RESTAURANT MANAGEMENT SYSTEM FOR NIMANSALA RESTAURANT

T.H.L.C CHANDRATHILAKE

November 2017

RESTAURANT MANAGEMENT SYSTEM FOR NIMANSALA RESTAURANT

T.H.L.C CHANDRATHILAKE

R091541

0915416

Name of the supervisor: Mr. H.J.K.S.C Wijerama

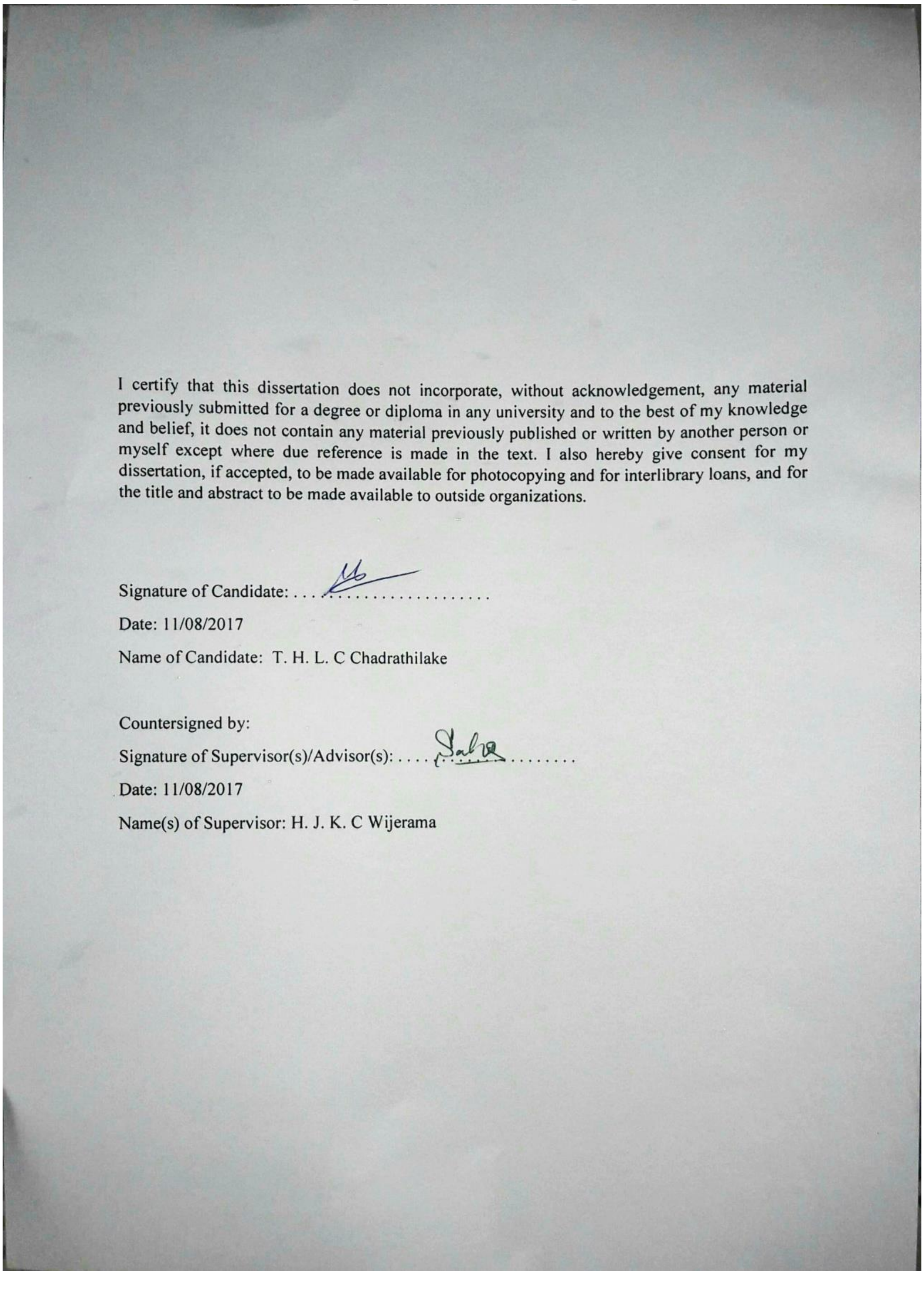
November 2017

BIT

UCSC

This dissertation is submitted in partial fulfillment of the requirement of the Degree of Bachelor of Information Technology of the University of Colombo School of Computing

DECLARATION



I certify that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a degree or diploma in any university and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, to be made available for photocopying and for interlibrary loans, and for the title and abstract to be made available to outside organizations.

Signature of Candidate: [illegible]

Date: 11/08/2017

Name of Candidate: T. H. L. C Chadrathilake

Countersigned by:

Signature of Supervisor(s)/Advisor(s): [illegible]

Date: 11/08/2017

Name(s) of Supervisor: H. J. K. C Wijerama

ABSTRACT

Information and Communication Technology is being improved daily to ensure the quality of Information which has turned out to be one of the most crucial aspects of nowadays businesses. Nowadays computing devices are equipped with greater processing power, memory capacity and many other technical capabilities.

Information Systems have been positively affecting many areas of businesses such as customer service, employee management, accounting and financial control, stock control and project management etc.

Nimansala Restaurant has been in the business for decades serving the people who live or travel around the Godagama, Homagama area. Apart from its main business as a food supplier, this organization provides some other amenities such as reception hall, liquor bar and it is decided to guest room services be started in near future.

Though the quality of the foods, beverages and other facilities provided by this company has been in an outstanding level, it has been a great challenge to provide quicker service for the customers with the current manual processes. It has also become a challenge to the management when it is needed to access the business information. With the daily increasing demand, it has become almost impossible to keep better control of finance and stocks, marketing, customer satisfaction and business growth while keeping employees not been unwantedly pressurized and preventing currently available human resources not been expanded superfluously.

Author was asked to analyze, design and develop a web based restaurant management system to unravel these identified problems and ameliorate the capabilities of the business. As discussed the implementation been done in few phases. In first phase, the crucial problems are solved and in following phases, new features are being added to the system to ameliorate the capabilities, availability and performance.

With the proposed system, it is expected 25% growth of revenue in next financial year while limiting the customer waiting time to maximum limit of 10 minutes and gain 8:2 ratio of positive and negative customer feedbacks.

ACKNOWLEDGEMENT

This will be considered as my opportunity to thank all the persons who have given a support to bring this project to this level. Starting from the parents, I would be privileged to have a mother and a father like them. Without their support, this system couldn’t have been taken to this level.

I have to give my sincere thanks to my supervisor, Mr. Sahan Wijerama who have supported me a lot in all the time I require. I should also be thankful to my Sisters for encouraging me and making my difficult times easier.

I would like to thank my best friends Mr. Sahan Wijerama and Mr. Shanuka Dilshan, who are also BIT graduates, for helping, guiding, and leading me through the project and whenever I had difficulties they are the people who gave me a relief and tracked me back in.

A special thank goes to the current staff members with whom I worked during the project for their support, and the many questions that they patiently answered while I went through the manual records. They were always ready and forthcoming with their suggestions.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1 - Introduction

1.1 The Client

Nimansala is mainly a Restaurant which provides foods, beverages and liquor. Further, this business has been providing reception hall facilities and they are planning to expand the business to provide guest room facilities as well.

1.2 Problem Domain

This business has gained a good customer reputation as they have been maintaining the quality of their products at an outstanding level. Customer base has grown very fast during the past decades. As well as the quality, the customers are expecting quicker service and especially customers are more concerned about First Come First Served (FCFS) policy.

“First-come, first-served (FCFS) – sometimes first-in, first-served and first-come, first choice – is a service policy whereby the requests of customers or clients are attended to in the order that they arrived, without other biases or preferences.” [1]

In order to achieve this, the company need to keep an accurate track of the customer orders, effectively communicate them to kitchen or bar and as soon the order is ready it should be delivered to the customer instantly.

When the reception hall operations are managed, the main task is handed over to kitchen. Kitchen staff should be able to deliver the required number of food portions (plates) on time.

While these processes should be efficient and accurate, management has to be aware of the business information timely and efficiently. While the sales are done, it is necessary to keep track on currently available stocks, their re-order levels, expiry dates to ensure smooth and proper business flow. In other way, management has to consider daily income, sales analysis and customer feedbacks for management decisions and short term/long term business planning.

1.3 Motivation for the Project

Nimansala Restaurant is currently operated by a manual system which completely depends on the capabilities of the staff. Sometimes it takes too long to serve a customer due to human errors or efficiency issues. Many customers were disappointed due to violation of FCFS policy.

As author has been informed during the analysis phase, management of this restaurant faced many difficulties due to lack of information. Cashier balance shortages, unavailability of ingredients and beverages due to expirations and stock consumptions, and day end conflicts have been frequent problems of this company.

If this company is aided by a proper Restaurant order management system, it will help the employees to minimize the human errors and efficiency issues and have an accurate track of the daily operations. This will help the management to retrieve real-time information accurately. With the great flexibility that the Information Systems can offer, it will be a great opportunity to management to find new ways of understanding the business related information.

1.4 Objectives of the Project

This Information System (IS) project is expected to immediately solve the critical business problems as the first priority. Eventually, it needs to be improved for new business capabilities and functionalities with the growth of the company.

Immediate goal of this project was to increase the revenue from 25% by the next financial year. Also it was expected to achieve 8:2 ratios among the positive and negative customer feedbacks by reducing the maximum waiting time for a typical order for maximum of 10 minutes.

Further, it was expected to reduce the employee stress and make any pre-defined information is available for the management within maximum of 5 minutes.

1.5 Scope of the Project

1.5.1 Scope of The Phase 1

This phase will mainly be focused on business problems that need to be resolved immediately.

1. Managing Common Information
    a. Designations
    b. Job Categories
    c. Staff Information
    d. Users and User Permissions
2. Stock Management
    a. Units
    b. Items
    c. Locations
    d. Item Batch
    e. Suppliers
    f. Stock Transfer Notes
    g. Direct GRNs
    h. Purchase Orders
    i. Goods Receive Notes
    j. Automated stock reduction records (on sales)
3. Sales and Order Management
    a. Restaurant/ Bar Order
    b. Reception Hall Reservation and Catering
    c. Issue Invoices
    d. FCFS Policy Control
    e. Customer Information Management
4. Kitchen Management
    a. Recipe Maintenance
    b. Kitchen Order Processing
    c. Kitchen Instructions

1.5.2 Scope of The Phase 2

This phase will be started after the successful implementation of the phase 1. This area mainly focused on immediate improvements that can be done to existing system and to the company.

1. Reception Hall and Room Reservations
    a. Room Reservation
    b. Hall Reservation
    c. Room Reservation related order processing

    d. Online Reservation
2. Sales and Order Management
    a. Mobile Application for Waiters
    b. Online Orders for take away (home delivery feature will be added in following phases)
3. Finance Management
    a. Accounts
    b. Inter Account Transfers
    c. Miscellanies Invoices
    d. Cash Vouchers
    e. Cash Drawer Management
4. Common Functionalities
    a. SMS Alerts
    b. Advanced Security
    c. Mobile Accessibility

1.6 Outline of the Chapters

1.6.1 Analysis

Under this chapter it is discussed how the system analysis have been carried out and the techniques used. And in this chapter the identified functional and non-functional requirements are discussed with relevant diagrams.

Further, current manual system and few existing software systems are explained and advantages and disadvantages will be reviewed.

Selected software development methodologies, technologies and other constraints will also be discussed in this chapter.

1.6.2 Design

The design of the software system will be comprehensively discussed in this chapter with the relevant diagrams which illustrate the system functionalities. Also, Identifying classes, relationships and object lifecycles will be discussed in design chapter.

1.6.3 Implementation

In this chapter, it will be discussed how the actual software system developed and techniques and technologies used.

Further in this chapter it will be discussed how the system is deployed on customer site, the configurations made and how the system transition has taken place. Required resources will also be discussed under this chapter. This chapter will have relevant diagrams to further explain the content.

1.6.4 Testing and Evaluation

In order to make sure the accuracy of the information processed in the system, the system need to be tested for different combination ions of cases. In this chapter, the testing techniques, test cases and their results will be discussed.

1.6.5 Conclusion

In this chapter, the actual effect of the proposed system to the business will be discussed and further, the project will be reviewed in project management view.

Further, a list of lessons learned will be discussed by understanding what went wrong and what went good. Suggestions will be made and decisions will be discussed to improve the future phases of same project as well as the other projects.

Chapter 2 - Analysis

2.1 Fact Gathering Techniques

This project was started with a meeting between the author and owner of Nimansala Restaurant. The domain area and the main business requirements have been discussed in this meeting.

Later, several analysis meetings were held with the client, author and the project supervisor. Most of the requirements have been identified and prioritized and planned in these set of meetings.

Further analysis has been done by analyzing the existing document formats, observing the day to day operations and discussing with users. Most of the non-functional requirements and some of the functional requirements have been raised by the users.

2.2 Analyzing the current manual system

When the operations of Nimansala Restaurant are analyzed, it was identified that these operations can be categorized into few areas.

2.2.1 Restaurant Operations

Nimansala Restaurant has the capacity of serving more than 100 people at a time. After 5:00 PM, the bar is also opened and more than 60% of this capacity is occupied on any given day. On Fridays, Saturdays and proceeding days of a holidays, this restaurant becomes more busy.

Taking orders and get them delivered to the customer should be done as much as quickly and following FCFS policy is vital in order to ensure the customer is delighted. Waiter has to visit the kitchen time to time to check the status of the order. If the waiter engaged in any other operation, there is no way of communicating which order is ready and which is not. Due to this communication gap, customer may have to wait unwantedly even their order is ready and there is a possibility of delivering the order to a new customer who requested same dish later.

2.2.2 Kitchen Operations

When a Kitchen Order is received from the restaurant or the reception hall, kitchen staff is responsible for delivering the order as quickly as possible while maintaining the quality of the product in highest level. Since the business is about food, nothing can be compromised.

Managing orders is an interesting and challenging task. Rather than struggling to deliver a time consuming order, it is clever to process and complete the quickly deliverable orders in higher priority even it breaks the FCFS policy. This has to be decided by overlooking all the orders available, the time estimations and the situational priorities.

Recording the ingredient consumptions and understanding the available stocks is also essential in this context.

2.2.3 Stores Management and Procuring

Stocks are the key of this business. Without a proper management of stocks, it is almost impossible to ensure the smooth run of the business.

When the stocks are consumed, the remaining balances of the stocks should be recorded and when those balances reach the reorder level, stores manager should place Purchase Orders to the suppliers. Or buy directly through the DGRN which is a hybrid version of PO and GRN. If a PO is placed, Goods should be received through concomitant GRNs.

2.2.4 Reception Hall Management

Reception Hall is Mostly provided free of charge (FOC) with the catering package. Package will be selected according to the number of plates which is going to serve and the menu wanted. Apart from that, there are some additional facilities which can be added to the reservation order. These can be FOC or Chargeable.

On the reserved day, an order should be made to the kitchen according to required number of plates and the menu. These orders should be ready on time and any delay will make the customer uncomfortable as it affects the whole agenda.

2.2.5 Finance and Accounting

At the end of day, cashier should balance the drawer with the invoice records and add the drawer collection to the relevant account.

Payments for Procurements should be made and remaining amounts should be banked.

Finance and Account matters are currently handled by the owner of the company, and it is required to prepare financial and revenue reports monthly, quarterly, bi-annually and yearly for different purposes such as taxation. These should be highly accurate as it effects directly to management decisions and inaccurate data may generate deceitful information which plows the company into unwanted legal problem.

2.2.6 Bar Management

During the bar open hours the orders received from the bar and reception hall should be delivered and due to nature of this domain, strict customer care is essential to avoid unwanted warm situations.

Liquor is purchased in different sized bottles and in bulk. Usually liquor is ordered by bottle size or number of milliliters. These different conditions of same item have different selling conditions. This nature of business has been challenging always when keeping an accurate track of stocks.

2.3 Existing similar systems

2.3.1 Floreant POS - Open source Point Of Sale For Restaurant

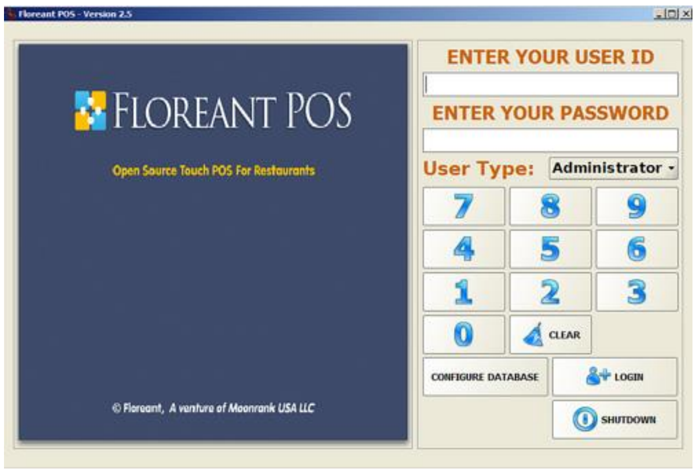
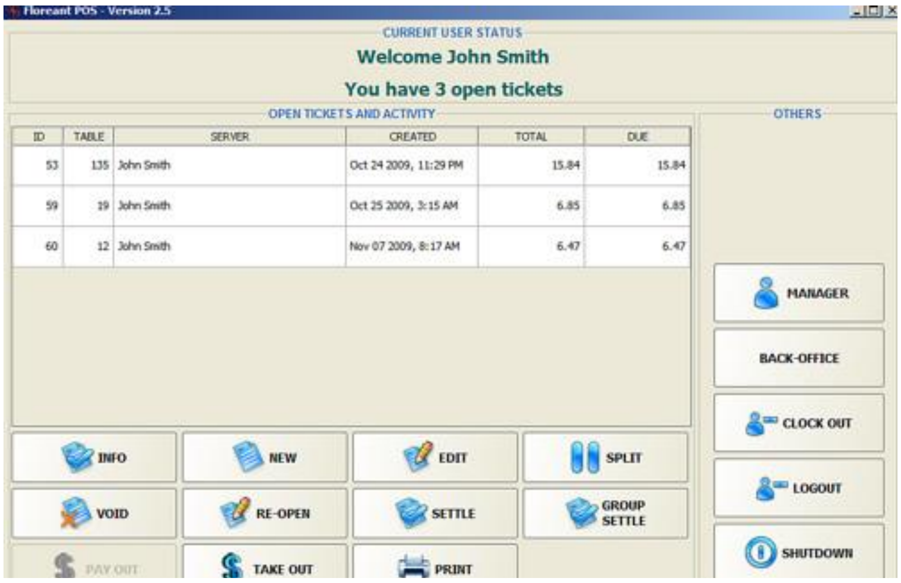


Figure 2-1: Floreant POS

“Floreant POS offers an ideal computer system for dining, restaurant management and franchise food service. Complete with detailed sales reporting, food cost and labor cost analysis, it provides intuitive touch screen ordering software for table-service, delivery, take-out and catering. This software has been released under” [2]

This is a free and open source Standalone System and Runs in Windows, Linux, Mac and Java supported Tablets. This has mainly focused on restaurant operations such as table order management, cooking instruction management, and daily sales and cash management. Shift based pricing can be considered as a valuable feature to manage happy hours concept.

This system doesn’t have stock management facility and reception hall management facility. In order to manage those particular areas, this system should be customized and improved.



Welcome John Smith

You have 3 open tickets

ID	TABLE	SERVER	CREATED	TOTAL	DUE
53	135	John Smith	Oct 24 2009, 11:29 PM	15.84	15.84
59	19	John Smith	Oct 25 2009, 3:15 AM	6.85	6.85
60	12	John Smith	Nov 07 2009, 8:17 AM	6.47	6.47

Figure 2-2: Main Screen of Floreant POS

2.3.2 Samba POS - Restaurant POS Software

This software also has mostly same features as in Floreant POS. This software is also windows based and version 4 of this is freely available and this software is open source. [3]

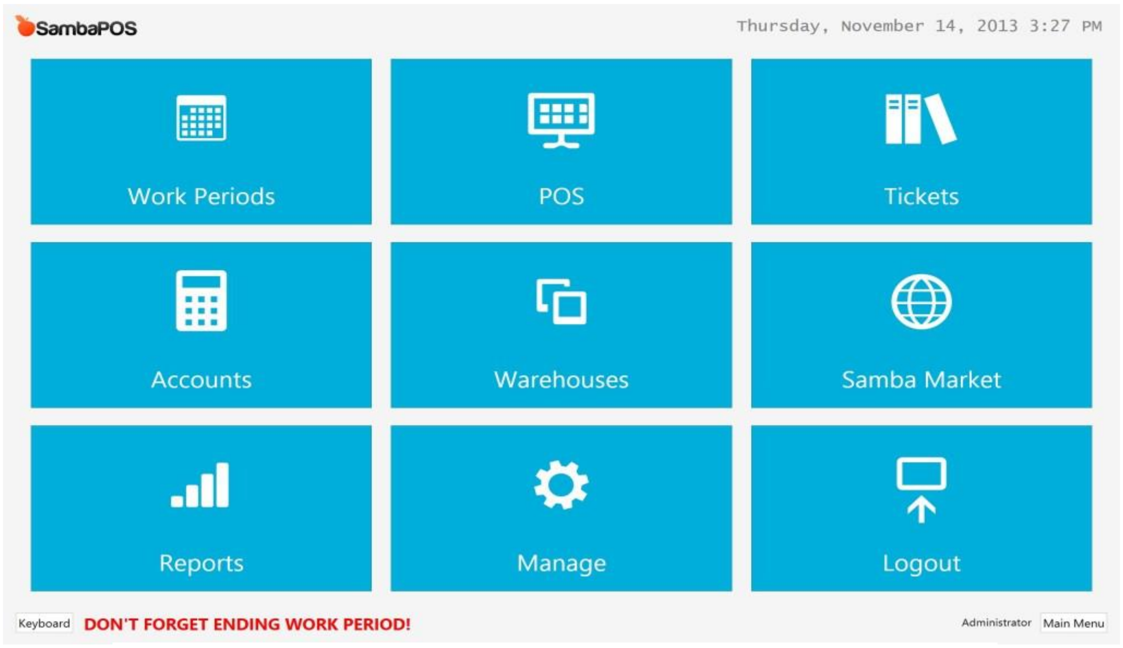
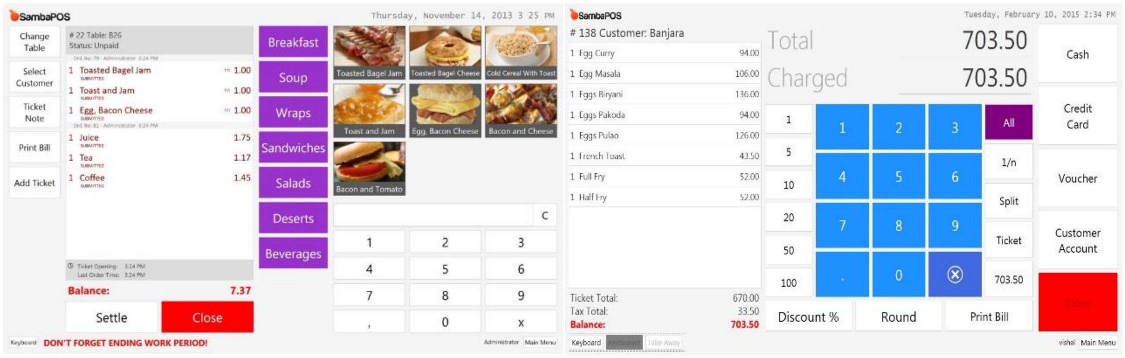


Figure 2-3: Samba POS Main Screen

This software comprehensively manage the restaurant and kitchen operations and the attractive user interfaces are sleek, modern and user friendly therefore User experience with this software is in excellent level.



22 Table: B26
Status: Unpaid

1	Toasted Bagel Jam	1.00
1	Toast and Jam	1.00
1	Egg, Bacon Cheese	1.00
1	Juice	1.75
1	Tea	1.17
1	Coffee	1.45

Ticket Opening: 3:24 PM
Last Order Time: 3:24 PM
Balance: 7.37

138 Customer: Banjara

1	Egg Curry	94.00
1	Egg Masala	106.00
1	Eggs Biryani	136.00
1	Eggs Pakoda	94.00
1	Eggs Pulao	126.00
1	French Toast	43.50
1	Full Fry	52.00
1	Half Fry	52.00

Ticket Total: 670.00
Tax Total: 33.50
Balance: 703.50

Total 703.50
Charged 703.50

Figure 2-4: Order Placement and Payment Screens of SambaPOS

Restaurant table layouts can be customized graphically and it is possible to track the occupied tables, seats and the orders made by each table separately.

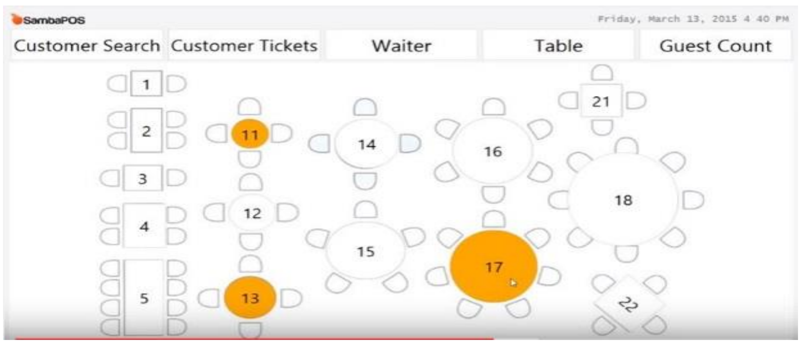


Figure 2-5: Custom Table Layout of Samba POS

2.4 Functional Requirements

2.4.1 Managing Common Information

1. Designations
2. Job Categories
3. Staff Information
4. Users and User Permissions

2.4.2 Stock Management

1. Units
2. Suppliers
3. Configure Items
   a. Item Locations
   b. Item Units and Conversions
   c. Suppliers
   d. Re-Order Level Setups
4. Item Batch
   a. Manually Create Item Batch
   b. Merge two Item Batches
   c. Manual Reconciliations
5. Stock Transfer Notes
6. Purchase Orders (PO)
7. Goods Receive Notes (GRN)
8. Direct Goods Receive Notes (DGRN)
9. Automated stock reduction records (on sales)

2.4.3 Sales and Order Management

1. Restaurant/ Bar Order
2. Reception Hall Reservation and Catering
3. Issue Invoices
4. FCFS Policy Control
5. Kitchen Order Ticket (KOT) Processing

6. Bar Order Ticket (BOT) Processing
7. Customer Information Management

2.5 Non-Functional Requirements

Online Availability and Real-time accessibility

This system should be accessible through the internet so that the management can view the business information remotely.

Quality of Information

The information provided by this system should be accurate, flexible, relevant, and reliable. Complete Information should be retrieved timely in economical way and should be represented as simply as possible allowing users to verify the reliability.

High security and workflow restrictions

User friendliness and higher performances

2.6 Use case diagram for entire system.

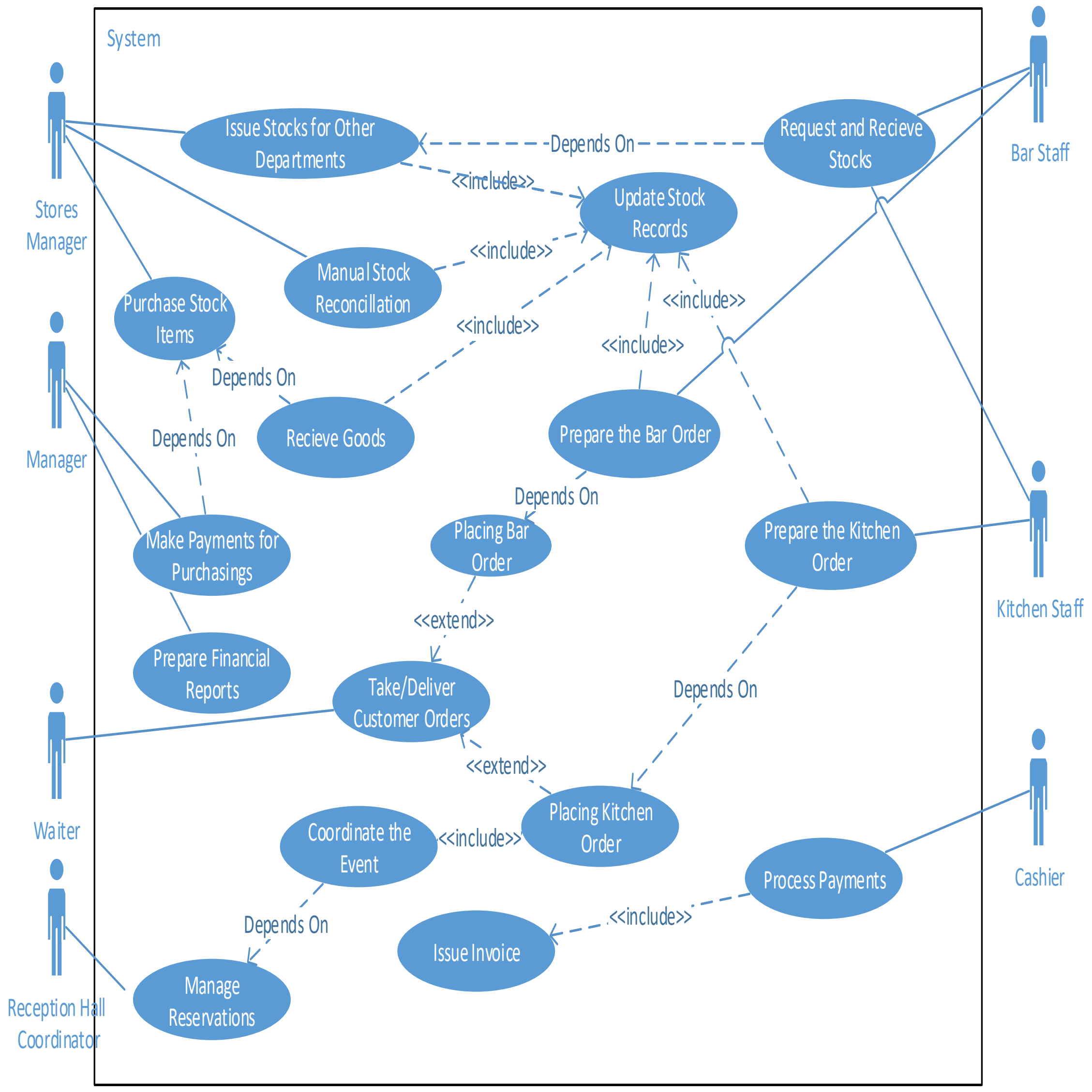


Figure 2-6: Use case diagram for entire system

Chapter 3 - Design

3.1 Design Strategy

3.1.1 Software Development Methodology

When developing a system, a selection of proper development methodology is essential to ensure the quality of the product and to effectively use the resources.

There are several development methodologies

Waterfall Development Model

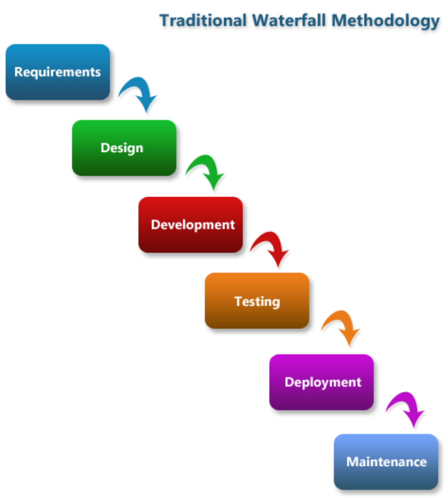


Figure 3-1: Waterfall Model

This methodology can be considered as the most basic software development methodologies. In this the whole software development project is divided in to typically six phases which are Requirement Analysis, Design, Development, Testing, Deployment and Maintenance. This methodology Emphasis on documents, time planning and budget planning.

In this methodology, it is not possible to deliver a tangible product to the customer until end of the project as this methodology does not allow customer to interact after the requirement analysis phase. Tight control over the tasks has to be maintained as this methodology may easily lead the project to fail.

Agile Development Methodologies

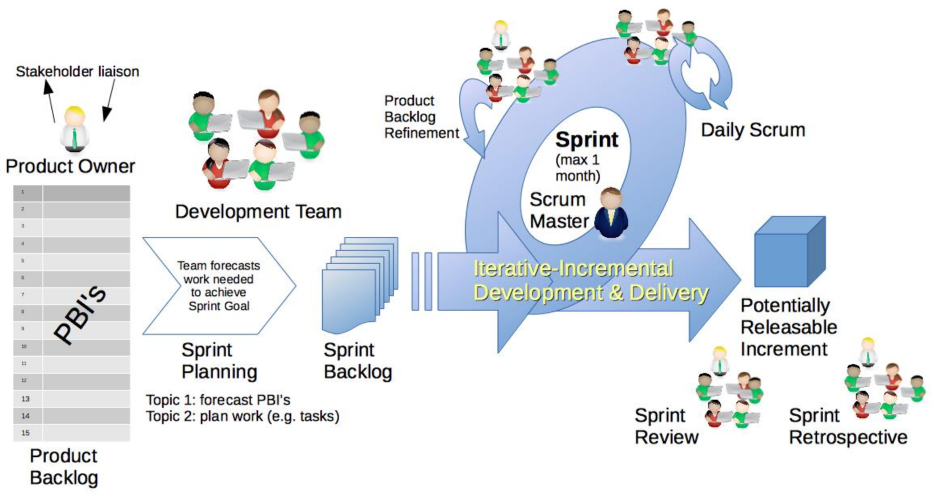


Figure 3-2: Agile Development - Scrum

Agile development is a software development methodology in which, requirements and the software solutions progress through sprints.

There are several agile development methodologies such as Kanban, Dynamic systems Development Method (DSDM) and Scrum which is the mostly used methodology.

Agile teams are adaptive planning and evolutionary development. It encourages rapid and flexible response to change. This methodology assures the early delivery of product and continuous improvements are encouraged.

To make this methodology successful, collaborative work of several cross-functional teams is obligatory.

Rapid application development

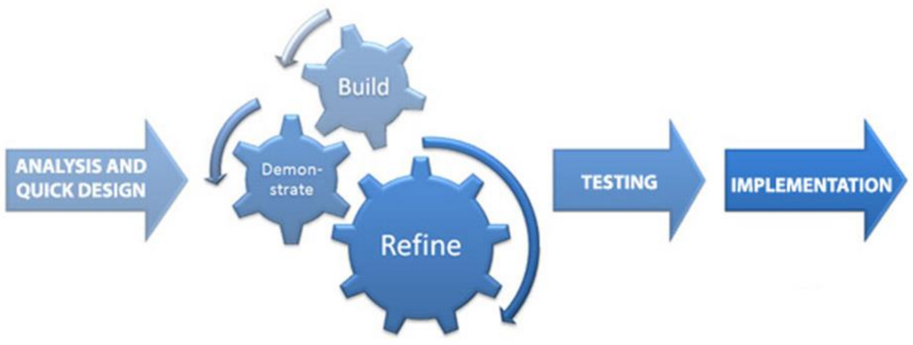


Figure 3-3: Rapid Application Development

Rapid Application Development also known as James Martin's approach to rapid development, is widely used among the smaller and mid-scale software development projects as this methodology facilitates high quality developments to be delivered fast and cost effectively.

This methodology depends on iterative development concept where involvement of users is offered in analyze and test the each iteration. Therefore this methodology does not require highly experienced development staff like in waterfall methodology to well understand the user requirements.

For this project RAD have been chosen as the best suitable development methodology by considering the capacity of development, project scope and nature of requirements.

3.1.2 Development Platform

When developing software, it is essential to identify the development platform, before defining the architecture and the development technologies. Depending on development platform, the capabilities of the system may fluctuate.

Standalone PC Application

This type of software applications have been from the beginning of the software era before any other application type. Rather than the resident computer hardware, software and configurations, mostly no any other dependency effected for this kind of applications. Higher control over the hardware is core advantage in scenarios like manufacturing equipment handling.

These applications are been abandoned presently due to lack of mobility and difficulty of maintenance. But in some cases, these applications are yet more suitable.

Mobile Applications

Mobile technologies have been evolving rapidly from a device which only facilitate for making calls, to a device which powered by higher computing capabilities and other modern technologies such as super-fast internet, interactive maps, Global Positioning System (GPS) and High resolution camera.

Mobile applications developers may use one or more of these capabilities and conglomerate them creatively to deliver new level of services. There are several mobile platforms popularly used and Apple IOS, Android and Windows Mobile are some of them.

In order to implement a multi user application on mobile platform, it is needed to host a web server as the central communication point.

Web Based Applications

Web based applications have been in the industry for few decades now and the technologies have been evolving rapidly to ensure security, user friendliness, ease of development and higher performances of the system.

With all the capabilities of standalone applications, these application support mobility in highest level. Same web application can be accessed from almost all devices such as computers, mobile tabs and mobile phones because of Responsive user interface design technology. These applications are compatible with most PC and Mobile operating systems which is identified as a challenge in both mobile applications and standalone applications.

According to requirement of Nimansala Restaurant, Web Based Application have been selected as most suitable approach as both other approaches are either incompetent or time consuming and costly.

3.2 Engineering Strategy

3.2.1 Software Architectural Pattern

Client–Server model

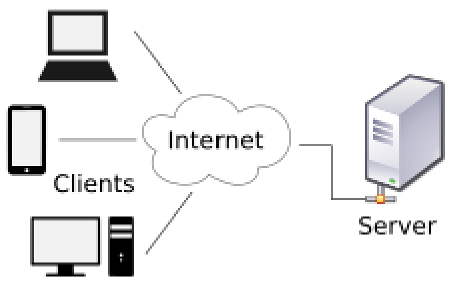


Figure 3-4: Client Server Architecture

This architecture isolates the software system into two portions client and server, in which the client makes requests to the server. Databases and Business logic resides in server in most scenarios while the presentation logic and other data manipulations are accommodated in client application.

Multilayered architecture

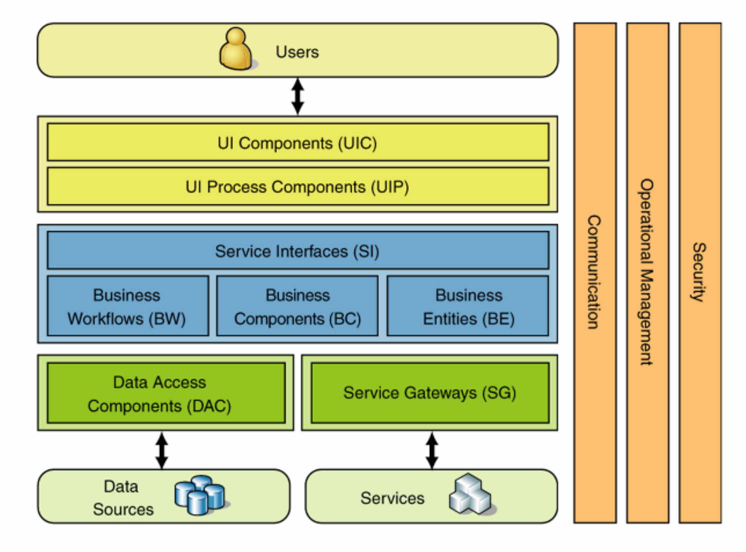


Figure 3-5: Multilayered Architecture

In Multilayered architecture, related functionalities are grouped and identified as distinct layers. Interactions among the layers are clear and loosely coupled. This layering benefits a great level of flexibility and maintainability.

Model-View-Controller (MVC)

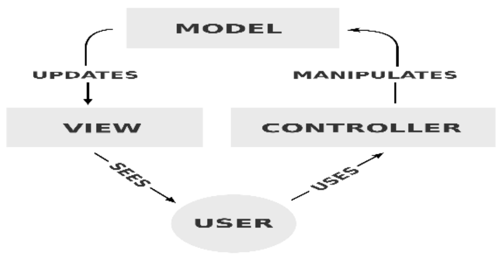


Figure 3-6: MVC Architecture

This architecture is widely used in modern software systems as this supports object oriented concepts well. This can be considered as inherited from both above architectures. Most of modern technologies such as software development frameworks are based on this. This architecture supports enhancement of any component without affecting the rest.

This architecture has been selected for this project as it is simple architecture which supports object oriented development as well.

3.2.2 Development Strategy

As the architecture, platform and architectural pattern have been selected, it was decided to develop the software system from the scratch as open source software's may not comply all these aspects.

3.3 Class Diagram

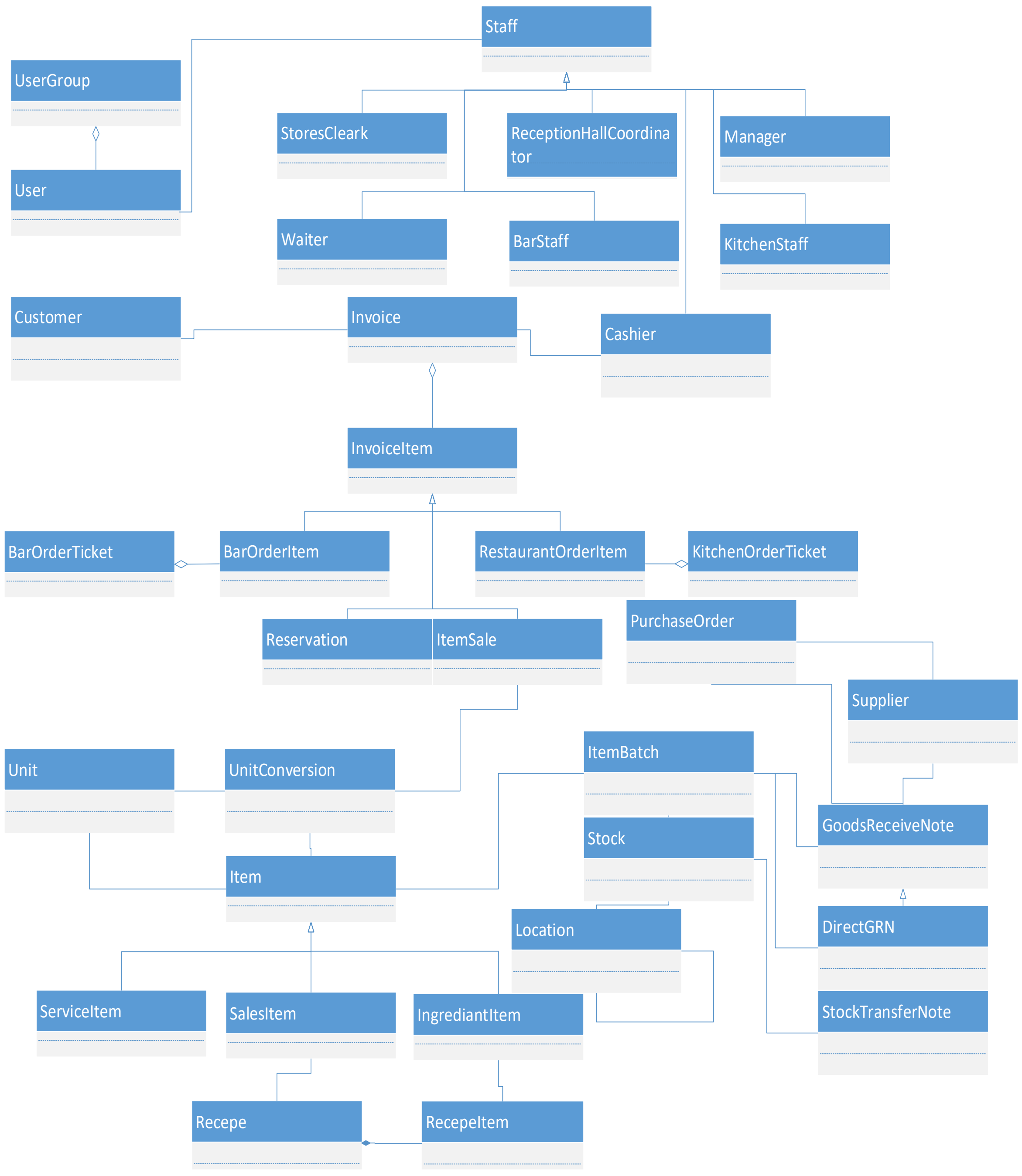


Figure 3-7: Class Diagram

3.4 Entity Relationship Diagram

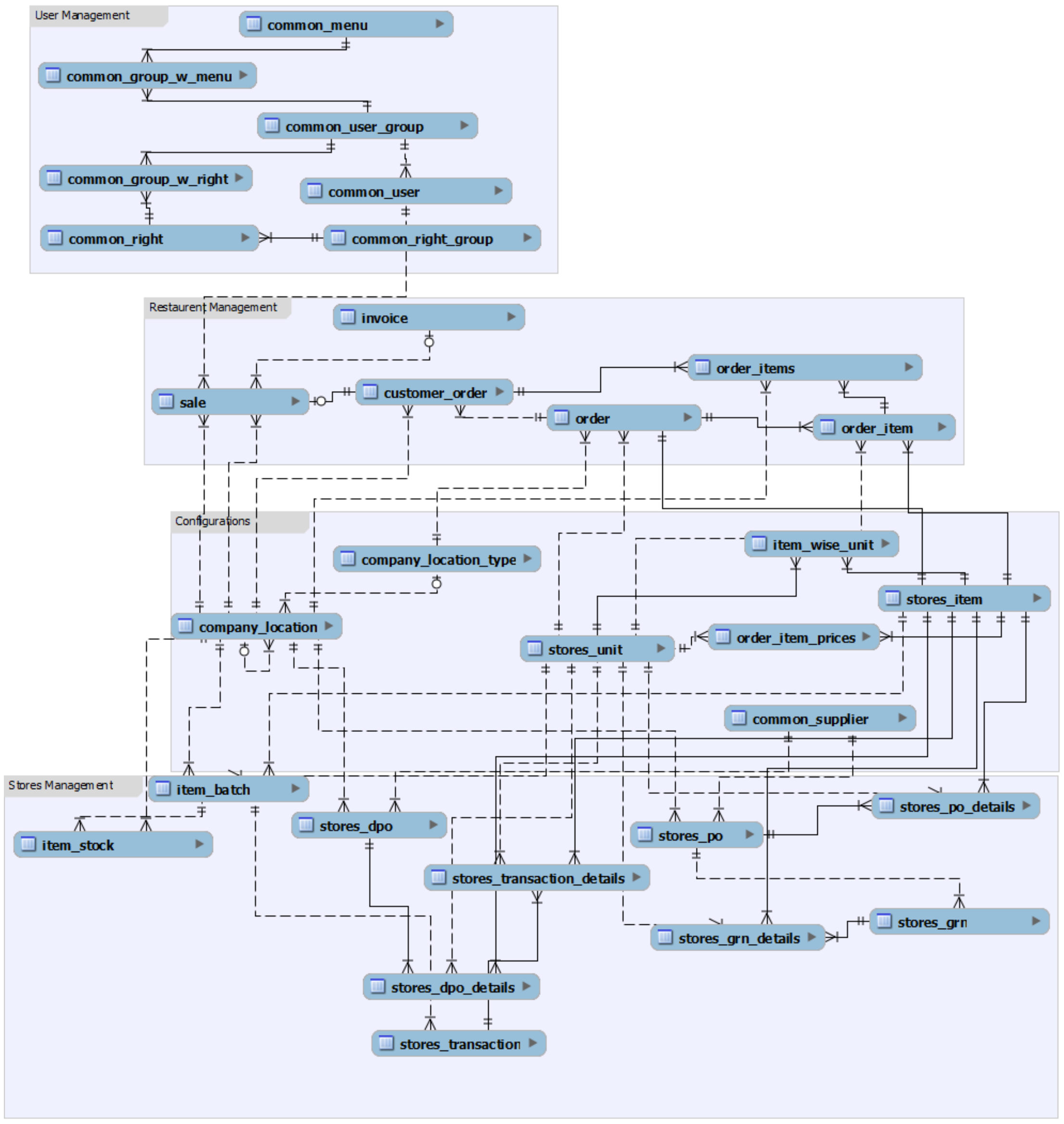


Figure 3-8: EER Diagram

3.5 User Interface Designing

3.5.1 Login form

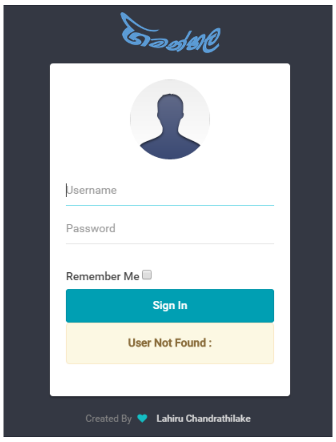
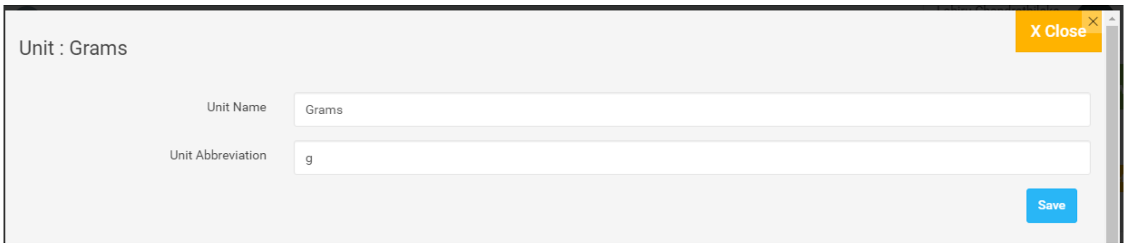


Figure 3-9: Login Form UI

3.5.2 Simple Master File

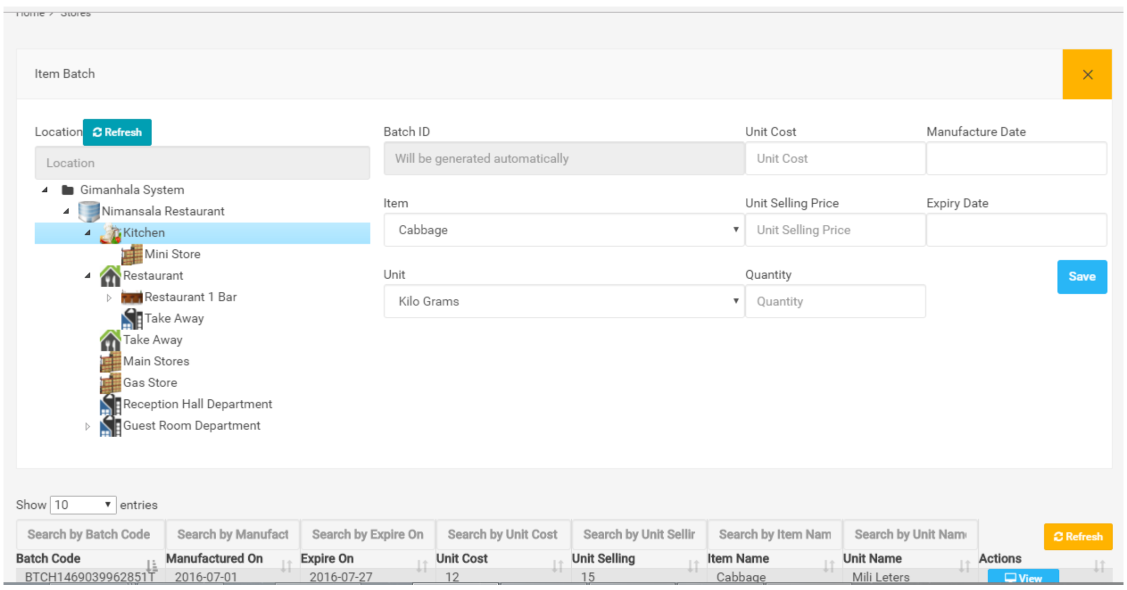

Unit : Grams

Unit Name: Grams

Unit Abbreviation: g

Figure 3-10: Simple Master File UI

3.5.3 Semi Complex Master File



Batch ID: Will be generated automatically

Unit Cost: Unit Cost

Manufacture Date:

Item: Cabbage

Unit Selling Price: Unit Selling Price

Expiry Date:

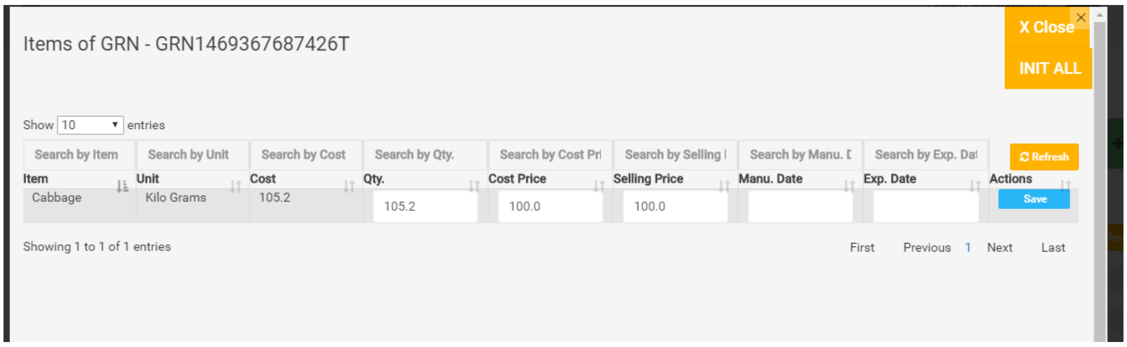
Unit: Kilo Grams

Quantity: Quantity

Batch Code	Manufactured On	Expire On	Unit Cost	Unit Selling	Item Name	Unit Name	Actions
BTCH1469039962851T	2016-07-01	2016-07-27	12	15	Cabbage	Mili Leters	View

Figure 3-11:Semi Complex Master File UI

3.5.4 Complex Transaction Form


Items of GRN - GRN1469367687426T

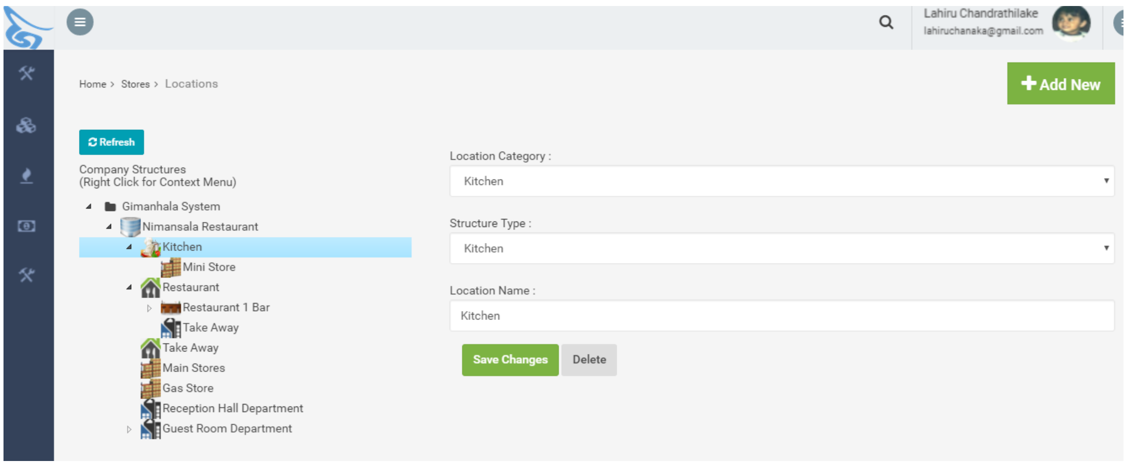
Show 10 entries

Item	Unit	Cost	Qty.	Cost Price	Selling Price	Manu. Date	Exp. Date	Actions
Search by Item	Search by Unit	Search by Cost	Search by Qty.	Search by Cost Pri	Search by Selling I	Search by Manu. D	Search by Exp. Dat	Refresh
Cabbage	Kilo Grams	105.2	105.2	100.0	100.0			Save

Showing 1 to 1 of 1 entries

Figure 3-12: Complex Transaction Form UI

3.5.5 Master File with Tree Structure



Company Structures
(Right Click for Context Menu)

- Gimanhala System
  - Nimansala Restaurant
    - Kitchen
      - Mini Store
    - Restaurant
      - Restaurant 1 Bar
      - Take Away
    - Take Away
    - Main Stores
    - Gas Store
    - Reception Hall Department
    - Guest Room Department

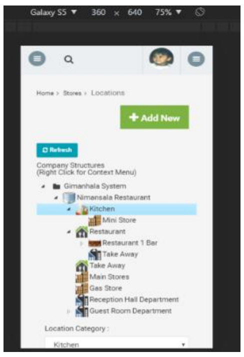
Location Category : Kitchen

Structure Type : Kitchen

Location Name : Kitchen

Figure 3-13: Master File with Tree Structure UI

3.5.6 Mobile Responsiveness



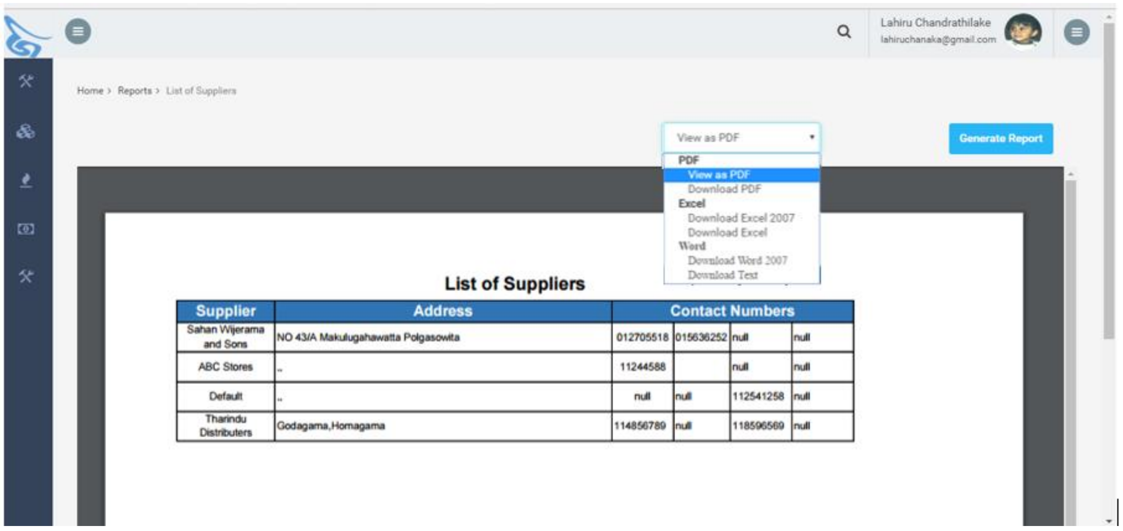
Company Structures
(Right Click for Context Menu)

- Gimanhala System
  - Nimansala Restaurant
    - Kitchen
      - Mini Store
    - Restaurant
      - Restaurant 1 Bar
      - Take Away
    - Take Away
    - Main Stores
    - Gas Store
    - Reception Hall Department
    - Guest Room Department

Location Category : Kitchen

Figure 3-14: Responsive UI

3.5.7 Report Viewer



List of Suppliers

Supplier	Address	Contact Numbers			
Sahan Wijerama and Sons	NO 43/A Makulugahawatta Polgasowita	012705518	015636252	null	null
ABC Stores	..	11244588		null	null
Default	..	null	null	112541258	null
Tharindu Distributers	Godagama,Homagama	114856789	null	118596569	null

Figure 3-15: Report Viewer UI

s

Chapter 4 - Implementation

4.1 Introduction

As the system has been developed using prototypes, implementation stage has been started before completing the analysis phase. After the completion of design stage, implementation has been accelerated and most of the back-end programming has been done this duration.

In order to make the system look attractive and user friendly, some re-usable front end libraries have been used including bootstrap and JQuery. This has significantly reduced the development time of the system. as it was decided to base this system on MVC architecture, JSON technology have been used in order to bring the MVC touch to the front end as well. Most of functionalities are based on AJAX, which allowed the system to minimize the amount of data transferred back and forth.

4.2 Implementation Environment

The implementation environment has to be powerful enough to quickly compile the source code in order to speed up the development process. The environment used for this project had following configurations.

Hardware Specification

Processor	: Intel Core i3 2.30 GHz
RAM	: 8GB
Hard Disk	: 500GB

Software Specification

Operating System	: Microsoft Windows 8.0
Virtual Machines	: JAVA 1.8 VM
Database Servers	: MySQL 5.6
Web Servers	: Apache Tomcat 7.0

4.3 Technologies and Tools Used

JAVA

Java is a very powerful matured programming language which has the ability to run in different platforms such as web, standalone, embedded and mobile. Java is a lightweight programming language which encourages the object oriented programming.

JAVA EE

Java EE is an implementation of JAVA, specially designed to have HTTP and web system capabilities.

My SQL

My SQL is the most popular free and open source database management system available in the industry. Though MySQL is equipped with the basic features of a typical database management system, still is capable enough for many general business solutions.

STS (Spring Tool Suite) IDE

This is a modern IDE which has many build in language capabilities. This IDE is based on eclipse platform and can be considered as a much user friendly environment to develop JAVA based systems.

My SQL Work Bench

My SQL workbench is the official GUI tool to manage MY SQL databases. It consists of the diagraming capabilities as well. The table structure has been designed by using this feature and ability to reverse and forward engineer the design, is the major advantage of this tool.

JAVA Script

Java Scripts client side is a web programming language which plays a major role in modern systems to make them look better and perform efficiently. As java scripts are amazingly light weight and runs on web browser, JavaScript are used to perform most of the calculations in order to reduce the server load.

JQuery

JQuery is a java script based framework which made java script capabilities easy to access. There are many pre written methods and components which can be used to minimized the programming effort while extending the capabilities of the developed software.

Bootstrap

Bootstrap is a CSS and Java Script based front end framework mainly focused on User interface and user experience management. Like in JQuery, there are many bootstrap based components available freely so that they can be used to improve the user experience of the system.

AJAX

AJAX stands for asynchronous java script and xml. It enables the web pages to communicate with the servers in background and make the necessary changed in the interfaces depending on the data received. This makes the information secure and emphasizes the efficiency by reducing the data transferred.

JSON and GSON

JSON is a data transfer standard and used to transfer data from backend to front end in an organized manner. GSON is a java library developed by Google which has the capability of converting the Java Objects in to JSON Strings as it is. This enables the front end java script to access the Objects, same way like in Java End.

Microsoft Power Point and Picture Manager

Though these are not much capable graphic designing tools, capable enough for some decent graphics be created. Following are some graphics created using these two tools.

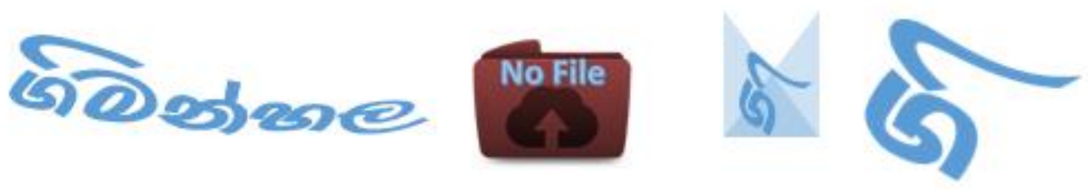


Figure 4-1: Graphics Created

4.4 Major Code Segments

Almost all the types of programming have been used in this system. These can be categorized mainly as Front end, Server Side and DB programming.

4.4.1 Front End Programming

Most of the functionalities have been handled in client end to reduce the load of the Server. For an Example Data Table Generation have been developed as a re-useable java script method which add all the capabilities in to all grids in the system. All data grids of the system have pagination, and real time search facilities.

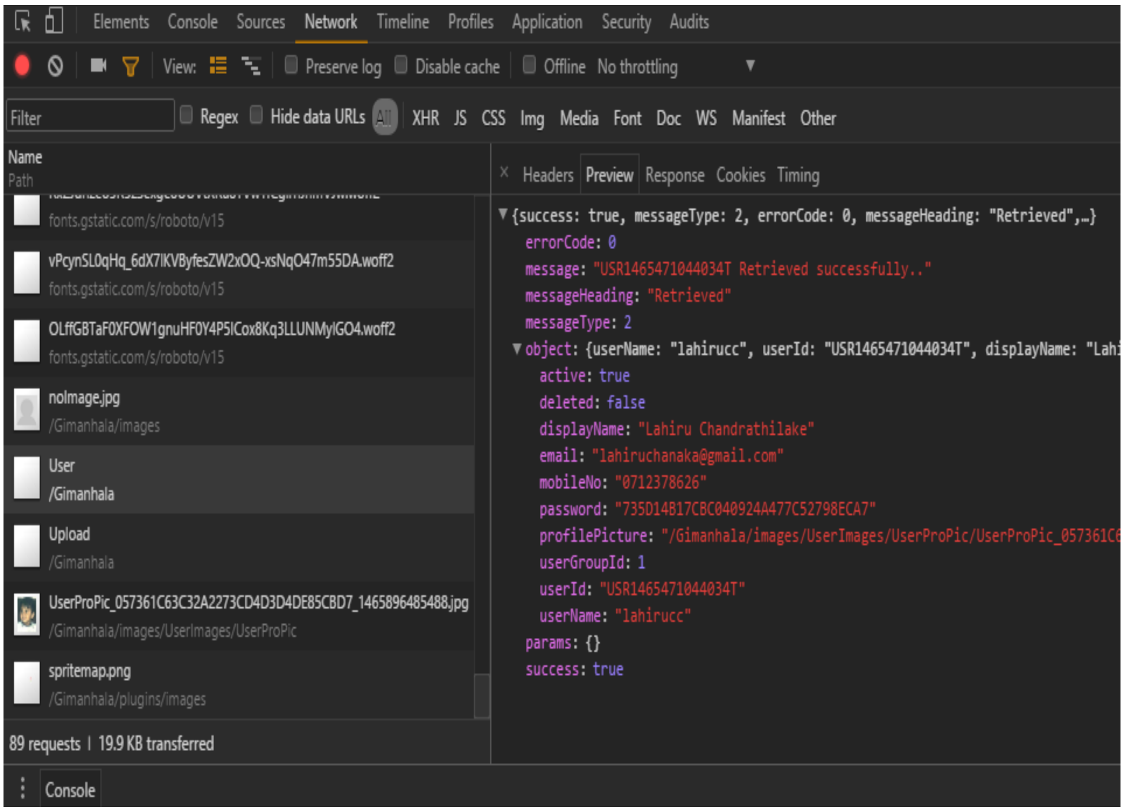
```
// Data table

function addColumns(colList,dataTableId) {
	refLoaded = false;
	var columns = '<thead><tr>';
	for (var i = 0; i < colList.length; i++) {
		columns += '<th style="width:auto;padding: 0px;margin:
0px"><b>'
				+ colList[i] + '</b></th>';
	}
	columns += '</tr><tr>';
	for (var i = 0; i < colList.length; i++) {
		columns += '<td style="width:auto;padding: 0px;margin:
0px"><b>'
				+ colList[i] + '</b></td>';
	}
	columns += '</tr></thead><tbody></tbody>';
	$(dataTableId).html(columns);
}

function loadTable(PageDetails) {
	var dataId = PageDetails[0];
	var dataTableId = '#' + dataId;
	var sAjaxSource = PageDetails[1];
	var colList = PageDetails[2];
	refLoaded = false;
	addColumns(colList,dataTableId);
	var  DataTable = $(dataTableId).dataTable({
		bSort : true,
		"sPaginationType" : "full_numbers",
		"bProcessing" : true,
		"bAutoWidth" : false,
		"bServerSide" : true,
		"sAjaxSource" : sAjaxSource
	});
	// Setup - add a text input to each footer cell
	$(dataTableId + ' thead th')
			.each(
				function() {
					var title = $(dataTableId + ' thead
th').eq(
      $(this).index()).text();
                                      if (title === "Image" || title ===
"Photo"
                                            || title === "Photos" || title
=== "Icon"|| title === "Images"|| title === "File" || title === "Related"||
title === "") {
                                            $(this).html("");
                                      } else if (title === "Actions") {
                                            $(this).html('<a type="button"
class="btn btn-warning btn-sm" style="float:right"
onclick="reloadTable('+dataId+')"><i class="fa fa-refresh"></i>
Refresh</a>');
                                      } else {
                                            $(this)
                                      .html(
                                      '<input type="text" class="form-
control" placeholder="Search by '+ title'" id ='+ dataId+ $(this).index()
             + ' onkeyup="tableSearch(event,\''+ dataTableId+
$(this).index() + '\','+ $(this).index() + ','+dataId+')" />');
                                      }
                                });
      var html = '';
      $(dataTableId + '_filter').html(html);
      return DataTable;
}

function tableSearch(event, id, index,DataTable) {
      refLoaded = false;
      DataTable.fnFilter($(id).val(), index);
}

function reloadTable(DataTable) {
      refLoaded = false;
      DataTable.fnDraw(true);
      // DataTable.fnStandingRedraw();
}
```

In order to validate and submit forms, flowing code segment have been re used.

```
function validateAndSubmit(form, url, successFunction, failedFunction) {
      $(form).data('bootstrapValidator').validate();
      var isValid = $(form).data('bootstrapValidator').isValid();
      if (isValid) {
            $.post(url,$(form).serializeArray()).done(function(data) {
                  console.log(data);
                  successFunction(data);
            });
      }
}
```

All responses have been unified by wrapping the requested information inside a common object called server response.

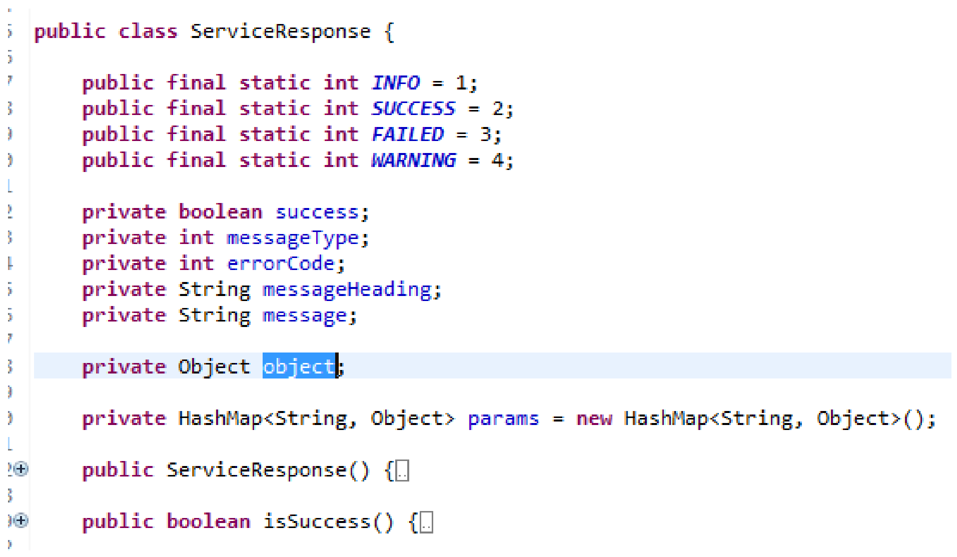
The JSON generated from this, carries Meta information about the request. Any error messages, status and the requested data are inside this object.



```
Name
Path
fonts.gstatic.com/s/roboto/v15
vPcynSL0qHq_6dX7IKVByfesZW2xOQ-xsNqO47m55DA.woff2
fonts.gstatic.com/s/roboto/v15
OLffGBTaF0XFOW1gnuHF0Y4P5ICox8Kq3LLUNMyIGO4.woff2
fonts.gstatic.com/s/roboto/v15
noImage.jpg
/Gimanhala/images
User
/Gimanhala
Upload
/Gimanhala
UserProPic_057361C63C32A2273CD4D3D4DE85CBD7_1465896485488.jpg
/Gimanhala/images/UserImages/UserProPic
spritemap.png
/Gimanhala/plugins/images
89 requests | 19.9 KB transferred
```

```
▼{success: true, messageType: 2, errorCode: 0, messageHeading: "Retrieved",…}
   errorCode: 0
   message: "USR1465471044034T Retrieved successfully.."
   messageHeading: "Retrieved"
   messageType: 2
 ▼object: {userName: "lahirucc", userId: "USR1465471044034T", displayName: "Lahi
     active: true
     deleted: false
     displayName: "Lahiru Chandrathilake"
     email: "lahiruchanaka@gmail.com"
     mobileNo: "0712378626"
     password: "735D14B17CBC040924A477C52798ECA7"
     profilePicture: "/Gimanhala/images/UserImages/UserProPic/UserProPic_057361C6
     userGroupId: 1
     userId: "USR1465471044034T"
     userName: "lahirucc"
   params: {}
   success: true
```

Figure 4-2: Preview of the Received JSON



```java
public class ServiceResponse {

    public final static int INFO = 1;
    public final static int SUCCESS = 2;
    public final static int FAILED = 3;
    public final static int WARNING = 4;

    private boolean success;
    private int messageType;
    private int errorCode;
    private String messageHeading;
    private String message;

    private Object object;

    private HashMap<String, Object> params = new HashMap<String, Object>();

    public ServiceResponse() {

    public boolean isSuccess() {
```

Figure 4-3: Service Response Class Definition

This response is interpreted in java script

```
function nodeSelected(node){
	resetForm();
	$.post("/Gimanhala/Location", {
		action : "single",
		id : companyDiv_cleanId(node.id)
	}).done(function(serverResponse) {
		if(!serverResponse.success){
			
	showMessage(serverResponse.messageType,serverResponse.messageHeading,
serverResponse.message);
		}else{
			
	$("#locationType").val(serverResponse.object.locationType.id);
			$("#category").val($('option:selected',
$("#locationType")).attr("category"));

	$("#locationType").val(serverResponse.object.locationType.id);
			$("#action").val("edit");
			$("#parent").val(serverResponse.object.parent);
			$("#key").val(serverResponse.object.id);
			$("#name").val(serverResponse.object.name);
			
		}
	});
}
```

4.4.2 Server Side Programming

At Server Side, Object Oriented Programming have been Implemented.

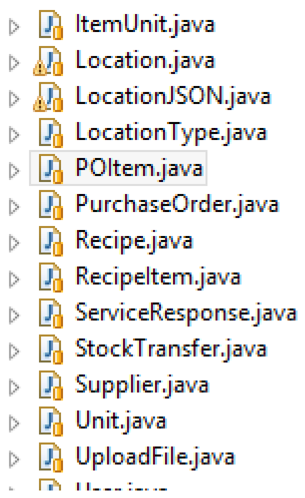

ItemUnit.java
Location.java
LocationJSON.java
LocationType.java
POItem.java
PurchaseOrder.java
Recipe.java
RecipeItem.java
ServiceResponse.java
StockTransfer.java
Supplier.java
Unit.java
UploadFile.java

Figure 4-4: Model Classes

In order to process Jasper Reports, a Common Platform have been created

```
protected void doPost(HttpServletRequest request, HttpServletResponse
response)
			throws ServletException, IOException {
		Map parameters = HTMLUtils.getAllParams(request);
		String userName= "unauth";
		boolean authFailed = true;
		try {
			User user = (User) 
request.getSession().getAttribute("user");
			if (user != null) {
				authFailed = false;
				userName= user.getUserName();
				
			}
		} catch (Exception ex) {
			ex.printStackTrace();
			authFailed = false;
		}
		
		if (authFailed) {
			String nextJSP = "/error.jsp?number=401";
			RequestDispatcher dispatcher = 
getServletContext().getRequestDispatcher(nextJSP);
			dispatcher.forward(request, response);
			return;
		}
		parameters.put("userName",userName);
		System.out.println(userName);
		fileName = parameters.get("fileName")+"  "+userName+" 
"+Format.TimetoText(new Date());
		try {
			String path = 
getServletContext().getRealPath("/reportfiles/" + 
parameters.get("reportName"));
			System.out.println(path);

			JasperPrint jasperPrint = 
ReportService.getJasperPrint(parameters, path);

			String outPut = "" + parameters.get("outPut");
			
			if (outPut.equals("PDFV")) {
				givePDFOutPut(parameters.get("displayName"), 
response, jasperPrint);
			}else if (outPut.equals("PDFD")) {
				givePDFDOutPut(parameters.get("displayName"), 
response, jasperPrint);
			}else if (outPut.equals("EXCELXD")) {
				giveExcelXOutPut(parameters.get("displayName"), 
response, jasperPrint);
			}else if (outPut.equals("EXCELD")) {
				giveExcelOutPut(parameters.get("displayName"), 
response, jasperPrint);
			}else if (outPut.equals("TEXT")) {
				giveTextOutPut(parameters.get("displayName"), 
response, jasperPrint);
			}else if (outPut.equals("WORD")) {
				giveWordOutPut(parameters.get("displayName"), 
response, jasperPrint);
			}
			
			// response.setHeader("Content-Disposition","inline; 
filename=Here
```

```
                    // is the Amazing PDF");

             } catch (Exception e) {
                    e.printStackTrace();
                    String nextJSP = "/error.jsp?number=505";
                    RequestDispatcher dispatcher =
getServletContext().getRequestDispatcher(nextJSP);
                    dispatcher.forward(request, response);
             }

       }
```

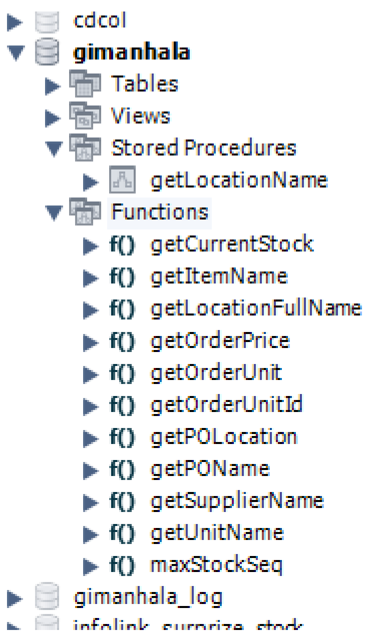
HTML form for Report parameters

```
<div class="row">
                  <div class="col-md-12">
                    <form id="form" action="/Gimanhala/Reports" class="form-
horizontal" target="reportViewer" method="post" onended="iframeLoaded()">


                                 <input id="reportName" type="hidden"
name="reportName" value="SupplierList.jrxml">
                                 <input id="fileName" type="hidden"
name="fileName" value="Supplier List">

                          <div class="col-md-12" >
                                 <div class="col-md-6" ></div>
                                 <div class="col-md-3" >
                                        <div class="form-group" style="float:
right;">
                                               <select id="outPut" name
="outPut" class="form-control" placeholder="Select Type" >
                                                      <optgroup label="PDF">
                                                             <option
value="PDFV" selected><i class="fa fa-file-pdf"></i> View as PDF</option>
                                                             <option
value="PDFD" ><i class="fa fa-file-pdf"></i> Download PDF </option>
                                                      </optgroup>
                                                      <optgroup
label="Excel">
                                                             <option
value="EXCELXD"><i class="fa fa-file-excel"></i> Download Excel
2007</option>
                                                             <option
value="EXCELD"><i class="fa fa-file-excel"></i> Download Excel</option>
                                                      </optgroup>
                                                      <optgroup label="Word"
class="fa fa-file-pdf-o">
                                                             <option
value="WORD"><i class="fa fa-file-pdf-o"></i> Download Word 2007</option>
                                                             <option
value="TEXT"><i class="fa fa-file-pdf"></i> Download Text</option>
                                                      </optgroup>


                                               </select>

                                        </div>
                                  </div>
                                  <div class="col-md-3" >
                                        <button id="btnReset" class="btn btn-
success" type="reset" onclick="resetForm()" style="float: right;
display:none;"><i class="fa fa-refresh"></i></button>
                                        <button id="btnShowReport" class="btn
btn-info" type="submit" onclick="submitForm()" style="float:
right;">Generate Report</button>
                                  </div>



                           </div>
                           </form>

                    </div>
```

Whatever the input boxes mentioned in this form will be directly mapped in to jasper report as a parameter. So that reports can be hosted by considering only the jasper report and the html form. No middle codes are needed.

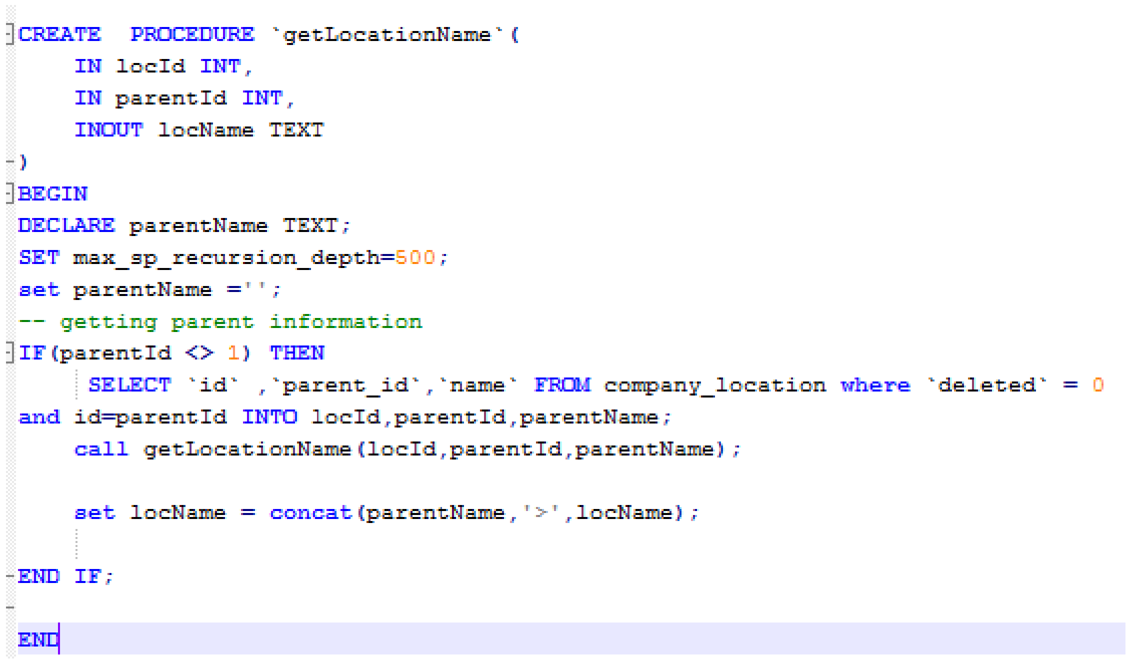
4.4.3 DB Programming

DB Program scripts have the capability of accessing the database quickly and manipulate quickly before sent to the application. This has also been used to reduce un-wanted effort of application.

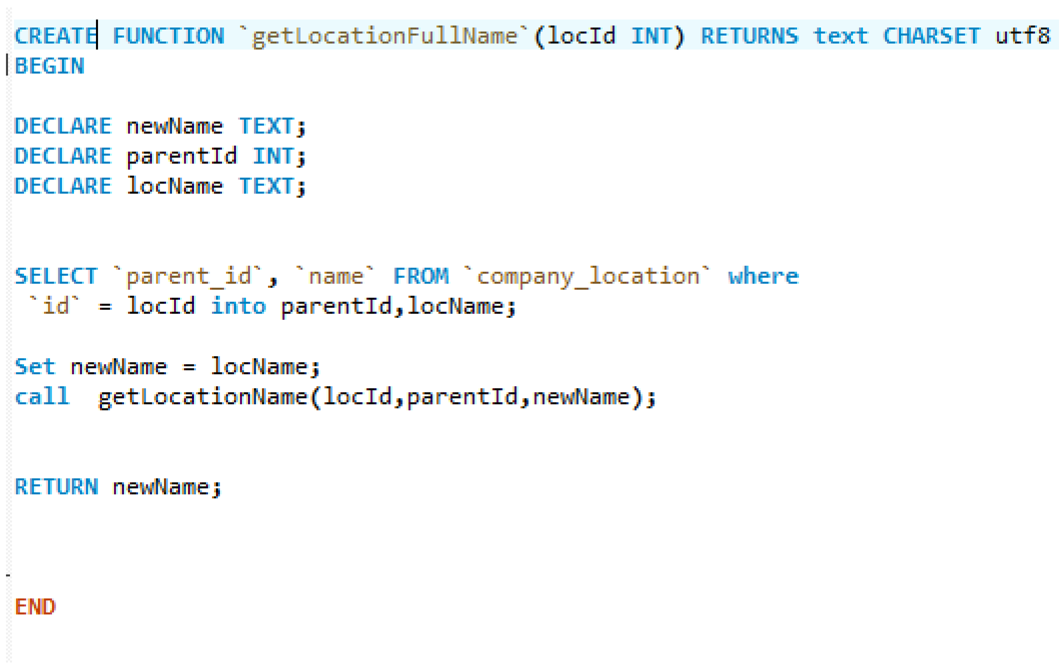


```
▶ cdcol
▼ gimanhala
    ▶ Tables
    ▶ Views
    ▼ Stored Procedures
        ▶ getLocationName
    ▼ Functions
        ▶ f() getCurrentStock
        ▶ f() getItemName
        ▶ f() getLocationFullName
        ▶ f() getOrderPrice
        ▶ f() getOrderUnit
        ▶ f() getOrderUnitId
        ▶ f() getPOLocation
        ▶ f() getPOName
        ▶ f() getSupplierName
        ▶ f() getUnitName
        ▶ f() maxStockSeq
▶ gimanhala_log
▶ infolink_surprize_stock
```

Figure 4-5: DB Programming Scripts



```
CREATE  PROCEDURE `getLocationName`(
    IN locId INT,
    IN parentId INT,
    INOUT locName TEXT
)
BEGIN
DECLARE parentName TEXT;
SET max_sp_recursion_depth=500;
set parentName ='';
-- getting parent information
IF(parentId <> 1) THEN
     SELECT `id` ,`parent_id`,`name` FROM company_location where `deleted` = 0
and id=parentId INTO locId,parentId,parentName;
    call getLocationName(locId,parentId,parentName);

    set locName = concat(parentName,'>',locName);

END IF;

END
```



```
CREATE FUNCTION `getLocationFullName`(locId INT) RETURNS text CHARSET utf8
BEGIN

DECLARE newName TEXT;
DECLARE parentId INT;
DECLARE locName TEXT;


SELECT `parent_id`, `name` FROM `company_location` where
 `id` = locId into parentId,locName;

Set newName = locName;
call  getLocationName(locId,parentId,newName);


RETURN newName;



END
```

Chapter 5 – Evaluation

5.1 Introduction

After developing a software system, it is needed to test that software system to make sure it is working as expected. Simply this is the definition of software system testing. But it covers a wider area in practical situations. System Testing can be classified under two different types. Those are Black box Testing and White box Testing.

5.1.2 White box Testing

In this mode of testing, the system tester is totally aware of the internal functionality and behavior of the system. In this kind of a testing it can be assured that the expected aspects of the system are fully working.

5.1.3 Black box Testing

In Black box testing, the tester should be or should pretend to be totally unaware of the internal functionality of the system. All possible inputs are tested on the system to make sure the system is not doing following things

- Accepting Invalid/Wrong/Erroneous Inputs from user.
- Reacting for unknown commands
- Produce wrong information
- Damage/ delete old data

When discussing about testing, there are some things which needs to be taken in to consideration.

Test Cases

Test cases are written to make sure all aspects of a system are tested. Usually in these cases it is included way of testing and expected result for each activity of a system

Unit Testing

When a system is loosely coupled and highly cohesive, components can be tested individually to make sure that they are working as expected. This enables the tester to focus on each component and test.

Integration Testing

When the Unit testing is completed, the inter connection of modules are tested under integration testing. This can clearly identify the confliction between modules and make sure the modules are communicating properly.

System Testing

Finally Whole system is considered as one unit and tested for overall functionality. This assures that the system is well inter-connected and working fine.

5.2 System Test Cases

5.2.1 Login Form Test Cases

Login Form			
Condition	Expected Result	Tested	Ok
Login with an Empty Username	Relevant Message and Return	Yes	Yes
Empty Password	Relevant Message and Return	Yes	Yes
Wrong Username	Relevant Message and Return	Yes	Yes
SQL Injection Try	Relevant Message and Return	Yes	Yes
Wrong Password	Relevant Message and Return	Yes	Yes
Unknown Errors	Relevant Message and Return	Yes	Yes
Correct Credentials and Inactive User	Relevant Message and Return	Yes	Yes
Correct Credentials and Active User	Relevant Message Successful Login	Yes	Yes
Hit Enter Key in User Name	Focus to Password	Yes	Yes
Hit Enter Key in Password	Login	Yes	Yes
Click Login Button	Login	Yes	Yes
Successful Login	Enable Relevant Links	Yes	Yes

Table 5-1: Login Form Test Cases

5.2.2 User Groups Test Cases

User Groups			
Condition	Expected Result	Tested	Ok
Insert/Update Action Without Group Name	Relevant Message and Return	Yes	Yes
Insert/Update Action With a Valid Group Name	Continue Action	Yes	Yes
Insert/Update Action With an Existing Group Name	Relevant Message and Return	Yes	Yes
Insert/Update DB Error	Relevant Message and Return	Yes	Yes
User Groups - Links to Other Forms			
Condition	Expected Result	Tested	Ok
Not Having View Rights of Access Control Form	Disable Access Control Button	Yes	Yes
Having View Rights of Access Control Form	Enable Access Control Button	Yes	Yes

Table 5-2 : User Groups Test Cases

5.2.3 User Management Test Cases

User Management			
Condition	Expected Result	Tested	Ok
Insert/Update Action Without User Name	Relevant Message and Return	yes	yes
Insert/Update Action With a Valid User Name	Continue Action	yes	yes
Insert/Update Action With An Existing User Name	Relevant Message and Return	yes	yes
Insert/Update DB Error	Relevant Message and Return	yes	yes
Insert without Typing Password	Relevant Message and Return	yes	yes
Insert without Retyping Password	Relevant Message and Return	yes	yes
Password and Retyped Password Not Matching	Relevant Message and Return	yes	yes
Entering User Details Successfully	Continue Saving Action	yes	yes
Update Without entering Password	Update only the other details	yes	yes
Update with Password	Validate Password and Save	yes	yes
Attempting to Activate/Deactivate Without Selecting User	Relevant Message and Return	yes	yes
Attempting to Activate/Deactivate After Selecting User	Continue Activating/Deactivation Process	yes	yes
Attempting to View Personal Details Without Selecting User	Relevant Message and Return	yes	yes
Attempting to View Personal Details After Selecting User	Continue Activating/Deactivation Process	yes	yes
Attempting to Control Access Without Selecting User	Relevant Message and Return	yes	yes
Attempting to Control Access After Selecting User	Continue Activating/Deactivation Process	yes	yes
Showing Details	Password Fields Empty	yes	yes
Showing Details	Show Data in Relevant Fields	yes	yes
Showing Details	Select User Group in Combo	yes	yes
Attempting to Insert/Update without selecting user group	Relevant Message and Return	yes	yes
Attempting to Insert/Update with a Valid user group	Continue Saving Action	yes	yes
Click on Refresh Group Button	Reload User Groups Combo	yes	yes
User Management - Special Activities			
Condition	Expected Result	Tested	Ok
Not Having Rights to Activate User	Disable Activation Button	Yes	Yes
Having Rights to Activate User	Enable Activation Button	Yes	Yes

Table 5-3: User Management Test Case

Chapter 6 - Conclusion

6.1 Introduction

This project has been a great opportunity to gain the live experience of developing a software system to an actual client with requirements of solutions for real business problems. In this chapter it is expected to discuss the experience of the completed phase, the plan for future enhancements and developments and the lessons learned.

Restaurant and Inventory management systems are two types of systems that have been IT industry for decades. These systems have powered many food and hospitality suppliers and especially most of the organizations are equipped with an inventory control system. It is observed that most systems available in the market are either highly expensive or are not having the flexibility of incorporating these two main functionalities with various other functionalities such as accounts, human capital management etc.

As the first phase, the mission critical functionalities of Nimansala Restaurant have been addressed and the second phase which include the enhancements and functionalities that have less priority, have been started. During the second phase, reception hall management, room reservation functionality and accounts functionalities are expected to be addressed. Further, many none functional requirements are expected to be incorporated with the system within the second phase.

6.2 Lessons Learnt

As the Software systems are intangible, it is not easy to make the client imagine the actual system and suggest the needed changes. In this project, prototype based development approach has been incorporated, and it helped the customer to visualize the system and map it in to own business requirement. There have been some tiny points that turned the whole system design around by 360 degrees. The relationships between the classes have been redefined by some requirement changes.

It was identified as more effective to build the basic data structure and relationships first and add the additional features later. So that the flow of data is much clear and the enhancements have specific scope and goals.

It is important to identify the whole requirement and separate into phases depending on the priority so that each phase has a usable delivery. This helps to control the project and smoothly build the solution.

The system requirement identification can be simplified by using correct OOAD (object oriented analysis and designing) techniques, standard conventions and by making the modules highly cohesive and loosely coupled.

6.3 Critical Assessment of the Project

This system has been built more tailor made for the customer requirement while maintaining the usability for general scenarios as well. Initially two existing similar systems, Floreant POS and Samba POS have been identified to compete with this solution.

This system has been built with more capabilities than these systems, though it has been identified that there should be much improvements user experience wise when compared to these two commercial systems. Samba POS provided more graphical table arrangement and though the backend, data structure and UI of Gimanhala has way more capability of handling scenarios, it can be better presented to the user so that users get more understanding about the location arrangement.

Location wise stock management was a key feature of Gimanhala system which was not been addressed in other two systems. The reports of the Gimanhala came out to be eye catching and easy to understand and that can be considered as a definite advantage of this system from the management users point of view.

Though there are no P0 and P1 level bugs found in Gimanhala System, It need to be improved quality wise. Especially, the scope of validation should be improved compared to two other systems.

6.4 Future Enhancements (Phase 2)

- Reception Hall and Room Reservations
    - Room Reservation
    - Hall Reservation
    - Room Reservation related order processing
    - Online Reservation
- Sales and Order Management
    - Mobile Application for Waiters/Stuarts
    - Online Orders for take away (home delivery feature will be added in following phases)
- Finance Management
    - Accounts
    - Inter Account Transfers
    - Miscellanies Invoices
    - Cash Vouchers
    - Cash Drawer Management
- Common Functionalities
    - SMS Alerts
    - Advanced Security
    - Mobile Accessibility

References

[1] Wikipedia, "First-come, first-served," [Online]. Available: https://en.wikipedia.org/wiki/First-come,_first-served.

[2] Floreant POS, "Floreant POS - Screen Shots," [Online]. Available: http://www.floreantpos.org/screenshots/.

[3] SambaPOS, "Free Restaurant POS Software - SambaPOS," SambaPOS, [Online]. Available: https://sambapos.com/.

Appendix A - System Documentation

In this chapter, the technical information about the system is discussed. The procedures to Set-Up the Development Environment and Production Environment are explained.

This project is mainly based on Java EE technologies and other popular front end web technologies like CSS, JQuery and Boostrap etc. STS IDE has been used as the development environment. In order to modify the source, the development environment needs to be set up properly.

Development Environment Specification

Processor: Intel Core i3 2.30 GHz

RAM: 8GB

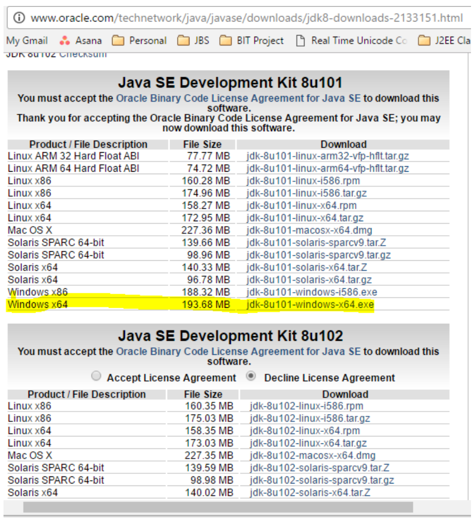
Hard Disk: 500GB

Operating System: Microsoft Windows 8.0

Setting up the Development Environment

In order to modify and compile the code, Java development kit (JDK) has to be installed and configured. For running the system in production environment, it is sufficient to install Java run time environment (JRE) and it is advised to avoid installing JDK on production environment to avoid any security issues.

1. Download and Install JDK and/or JRE.



Java SE Development Kit 8u101

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.77 MB	jdk-8u101-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.72 MB	jdk-8u101-linux-arm64-vfp-hflt.tar.gz
Linux x86	160.28 MB	jdk-8u101-linux-i586.rpm
Linux x86	174.96 MB	jdk-8u101-linux-i586.tar.gz
Linux x64	158.27 MB	jdk-8u101-linux-x64.rpm
Linux x64	172.95 MB	jdk-8u101-linux-x64.tar.gz
Mac OS X	227.36 MB	jdk-8u101-macosx-x64.dmg
Solaris SPARC 64-bit	139.66 MB	jdk-8u101-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	98.96 MB	jdk-8u101-solaris-sparcv9.tar.gz
Solaris x64	140.33 MB	jdk-8u101-solaris-x64.tar.Z
Solaris x64	96.78 MB	jdk-8u101-solaris-x64.tar.gz
Windows x86	188.32 MB	jdk-8u101-windows-i586.exe
Windows x64	193.68 MB	jdk-8u101-windows-x64.exe

Java SE Development Kit 8u102

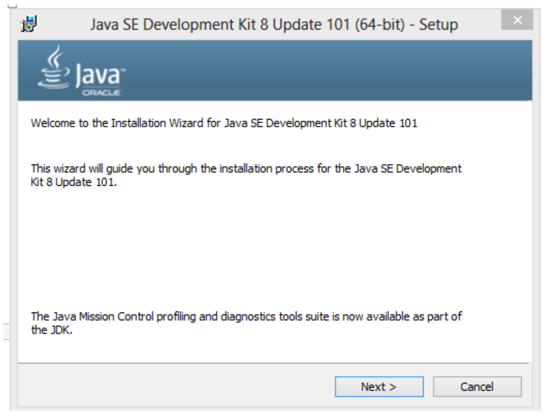
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Accept License Agreement [✓ Decline License Agreement]

Product / File Description	File Size	Download
Linux x86	160.35 MB	jdk-8u102-linux-i586.rpm
Linux x86	175.03 MB	jdk-8u102-linux-i586.tar.gz
Linux x64	158.35 MB	jdk-8u102-linux-x64.rpm
Linux x64	173.03 MB	jdk-8u102-linux-x64.tar.gz
Mac OS X	227.35 MB	jdk-8u102-macosx-x64.dmg
Solaris SPARC 64-bit	139.59 MB	jdk-8u102-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	98.98 MB	jdk-8u102-solaris-sparcv9.tar.gz
Solaris x64	140.02 MB	jdk-8u102-solaris-x64.tar.Z

Figure A-1: JDK Download

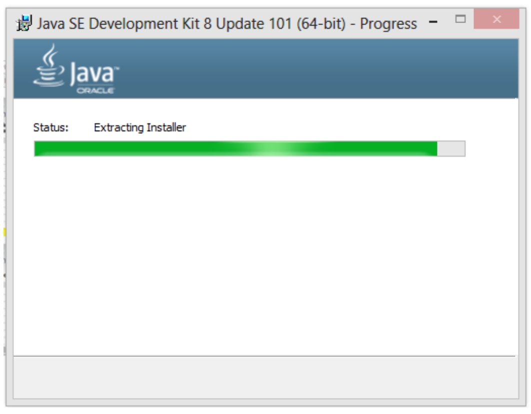
http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html


Java SE Development Kit 8 Update 101 (64-bit) - Setup

Welcome to the Installation Wizard for Java SE Development Kit 8 Update 101

This wizard will guide you through the installation process for the Java SE Development Kit 8 Update 101.

The Java Mission Control profiling and diagnostics tools suite is now available as part of the JDK.

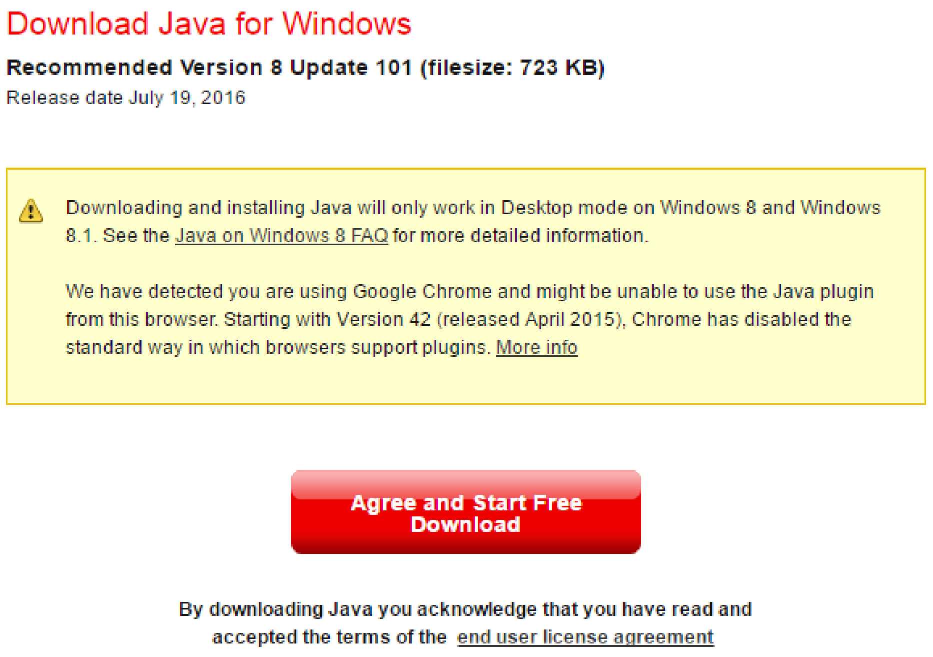

Java SE Development Kit 8 Update 101 (64-bit) - Progress

Status: Extracting Installer

Figure A-2: JDK Installation

After completing the JDK installation, JRE setup will be started automatically. Java RE need to be downloaded from Java website for those who wish avoid JDK installation for production environments.

http://www.java.com/en/download/win8.jsp


Download Java for Windows

Recommended Version 8 Update 101 (filesize: 723 KB)

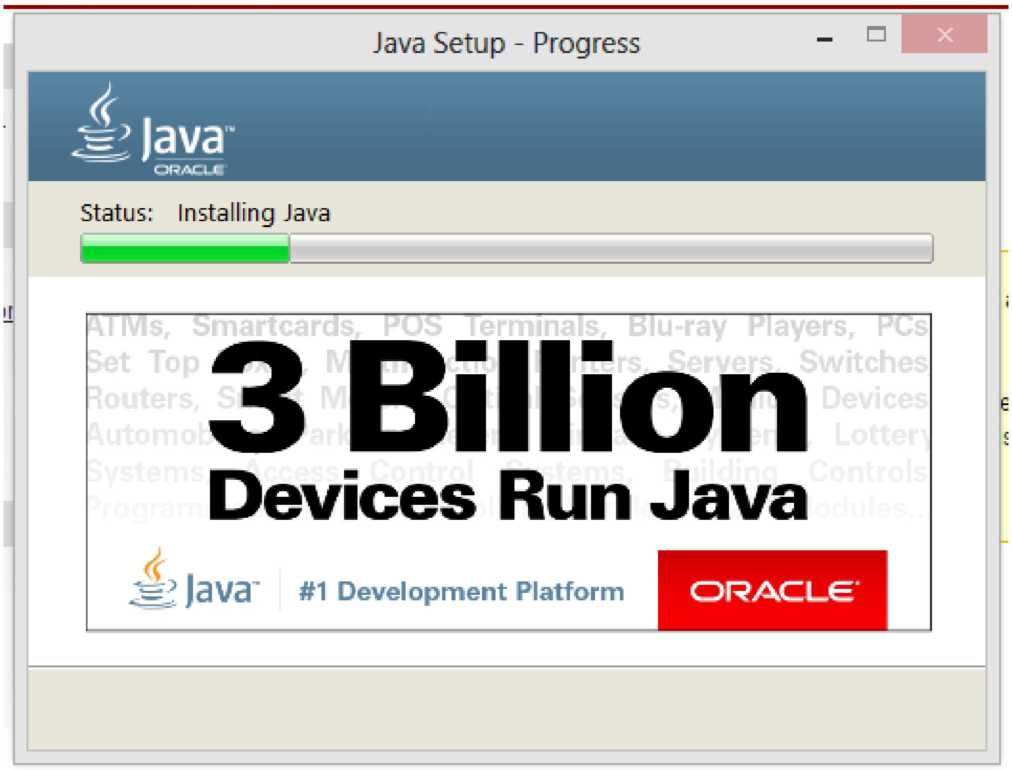
Release date July 19, 2016

Downloading and installing Java will only work in Desktop mode on Windows 8 and Windows 8.1. See the Java on Windows 8 FAQ for more detailed information.

We have detected you are using Google Chrome and might be unable to use the Java plugin from this browser. Starting with Version 42 (released April 2015), Chrome has disabled the standard way in which browsers support plugins. More info

By downloading Java you acknowledge that you have read and accepted the terms of the end user license agreement

Figure A-3: JRE download

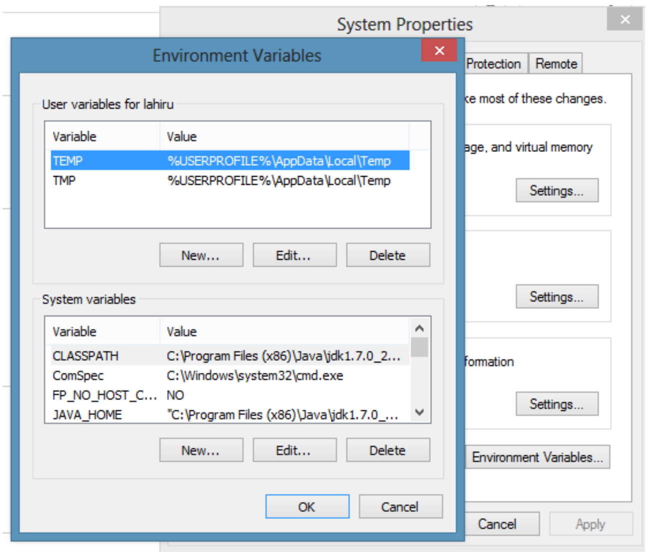


Java Setup - Progress

Status: Installing Java

Figure A-4 JRE Installation

Create the JAVA_HOME, Path and CLASSPATH Environment Variables in Windows



System Properties

Environment Variables

User variables for lahiru

Variable	Value
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

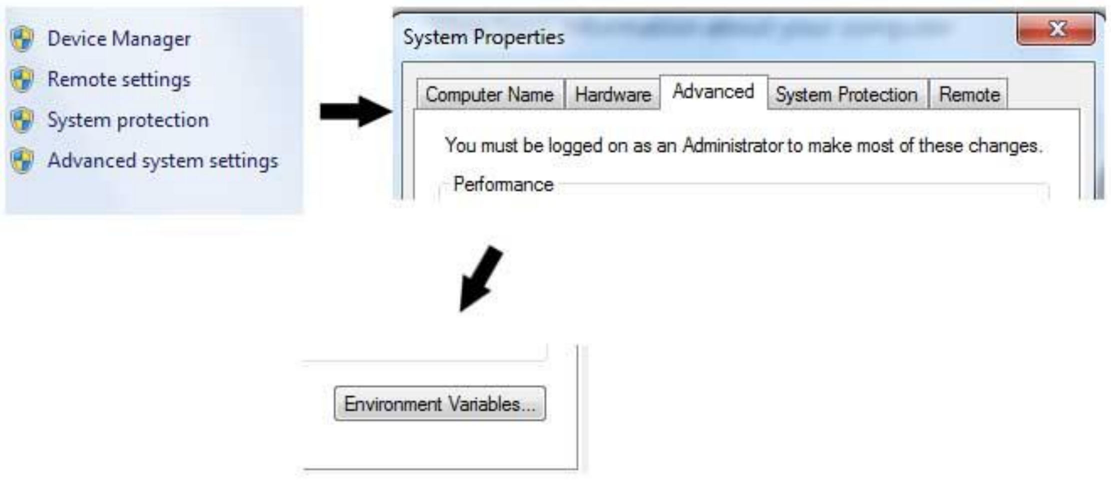
System variables

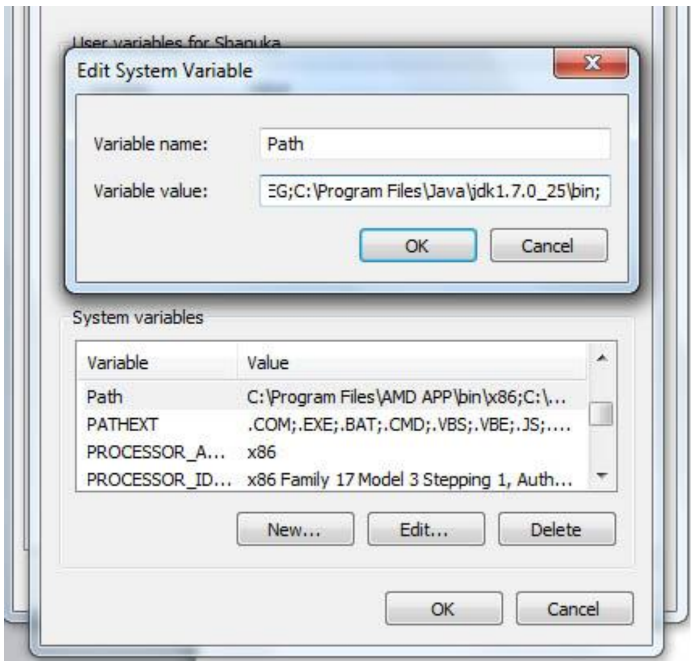
Variable	Value
CLASSPATH	C:\Program Files (x86)\Java\jdk1.7.0_2...
ComSpec	C:\Windows\system32\cmd.exe
FP_NO_HOST_C...	NO
JAVA_HOME	"C:\Program Files (x86)\Java\jdk1.7.0_...

Figure A-5: Setting up Environment Variables

Now it is needed to setup windows environmental variables to work Java properly. Right click on the “My Computer” icon. →Then select properties → on the left side menu click “Advanced system settings”.

System Properties window will be opened. Select "Advanced" tab and click "Environment Variables" button situated on the right bottom corner. (Figure A.7)





Variable name: Path

Variable value: ΞG;C:\Program Files\Java\jdk1.7.0_25\bin;

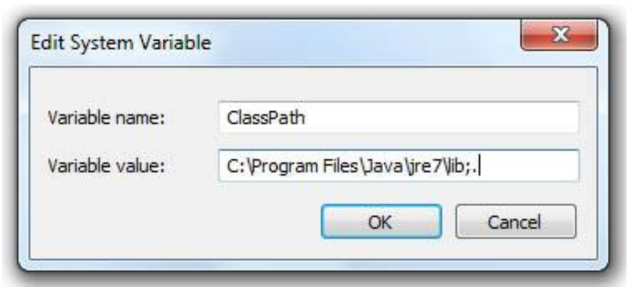
Variable	Value
Path	C:\Program Files\AMD APP\bin\x86;C:\...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....
PROCESSOR_A...	x86
PROCESSOR_ID...	x86 Family 17 Model 3 Stepping 1, Auth...

Figure A-6: Path Variable

Find the "Path" variable from the System variables table and press edit button. Afterwards it will be opened a dialog box with two inputs called "Edit System variable"

Click on the text in “variable value” input field and press key board “End” button to go end of the text. Now you should put semicolon (;) to end of the text.

Then go to the installation folder of java and copy path of the bin folder in the JDK 1.7.0 folder. Put this path to after the semicolon and add another semicolon to end of the path. Now click “Ok” button to set the path. (Figure A.8)


Edit System Variable

Variable name: ClassPath

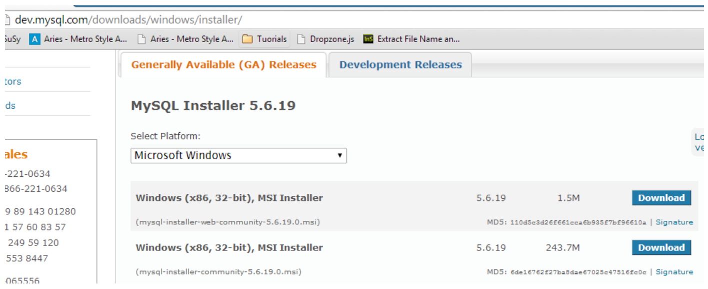
Variable value: C:\Program Files\Java\jre7\lib;.

Figure A-7: Class path variable

Next click the “New” button and put the variable name as “CLASPATH”. Insert the path of “java\jre7\lib” as variable value and put semicolon and full stop to end of the path and click “Ok” and “Ok” to completing configuration of java. (Figure A.9)

1. Install MySQL

   Download MySQL Server from oracle web site.


dev.mysql.com/downloads/windows/installer/

-221-0634
866-221-0634
9 89 143 01280
1 57 60 83 57
249 59 120
553 8447
065556

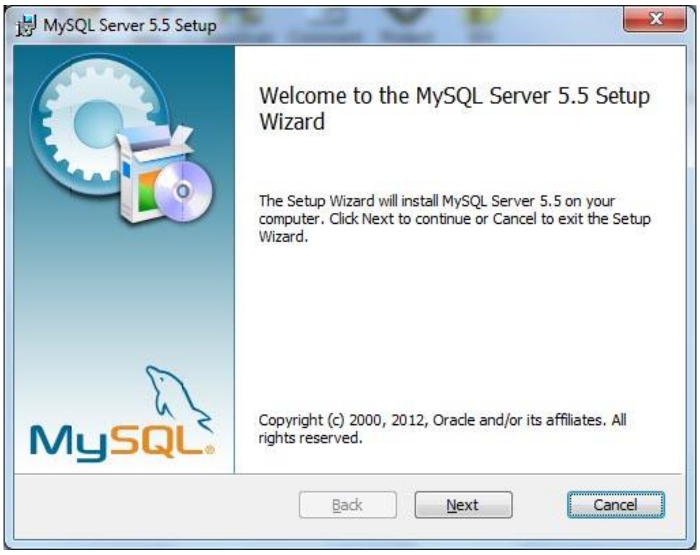
MySQL Installer 5.6.19

Select Platform: Microsoft Windows

Windows (x86, 32-bit), MSI Installer	5.6.19	1.5M	Download
(mysql-installer-web-community-5.6.19.0.msi)		MD5: 110d5c3d26f661cca6b935f7bf96610a | Signature	
Windows (x86, 32-bit), MSI Installer	5.6.19	243.7M	Download
(mysql-installer-community-5.6.19.0.msi)		MD5: 6de16762f27ba8dae67025c47516fc0c | Signature	

Figure A-8: Download My SQL

The welcome screen as follows.


MySQL Server 5.5 Setup

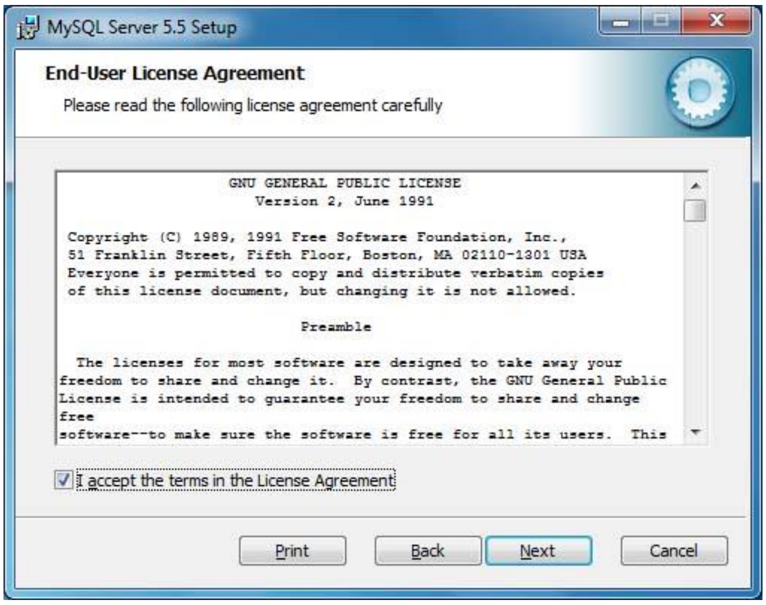
Welcome to the MySQL Server 5.5 Setup Wizard

The Setup Wizard will install MySQL Server 5.5 on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

Copyright (c) 2000, 2012, Oracle and/or its affiliates. All rights reserved.

Figure A-9: MySQL Welcome

Select I accept and click next button


MySQL Server 5.5 Setup

End-User License Agreement

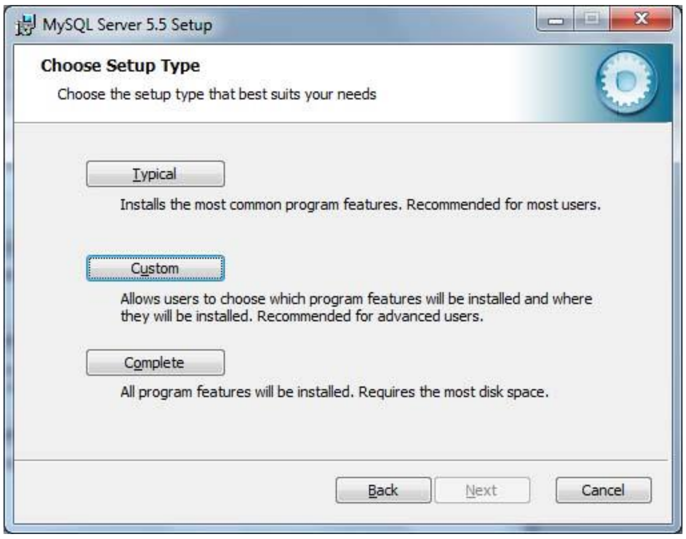
Please read the following license agreement carefully

```
                    GNU GENERAL PUBLIC LICENSE
                       Version 2, June 1991

 Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                            Preamble

  The licenses for most software are designed to take away your
freedom to share and change it.  By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change
free
software--to make sure the software is free for all its users.  This
```

[✓] I accept the terms in the License Agreement

Figure A-10: MySQL Agreement

Select “Complete” button and click Next button for custom setting.


MySQL Server 5.5 Setup

Choose Setup Type

Choose the setup type that best suits your needs

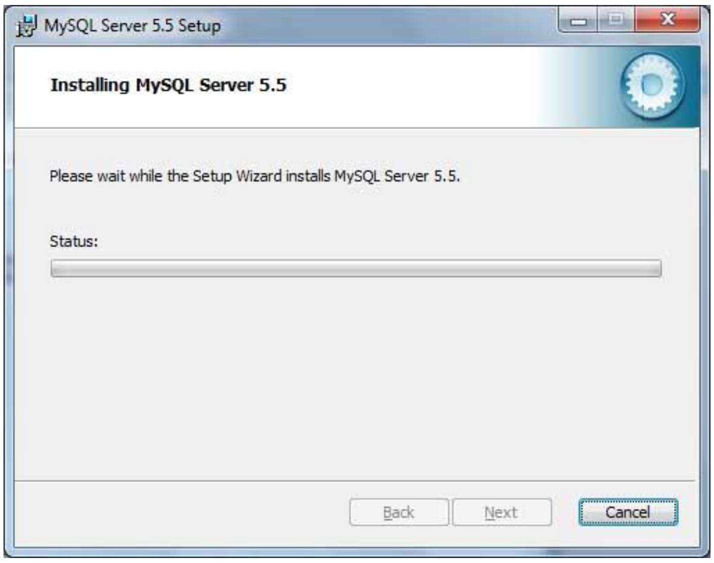
Installs the most common program features. Recommended for most users.

Allows users to choose which program features will be installed and where they will be installed. Recommended for advanced users.

All program features will be installed. Requires the most disk space.

Figure A-11: MySQL Setup Screen

Click "Install" button on the next window and you will see installing MySQL server on the server computer. After installation completed, click "Finish" button


MySQL Server 5.5 Setup

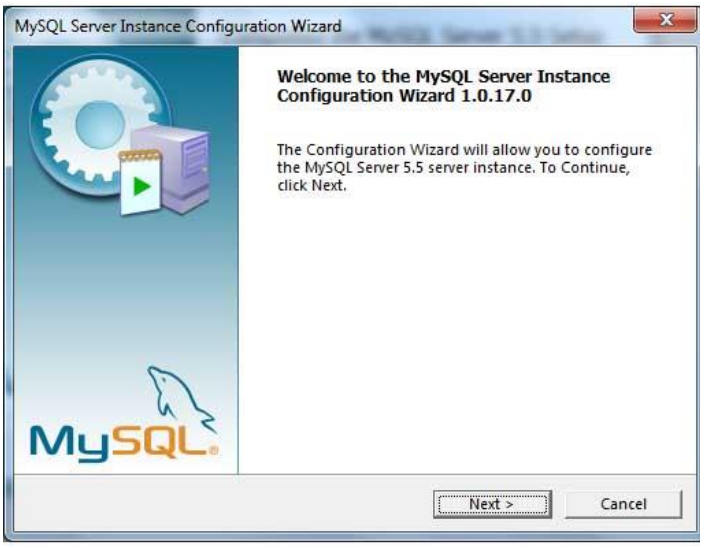
Installing MySQL Server 5.5

Please wait while the Setup Wizard installs MySQL Server 5.5.

Status:

Figure A-12: My SQL Installing

Afterwards MySQL configuration setup will be loaded for configure install MySQL server.

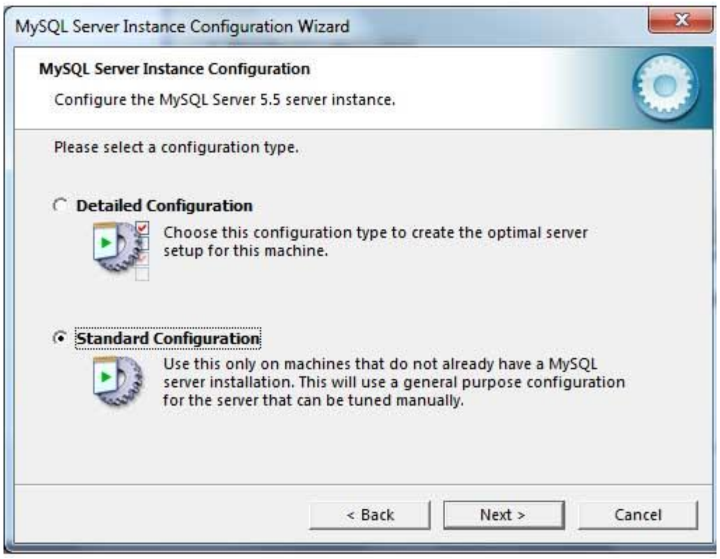

MySQL Server Instance Configuration Wizard

Welcome to the MySQL Server Instance Configuration Wizard 1.0.17.0

The Configuration Wizard will allow you to configure the MySQL Server 5.5 server instance. To Continue, click Next.

Figure A-13: MySQL Instance Setup

Select "Standard configuration" from next window and click "Next" for proceed configuration.


MySQL Server Instance Configuration Wizard

MySQL Server Instance Configuration

Configure the MySQL Server 5.5 server instance.

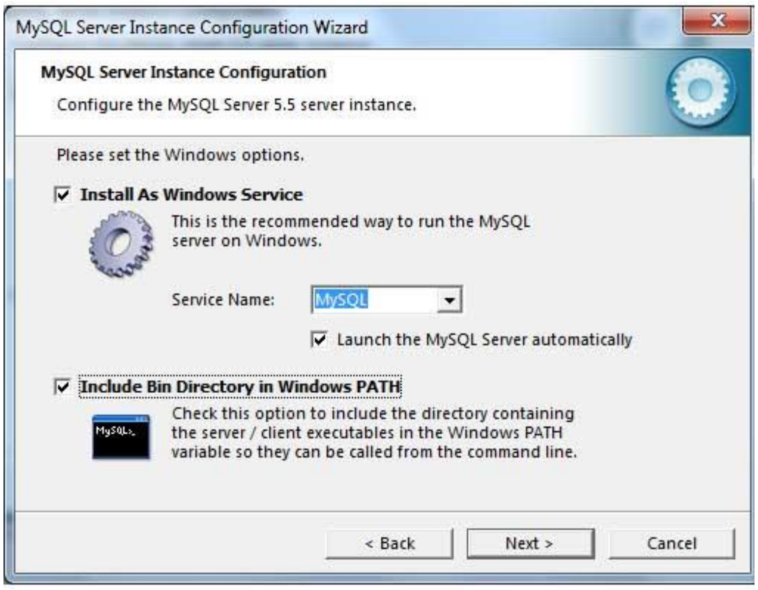
Please select a configuration type.

( ) **Detailed Configuration**
Choose this configuration type to create the optimal server setup for this machine.

(•) **Standard Configuration**
Use this only on machines that do not already have a MySQL server installation. This will use a general purpose configuration for the server that can be tuned manually.

Figure A-14: MySQL Configuration Screen

You must tick all of option in the "Windows Option" window and click "Next" button


MySQL Server Instance Configuration Wizard

MySQL Server Instance Configuration

Configure the MySQL Server 5.5 server instance.

Please set the Windows options.

[✓] Install As Windows Service

This is the recommended way to run the MySQL server on Windows.

Service Name: MySQL

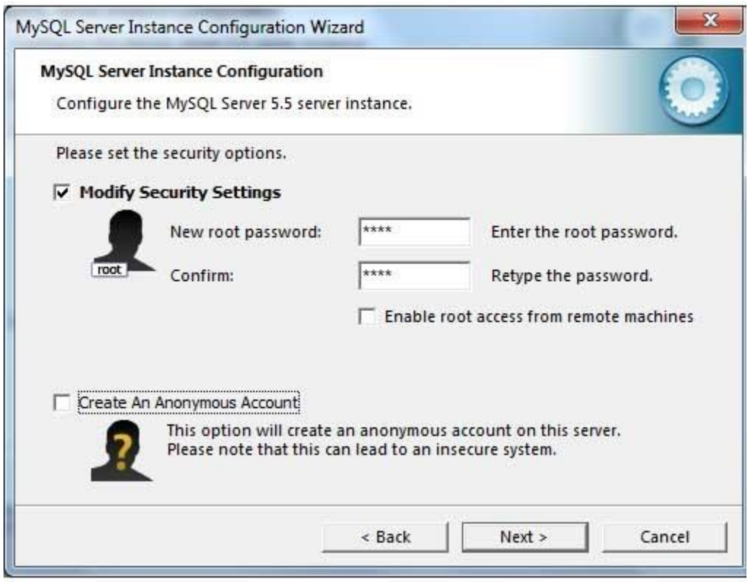
[✓] Launch the MySQL Server automatically

[✓] Include Bin Directory in Windows PATH

Check this option to include the directory containing the server / client executables in the Windows PATH variable so they can be called from the command line.

Figure A-15: My SQL Service Configuration

Enter root user password two times and click “Next”


MySQL Server Instance Configuration Wizard

MySQL Server Instance Configuration

Configure the MySQL Server 5.5 server instance.

Please set the security options.

[✓] Modify Security Settings

New root password: **** Enter the root password.

Confirm: **** Retype the password.

[ ] Enable root access from remote machines

[ ] Create An Anonymous Account

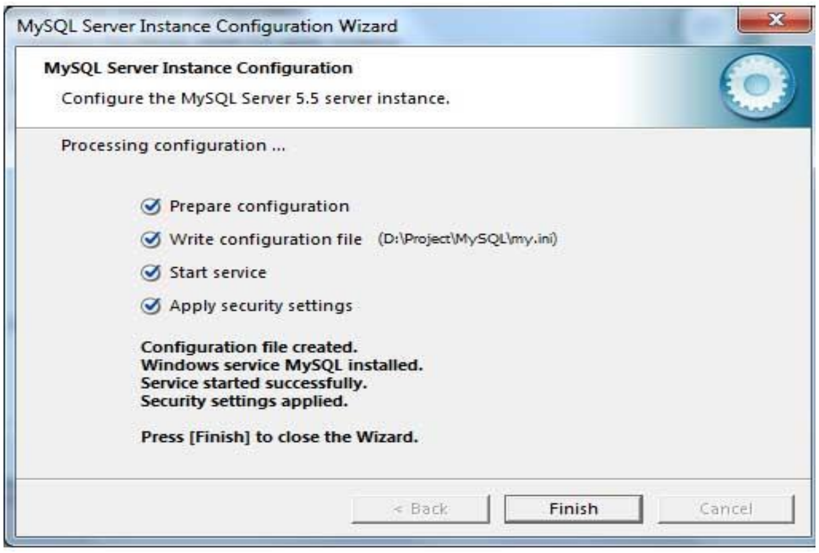
This option will create an anonymous account on this server. Please note that this can lead to an insecure system.

Figure A-16: My SQL root account setup

Step 8

Click “Execute” button. You will see MySQL setup is processing your configuration.

Then click “Finish” to complete.



MySQL Server Instance Configuration Wizard

MySQL Server Instance Configuration

Configure the MySQL Server 5.5 server instance.

Processing configuration ...

✓ Prepare configuration
✓ Write configuration file (D:\Project\MySQL\my.ini)
✓ Start service
✓ Apply security settings

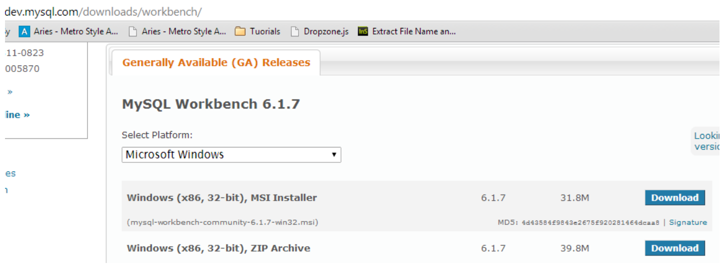
Configuration file created.
Windows service MySQL installed.
Service started successfully.
Security settings applied.

Press [Finish] to close the Wizard.

Figure A-17: MySQL Setup Finalization Screen

2. Install MySQL Workbench

   Download and install MySQL workbench from oracle website.



dev.mysql.com/downloads/workbench/

Generally Available (GA) Releases

MySQL Workbench 6.1.7

Select Platform: Microsoft Windows

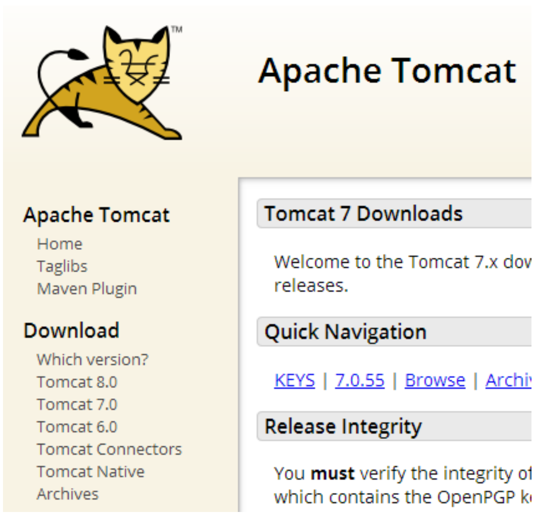
Windows (x86, 32-bit), MSI Installer	6.1.7	31.8M	Download
(mysql-workbench-community-6.1.7-win32.msi)			MD5: 4d43584f9843e2675f920281464dcaa8 | Signature
Windows (x86, 32-bit), ZIP Archive	6.1.7	39.8M	Download

Figure A-18: Download MySQL Work Bench

3. Install Tomcat server

   Download apache tomcat files from apache website and unzip it.

   Go to unzipped folder and run startup.bat file in bin folder.

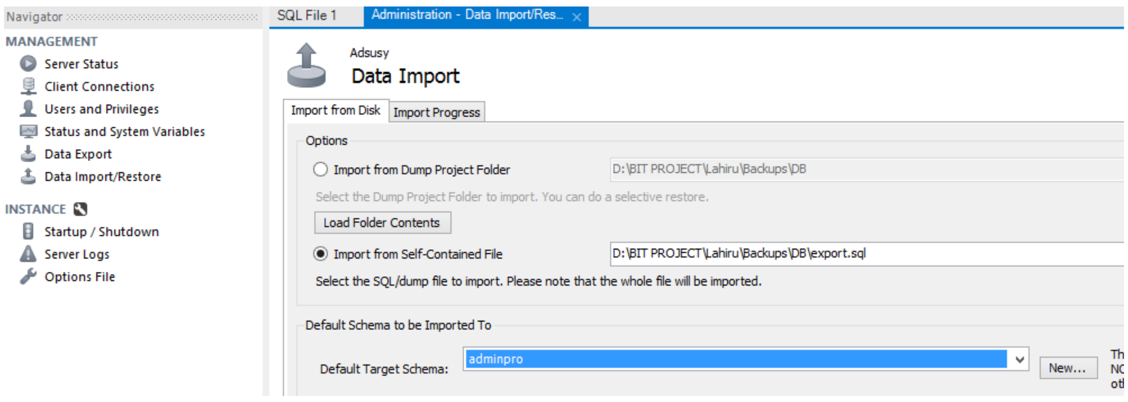


Welcome to the Tomcat 7.x dov
releases.

You **must** verify the integrity o
which contains the OpenPGP k

Figure A-19: Download Tomcat

4. Host DB in MySQL Server
   Restore the given backup to gimanhala database of MySQL Server using MySQL Workbench.



Import from Dump Project Folder: D:\BIT PROJECT\Lahiru\Backups\DB

Select the Dump Project Folder to import. You can do a selective restore.

Import from Self-Contained File: D:\BIT PROJECT\Lahiru\Backups\DB\export.sql

Select the SQL/dump file to import. Please note that the whole file will be imported.

Default Target Schema: adminpro

Figure A-20: Hosting Database File

5. Host Application in Tomcat Server
   Copy the given .War file to webapps folder of tomcat directory. And restart the tomcat server

Minimum Hardware and software Requirements

Hardware requirements – For Server

- 2.8 Dual Core GHz processor or higher
- 1GB RAM
- 50GB free HDD space
- General Key board and Mouse
- Internet Modem/Router with Internet Connection

Hardware requirements – For Client PC

- 2.0 GHz processor or higher
- 512MB RAM
- General Key board and Mouse
- Internet Modem/Router with Internet Connection
- Printer

Software requirements – For Server

- MS Window Server or Linux Server OS.
- Java Runtime Environment 1.7 or higher
- MySQL 5.5 or higher
- Apache Tomcat Server

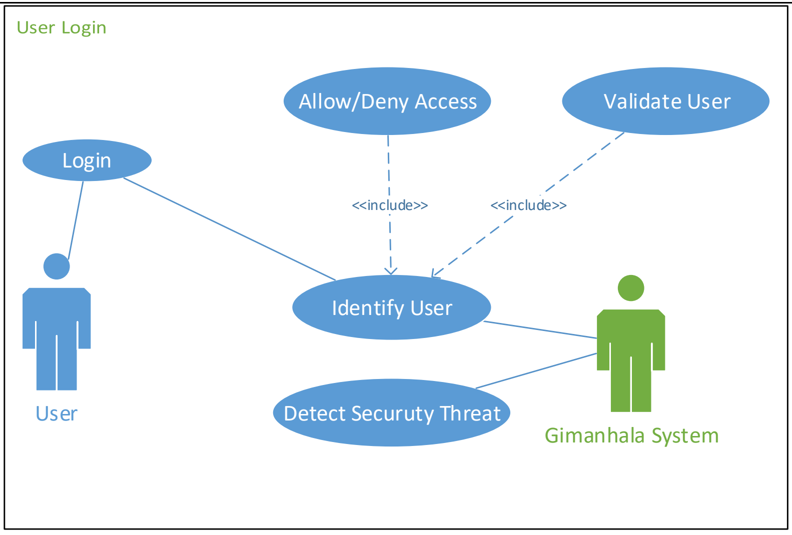
Software requirements – For Client PC

1. MS Window, Mac or Linux PC OS.
2. Google Chrome Web Browser

Appendix B - Design Documentation

USE CASE Narratives

- Login

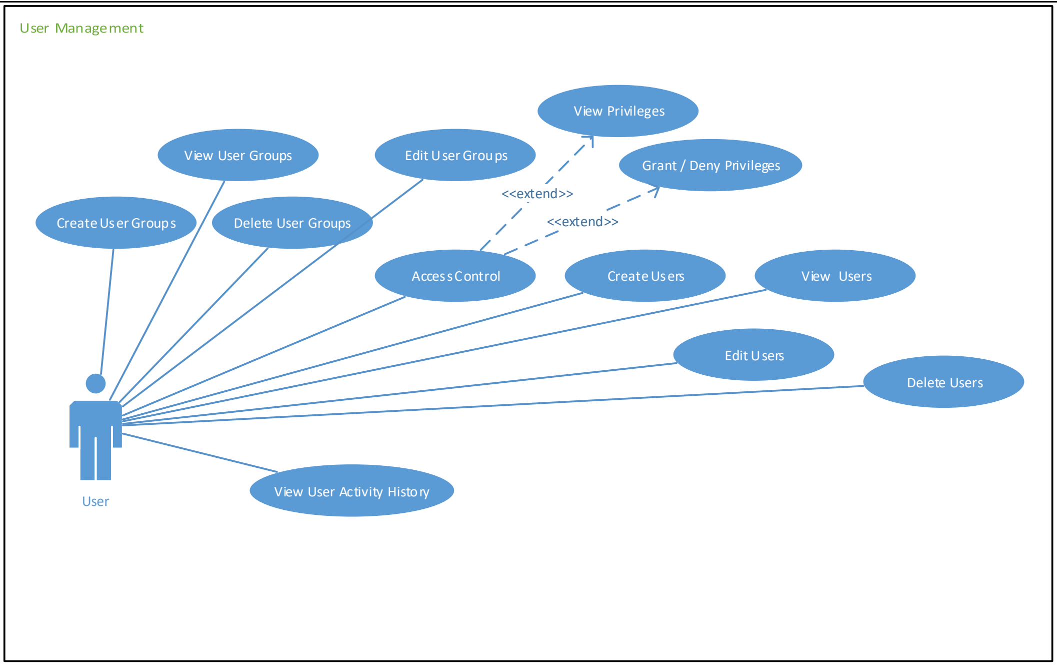


Use Case Name	Login
Objective	Initialize login process of user
Pre-Condition	User Enters Username and Password
Condition On Success	Redirect to Home Page
Condition On Failure	Show Relevant Error Message
Main Scenario	User clicks on login button. Then the system will get the user credentials and send it for verification

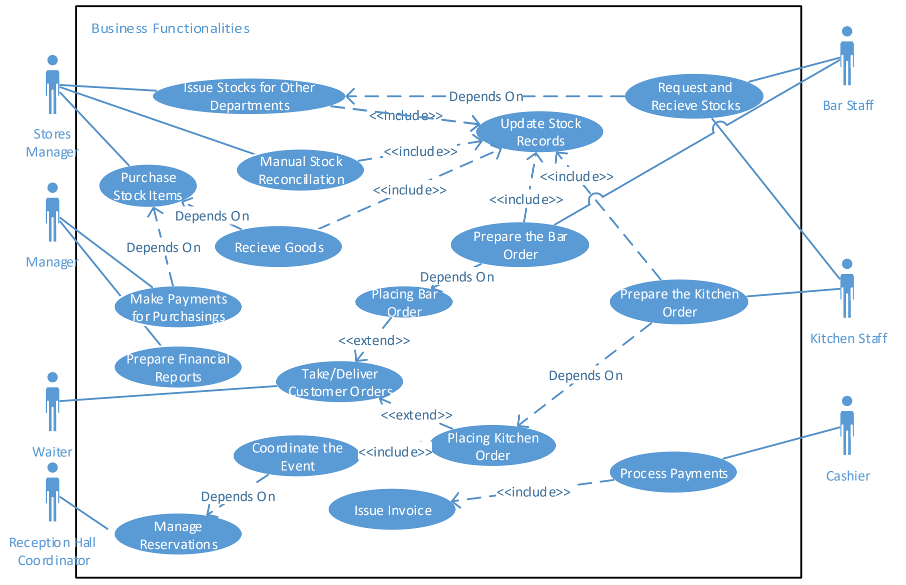
Use Case Name	Detect Security Threat
Objective	Identify security attacks and prevent the system from security threats
Pre-Condition	User try Brute force or SQL Injection Attack
Condition On Success	Show Message and Block access
Condition On Failure	Reload the page
Main Scenario	System identify above attacks separately and takes necessary actions

Use Case Name	Identify User
Objective	Recognize User and Allow /Deny Access
Pre-Condition	User try to login
Condition On Success	Grant Access to user
Condition On Failure	Deny Access for user
Main Scenario	Check the validity of username and password Check the availability of user Match the password

	Check the privileges Allow /Deny System Functionalities Showing Relevant Messages Process Login
Use Case Name	Allow/Deny Access
Objective	Allow users to perform actions which are assigned and deny user from other activities
Pre-Condition	User attempts to login
Condition On Success	Enable Functionality
Condition On Failure	Disable Functionality
Main Scenario	Read User Privileges from Database Enable/Disable Links Allow/Deny in user activities
Use Case Name	Validate User
Objective	Make sure the Validity of the user
Pre-Condition	User attempts to login
Condition On Success	Login User
Condition On Failure	Redirect to login
Main Scenario	Check username and password Encrypt Username and Password Check User Availability Check password Check User Status Check User Group Status



Use Case Name	User management
Objective	Identify the correct user
Pre-Condition	System access given
Condition On Success	login
Condition On Failure	Login blocked
Main Scenario	Checking the authorization

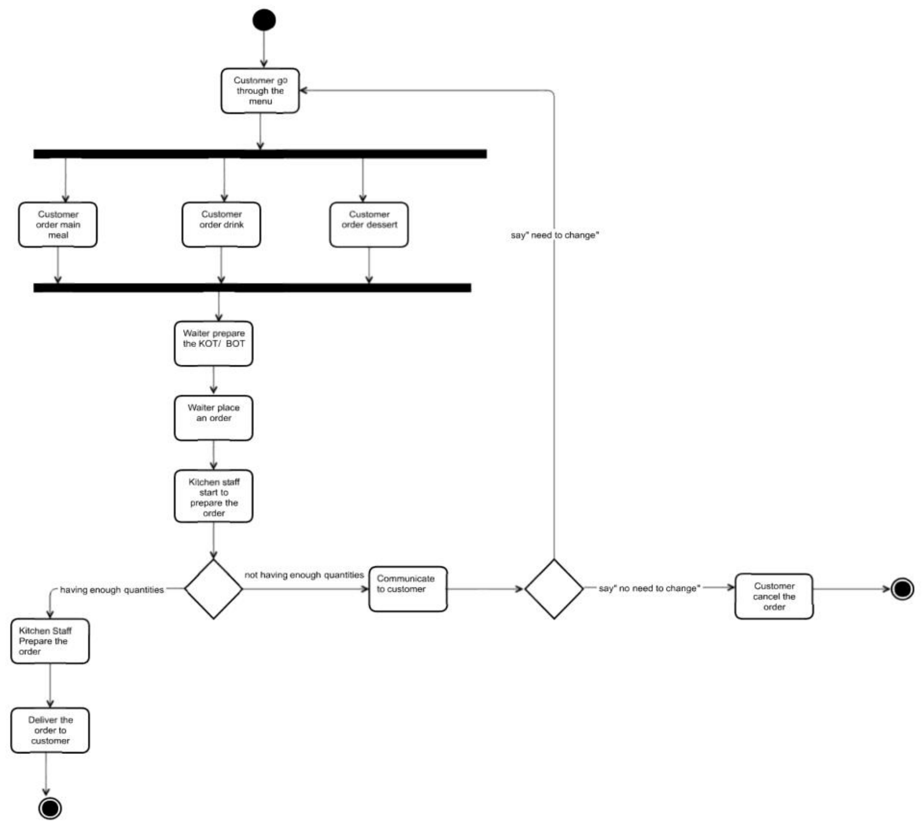


Use Case Name	Order management
Objective	Identify / verify each business process
Pre-Condition	User rights given

Condition On Success	Next activity loaded
Condition On Failure	Error message shown
Main Scenario	Processing order

Activity Diagram for Order Management Module

Figure B.1 below depicts the activity diagram for order management module for Gimanhala.



Customer go through the menu
Customer order main meal
Customer order drink
Customer order dessert
say" need to change"
Waiter prepare the KOT/ BOT
Waiter place an order
Kitchen staff start to prepare the order
having enough quantities
not having enough quantities
Communicate to customer
say" no need to change"
Customer cancel the order
Kitchen Staff Prepare the order
Deliver the order to customer

Figure B-1: activity diagram for order management module

Sequence Diagram for Order Management Module

Figure B.2 below depicts the activity diagram for order management module for Gimanhala.

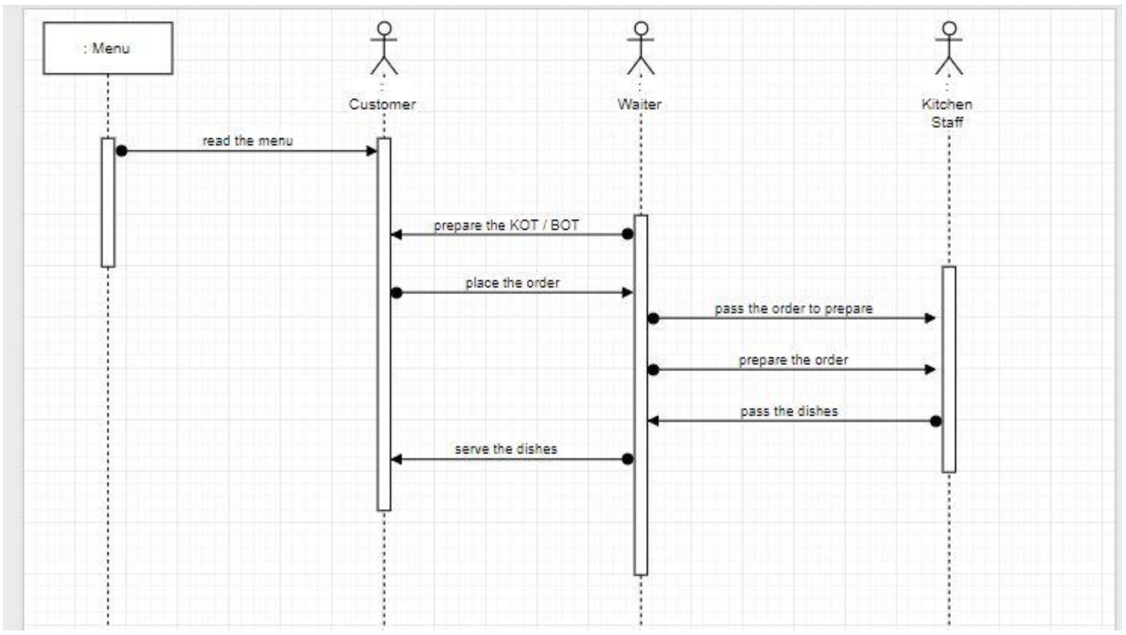


Figure B-2: sequence diagram for order management module

Appendix C - User Documentation

Getting Started

Enter the given URL in web browser and following login page will be shown

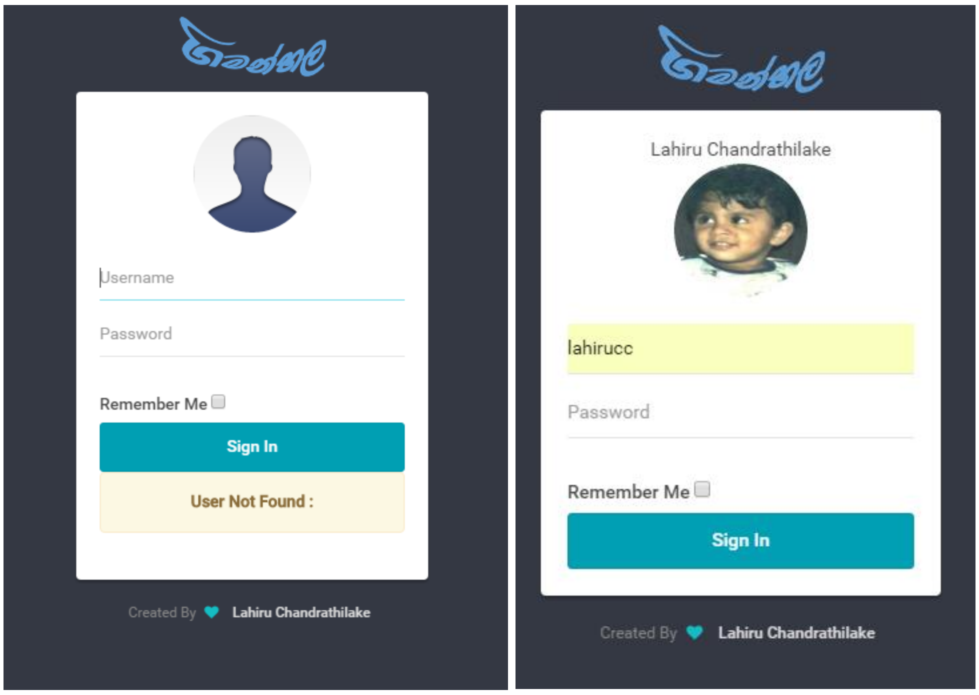


Figure C-1: login page

As the user name entered, system will validate the user. When entered after typing the password, the system will validate the credentials and login to the system if successful

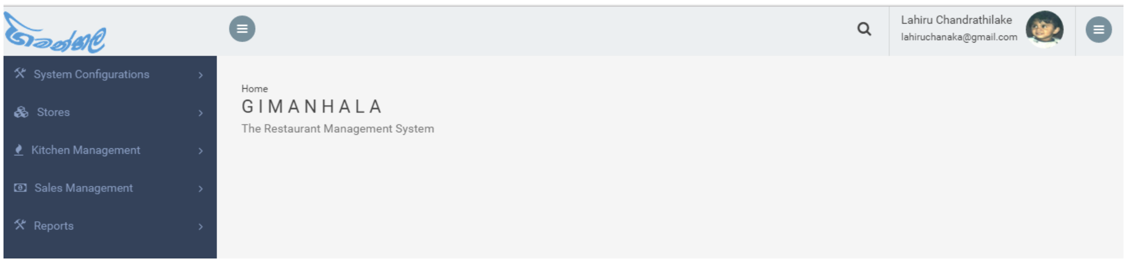


Figure C-2: Home Page

Menu is placed in left side, which has mainly 5 areas.

System configuration area has links to setup user groups, users, access rights, company location types and location structure. These locations are used in stores management module to manage separate stocks.

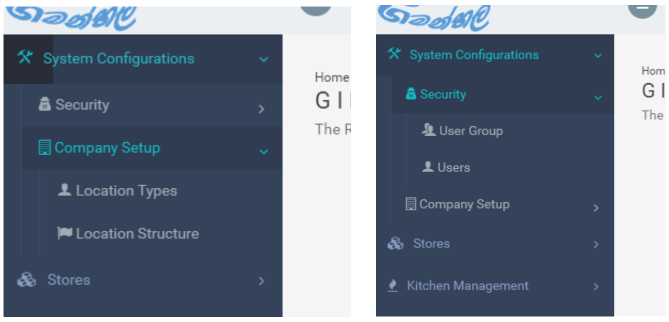
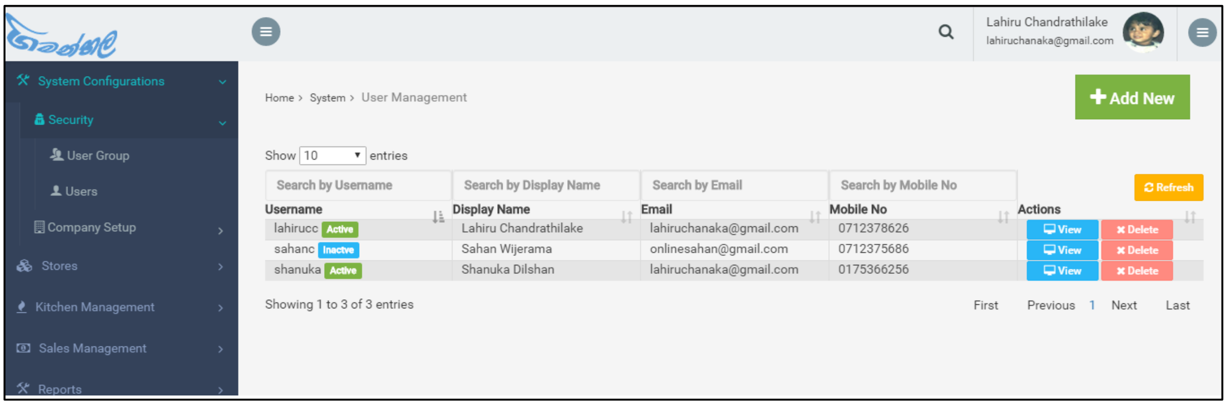


Figure C-3: System Configuration Menu



Username	Display Name	Email	Mobile No	Actions
lahirucc Active	Lahiru Chandrathilake	lahiruchanaka@gmail.com	0712378626	View Delete
sahanc Inactive	Sahan Wijerama	onlinesahan@gmail.com	0712375686	View Delete
shanuka Active	Shanuka Dilshan	lahiruchanaka@gmail.com	0175366256	View Delete

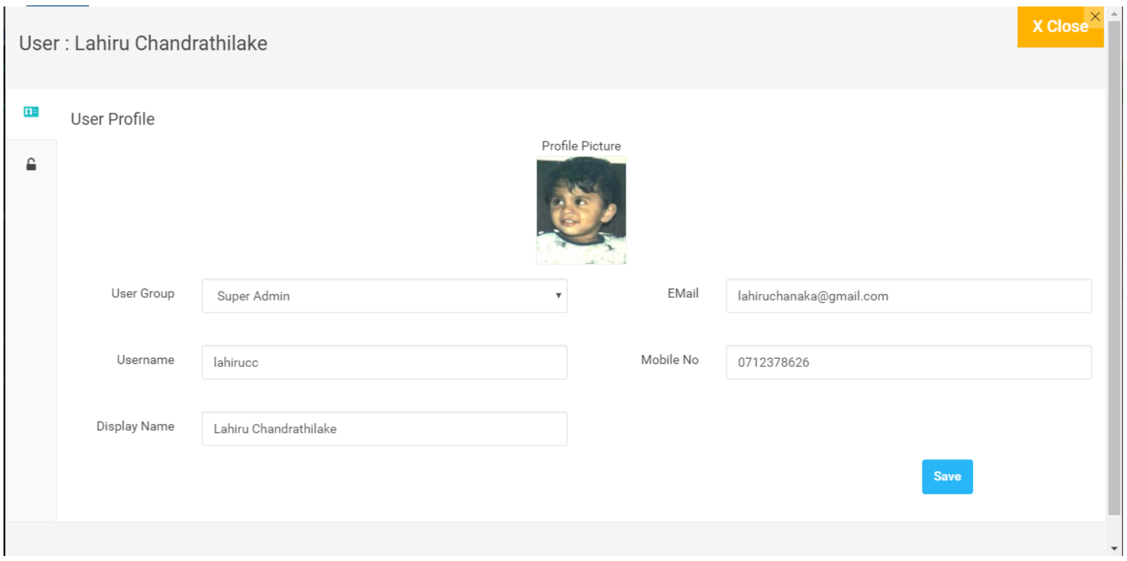
Figure C-4: User Management Grid

There are few main components that can be seen in this interface

- Add New Button

  By clicking on this, a form to create new record will be opened and there is a save button at the bottom of each form for saving the records. For view button as well this functionality is same and additionally the relevant data will be loaded to form

- Search Boxes at the top of each column
  These boxes can be seen in all data tables used in this system. These are optimized to search a record by typing any part of the text
- Action Buttons at the last column
  Different actions that can be done to records, are listed in last column which named as action column



User Group	Super Admin	EMail	lahiruchanaka@gmail.com
Username	lahirucc	Mobile No	0712378626
Display Name	Lahiru Chandrathilake		

Figure C-5: New User Form

User Configuration can be done by clicking on padlock icon at the left side

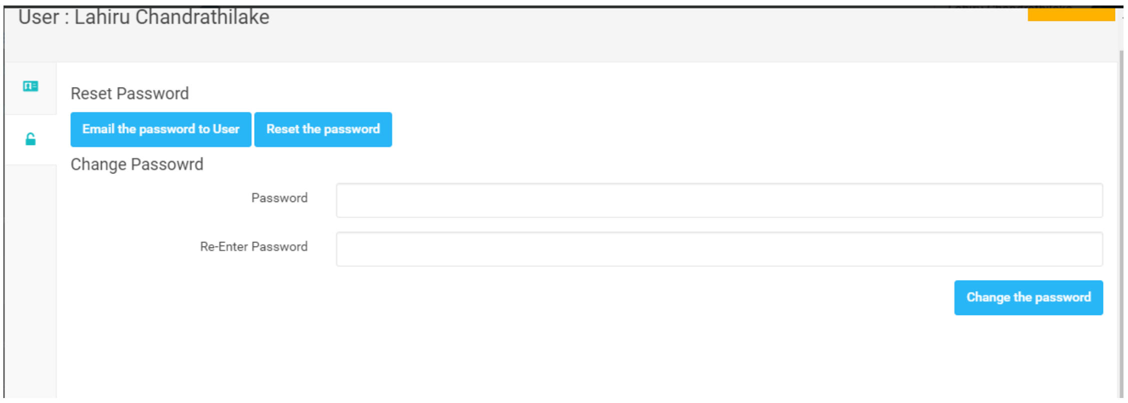
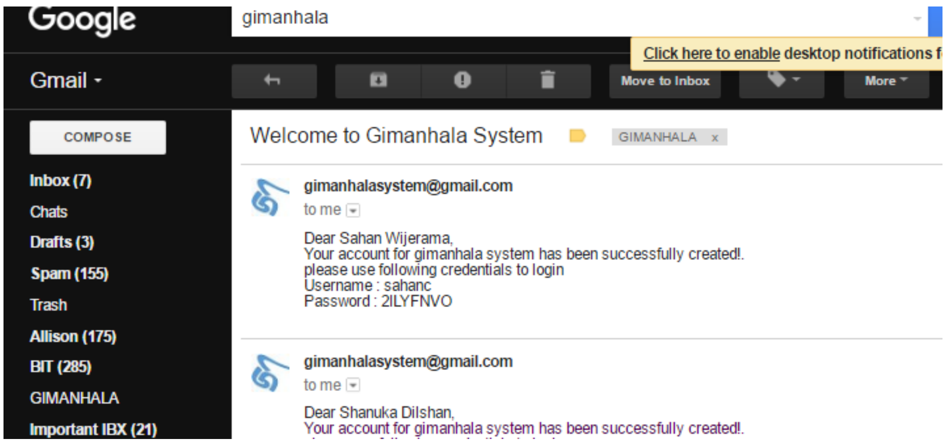


Figure C-6: User Configuration Page

When a user is created, user will receive an email with the generated password. Users are advised to change the password immediately.



gimanhalasystem@gmail.com
to me

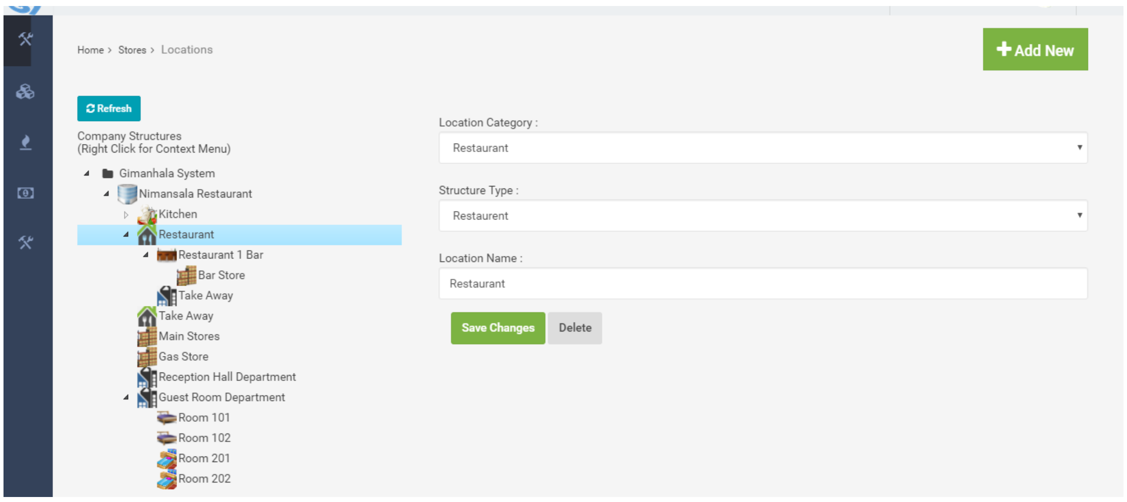
Dear Sahan Wijerama,
Your account for gimanhala system has been successfully created!.
please use following credentials to login
Username : sahanc
Password : 2ILYFNVO

gimanhalasystem@gmail.com
to me

Dear Shanuka Dilshan,
Your account for gimanhala system has been successfully created!.

Figure C-7: User Email

Stock Locations can be managed using following screen



Company Structures
(Right Click for Context Menu)

- Gimanhala System
  - Nimansala Restaurant
    - Kitchen
    - Restaurant
      - Restaurant 1 Bar
        - Bar Store
      - Take Away
    - Take Away
    - Main Stores
    - Gas Store
    - Reception Hall Department
    - Guest Room Department
      - Room 101
      - Room 102
      - Room 201
      - Room 202

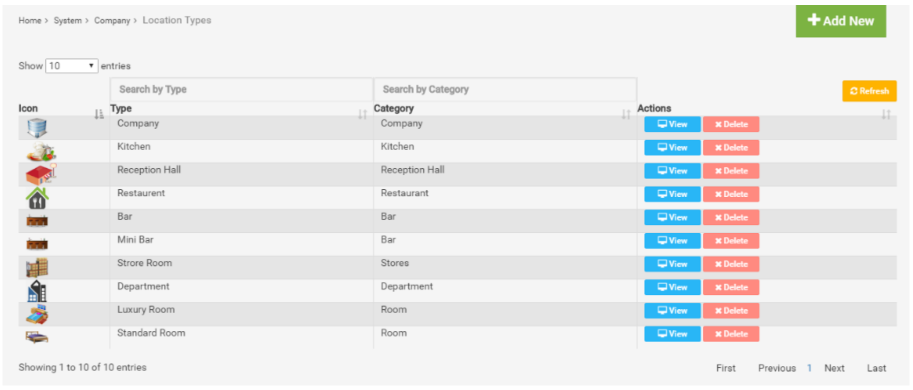
Location Category : Restaurant

Structure Type : Restaurent

Location Name : Restaurant

Figure C-8: Location Structure Screen

Structure Types can be defined by using following screen



Show 10 entries

Icon	Type	Category	Actions
	Company	Company	View Delete
	Kitchen	Kitchen	View Delete
	Reception Hall	Reception Hall	View Delete
	Restaurent	Restaurant	View Delete
	Bar	Bar	View Delete
	Mini Bar	Bar	View Delete
	Strore Room	Stores	View Delete
	Department	Department	View Delete
	Luxury Room	Room	View Delete
	Standard Room	Room	View Delete

Showing 1 to 10 of 10 entries

First Previous 1 Next Last

Figure C-9: Location Structure Types

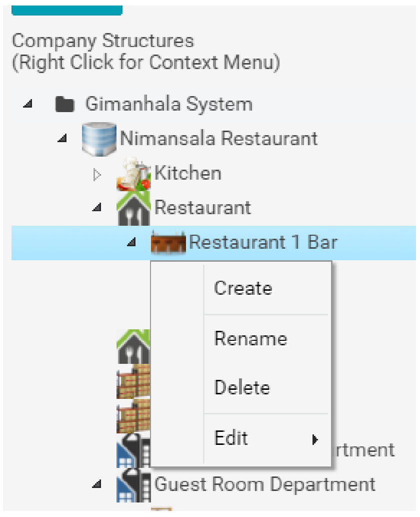


Figure C-10: Location Structure Context Menu

There is a context menu in the location structure tree, it can be obtained by right clicking on it. This Tree Structure Supports Copy/Paste and Drag & Drop.

It is very easy to move or copy a location inside to another location.

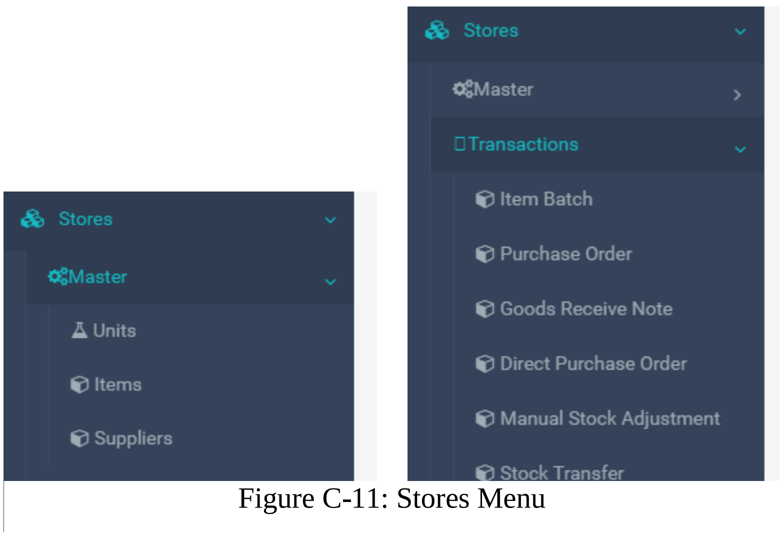
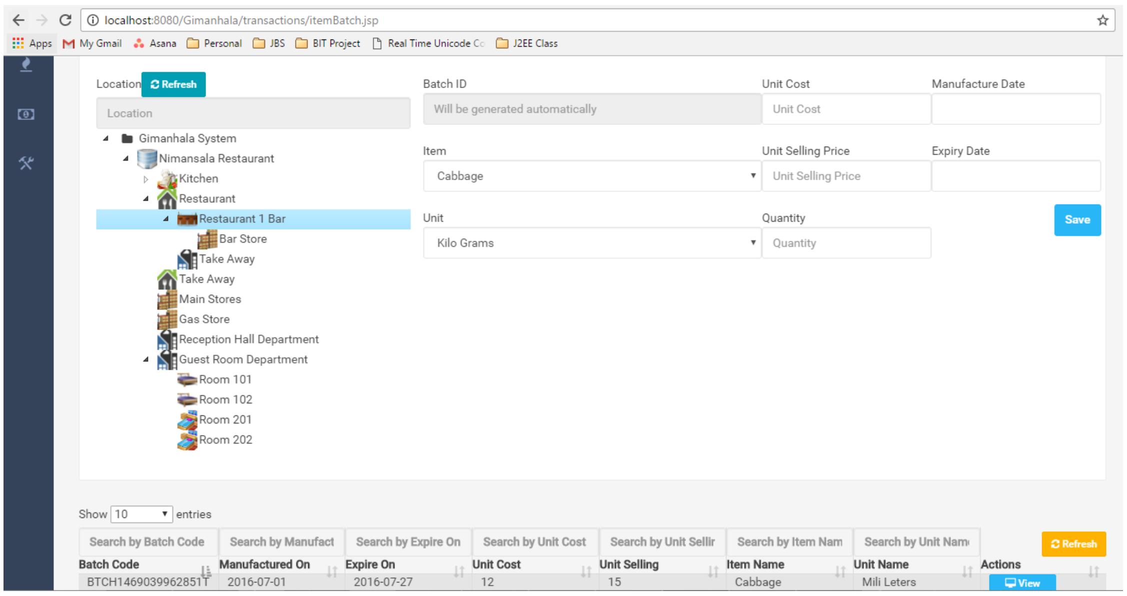


Figure C-11: Stores Menu

By master files, users can define the Units used to measure the items, list of items and the suppliers.

Transactions

- Item Batch
  Each purchasing of an item is considered as a new batch. Batches can be created manually as well. This screen is used for managing batches


localhost:8080/Gimanhala/transactions/itemBatch.jsp

Apps My Gmail Asana Personal JBS BIT Project Real Time Unicode Co J2EE Class

Location Refresh

Location

- Gimanhala System
  - Nimansala Restaurant
    - Kitchen
    - Restaurant
      - Restaurant 1 Bar
        - Bar Store
      - Take Away
    - Take Away
    - Main Stores
    - Gas Store
    - Reception Hall Department
    - Guest Room Department
      - Room 101
      - Room 102
      - Room 201
      - Room 202

Batch ID: Will be generated automatically
Unit Cost: Unit Cost
Manufacture Date:
Item: Cabbage
Unit Selling Price: Unit Selling Price
Expiry Date:
Unit: Kilo Grams
Quantity: Quantity
Save

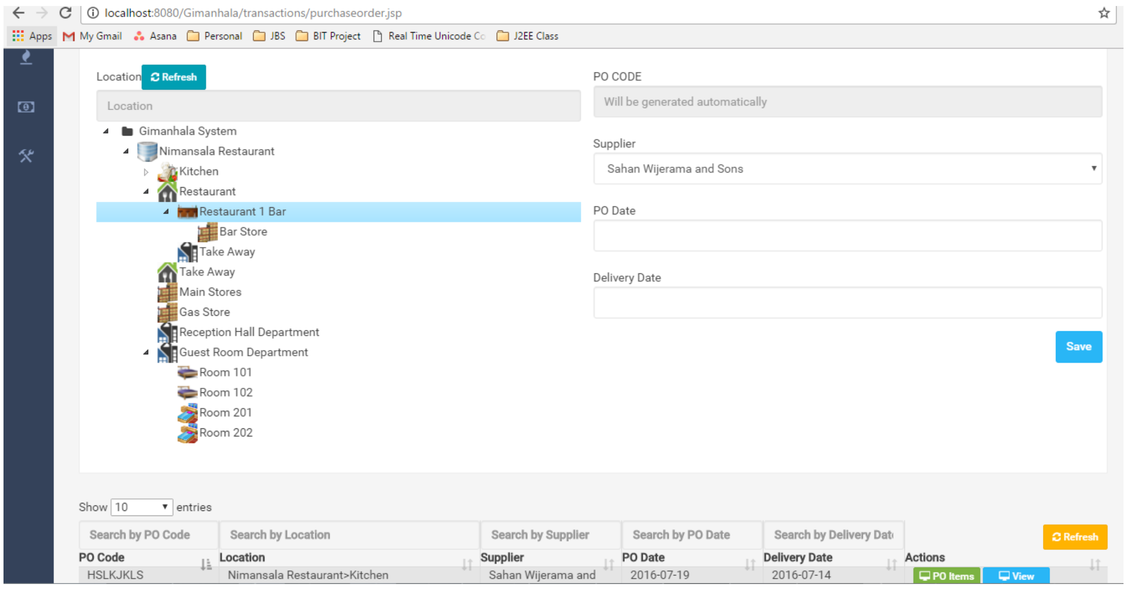
Show 10 entries

Refresh

Batch Code	Manufactured On	Expire On	Unit Cost	Unit Selling	Item Name	Unit Name	Actions
Search by Batch Code	Search by Manufact	Search by Expire On	Search by Unit Cost	Search by Unit Sellir	Search by Item Nam	Search by Unit Nam	
BTCH1469039962851T	2016-07-01	2016-07-27	12	15	Cabbage	Mili Leters	View

Figure C-12: Item Batch Form

- Purchase Order
  Purchase Orders are placed to the suppliers to request goods. These purchase orders are usually settled with one or more goods receive notes

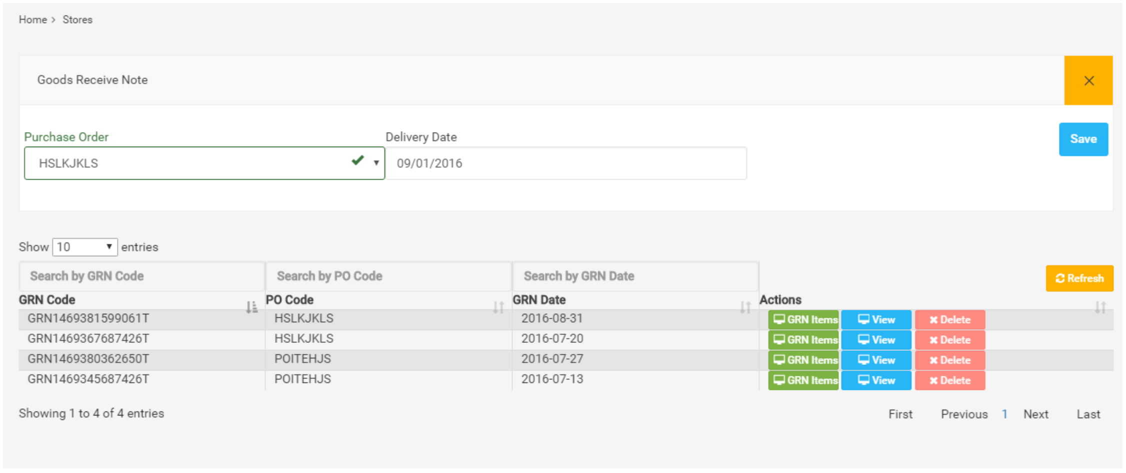


PO CODE	Will be generated automatically
Supplier	Sahan Wijerama and Sons
PO Date	
Delivery Date	

PO Code	Location	Supplier	PO Date	Delivery Date	Actions
HSLKJKLS	Nimansala Restaurant>Kitchen	Sahan Wijerama and	2016-07-19	2016-07-14	PO Items View

Figure C-13: Purchase Order Form

- Goods Receive Note
  GRN can be used to receive the goods requested in Purchase Order. When goods entered in this note, batches are created for each item.

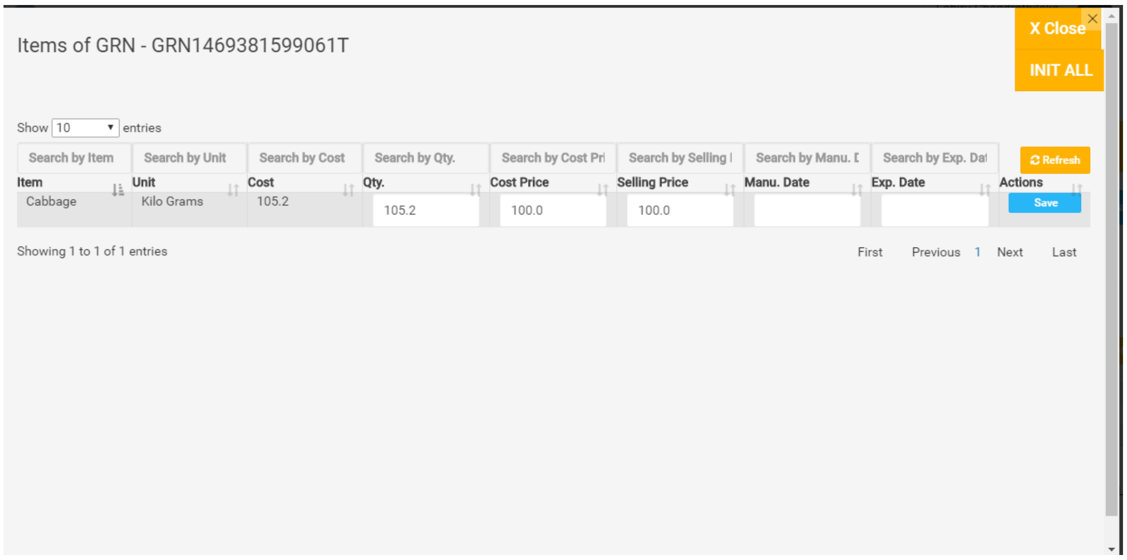


Purchase Order	HSLKJKLS
Delivery Date	09/01/2016

GRN Code	PO Code	GRN Date	Actions
GRN1469381599061T	HSLKJKLS	2016-08-31	GRN Items View Delete
GRN1469367687426T	HSLKJKLS	2016-07-20	GRN Items View Delete
GRN1469380362650T	POITEHJS	2016-07-27	GRN Items View Delete
GRN1469345687426T	POITEHJS	2016-07-13	GRN Items View Delete

Showing 1 to 4 of 4 entries   First Previous 1 Next Last

Figure C-14: Good Receive Note Form

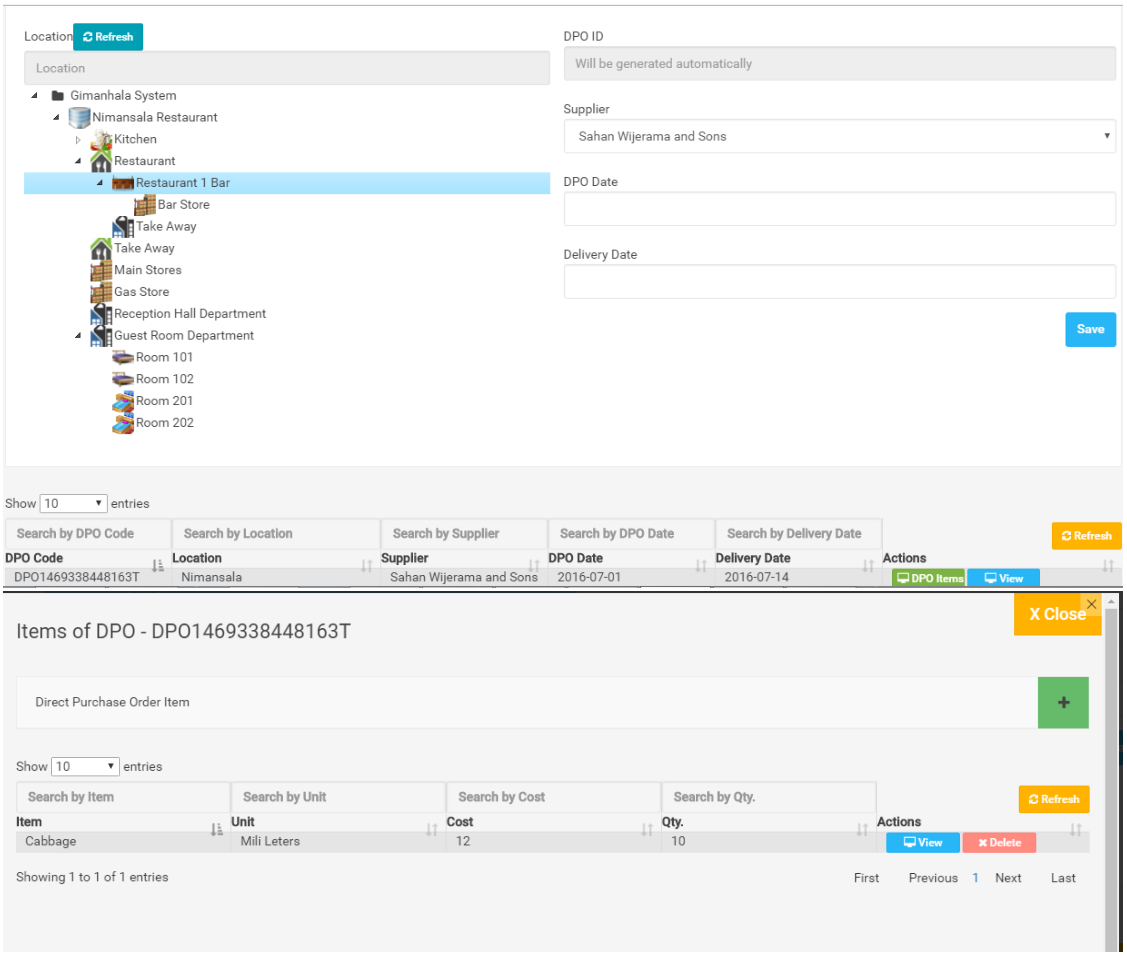

Items of GRN - GRN1469381599061T

Item	Unit	Cost	Qty.	Cost Price	Selling Price	Manu. Date	Exp. Date	Actions
Search by Item	Search by Unit	Search by Cost	Search by Qty.	Search by Cost Pri	Search by Selling I	Search by Manu. D	Search by Exp. Dat	
Cabbage	Kilo Grams	105.2	105.2	100.0	100.0			Save

Showing 1 to 1 of 1 entries

Figure C-15: Form to Add Item to GRN

- Direct Purchase Order.
  This is a fusion of above two notes to allow user to quickly enter the PO and GRN at the same time. This is suitable for ad-hoc purchasing done from various vendors



DPO ID: Will be generated automatically

Supplier: Sahan Wijerama and Sons

DPO Date:

Delivery Date:

DPO Code	Location	Supplier	DPO Date	Delivery Date	Actions
Search by DPO Code	Search by Location	Search by Supplier	Search by DPO Date	Search by Delivery Date	
DPO1469338448163T	Nimansala	Sahan Wijerama and Sons	2016-07-01	2016-07-14	DPO Items View

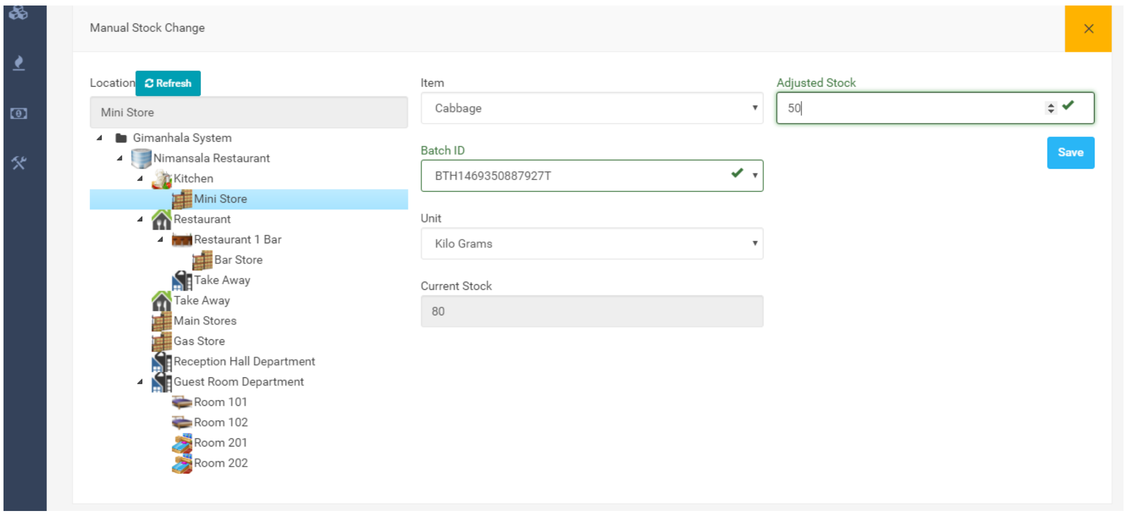
Items of DPO - DPO1469338448163T

Item	Unit	Cost	Qty.	Actions
Search by Item	Search by Unit	Search by Cost	Search by Qty.	
Cabbage	Mili Leters	12	10	View Delete

Showing 1 to 1 of 1 entries

Figure C-16: Direct Purchase Order Screens

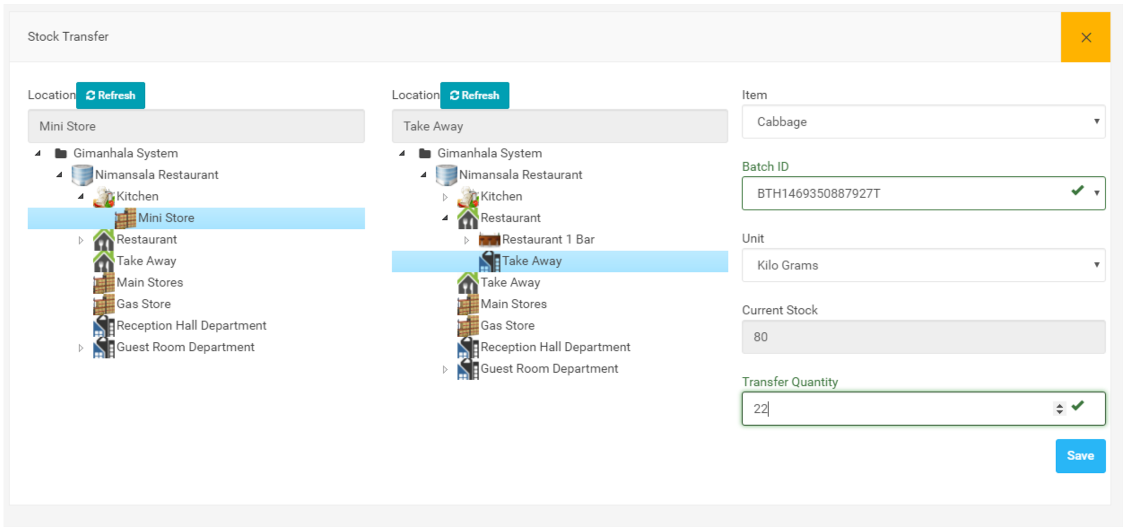
- Manual Stock Adjustments
  This feature was given to allow superior users to manually reconcile the stock in any justifiable situation
  Both increments and deductions are allowed in this


Manual Stock Change

Location	Mini Store
Item	Cabbage
Batch ID	BTH1469350887927T
Unit	Kilo Grams
Current Stock	80
Adjusted Stock	50

Figure C-17: Manual Stock Adjustment Form

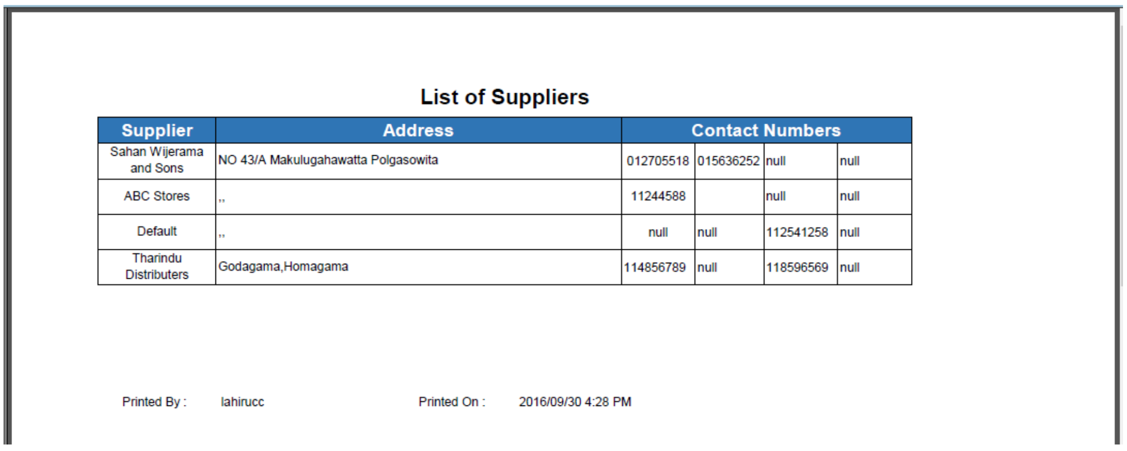
- Stock Transfer
  This interface can be used to move the stock between the stock locations.


Stock Transfer

Location	Mini Store
Location	Take Away
Item	Cabbage
Batch ID	BTH1469350887927T
Unit	Kilo Grams
Current Stock	80
Transfer Quantity	22

Figure C-18: Stock Transfer Form

Appendix D - Management Reports

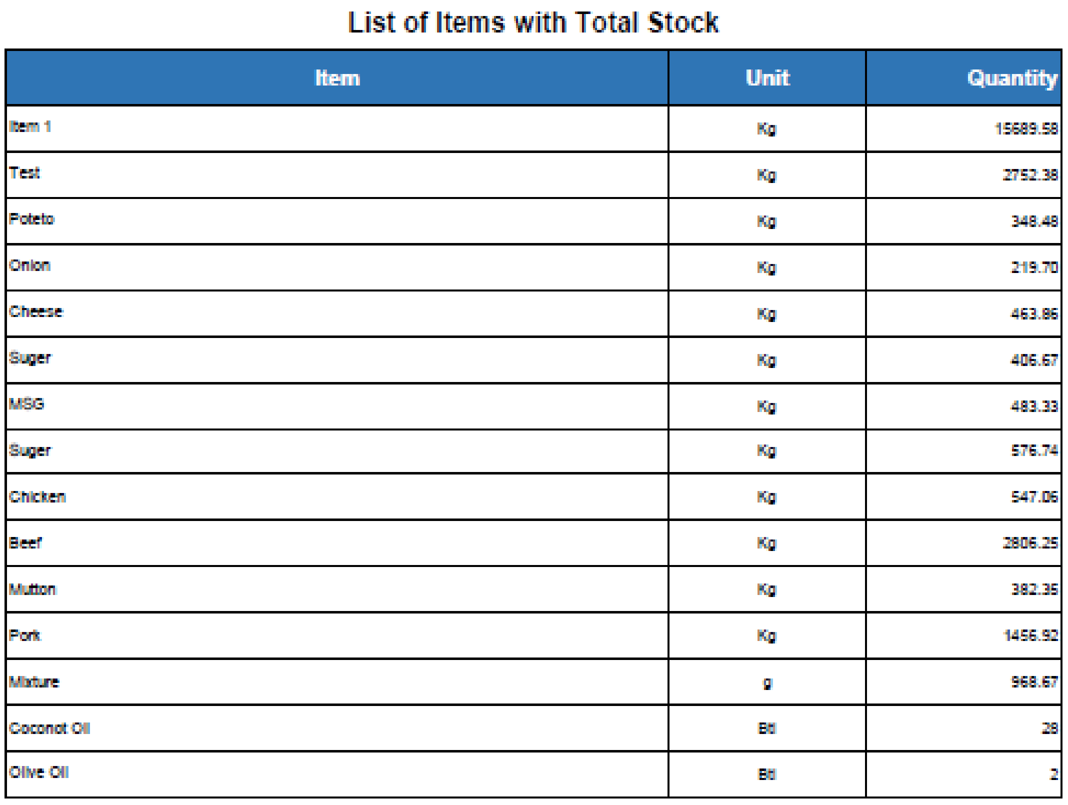


List of Suppliers

Supplier	Address	Contact Numbers			
Sahan Wijerama and Sons	NO 43/A Makulugahawatta Polgasowita	012705518	015636252	null	null
ABC Stores	,,	11244588		null	null
Default	,,	null	null	112541258	null
Tharindu Distributers	Godagama,Homagama	114856789	null	118596569	null

Printed By : lahirucc Printed On : 2016/09/30 4:28 PM

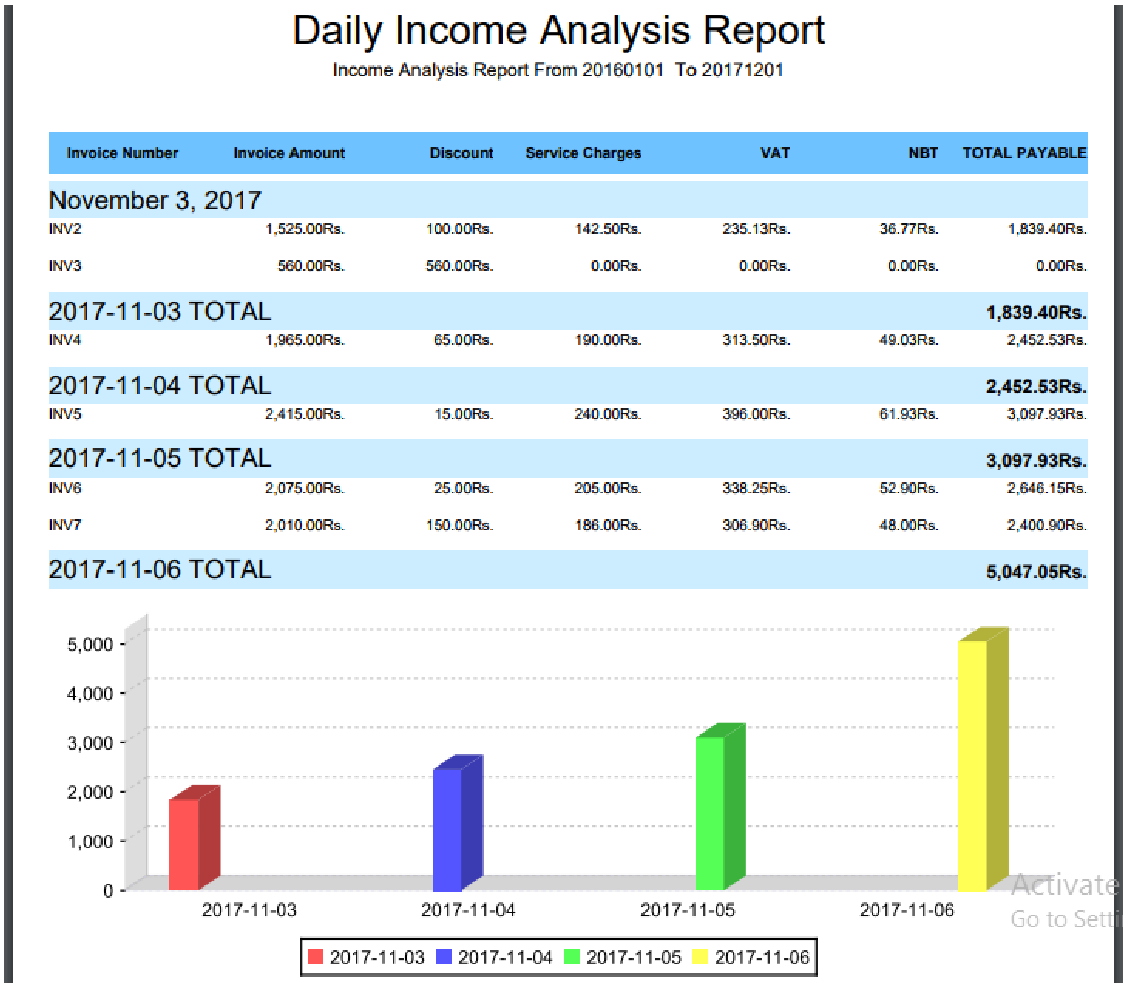
Figure D-1: List of Suppliers Report



List of Items with Total Stock

Item	Unit	Quantity
Item 1	Kg	15689.58
Test	Kg	2752.38
Poteto	Kg	348.48
Onion	Kg	219.70
Cheese	Kg	463.86
Suger	Kg	406.67
MSG	Kg	483.33
Suger	Kg	576.74
Chicken	Kg	547.06
Beef	Kg	2806.25
Mutton	Kg	382.35
Pork	Kg	1456.92
Mixture	g	968.67
Coconut Oil	Btl	28
Olive Oil	Btl	2

Figure D-2: Item List with Total Available Stock



Daily Income Analysis Report

Income Analysis Report From 20160101 To 20171201

Invoice Number	Invoice Amount	Discount	Service Charges	VAT	NBT	TOTAL PAYABLE
November 3, 2017						
INV2	1,525.00Rs.	100.00Rs.	142.50Rs.	235.13Rs.	36.77Rs.	1,839.40Rs.
INV3	560.00Rs.	560.00Rs.	0.00Rs.	0.00Rs.	0.00Rs.	0.00Rs.
2017-11-03 TOTAL						**1,839.40Rs.**
INV4	1,965.00Rs.	65.00Rs.	190.00Rs.	313.50Rs.	49.03Rs.	2,452.53Rs.
2017-11-04 TOTAL						**2,452.53Rs.**
INV5	2,415.00Rs.	15.00Rs.	240.00Rs.	396.00Rs.	61.93Rs.	3,097.93Rs.
2017-11-05 TOTAL						**3,097.93Rs.**
INV6	2,075.00Rs.	25.00Rs.	205.00Rs.	338.25Rs.	52.90Rs.	2,646.15Rs.
INV7	2,010.00Rs.	150.00Rs.	186.00Rs.	306.90Rs.	48.00Rs.	2,400.90Rs.
2017-11-06 TOTAL						**5,047.05Rs.**

Figure D-3: Daily Income Report

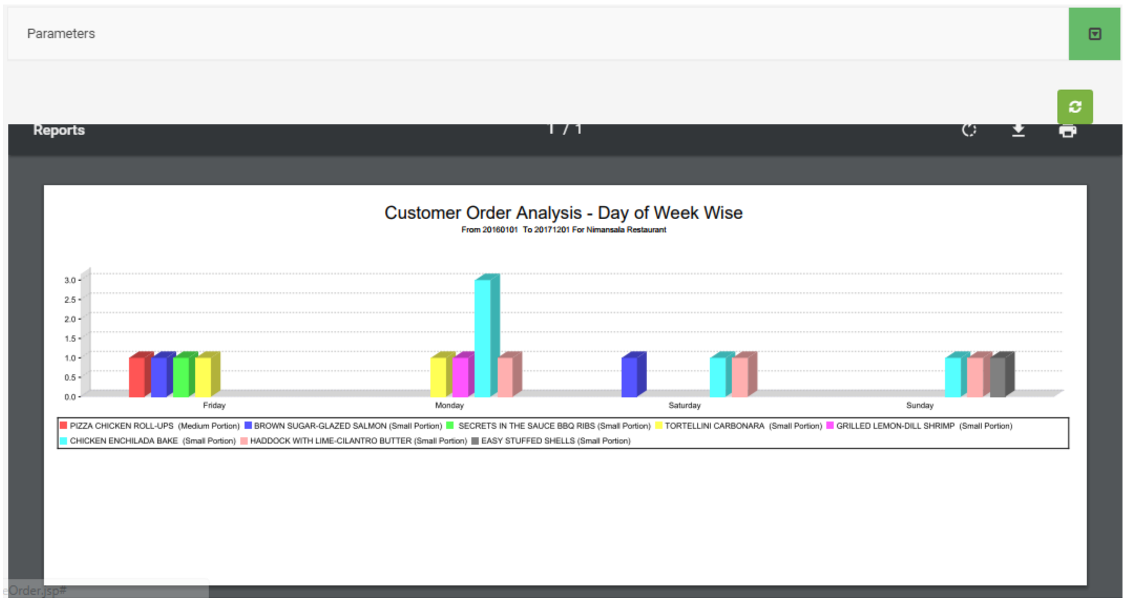
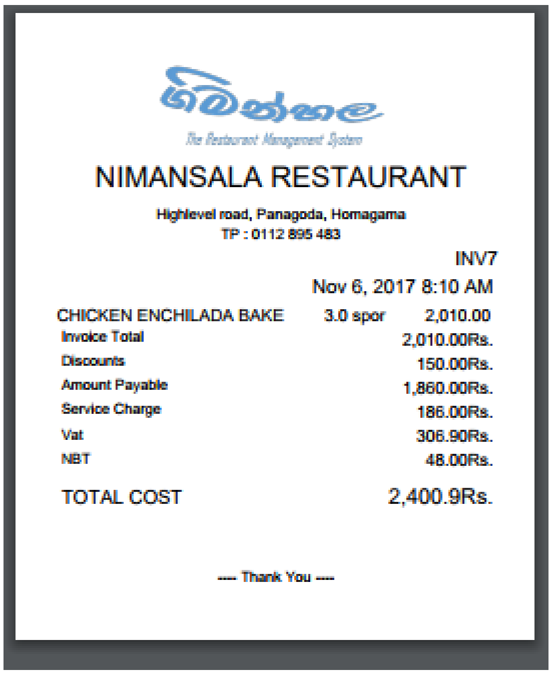


Figure D-4: Customer Order Analysis Report



The Restaurant Management System

NIMANSALA RESTAURANT

Highlevel road, Panagoda, Homagama
TP : 0112 895 483

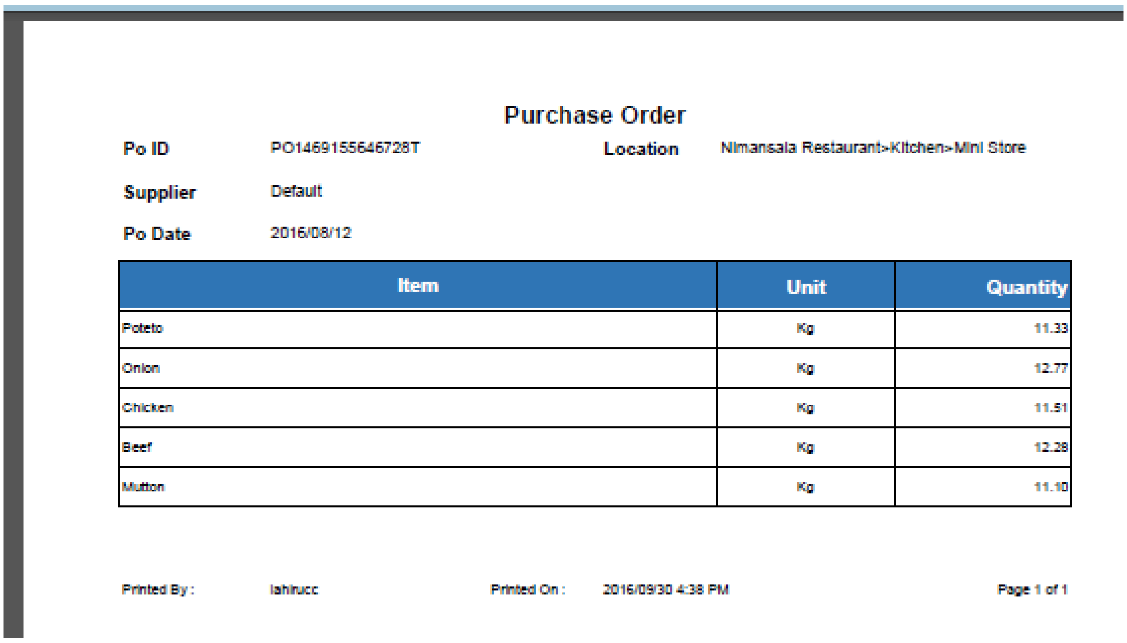
INV7

Nov 6, 2017 8:10 AM

CHICKEN ENCHILADA BAKE	3.0 spor	2,010.00
Invoice Total		2,010.00Rs.
Discounts		150.00Rs.
Amount Payable		1,860.00Rs.
Service Charge		186.00Rs.
Vat		306.90Rs.
NBT		48.00Rs.
TOTAL COST		2,400.9Rs.

---- Thank You ----

Figure D-5: Invoice Print



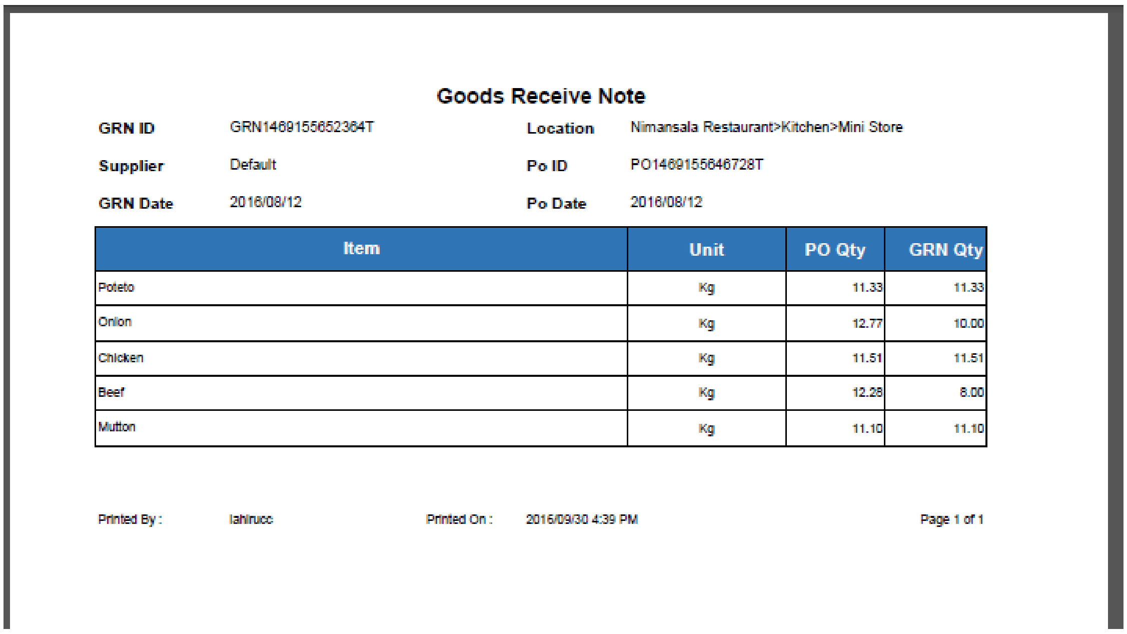
Purchase Order

Po ID	PO1469155646728T	Location	Nimansala Restaurant->Kitchen->Mini Store
Supplier	Default		
Po Date	2016/08/12		

Item	Unit	Quantity
Poteto	Kg	11.33
Onion	Kg	12.77
Chicken	Kg	11.51
Beef	Kg	12.28
Mutton	Kg	11.10

Printed By : lahirucc Printed On : 2016/09/30 4:38 PM Page 1 of 1

Figure D-6: Purchase Order Print



Goods Receive Note

GRN ID	GRN1469155652364T	Location	Nimansala Restaurant>Kitchen>Mini Store
Supplier	Default	Po ID	PO1469155646728T
GRN Date	2016/08/12	Po Date	2016/08/12

Item	Unit	PO Qty	GRN Qty
Poteto	Kg	11.33	11.33
Onion	Kg	12.77	10.00
Chicken	Kg	11.51	11.51
Beef	Kg	12.28	8.00
Mutton	Kg	11.10	11.10

Printed By : lahirucc Printed On : 2016/09/30 4:39 PM Page 1 of 1

Figure D-7: Goods Receive Note Print

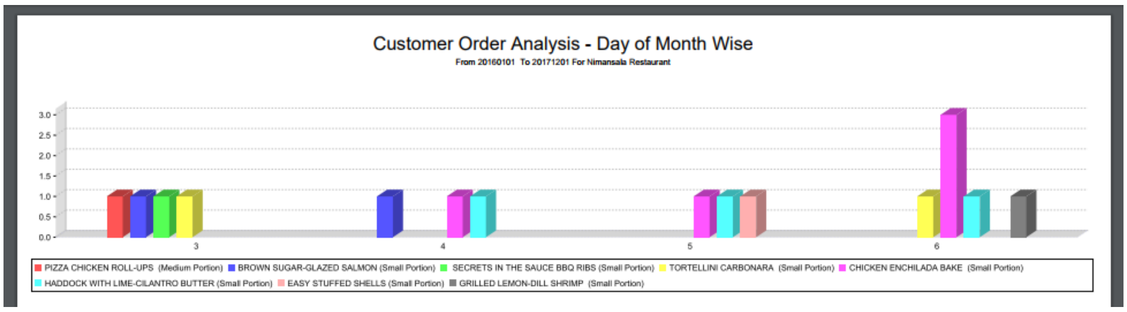


Figure D-8: Customer Order Analysis Report –Day of the month

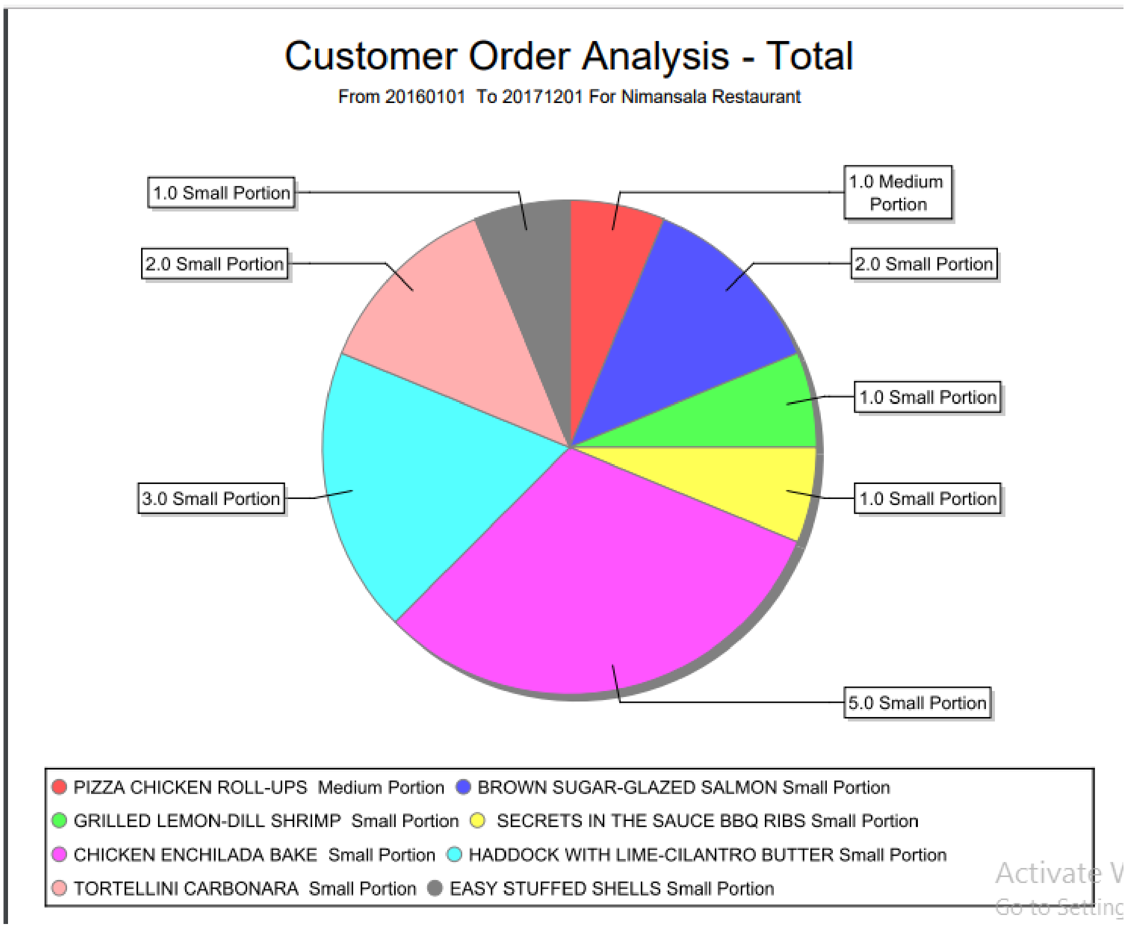


Figure D-9: Customer Order Analysis Report - Total

Appendix E – Test Results

Condition	Pass
User Groups	
Can User Group List Be Viewed	Yes
Can User Group Grid be filtered	Yes
Is the list blocked for un authorized users	Yes
Can a new user group be created	Yes
Is the uniqueness of group name is validated	No
Can a record be viewed	Yes
Can a Record be amended	Yes
Can a user group be deleted	Yes
Can the Rights be viewed	Yes
Can the Rights be Granted	Yes
Can the Rights be Revoked	Yes
Can the user group be deactivated	Yes
Can the user group be activated	Yes
Users	
Can User List be viewed	Yes
Can user grid be filtered	Yes
Can a user information be viewed	Yes
Can a new user be created	Yes
Is the welcome email generated	Yes
Is the uniqueness of the username validated	Yes
Can a user be activated	Yes
Can a user be deactivated	Yes
Can a user be deleted	Yes
Can the user information be amended	Yes
Can the password be reset	Yes
Can the password be sent to email	Yes
Can the password be reset with auto generated password	Yes
Can the user be assigned to multiple user group	No
Are the user activities are Audited	Yes
Does the system restrict login for wrong credentials	Yes
Does the system allow to login with right credentials	Yes
Does the system recognize the user after entering the username	Yes
Does the login form validated for blank fields	Yes
Is the menu filtered according to the rights	Yes

Location Type	
Can Location Type List be viewed	Yes
Can a location Type be viewed/ amended / deleted	Yes/Yes/Yes
Is it possible to upload an icon	Yes
Location Type	
Can the Location Structure be viewed as a tree structure	Yes
Can the Structure be re arranged by dragging and dropping	Yes
Can the Structure be Copied and Pasted by using the context menu	Yes
Can a Node Be Renamed using the context menu	Yes
Can a Node be viewed in form	Yes
Does the Icon picked from location type	Yes
Does the system restrict saving two nodes in same name under same parent	No

Units	
Can Unit List be viewed	Yes
Can a Unit be viewed/ amended / deleted	Yes/Yes/Yes
Suppliers	
Can Supplier List be viewed	Yes
Can a be viewed/ amended / deleted	Yes/Yes/Yes

Items	
Can Item List be viewed	Yes
Can the Item grid be filtered	Yes
Can Item be viewed/ amended / deleted	Yes/Yes/Yes
Can the current stock of item be viewed	Yes
Can the Current batches of an item be viewed	Yes
Can the re-order level and expiry notification level be set	Yes
Is the auto generated messages received	No
Can Image be uploaded for Item	Yes
Can multiple units and conversion ratio be configured for items	Yes

Item Batch	
Can Item Batch List be viewed	Yes
Can Item Batch be deleted	Yes
Is the stock adjusted when a batch deleted	Yes
Can a new Item Batch be created	Yes
Is the stock updated with new batch	Yes
Can the location of batch be changed	Yes
Is the stock moved when location changed	Yes

Purchase Order	
Can view the previous PO list	Yes
Can view single PO	Yes
Can view item details of PO	Yes
Can PO be edited before goods received	Yes
Can PO Items be added/ removed/ amended	Yes/Yes/Yes
Does the system restrict the PO be edited after goods received	Yes
Can a PO be printed	Yes
Goods Receive Note	
Can view the previous GRNs	Yes
Can view the pending PO	Yes
Are the received PO hidden	Yes
Can Amounts be given for Receiving Items	Yes
Can Manufacture and Expiry Dates Be Setup	Yes
Are the batches created when placed a GRN	Yes
Can a GRN be Edited	Yes
Can a GRN be Deleted	Yes
Does the system revert the created batches when GRN amended or deleted	Yes
Can items be added to existing batches	No
Can the GRN be printed	Yes

Appendix F - Code Listing

Stores Manager Service is the Core of this system. It manages the most of the stores related functionalities.

```
package com.gimanhala.services;

import java.text.Normalizer.Form;

import com.gimanhala.dto.DPO;
import com.gimanhala.dto.DPOItem;
import com.gimanhala.dto.GRN;
import com.gimanhala.dto.GRNItem;
import com.gimanhala.dto.ItemBatch;
import com.gimanhala.dto.POItem;
import com.gimanhala.dto.PurchaseOrder;
import com.gimanhala.dto.ServiceResponse;
import com.gimanhala.util.Format;
import com.gimanhala.util.HTMLUtils;
import com.gimanhala.util.NewId;
import com.gimanhala.util.Settings;
import com.gimnhala.dao.company_location_type;
import com.gimnhala.dao.item_batch;
import com.gimnhala.dao.item_stock;
import com.gimnhala.dao.stores_dpo;
import com.gimnhala.dao.stores_dpo_details;
import com.gimnhala.dao.stores_grn;
import com.gimnhala.dao.stores_grn_details;
import com.gimnhala.dao.stores_po;
import com.gimnhala.dao.stores_po_details;

public class StockTransactionService {

	// ////////////////////// PURCHASE ORDER
	// ////////////////////////////////////////
	public static ServiceResponse getPOList() {
		ServiceResponse response = new ServiceResponse();
		try {
			DBManagerService.connect();
			PurchaseOrder[] pos = stores_po
					.getPurchaseOrdersForQuery("where deleted=0");
			response.setObject(pos);
			DBManagerService.closeConnection();
			if (pos.length > 0) {
				response.setSuccess(true);
				response.setMessageType(ServiceResponse.INFO);
				response.setMessageHeading("PO Retrieved");
				response.setMessage("!");
			} else {
				response.setSuccess(false);
				response.setMessageType(ServiceResponse.WARNING);
				response.setMessageHeading("No POs Defined");
				response.setMessage("Please add POs");
			}
		} catch (Exception e) {
			response.setSuccess(true);
			response.setMessageType(ServiceResponse.FAILED);
			response.setMessageHeading("Error in Loading Location Types!");
			if (Settings.sqlErrosInUI) {
				response.setMessage(e.getMessage());
			} else {
				response.setMessage("Server Error");
			}
		}
		return response;
	}

	public static ServiceResponse getPO(String id) {
		PurchaseOrder po;
		ServiceResponse response = new ServiceResponse();
		try {
```

```
                        DBManagerService.connect();
                        po = stores_po.getPurchaseOrderById(id);
                        DBManagerService.closeConnection();
                        response.setSuccess(true);
                        response.setMessageType(ServiceResponse.SUCCESS);
                        response.setMessageHeading("PO Retrieved");
                        response.setMessage(po.getName());
                        response.setObject(po);
                } catch (Exception e) {
                        response.setSuccess(true);
                        response.setMessageType(ServiceResponse.FAILED);
                        response.setMessageHeading("Error!");
                        if (Settings.sqlErrosInUI) {
                                response.setMessage(e.getMessage());
                        } else {
                                response.setMessage("Server Error");
                        }
                }
                return response;
        }

        public static ServiceResponse insertPO(PurchaseOrder po) {
                ServiceResponse response = new ServiceResponse();
                try {
                        DBManagerService.connect();
                        po.setName(NewId.GetId("PO"));
                        stores_po.insertRecord(po);
                        ;
                        DBManagerService.closeConnection();
                        response.setSuccess(true);
                        response.setMessageType(ServiceResponse.SUCCESS);
                        response.setMessageHeading("PO Saved!");
                        response.setMessage(po.getName());
                        response.setObject(po);
                } catch (Exception e) {
                        response.setSuccess(true);
                        response.setMessageType(ServiceResponse.FAILED);
                        response.setMessageHeading("Error!");
                        if (Settings.sqlErrosInUI) {
                                response.setMessage(e.getMessage());
                        } else {
                                response.setMessage("Server Error");
                        }
                }
                return response;
        }

        public static ServiceResponse updatePO(PurchaseOrder po) {
                ServiceResponse response = new ServiceResponse();
                try {
                        DBManagerService.connect();
                        stores_po.updateRecord(po);
                        DBManagerService.closeConnection();
                        response.setSuccess(true);
                        response.setMessageType(ServiceResponse.SUCCESS);
                        response.setMessageHeading("PO Updated!");
                        response.setMessage(po.getName());
                        response.setObject(po);
                } catch (Exception e) {
                        response.setSuccess(true);
                        response.setMessageType(ServiceResponse.FAILED);
                        response.setMessageHeading("Error!");
                        if (Settings.sqlErrosInUI) {
                                response.setMessage(e.getMessage());
                        } else {
                                response.setMessage("Server Error");
                        }
                }
                return response;
        }

        public static ServiceResponse deletePO(String id) {
                ServiceResponse response = new ServiceResponse();
                try {
                        DBManagerService.connect();
                        stores_po.deleteRecord(id);
                DBManagerService.closeConnection();
                response.setSuccess(true);
                response.setMessageType(ServiceResponse.SUCCESS);
                response.setMessageHeading("PO Deleted");
                response.setMessage("!");
        } catch (Exception e) {
                response.setSuccess(true);
                response.setMessageType(ServiceResponse.FAILED);
                response.setMessageHeading("Error!");
                if (Settings.sqlErrosInUI) {
                        response.setMessage(e.getMessage());
                } else {
                        response.setMessage("Server Error");
                }
        }
        return response;
}

public static ServiceResponse getPOByName(String refName) {
        PurchaseOrder po;
        ServiceResponse response = new ServiceResponse();
        try {
                DBManagerService.connect();
                po = stores_po.getPurchaseOrderByName(refName);
                DBManagerService.closeConnection();
                response.setSuccess(true);
                response.setMessageType(ServiceResponse.SUCCESS);
                response.setMessageHeading("PO Retrieved");
                response.setMessage(po.getName());
                response.setObject(po);
        } catch (Exception e) {
                response.setSuccess(true);
                response.setMessageType(ServiceResponse.FAILED);
                response.setMessageHeading("Error!");
                if (Settings.sqlErrosInUI) {
                        response.setMessage(e.getMessage());
                } else {
                        response.setMessage("Server Error");
                }
        }
        return response;
}

// //////////////////////////////////// POITEMS

public static ServiceResponse insertPOItem(POItem item) {
        ServiceResponse response = new ServiceResponse();
        try {
                DBManagerService.connect();
                stores_po_details.insertRecord(item);
                ;
                DBManagerService.closeConnection();
                response.setSuccess(true);
                response.setMessageType(ServiceResponse.SUCCESS);
                response.setMessageHeading("PO Item Saved!");
                response.setMessage("");
                response.setObject(item);
        } catch (Exception e) {
                response.setSuccess(true);
                response.setMessageType(ServiceResponse.FAILED);
                response.setMessageHeading("Error!");
                if (Settings.sqlErrosInUI) {
                        response.setMessage(e.getMessage());
                } else {
                        response.setMessage("Server Error");
                }
        }
        return response;
}

public static ServiceResponse updatePOItem(POItem item) {
        ServiceResponse response = new ServiceResponse();
        try {
                DBManagerService.connect();
                stores_po_details.updateRecord(item);
                DBManagerService.closeConnection();
```

```
				response.setSuccess(true);
				response.setMessageType(ServiceResponse.SUCCESS);
				response.setMessageHeading("PO Item Updated!");
				response.setMessage("");
				response.setObject(item);
			} catch (Exception e) {
				response.setSuccess(true);
				response.setMessageType(ServiceResponse.FAILED);
				response.setMessageHeading("Error!");
				if (Settings.sqlErrosInUI) {
					response.setMessage(e.getMessage());
				} else {
					response.setMessage("Server Error");
				}
			}
			return response;
		}

		public static ServiceResponse deletePOItem(String poId, String itemId) {
			ServiceResponse response = new ServiceResponse();
			try {
				DBManagerService.connect();
				stores_po_details.deleteRecord(poId, itemId);
				DBManagerService.closeConnection();
				response.setSuccess(true);
				response.setMessageType(ServiceResponse.SUCCESS);
				response.setMessageHeading("PO Item Deleted");
				response.setMessage("!");
			} catch (Exception e) {
				response.setSuccess(true);
				response.setMessageType(ServiceResponse.FAILED);
				response.setMessageHeading("Error!");
				if (Settings.sqlErrosInUI) {
					response.setMessage(e.getMessage());
				} else {
					response.setMessage("Server Error");
				}
			}
			return response;
		}

		public static ServiceResponse getPOItemList(String poId) {
			ServiceResponse response = new ServiceResponse();
			try {
				DBManagerService.connect();
				POItem[] pos = stores_po_details.getPoItems(poId);
				response.setObject(pos);
				DBManagerService.closeConnection();
				if (pos.length > 0) {
					response.setSuccess(true);
					response.setMessageType(ServiceResponse.INFO);
					response.setMessageHeading("PO Items Retrieved");
					response.setMessage("!");
				} else {
					response.setSuccess(false);
					response.setMessageType(ServiceResponse.WARNING);
					response.setMessageHeading("No PO Items Defined");
					response.setMessage("Please add PO Items");
				}
			} catch (Exception e) {
				response.setSuccess(true);
				response.setMessageType(ServiceResponse.FAILED);
				response.setMessageHeading("Error in Loading PO Items!");
				if (Settings.sqlErrosInUI) {
					response.setMessage(e.getMessage());
				} else {
					response.setMessage("Server Error");
				}
			}
			return response;
		}

		public static ServiceResponse getPOItem(String poId, String itemId) {
			POItem poi;
			ServiceResponse response = new ServiceResponse();
			try {
```

```
                        DBManagerService.connect();
                        poi = stores_po_details.getPOItemById(poId, itemId);
                        DBManagerService.closeConnection();
                        response.setSuccess(true);
                        response.setMessageType(ServiceResponse.SUCCESS);
                        response.setMessageHeading("PO Item Retrieved");
                        response.setMessage("");
                        response.setObject(poi);
                } catch (Exception e) {
                        response.setSuccess(true);
                        response.setMessageType(ServiceResponse.FAILED);
                        response.setMessageHeading("Error!");
                        if (Settings.sqlErrosInUI) {
                                response.setMessage(e.getMessage());
                        } else {
                                response.setMessage("Server Error");
                        }
                }
                return response;
        }

        // ////////////////////////////////////// DIRECT PURCHASE ORDER
        // //////////////////////////////////////////
        public static ServiceResponse getDPOList() {
                ServiceResponse response = new ServiceResponse();
                try {
                        DBManagerService.connect();
                        DPO[] pos = stores_dpo.getDPOsForQuery("where deleted=0");
                        response.setObject(pos);
                        DBManagerService.closeConnection();
                        if (pos.length > 0) {
                                response.setSuccess(true);
                                response.setMessageType(ServiceResponse.INFO);
                                response.setMessageHeading("DPO Retrieved");
                                response.setMessage("!");
                        } else {
                                response.setSuccess(false);
                                response.setMessageType(ServiceResponse.WARNING);
                                response.setMessageHeading("No DPOs Defined");
                                response.setMessage("Please add DPOs");
                        }
                } catch (Exception e) {
                        response.setSuccess(true);
                        response.setMessageType(ServiceResponse.FAILED);
                        response.setMessageHeading("Error in Loading DPO!");
                        if (Settings.sqlErrosInUI) {
                                response.setMessage(e.getMessage());
                        } else {
                                response.setMessage("Server Error");
                        }
                }
                return response;
        }

        public static ServiceResponse getDPO(String id) {
                DPO dpo;
                ServiceResponse response = new ServiceResponse();
                try {
                        DBManagerService.connect();
                        dpo = stores_dpo.getDPOById(id);
                        DBManagerService.closeConnection();
                        response.setSuccess(true);
                        response.setMessageType(ServiceResponse.SUCCESS);
                        response.setMessageHeading("DPO Retrieved");
                        response.setMessage(dpo.getName());
                        response.setObject(dpo);
                } catch (Exception e) {
                        response.setSuccess(true);
                        response.setMessageType(ServiceResponse.FAILED);
                        response.setMessageHeading("Error!");
                        if (Settings.sqlErrosInUI) {
                                response.setMessage(e.getMessage());
                        } else {
                                response.setMessage("Server Error");
                        }
                }
```

```
		return response;
	}

	public static ServiceResponse insertDPO(DPO dpo) {
		ServiceResponse response = new ServiceResponse();
		try {
			DBManagerService.connect();
			dpo.setName(NewId.GetId("DPO"));
			stores_dpo.insertRecord(dpo);
			DBManagerService.closeConnection();
			response.setSuccess(true);
			response.setMessageType(ServiceResponse.SUCCESS);
			response.setMessageHeading("DPO Saved!");
			response.setMessage(dpo.getName());
			response.setObject(dpo);
		} catch (Exception e) {
			response.setSuccess(true);
			response.setMessageType(ServiceResponse.FAILED);
			response.setMessageHeading("Error!");
			if (Settings.sqlErrosInUI) {
				response.setMessage(e.getMessage());
			} else {
				response.setMessage("Server Error");
			}
		}
		return response;
	}

	public static ServiceResponse updateDPO(DPO dpo) {
		ServiceResponse response = new ServiceResponse();
		try {
			DBManagerService.connect();
			stores_dpo.updateRecord(dpo);
			DBManagerService.closeConnection();
			response.setSuccess(true);
			response.setMessageType(ServiceResponse.SUCCESS);
			response.setMessageHeading("DPO Updated!");
			response.setMessage(dpo.getName());
			response.setObject(dpo);
		} catch (Exception e) {
			response.setSuccess(true);
			response.setMessageType(ServiceResponse.FAILED);
			response.setMessageHeading("Error!");
			if (Settings.sqlErrosInUI) {
				response.setMessage(e.getMessage());
			} else {
				response.setMessage("Server Error");
			}
		}
		return response;
	}

	public static ServiceResponse deleteDPO(String id) {
		ServiceResponse response = new ServiceResponse();
		try {
			DBManagerService.connect();
			stores_po.deleteRecord(id);
			DBManagerService.closeConnection();
			response.setSuccess(true);
			response.setMessageType(ServiceResponse.SUCCESS);
			response.setMessageHeading("PO Deleted");
			response.setMessage("!");
		} catch (Exception e) {
			response.setSuccess(true);
			response.setMessageType(ServiceResponse.FAILED);
			response.setMessageHeading("Error!");
			if (Settings.sqlErrosInUI) {
				response.setMessage(e.getMessage());
			} else {
				response.setMessage("Server Error");
			}
		}
		return response;
	}

	public static ServiceResponse getDPOByName(String refName) {
		DPO dpo;
		ServiceResponse response = new ServiceResponse();
		try {
			DBManagerService.connect();
			dpo = stores_dpo.getDPOByName(refName);
			DBManagerService.closeConnection();
			response.setSuccess(true);
			response.setMessageType(ServiceResponse.SUCCESS);
			response.setMessageHeading("DPO Retrieved");
			response.setMessage(dpo.getName());
			response.setObject(dpo);
		} catch (Exception e) {
			response.setSuccess(true);
			response.setMessageType(ServiceResponse.FAILED);
			response.setMessageHeading("Error!");
			if (Settings.sqlErrosInUI) {
				response.setMessage(e.getMessage());
			} else {
				response.setMessage("Server Error");
			}
		}
		return response;
	}

	// ///////////////////////////////// POITEMS

	public static ServiceResponse insertDPOItem(DPOItem item) {
		ServiceResponse response = new ServiceResponse();
		try {
			DBManagerService.connect();
			String bthId = NewId.GetId("BTH");
			item.setId(bthId);
			stores_dpo_details.insertRecord(item);
			item_batch.insertRecord(item);
			item_stock.setStock(bthId, item.getLocationId(), item.getUnitId(),
					item.getQuantity(), "" + item.getDpoId(),
					"Direct Purchase Order", "3");
			DBManagerService.closeConnection();
			response.setSuccess(true);
			response.setMessageType(ServiceResponse.SUCCESS);
			response.setMessageHeading("DPO Item Saved!");
			response.setMessage("");
			response.setObject(item);
		} catch (Exception e) {
			response.setSuccess(true);
			response.setMessageType(ServiceResponse.FAILED);
			response.setMessageHeading("Error!");
			if (Settings.sqlErrosInUI) {
				response.setMessage(e.getMessage());
			} else {
				response.setMessage("Server Error");
			}
		}
		return response;
	}

	public static ServiceResponse updateDPOItem(DPOItem item) {
		ServiceResponse response = new ServiceResponse();
		try {
			DBManagerService.connect();
			stores_dpo_details.updateRecord(item);

			Format.jsonOutput(item);

			item_batch.updateRecord(item);
			item_stock.setStock(item.getId(), item.getLocationId(),
					item.getUnitId(), item.getQuantity(), "" + item.getDpoId(),
					"Direct Purchase Order Update", "4");
			DBManagerService.closeConnection();
			response.setSuccess(true);
			response.setMessageType(ServiceResponse.SUCCESS);
			response.setMessageHeading("DPO Item Updated!");
			response.setMessage("");
			response.setObject(item);
		} catch (Exception e) {
			response.setSuccess(true);
			response.setMessageType(ServiceResponse.FAILED);
			response.setMessageHeading("Error!");
			if (Settings.sqlErrosInUI) {
				response.setMessage(e.getMessage());
			} else {
				response.setMessage("Server Error");
			}
		}
		return response;
	}

	public static ServiceResponse deleteDPOItem(String dpoId, String itemId) {
		ServiceResponse response = new ServiceResponse();
		try {
			DBManagerService.connect();
			stores_dpo_details.deleteRecord(dpoId, itemId);
			DBManagerService.closeConnection();
			response.setSuccess(true);
			response.setMessageType(ServiceResponse.SUCCESS);
			response.setMessageHeading("DPO Item Deleted");
			response.setMessage("!");
		} catch (Exception e) {
			response.setSuccess(true);
			response.setMessageType(ServiceResponse.FAILED);
			response.setMessageHeading("Error!");
			if (Settings.sqlErrosInUI) {
				response.setMessage(e.getMessage());
			} else {
				response.setMessage("Server Error");
			}
		}
		return response;
	}

	public static ServiceResponse getDPOItemList(String dpoId) {
		ServiceResponse response = new ServiceResponse();
		try {
			DBManagerService.connect();
			DPOItem[] dpos = stores_dpo_details.getPoItems(dpoId);
			response.setObject(dpos);
			DBManagerService.closeConnection();
			if (dpos.length > 0) {
				response.setSuccess(true);
				response.setMessageType(ServiceResponse.INFO);
				response.setMessageHeading("DPO Items Retrieved");
				response.setMessage("!");
			} else {
				response.setSuccess(false);
				response.setMessageType(ServiceResponse.WARNING);
				response.setMessageHeading("No DPO Items Defined");
				response.setMessage("Please add DPO Items");
			}
		} catch (Exception e) {
			response.setSuccess(true);
			response.setMessageType(ServiceResponse.FAILED);
			response.setMessageHeading("Error in Loading DPO Items!");
			if (Settings.sqlErrosInUI) {
				response.setMessage(e.getMessage());
			} else {
				response.setMessage("Server Error");
			}
		}
		return response;
	}

	public static ServiceResponse getDPOItem(String dpoId, String itemId) {
		DPOItem dpoi;
		ServiceResponse response = new ServiceResponse();
		try {
			DBManagerService.connect();
			dpoi = stores_dpo_details.getDPOItemById(dpoId, itemId);
			DBManagerService.closeConnection();
			response.setSuccess(true);
			response.setMessageType(ServiceResponse.SUCCESS);
			response.setMessageHeading("DPO Item Retrieved");
			response.setMessage("");
```

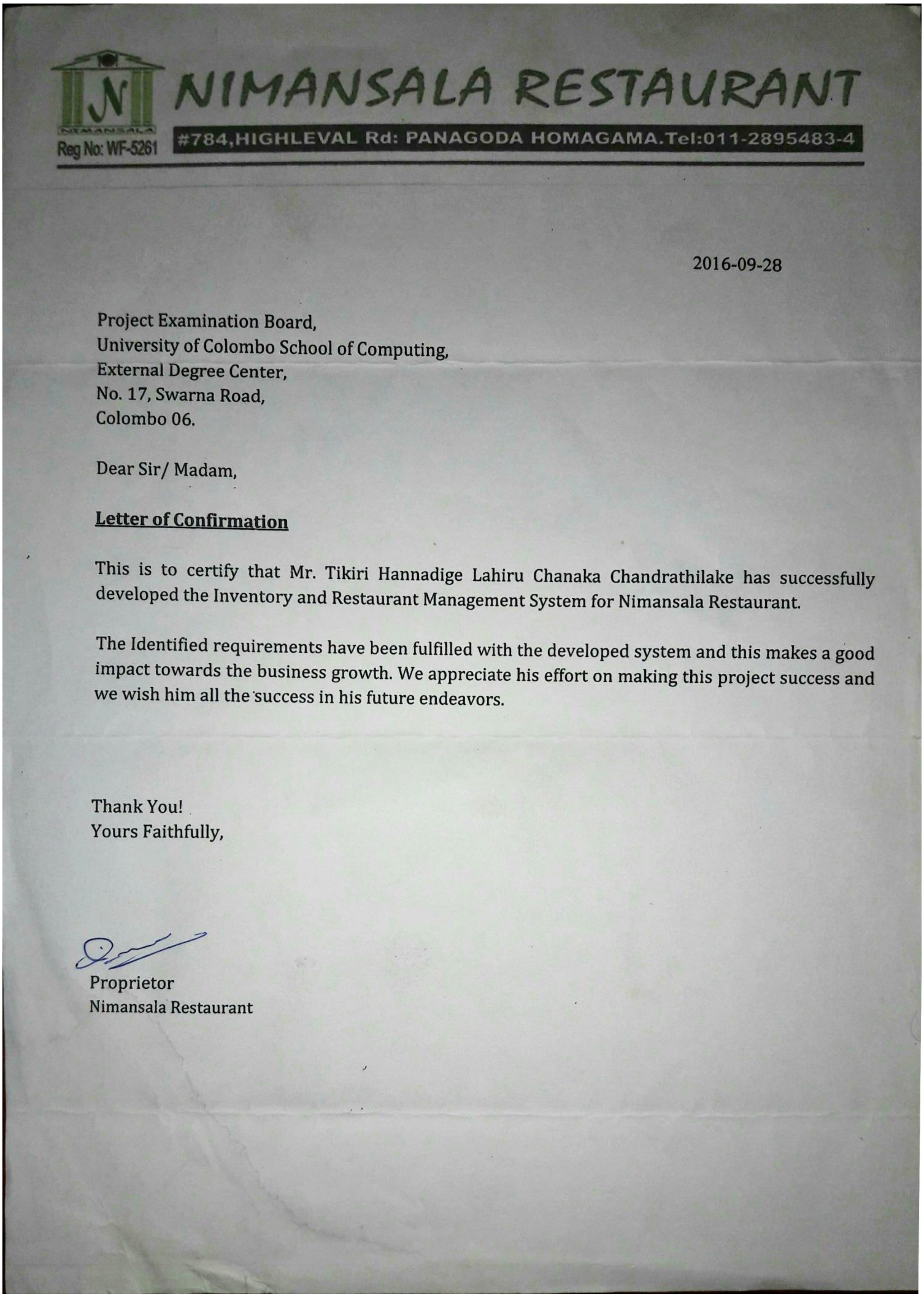
```
				response.setObject(dpoi);
			} catch (Exception e) {
				response.setSuccess(true);
				response.setMessageType(ServiceResponse.FAILED);
				response.setMessageHeading("Error!");
				if (Settings.sqlErrosInUI) {
					response.setMessage(e.getMessage());
				} else {
					response.setMessage("Server Error");
				}
			}
			return response;
	}

	// ////////////////////////////////////// GOOD RECEIVE NOTE
	// ////////////////////////////////////////
	public static ServiceResponse getGRNList() {
			ServiceResponse response = new ServiceResponse();
			try {
				DBManagerService.connect();
				GRN[] pos = stores_grn.getGRNsForQuery("where deleted=0");
				response.setObject(pos);
				DBManagerService.closeConnection();
				if (pos.length > 0) {
					response.setSuccess(true);
					response.setMessageType(ServiceResponse.INFO);
					response.setMessageHeading("GRN Retrieved");
					response.setMessage("!");
				} else {
					response.setSuccess(false);
					response.setMessageType(ServiceResponse.WARNING);
					response.setMessageHeading("No GRNs Defined");
					response.setMessage("Please add GRNs");
				}
			} catch (Exception e) {
				response.setSuccess(true);
				response.setMessageType(ServiceResponse.FAILED);
				response.setMessageHeading("Error in Loading GRN!");
				if (Settings.sqlErrosInUI) {
					response.setMessage(e.getMessage());
				} else {
					response.setMessage("Server Error");
				}
			}
			return response;
	}

	public static ServiceResponse getGRN(String id) {
			GRN GRN;
			ServiceResponse response = new ServiceResponse();
			try {
				DBManagerService.connect();
				GRN = stores_grn.getGRNById(id);
				DBManagerService.closeConnection();
				response.setSuccess(true);
				response.setMessageType(ServiceResponse.SUCCESS);
				response.setMessageHeading("GRN Retrieved");
				response.setMessage(GRN.getName());
				response.setObject(GRN);
			} catch (Exception e) {
				response.setSuccess(true);
				response.setMessageType(ServiceResponse.FAILED);
				response.setMessageHeading("Error!");
				if (Settings.sqlErrosInUI) {
					response.setMessage(e.getMessage());
				} else {
					response.setMessage("Server Error");
				}
			}
			return response;
	}

	public static ServiceResponse insertGRN(GRN GRN) {
			ServiceResponse response = new ServiceResponse();
			try {
				DBManagerService.connect();
```

```
                GRN.setName(NewId.GetId("GRN"));
                stores_grn.insertRecord(GRN);
                DBManagerService.closeConnection();
                response.setSuccess(true);
                response.setMessageType(ServiceResponse.SUCCESS);
                response.setMessageHeading("GRN Saved!");
                response.setMessage(GRN.getName());
                response.setObject(GRN);
        } catch (Exception e) {
                response.setSuccess(true);
                response.setMessageType(ServiceResponse.FAILED);
                response.setMessageHeading("Error!");
                if (Settings.sqlErrosInUI) {
                        response.setMessage(e.getMessage());
                } else {
                        response.setMessage("Server Error");
                }
        }
        return response;
}

public static ServiceResponse updateGRN(GRN GRN) {
        ServiceResponse response = new ServiceResponse();
        try {
                DBManagerService.connect();
                stores_grn.updateRecord(GRN);
                DBManagerService.closeConnection();
                response.setSuccess(true);
                response.setMessageType(ServiceResponse.SUCCESS);
                response.setMessageHeading("GRN Updated!");
                response.setMessage(GRN.getName());
                response.setObject(GRN);
        } catch (Exception e) {
                response.setSuccess(true);
                response.setMessageType(ServiceResponse.FAILED);
                response.setMessageHeading("Error!");
                if (Settings.sqlErrosInUI) {
                        response.setMessage(e.getMessage());
                } else {
                        response.setMessage("Server Error");
                }
        }
        return response;
}

public static ServiceResponse deleteGRN(String id) {
        ServiceResponse response = new ServiceResponse();
        try {
                DBManagerService.connect();
                stores_po.deleteRecord(id);
                DBManagerService.closeConnection();
                response.setSuccess(true);
                response.setMessageType(ServiceResponse.SUCCESS);
                response.setMessageHeading("PO Deleted");
                response.setMessage("!");
        } catch (Exception e) {
                response.setSuccess(true);
                response.setMessageType(ServiceResponse.FAILED);
                response.setMessageHeading("Error!");
                if (Settings.sqlErrosInUI) {
                        response.setMessage(e.getMessage());
                } else {
                        response.setMessage("Server Error");
                }
        }
        return response;
}

public static ServiceResponse getGRNByName(String refName) {
        GRN GRN;
        ServiceResponse response = new ServiceResponse();
        try {
                DBManagerService.connect();
                GRN = stores_grn.getGRNByName(refName);
                DBManagerService.closeConnection();
                response.setSuccess(true);
```

```
			response.setMessageType(ServiceResponse.SUCCESS);
			response.setMessageHeading("DPO Retrieved");
			response.setMessage(GRN.getName());
			response.setObject(GRN);
		} catch (Exception e) {
			response.setSuccess(true);
			response.setMessageType(ServiceResponse.FAILED);
			response.setMessageHeading("Error!");
			if (Settings.sqlErrosInUI) {
				response.setMessage(e.getMessage());
			} else {
				response.setMessage("Server Error");
			}
		}
		return response;
	}



	// ///////////////////////////////// GRN ITEMS
//
	public static ServiceResponse insertGRNItem(GRNItem item) {
		ServiceResponse response = new ServiceResponse();
		try {
			DBManagerService.connect();
			String bthId = NewId.GetId("BTH");
			item.setId(bthId);
			stores_grn_details.insertRecord(item);
			item_batch.insertRecord(item);
			item_stock.setStock(bthId, item.getLocationId(), item.getUnitId(),
					item.getQuantity(), "" + item.getGrnId(),
					"GRN INSERT", "5");
			DBManagerService.closeConnection();
			response.setSuccess(true);
			response.setMessageType(ServiceResponse.SUCCESS);
			response.setMessageHeading("GRN Item Saved!");
			response.setMessage("");
			response.setObject(item);
		} catch (Exception e) {
			response.setSuccess(true);
			response.setMessageType(ServiceResponse.FAILED);
			response.setMessageHeading("Error!");
			if (Settings.sqlErrosInUI) {
				response.setMessage(e.getMessage());
			} else {
				response.setMessage("Server Error");
			}
		}
		return response;
	}
//
	public static ServiceResponse updateGRNItem(GRNItem item) {
		ServiceResponse response = new ServiceResponse();
		try {
			DBManagerService.connect();
			stores_grn_details.updateRecord(item);
			Format.jsonOutput(item);
			item_batch.updateRecord(item);
			item_stock.setStock(item.getId(), item.getLocationId(),
					item.getUnitId(), item.getQuantity(), "" + item.getGrnId(),
					"GRN Update", "6");
			DBManagerService.closeConnection();
			response.setSuccess(true);
			response.setMessageType(ServiceResponse.SUCCESS);
			response.setMessageHeading("GRN Item Updated!");
			response.setMessage("");
			response.setObject(item);
		} catch (Exception e) {
			response.setSuccess(true);
			response.setMessageType(ServiceResponse.FAILED);
			response.setMessageHeading("Error!");
			if (Settings.sqlErrosInUI) {
				response.setMessage(e.getMessage());
			} else {
				response.setMessage("Server Error");
			}
		}
		return response;
	}

	public static ServiceResponse deleteGRNItem(String GRNId, String itemId) {
		ServiceResponse response = new ServiceResponse();
		try {
			DBManagerService.connect();
			stores_GRN_details.deleteRecord(GRNId, itemId);
			DBManagerService.closeConnection();
			response.setSuccess(true);
			response.setMessageType(ServiceResponse.SUCCESS);
			response.setMessageHeading("GRN Item Deleted");
			response.setMessage("!");
		} catch (Exception e) {
			response.setSuccess(true);
			response.setMessageType(ServiceResponse.FAILED);
			response.setMessageHeading("Error!");
			if (Settings.sqlErrosInUI) {
				response.setMessage(e.getMessage());
			} else {
				response.setMessage("Server Error");
			}
		}
		return response;
	}

	public static ServiceResponse getGRNItemList(String GRNId) {
		ServiceResponse response = new ServiceResponse();
		try {
			DBManagerService.connect();
			GRNItem[] GRNs = stores_GRN_details.getPoItems(GRNId);
			response.setObject(GRNs);
			DBManagerService.closeConnection();
			if (GRNs.length > 0) {
				response.setSuccess(true);
				response.setMessageType(ServiceResponse.INFO);
				response.setMessageHeading("GRN Items Retrieved");
				response.setMessage("!");
			} else {
				response.setSuccess(false);
				response.setMessageType(ServiceResponse.WARNING);
				response.setMessageHeading("No GRN Items Defined");
				response.setMessage("Please add GRN Items");
			}
		} catch (Exception e) {
			response.setSuccess(true);
			response.setMessageType(ServiceResponse.FAILED);
			response.setMessageHeading("Error in Loading GRN Items!");
			if (Settings.sqlErrosInUI) {
				response.setMessage(e.getMessage());
			} else {
				response.setMessage("Server Error");
			}
	}
		return response;
	}
	public static ServiceResponse getGRNItem(String GRNId, String itemId) {
		GRNItem GRNi;
		ServiceResponse response = new ServiceResponse();
		try {
			DBManagerService.connect();
			GRNi = stores_GRN_details.getGRNItemById(GRNId, itemId);
			DBManagerService.closeConnection();
			response.setSuccess(true);
			response.setMessageType(ServiceResponse.SUCCESS);
			response.setMessageHeading("GRN Item Retrieved");
			response.setMessage("");
			response.setObject(dpoi);
		} catch (Exception e) {
			response.setSuccess(true);
			response.setMessageType(ServiceResponse.FAILED);
			response.setMessageHeading("Error!");
			if (Settings.sqlErrosInUI) {
				response.setMessage(e.getMessage());
			} else {
```

```
                        response.setMessage("Server Error");
                    }
                }
                return response;
            }

        }
```

Appendix G - Client Certificate



NIMANSALA RESTAURANT

NIMANSALA

Reg No: WF-5261

#784,HIGHLEVAL Rd: PANAGODA HOMAGAMA.Tel:011-2895483-4

2016-09-28

Project Examination Board,
University of Colombo School of Computing,
External Degree Center,
No. 17, Swarna Road,
Colombo 06.

Dear Sir/ Madam,

Letter of Confirmation

This is to certify that Mr. Tikiri Hannadige Lahiru Chanaka Chandrathilake has successfully developed the Inventory and Restaurant Management System for Nimansala Restaurant.

The Identified requirements have been fulfilled with the developed system and this makes a good impact towards the business growth. We appreciate his effort on making this project success and we wish him all the success in his future endeavors.

Thank You!
Yours Faithfully,

[illegible]

Proprietor
Nimansala Restaurant