptionPane.showMessageDialog(null, "Address is empty");

#### } else if (nic.getText().isEmpty()) {

#### JOptionPane.showMessageDialog(null, "NIC is empty");

#### } else if (CBBGender.getSelectedItem()=="Select") {

#### JOptionPane.showMessageDialog(null, "Gender is not selected");

#### } else if (CBStatus.getSelectedItem()=="Select") {

#### JOptionPane.showMessageDialog(null, "Civil Status is not selected");

#### } else if (address.getText().isEmpty()) {

#### JOptionPane.showMessageDialog(null, "Address is empty");

#### } else if (mobile.getText().isEmpty()) {

#### JOptionPane.showMessageDialog(null, "Mobile Number is empty");

#### } else if (email.getText().isEmpty()) {

#### JOptionPane.showMessageDialog(null, "Email is empty");

#### }else {

#### if(!dob.getText().matches("(0?[1-9]|[12][0-9]|3[01])/(0?[1-9]|1[012])/((19|20)\\d\\d)")){

#### JOptionPane.showMessageDialog(null, "DOB is invalid, enter in format dd/mm/yyyy");

#### }else if(!nic.getText().matches("\\d{9}v")){

#### JOptionPane.showMessageDialog(null, "Invalid NIC, enter in format XXXXXXXXXv");

#### }else if(!mobile.getText().matches("\\d{3}-\\d{7}")){

#### JOptionPane.showMessageDialog(null, "Invalid Number, enter in format XXX-XXXXXXX");

#### }else if(!email.getText().matches("^[a-zA-Z0-9\_+&\*-]+(?:\\."+ "[a-zA-Z0-9\_+&\*-]+)\*@" + "(?:[a-zA-Z0-9- ]+\\.)+[a-z" + "A-Z]{2,7}$")){

#### JOptionPane.showMessageDialog(null, "Invalid Email");

#### }else{

#### pst.executeUpdate();

#### JOptionPane.showMessageDialog(null, "Inserted Successfully");

#### }

#### Add Product

#### validation code

#### if (txtProID.getText().isEmpty()) {

#### JOptionPane.showMessageDialog(null, "Product ID is empty");

#### } else if (txtProName.getText().isEmpty()) {

#### JOptionPane.showMessageDialog(null, "Product Name is empty");

#### 

#### } else if (txtQuantity.getText().isEmpty()) {

#### JOptionPane.showMessageDialog(null, "Quantity is empty");

#### } else if (txtCapacity.getText().isEmpty()) {

#### JOptionPane.showMessageDialog(null, "Capacity is empty");

#### } else if (txtExpDate.getText().isEmpty()) {

#### JOptionPane.showMessageDialog(null, "Expiry Date is empty");

#### } else if ((String)CBCategory.getSelectedItem()=="Select") {

#### JOptionPane.showMessageDialog(null, "Category not Selected");

#### } else if (txtArrDate.getText().isEmpty()) {

#### JOptionPane.showMessageDialog(null, "Arrived Date is empty");

#### } else if (txtArrTime.getText().isEmpty()) {

#### JOptionPane.showMessageDialog(null, "Arrived Time is empty");

#### } else if (txtPrice.getText().isEmpty()) {

#### JOptionPane.showMessageDialog(null, "Price is empty");

#### } else {

#### pst.executeUpdate();

#### JOptionPane.showMessageDialog(null, "Inserted Successfully");

#### }

#### } catch (SQLIntegrityConstraintViolationException e) {

#### JOptionPane.showMessageDialog(null, "Duplicate ID Found");

#### } catch (SQLException | HeadlessException ex) {

#### JOptionPane.showMessageDialog(null, ex);

#### }

#### TV.showTableData(tableNewPro, "newproduct");

#### }

#### private void buttonDeleteActionPerformed(java.awt.event.ActionEvent evt) {

#### try {

#### String sql = "DELETE FROM newproduct WHERE ProductID = ?";

#### con = DriverManager.getConnection("jdbc:mysql://localhost/isproj", "root", "");

#### pst = con.prepareStatement(sql);

#### pst.setString(1,txtProID.getText());

#### pst.executeUpdate();

#### JOptionPane.showMessageDialog(null, "Deleted Successfully!");

#### 

#### } catch (SQLException | HeadlessException ex) {

#### JOptionPane.showMessageDialog(null, ex);

#### }

#### TV.showTableData(tableNewPro, "newproduct");

#### }

#### private void buttonUpdateActionPerformed(java.awt.event.ActionEvent evt) {

#### try {

#### String id = txtProID.getText();

#### String name = txtProName.getText();

#### String quan = txtQuantity.getText();

#### String cap = txtCapacity.getText();

#### String ed = txtExpDate.getText();

#### String cat = (String)CBCategory.getSelectedItem();

#### String ad = txtArrDate.getText();

#### String at = txtArrTime.getText();

#### String pri = txtPrice.getText();

#### String sql = "UPDATE newproduct SET ProductID='" + id + "', ProductName='" + name + "', Quantity='" + quan + "', Capacity='" + cap + "', ExpireDate='" + ed

#### + "', Category='" + cat + "', ArrivedDate='" + ad + "', ArrivedTime='" + at + "', Price='" + pri + "' WHERE ProductID='" + id + "'";

#### con = DriverManager.getConnection("jdbc:mysql://localhost/isproj", "root", "");

#### stmt = con.createStatement();

#### stmt.executeUpdate(sql);

#### JOptionPane.showMessageDialog(null, "Update Successfully");

#### Add Sales

#### validation code

**if (cusID.getText().isEmpty()) {**

**JOptionPane.showMessageDialog(null, "Customer ID is empty");**

**} else if (fname.getText().isEmpty()) {**

**JOptionPane.showMessageDialog(null, "First Name is empty");**

**} else if (lname.getText().isEmpty()) {**

**JOptionPane.showMessageDialog(null, "Last Name is empty");**

**} else if (address.getText().isEmpty()) {**

**JOptionPane.showMessageDialog(null, "Address is empty");**

**} else if (contact.getText().isEmpty()) {**

**JOptionPane.showMessageDialog(null, "Contact is empty");**

**} else if (email.getText().isEmpty()) {**

**JOptionPane.showMessageDialog(null, "Email is empty");**

**} else if (nic.getText().isEmpty()) {**

**JOptionPane.showMessageDialog(null, "NIC is empty");**

**} else {**

**if(!email.getText().matches("^[a-zA-Z0-9\_+&\*-]+(?:\\."+ "[a-zA-Z0-9\_+&\*-]+)\*@" + "(?:[a-zA-Z0-9-]+\\.)+[a-z" + "A-Z]{2,7}$")){**

**JOptionPane.showMessageDialog(null, "Invalid Email");**

**}else if(!nic.getText().matches("\\d{9}v")){**

**JOptionPane.showMessageDialog(null, "Invalid NIC, enter in format XXXXXXXXXv");**

**} else if (!contact.getText().matches("\\d{3}-\\d{7}")) {**

**JOptionPane.showMessageDialog(null, "Invalid Number, enter in format XXX-XXXXXXX");**

**} else {**

**pst.executeUpdate();**

**JOptionPane.showMessageDialog(null, "Inserted Successfully");**

**}**

**}**

**} catch (SQLIntegrityConstraintViolationException e) {**

**JOptionPane.showMessageDialog(null, "Duplicate ID Found");**

**} catch (SQLException | HeadlessException ex) {**

**JOptionPane.showMessageDialog(null, ex);**

#### Add Sales

#### validation code

**if (SaleID.getText().isEmpty()) {**

**JOptionPane.showMessageDialog(null, "Sale ID is empty");**

**} else if (CID.getText().isEmpty()) {**

**JOptionPane.showMessageDialog(null, "Customer ID is empty");**

**} else if (PID.getText().isEmpty()) {**

**JOptionPane.showMessageDialog(null, "Product ID is empty");**

**} else if (txtPrice.getText().isEmpty()) {**

**JOptionPane.showMessageDialog(null, "Price is empty");**

**} else if (txtSales.getText().isEmpty()) {**

**JOptionPane.showMessageDialog(null, "Sales is empty");**

**} else {**

**pst.executeUpdate();**

**JOptionPane.showMessageDialog(null, "Inserted Successfully");**

**}**

**} catch (SQLIntegrityConstraintViolationException e) {**

**JOptionPane.showMessageDialog(null, "Duplicate ID Found");**

#### Add Supplier

#### validation code

**if (txtSupID.getText().isEmpty()) {**

**JOptionPane.showMessageDialog(null, "Supplier ID is empty");**

**} else if (txtSupName.getText().isEmpty()) {**

**JOptionPane.showMessageDialog(null, "Supplier Name is empty");**

**} else if (txtEmergency.getText().isEmpty()) {**

**JOptionPane.showMessageDialog(null, "Emergency is empty");**

**} else if (txtEmail.getText().isEmpty()) {**

**JOptionPane.showMessageDialog(null, "Email is empty");**

**} else if (txtAddress.getText().isEmpty()) {**

**JOptionPane.showMessageDialog(null, "Address is empty");**

**} else {**

**pst.executeUpdate();**

**JOptionPane.showMessageDialog(null, "Inserted Successfully");**

**}**

**} catch (SQLIntegrityConstraintViolationException e) {**

**JOptionPane.showMessageDialog(null, "Duplicate ID Found");**

**} catch (SQLException | HeadlessException ex) {**

**JOptionPane.showMessageDialog(null, ex);**

#### Code for Daily Sales Calculate

**double total = price \* sales;**

**String tot = new Double(total).toString();**

**lblTotal.setText(tot);**

#### Code for Salary Calculate

**double totSal;**

**double basic = Double.parseDouble(txtBasicSal.getText());**

**double rate = Double.parseDouble(txtRate.getText());**

**double hours = Double.parseDouble(txtHours.getText());**

**totSal = basic + (rate \* hours);**

**String totSt = new Double(totSal).toString();**

**lblTotal.setText(totSt);**

#### Code for Generate Report

#### try{

#### String report="C:\\Users\\ASANKA\\Desktop\\JAVA\\ProjectIS\\src\\projectis\\sample.jrxml";

#### JasperReport jr=JasperCompileManager.compileReport(report);

#### JasperPrint jp=JasperFileManager.fillReport(jp,null,conn);

#### JasperViewer.vierReport(jp);

#### }

#### catch(Exception e)

#### {

#### JOptionPane.showMessageDialog(null, e);

## Testing

Table 1-4 Test Cases

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test case ID** | **Test case Description** | **Preconditions** | **Test procedure** | **Test inputs** | **Expected Results** |  |
| Test case 1 | Check for a valid password to login | Password field should not be null/empty. | The password field must be filled.  Password must contain 8 characters | xxxxx | Enter a valid password to the login successfully |
| Test case 2 | Check for a valid email address | Email field should not be null/empty. | The email field must be filled. Index of ‘**@**’should be lower than index of ‘**.** ’ | [xxxxxxx@xxx.com](mailto:xxxxxxx@xxx.com) | Add an email to the system database according to the Format |
| Test case 3 | Check for a valid contact number | Contact No field must be filled. | The Contact No field must be filled only with 10 numerical values/digits. | xxx-xxxxxxx | Add telephone number to the system database using only 10 digits |
| Test case 4 | Check for a valid NIC | NIC field must be filled. | NIC field must be filled only with 9 numerical values/digits in the places of x | xxxxxxxxxV | Add NIC to the system database according to the format |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test case 5 | Check if all the fields are filled | All the fields must be filled. | All the fields must be filled to proceed Otherwise the error message will display | - | Fill all the text fields |  |
| Test case 6 | Checking if the record already exists in the database | The relevant record must be in the system database. | When the user enter a new record, that particular record should be in the system.  If so display an error message | - | Enter a new record successfully |

# Evaluation

## Assessment of the Project results

The objectives set for this project were defined in the Introduction. As per that, the main objective for this entire project is to computerize and manage all the main functions of the company to make the activities to be carried out conveniently and efficiently. The main functions categorized as Employee Handling, Stock handlying, sales handling and supplier handling. As well each sub system comes out with a separate unique object to be fulfilled. As separate systems each sub system worked with or without minor faults. The Milco Outlet Handling system and is connected with visa processing management system also according to their functionalities.

## Lessons Learned

While doing this project a lot of valuable lessons were learned by us.

* One of the best lessons that was understood by us as a team how to share the duties equally among us and perform them to our level best.
* And also time management was one of the most crucial problems that we had to encounter. There were deadlines assigned to us and it was always important that the work was submitted beforehand.
* Developed coding skills in java and learned to self-study with the help of online tutorials and video documentations.
* It was understood by us how huge IT projects are done in the industry and gained a vast experience in project management.

## Future Work

This prototype project is going to be the automated software that the client is going to use and if the management decides to increase the usage of computers for their other tasks, this system would help them a lot. Furthermore we will be planning to make some changes to add email system to send e-mail to suppliers when the stock wants to refill automatically when the product limit gone down. As well as we planning to set a finger print machine to get employee attendance properly. So above mentioned suggestions would be the future expectations of this prototype system.

# Conclusion

Standalone solution for the company Milco outlet ,Ragama has been a successful project throughout the execution and completion.

The team managed to incorporate most, if not all, of the requirements given by the above company into the final product. Time constraint on the project was the biggest limitation the team has faced. Although, for the given time period, the team has successfully developed a fully functioning prototype system.

However, the system can be improved further by implementing performance improvements, better error handing, and fixes for various bugs that are yet to be discovered. Additional functionality can also be implemented in the future as the system is designed to be flexible and easily maintainable.

This project made the team gain confidence on working with the industry and acquires a great deal of information on following a software development life cycle model. The team also gained a higher level of expertise in both developing a software-based solution as well as in the world of business.

Each member of the team was responsible for the success of the project. In the end the team was able to achieve their goal in succeeding with their prototype system for the above company.

# References

1. SLIIT Course Web

#### [www.courseweb.sliit.lk](http://www.courseweb.sliit.lk/)

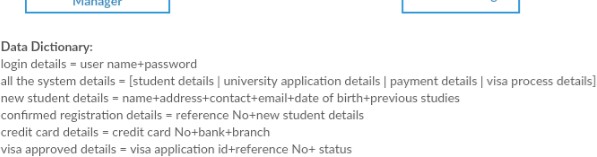
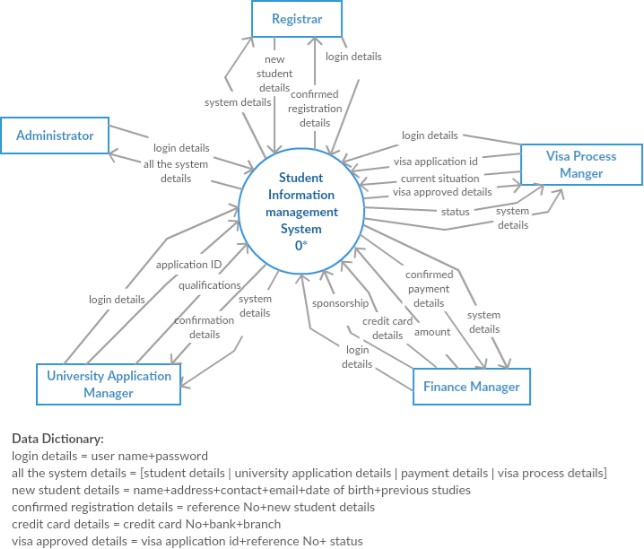


Figure A.1 Context Diagram

**Appendix A: Design Diagrams**



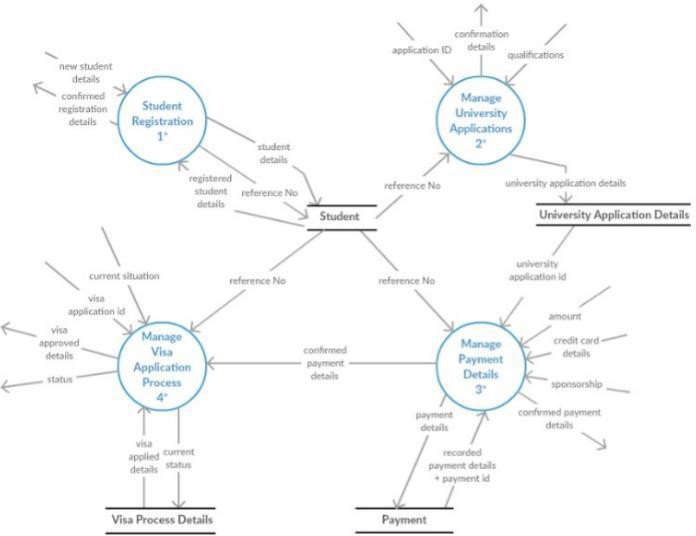


Figure A.2 Data Flow Diagram Level 1

### Student Registration

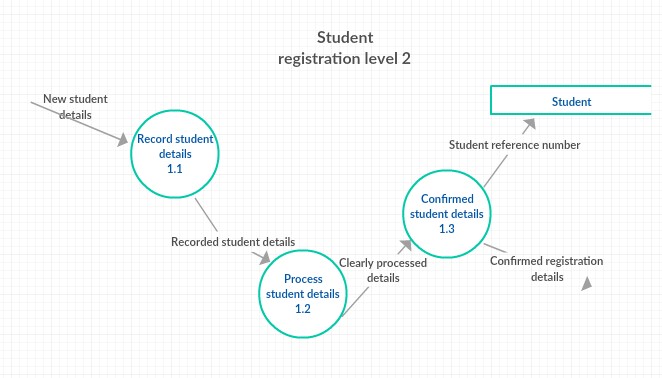


Figure A.3 Data Flow Diagram Level 2- Student Registration

### anagev University Applications

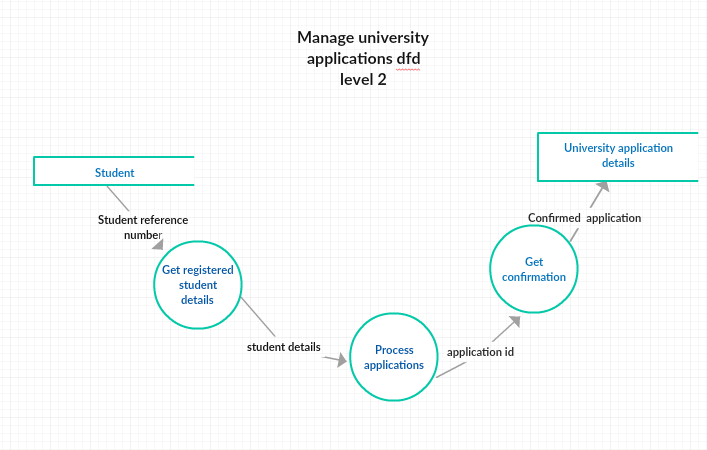


Figure A.4 Data Flow Diagram Level 2- University Application Management

### Manage Payment Details

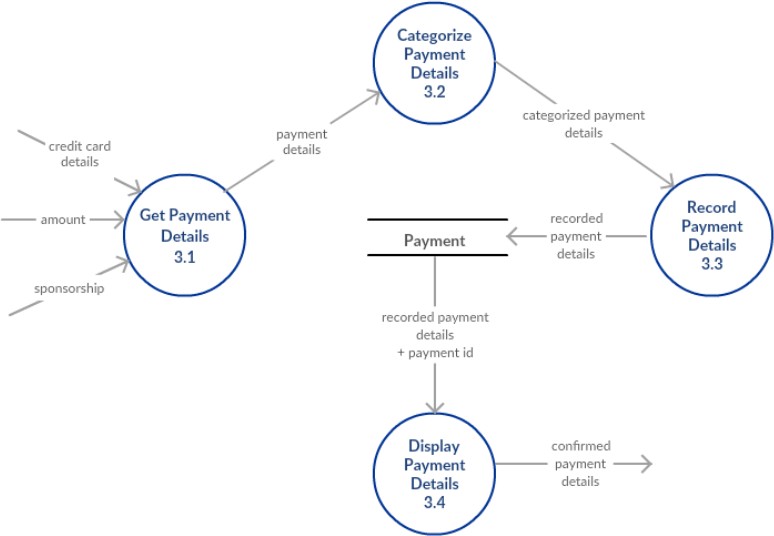


Figure A.5 Data Flow Diagram Level 2- Payment Handling

### Manage Visa Process Details

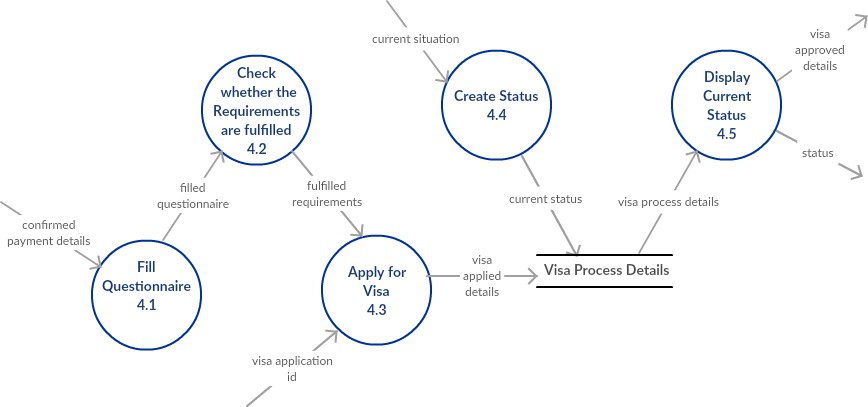


Figure A.6 Data Flow Diagram Level 2- Manage Visa Process Details

# Appendix C: Selected Code Listings

#### Code for Database Connection

public class TableView {

Connection con = null;

PreparedStatement pst = null;

ResultSet rs = null;

Statement stmt = null;

public void showTableData(JTable tab, String tabSQL) {

try {

con = DriverManager.getConnection("jdbc:mysql://localhost/isproj", "root", "");

String sql = "SELECT \* FROM "+tabSQL;

pst = con.prepareStatement(sql);

rs = pst.executeQuery();

tab.setModel(DbUtils.resultSetToTableModel(rs));

} catch (SQLException ex) {

JOptionPane.showMessageDialog(null, ex);

}

}

}