ROAD PREMIX MANAGEMENT SYSTEM

TAN CHENG CHAN

BACHELOR OF COMPUTER SCIENCE
(SOFTWARE ENGINEERING) WITH HONOURS

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG
BORANG PENGESAHAN STATUS TESIS
UNIVERSITI MALAYSIA PAHANG
BORANG PENGESAHAN STATUS TESIS

JUDUL: ROAD PREMIX MANAGEMENT SYSTEM
SESI PENGAJIAN: 2014/2015
SAYA TAN CHENG CHAN

Mengaku membenarkan tesis/ laporan PSM ini disimpan di Perpustakaan Universiti Malaysia Pahang dengan syarat-syarat kegunaan seperti berikut:

1. Tesis/Laporan adalah hakmilik Universiti Malaysia Pahang.
2. Perpustakaan Universiti Malaysia Pahang dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institut pengajian tinggi.
4. **Sila tandakan (√)

√	SULIT	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972) *
	TERHAD	(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan) *
	TIDAK TERHAD	

Disahkan Oleh

………………………………   ………………………………
Alamat tetap: Penyelia
333 Tmn Sri Delima, Blkg Balai Polis,   Dr. Mohammed Adam
16250, Wakaf Bharu. Kelantan.   Ibrahim Fakhreldin
Tarikh: Tarikh:

*Sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali sebab dan tempoh tesis/laporan ini perlu dikelaskan sebagai SULIT atau TERHAD.

1

SUPERVISOR DECLARATION

I hereby declare that I have read this thesis and in my opinion this thesis/report is sufficient in terms of scope and quality for the award of the degree of Bachelor of Computer Science (Software Engineering).

Signature: …………………………………………………….

Name of Supervisor: Dr. Mohammed Adam Ibrahim Fakhreldin

Date: 19 December 2014

STUDENT DECLARATION

I hereby declare that the work in this project is my own except for quotations and summaries which have been duly acknowledged. The thesis has not been accepted for any degree and is not concurrently submitted for award of other degree.

Signature: ………………………………………….

Name of Student: Tan Cheng Chan

ID Number: CB11034

Date:19 December 2014

ACKNOWLEDGMENT

ABSTRACT

This project deals with premix for road construction data from Seri Barat Mixed Sdn Bhd and its respective branches, Kuari Dinar Sdn Bhd, Damai Kuari Sdn Bhd, and Dimensi Timal Sdn Bhd in the state of Kelantan. The project is entitled as Road Premix Management System (RPMS). The objective of this project is to develop a Web-based managaement system for the companies in managing premix operations from different branches. There only administrator for main branch and branches that are defined as the main user in the system and there are no customer interactions with the system. The system is only integrated with sales and production and no others department even actually consists of three parts in the system. The system will determine the differences between the sales and production of premix according to the date. All the data entered will be saved in the database management system, called MySQL and Laravel framework is used for implementing the system.

ABSTRAK

Project ini berkaitan dengan premix data untuk pembinaan jalan raya oleh syarikat Seri Barat Mixed Sdn Bhd, dan cawangannya iaitu Kuari Dinar Sdn Bhd, Damai Kuari Sdn Bhd dan Dimensi Timal Sdn Bhd. Objektif projek ini adalah untuk membina sebuah system pengurusan berasaskan rangkain dalam menguruskan operasi premix tersebut. Project ini dinamakan sebagai Road Premix Management System (RPMS). Objektif utama projek ini adalah untuk membangunkan satu system aplikasi berasaskan Web untuk syarikat tersebut dalam menguruskan operasi premis dari cawangan-cawangan. Pengguna yg diutarakan dalam sistem aplikasi ini ialah pentadbir bagi setiap cawangan dan tiada perhubungan antara pelanggan. Sistem ini hanya bersepadu dengan jualan dan pengeluaran premix walaupun terdiri daripada tiga bahagian. Sistem in akan menentukan perbezaan diantara jualan dan pengeluaran premix mengikut tarikh yang dimasukkan. Seluruh data yang dimasukkan akan disimpan di dalam pengkalan data sistem pengurusan, iaitu MySQL dan Laravel Framework digunakan untuk implementasi sistem ini.

TABLE CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Term	**Short description**
CSS	Cascading Style Sheets
HTML	Hypertext Transfer Markup Language
RPMS	Road Premix Management System
RAD	Rapid Application Development
MVC	Model-View-Controller
SAP	*S*ystemanalyse und Programmentwicklung
SBMSD	Seri Barat Mixed Sdn Bhd

Chapter 1

INTRODUCTION

In this chapter is briefly discuss about the overview of this project. It contains six parts. The first part is background; follow by the problem statement. Next is the objectives where the project's goal is determined, followed by the scopes of the project. Lastly, is about the software development methodology of the system.

1.1   Background

Seri Barat Mixed Sdn Bhd is a main office of Dimensi Timal Sdn Bhd, Damai Kuari Sdn Bhd and Kuari Dinar Sdn Bhd which is located at different locations in the state of Kelantan with its major source of revenue continues to be derived from construction sector and road premix production. It is a registered Class "C" Contractor with PUSAKA BUMI, is presently managed by a professional management team with more than 20 years of experience in this sector of business.

However, the company has inefficient of communication between the branches. The main office need to manage the planning, collecting, and reporting the documents from each branch offices every day with different standard through email into a spreadsheet. This documentation, a major road premix production in premix plant, plays a vital role in producing a report. The road premix daily report forms collects information about sale, and production of each premix plant in each branch. During reporting, a clerk systematically collects, verifies, and analyses the documents to derive a date range, daily, monthly, and yearly report accordingly.

In order to replace the paper-based documentation, the main aim of this project is to develop a RPMS for Seri Barat Mixed Sdn Bhd in Web-based management system which enables operation between main branch and branch offices to communicate efficiently and effectively.

1.2   Problem Statement

Recognizing the problems to be solved is a good engineering method. The following problem statements have been discovered with Seri Barat Mixed Sdn Bhd:

i.   Disorganized of communication, sharing of information, and co-ordination among main office and the branch offices thus further relief the productivities of the company in terms of resources such as time.

ii.   Cost increased in collecting documents and lacking approaches through email to main office and produced a monthly report using spreadsheet.

iii. The company collected the data for sales and production every month and produced a differences of sales and productions quantities in metric tonnes of each branch using Microsoft Excel spreadsheet and had difficulties in analyzing, collecting, and managing data from each branch.

1.3   Objectives

The current problems in Seri Barat Mixed Sdn Bhd will be solved which involves three main objectives. The following objectives will be developed according to the development plan:

i.   To develop a Web-based management system for data entry and management of the data and information into a database by using Laravel Framwork and MySql.

ii.   To develop a Web-based management system for collecting on daily reports that dynamically extracting information from different districts into a customize form of tabular and graphical chart by using Google Chart.

iii. To calculate the differences of sales and productions of road premix in metric tonnes.

1.4   Scopes

The stated scopes have been identified for the development of Web-based management system. This will be categorized into two parts, in the scope and out of the scope. For in the scope, there are two main users that will compliment in the system which are administratively from the main branch and branch. Each of them would be considered within the scope and has their roles and responsibilities in sales management, and production management until the generation of daily report, monthly report, and yearly report of sales and productions.

i.   Administrator: Manager

Manager in main branch has accessibility to enter sale and production data. Main branch manager can generate report according to the date range, daily report, monthly report and yearly report. Only main branch can generate the report of each branch. The report composed of sales and productions of road premix.

ii.   Administrator: Branch Manager

Branch manager has access to enter sale and production data. Branch manager also can generate report according to the date range, daily report, monthly report and yearly report. The report composed of sales and productions of road premix.

1.5 Report Organizations

This report consists of four (4) chapters:

Chapter 1:   **Introduction** will shortly introduce the system. This chapter will comprise the problem statements, objectives, scopes and the summary.

Chapter 2:   **Literature Review** will explain in details the overview of the project flow. In this chapter, the project concepts, technology applied for the system, the manual system, and the existing systems which are related to the case study will be reviewed. This chapter will also review on the methodology of the project research concisely.

Chapter 3:   **Methodology**, further and deeper reviews of the overall approach and framework will be deliberated. This chapter also covers the details for the method, technique, hardware and software during the research process.

Chapter 4:   **Conclusion and future Works** will summarize the project findings as a whole, and discussed for any future enhancement for the research topic or technique. References and appendices will be added to the last part of this project.

1.6 Summary

As a conclusion, this chapter is shortly introducing the concept of Road Premix Management System in Web-based management of Seri Barat Mixed Sdn Bhd. All details about background, problem statement, objectives and scope are included with clearly explanation.

Chapter 2

LITERATURE REVIEW

In this chapter, the current and existing system, operating system, programming language, Integrated Development Environment (IDE), Relational Database Management System (RDBMS) and web server which are related to this project will be reviewed. This chapter will also review on the software development methodology of the project concisely.

2.1 Current and Existing System

2.1.1 SAP Enterprise Resource Planning

The main objective is to increase operating efficiency by improving business processes and decreasing costs (Nah, Lau, & Kuang 2001; Beheshti 2006). ERP allows different departments with diverse needs to communicate with each other by sharing the same information in a single system. ERP thus increases cooperation and interaction between all business units in an organization on this basis (Harrison, 2004).

As different departments across an institution share an integrated database, end users can access data in real time. Best-of-breed information technology such as web technologies, mobile phones, and on-line services offer additional benefits not only to the administration within an institution, but also to people who constantly interact with the institution – faculty, students, and staff (Murphy, 2004; Zornada & Velkavrh, 2005).

Business benefits

- Improve internal communications
- Reduce or eliminate manual processes

- Establish a self-service environment for employees
- Improve self-service environment for students and faculty
- Enable higher availability of administrative systems

Technical benefits

- Reduce or eliminate the need for backup or shadow systems
- Platform for re-engineering business practices and continued process improvements
- Develop and maintain consistent data definitions
- Provide accessible, user-friendly administrative and student support services
- Access to data in real time

2.1.2 Oracle E-Business Suite

Oracle's market-leading Enterprise Resource Planning (ERP) solutions which include Financials, Project Portfolio Management, Procurement, and Governance, Risk, and Compliance are proven, trusted foundations for core business operations.

Business benefits

- Get up and running faster with less upfront costs and investment risk
- Ensure consistent processes across all your location around the world
- Make more informed and data-driven business decisions
- Boosts user productivity and increase user adoption
- Eliminate the needs for expensive customizations

Technical benefits

- Oracle EBS has advanced in database technology
- Oracle EBS integrates company's business with over 100 integrated product modules for every aspect of the business including finance, human resources, customer relationship management and project management.
- Develop and maintain consistent data definitions

- Provide accessible, user-friendly administrative and student support services
- Access to data in real time

2.3 Software Development Methodology

2.3.1 Waterfall Model

The Waterfall Model was first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed fully before the next phase can begin. At the end of each phase, a review takes place to determine if the project is on the right path and whether or not to continue or discard the project. In waterfall model phases do not overlap.
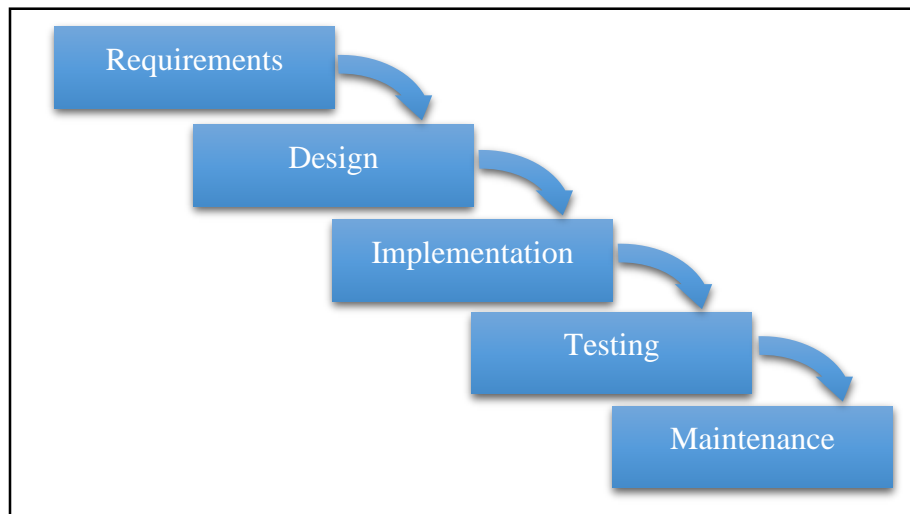
Waterfall Model Design

Figure 2. 1 Waterfall model

Figure 2.1 shows the waterfall model that consists of five main phases, Requirements, Design, Impementation, Testing and Maintenance.

Waterfall Model Five Phases

i. **Requirements**

Requirements of a user are gathered, analyzed and documented for preparation in the development process.

ii.   Design

The requirements gathered are evaluated and a proper implementation strategy is formulated according to the software environment. The design phase is categorized into two sections, i.e. system design and component design. The system design contains details and specifications of the whole system and explains how each component of the system will interact with others. The component design contains specifications as to how each component will work separately and how results from one component will travel to another. Individual coders are usually assigned to develop each component.

iii. Implementation

This phase is creating the components. The information gathered is applied in this step to create the actual working parts of the system. The design generated in the above phase is converted into machine language that the computers can actually understand and process.

iv.   Testing

The testing phase is where the software is checked for any errors or discrepancies. The testing of the software starts after the code is finished which is usually in the ending stages of implementation phase. Various different tools, software and strategies are used for testing the solution for an error free software.

v.   Maintenance

Maintenance is an ongoing process which may stretch from a few months to many years. It is a fact that all software has bugs no matter how cautiously it has been developed and tested. Furthermore, with the passage of time, requirements will also change and modifications or additions will be required to keep it effective. All this work comes under the umbrella term which is maintenance.

Pros

- Simple and easy to understand and use
- Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
- Phases are processed and completed one at a time.
- Works well for smaller projects where requirements are very well understood.
- Well understood milestones.

Cons

- No working software is produced until late during the life cycle.
- High amounts of risk and uncertainty.
- Poor model for long and ongoing projects.
- It is difficult to measure progress within stages.
- Cannot accommodate changing requirements.

2.3.2   Rapid Application Development (RAD) Model

RAD is a type of incremental model. In RAD model the components or functions are developed in parallel as if they were mini projects. The developments are time boxed, delivered and then assembled into a working prototype. This can quickly give the customer something to see and use and to provide feedback regarding the delivery and their requirements.
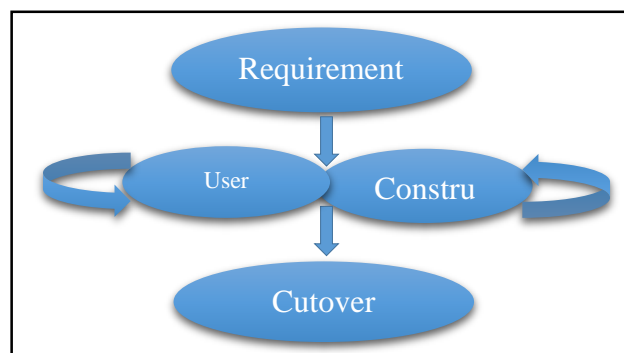
RAD Model Design

Figure 2. 2 RAD model

Figure 2.2 shows the RAD design that consists of four main phases, Requirements planning, User design, Construction, and Cutover phases.

RAD Model Four Phases

i. **Requirements planning phase**

In requirements planning phase, it combines elements of the system planning and systems analysis phases of the Systems Development Life Cycle (SDLC). Users, managers, and IT staff members discuss and agree on business needs, project scope, constraints, and system requirements. It ends when the team agrees on the key issues and obtains management authorization to continue.

ii. **User design phase**

During this phase, users interact with systems analysts and develop models and prototypes that represent all system processes, inputs, and outputs. The RAD groups or subgroups typically use a combination of Joint Application Development (JAD) techniques and CASE tools to translate user needs into working models. *User Design* is a continuous interactive process that allows users to understand, modify, and eventually approve a working model of the system that meets their needs.

iii. **Construction phase**

This phase focuses on program and application development task similar to the SDLC. In RAD, however, users continue to participate and can still suggest changes or improvements as actual screens or reports are developed. Its tasks are programming and application development, coding, unit-integration and system testing.

iv. **Cutover phase**

Cutover phase resembles the final tasks in the SDLC implementation phase, including data conversion, testing, changeover to the new system, and user training. Compared with traditional methods, the entire process is compressed. As a result, the new system is built, delivered, and placed in operation much sooner.

Pros

- Changing requirements can be accommodated.

- Progress can be measured.

- Iteration time can be short with use of powerful RAD tools.

- Reduced development time.

- Increases reusability of components

Cons

- Dependency on technically strong team members for identifying business requirements.

- Only system that can be modularized can be built using RAD.

- High dependency on modeling skills.

- Management complexity is more.

- Requires user involvement throughout the life cycle.

2.4   Design Pattern

2.4.1   Model-View-Controller (MVC)

Software design patterns provide general solution on designing and developing of a system. The design patterns proposed regardless of the used of programming languages. Nevertheless, programming languages will affluent the design patterns as the conceptual background of design patterns that emphasized concept of object-oriented programming (Akbiyikh & Tercan, 2010).

The fundamental of MVC is well known software design pattern that enable easier segregation of three layers of the MVC, the model, the view, and the controller. The design pattern is good for architecting interactive web applications, however designing MVC is complicated by the fact that current technologies encourage web-developers to apply as early as in the user design phase (Leff, 2001).
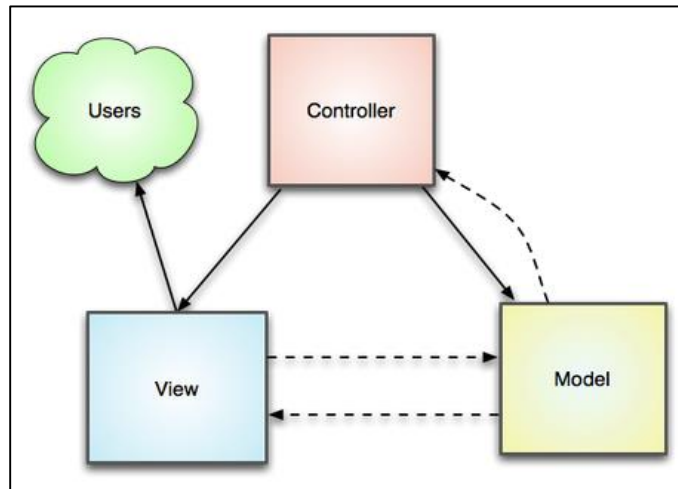
Figure 2. 3 MVC Archictecture

Figure 2.3 shows the relationship MVC pattern based on developing an application in three parts: The view, the model, and the controller.

i. Model

The model holds all the data, state and application logic. The model is responsible for the view and controller. This includes the holding of the data, state and application logic (Freeman et al., 2004). Therefore, it will represent the business logic of the applications for implementation.

ii. View

The view renders the presentation from the model directly. It is responsible for using the information it has available to produce any presentational interface your application might need. For example, as the Model layer returns a set of data, the view would use it to render a HTML page containing it, or a XML formatted result for others to consume. The View layer is not only limited to HTML or text representation of the data. It can be used to deliver a wide variety of formats depending on your needs, such as videos, music, documents and any other format you can think of.

iii. Controller

The Controller layer handles requests from users. It is responsible for rendering a response with the aid of both the Model and the View layer. A

controller can be seen as a manager that ensures that all resources needed for completing a task are delegated to the correct workers. It waits for petitions from clients, checks their validity according to authentication or authorization rules, delegates' data fetching or processing to the model, selects the type of presentational data that the clients are accepting, and finally delegates the rendering process to the View layer.

2.4.2   Three-tier Architecture

Three-tier architecture is a client-server architecture consists of three layers; Presentation, Business and Data.

i.   Presentation layer

The presentation layer records the request from the client, applies the necessary business logic and its output is transferred to the business layer.

ii.   **Business layer**

The business layer performs processing data, exchanging data, and controlling operations. This layer responds to request from the presentation layer and sends these to data layer and vice versa.

iii.   Data layer

The data layer provides relationship between the system and the database. This layer processes the data sent by the business layer and afterwards, it prepares and sends to the database.
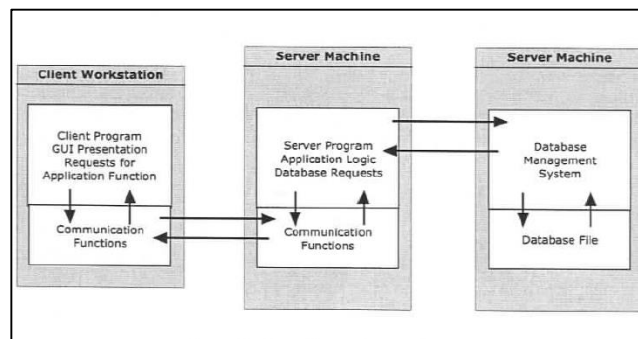

Figure 2. 4 Three-tier System Architecture

Figure 2.4 shows the relationship of business layer onwards data layer in Three-tier system architecture.

2.5   Programming Language

2.5.1   Laravel Framework

Laravel is a free, open source PHP web application framework, designed for the development of model–view–controller (MVC) web applications. Laravel is released under the MIT license, with its source code hosted on GitHub. According to a December 2013 developers survey on PHP frameworks popularity, Laravel is listed as the most popular PHP framework in 2013, followed by Phalcon, Symfony2, CodeIgniter and others. At the same time, as of August 2014 Laravel is the most popular and watched PHP project on GitHub.

Features

- Eloquent ORM (object-relational mapping) is an advanced PHP implementation of the active record pattern, providing internal methods for enforcing constraints to the relationships between database objects. Laravel's query builder, Fluent, is natively supported by Eloquent.

- Application logic is part of developed applications, either by using controllers, or as part of route declarations. Syntax used for definitions is similar to the one used by Sinatra framework.

- Reverse routing defines a relationship between links and routes, making it possible for later changes to routes to be automatically propagated into relevant links. When links are created by using names of existing routes, appropriate uniform resource identifiers (URIs) are automatically created by Laravel.
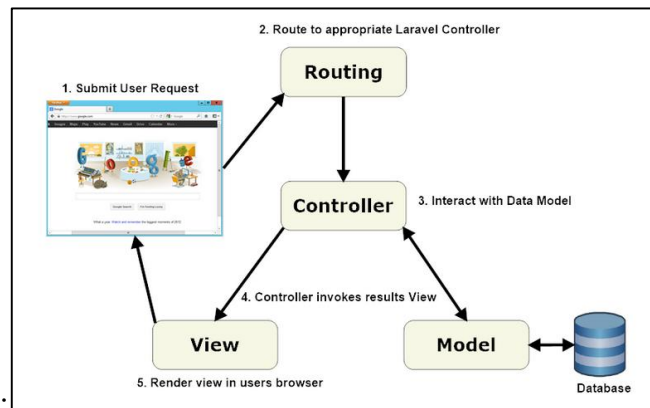
Figure 2. 5 Laravel Architecture

Figure 2.5 is a typical Laravel application consists of the above mentioned MVC components, as you can see below:

When interacting with a Laravel application, a browser sends a request, which is received by a web server and passed on to the Laravel routing engine. The Laravel router receives the request and redirects to the appropriate controller class method based on the routing URL pattern.

The controller class then takes over. In some cases, the controller will immediately render a view, which is a template that gets converted to HTML and sent back to the browser. More commonly for dynamic sites, the controller interacts with a model, which is a PHP object that represents an element of the application (such as a user, blog post) and is in charge of communicating with the database. After invoking the model, the controller then renders the final view (HTML, CSS, and images) and returns the complete web page to the user's browser.

Laravel promotes the concept that models, views, and controllers should be kept quite separate by storing the code for each of these elements as separate files in separate directories. This is where the Laravel directory structure comes into play.

Design patterns such as MVC are created to make a developer's life easier. This is where Laravel scores over plain PHP which doesn't follow any sort of paradigm. If this discussion seems a bit abstract right now, worry not! After you start working with Laravel, you won't even realize that you are working in a design pattern. It all becomes natural to you after a while.

26

2.6   Database Management System

2.6.1   PDO

The database query builder provides a convenient, fluent interface to creating and running database queries. It can be used to perform most database operations in your application, and works on all supported database systems. The Laravel query builder uses PDO parameter binding throughout to protect your application against SQL injection attacks. There is no need to clean strings being passed as bindings.

Laravel makes connecting with databases and running queries extremely simple. The database configuration file is app/config/database.php. In this file you may define all of your database connections, as well as specify which connection should be used by default. Examples for all of the supported database systems are provided in this file.
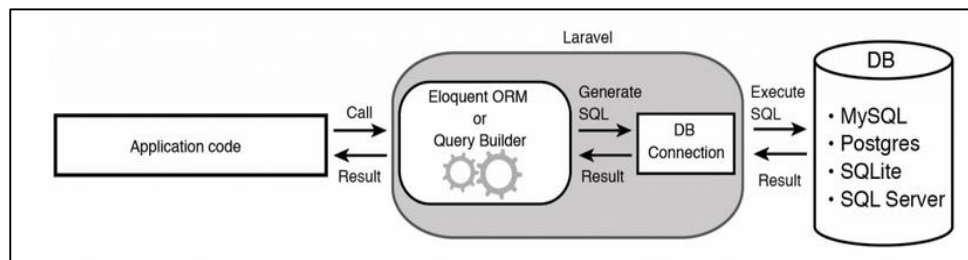

Figure 2. 6 Database operation structure

Figure 2.6 show the Laravel supports four database systems: MySQL, Postgres, SQLite, and SQL Server.

2.7   Web Server

2.7.1   Xampp

XAMPP is a free and open source cross-platform web server solution stack package, consisting mainly of the Apache HTTP Server, MySQL database, and interpreters for scripts written in the PHP and Perl programming languages.

2.8   Summary

This chapter is about the review for the RPMS of Seri Barat Mixed Sdn Bhd. There are some brief explanation of enterprise resource planning system, researches that had been developed by others and present projects which are related to the current project. Also it is included the explanation of development tools, and software development methodology.

CHAPTER 3

METHODOLOGY

This chapter is discusses about the implementing of RAD in developing Road Premix Management System for Seri Barat Mixed Sdn Bhd. Managing the system process manually is a tedious task, thus a Web-based managemnt system to manage the system is proposed to be developed using Rapid Application Development model. There are three sections involve in this chapter. The first section is introduced about introduction of RAD methodology and the reasons of selecting the methodology. The next section is discusses about implementation of the methodology. The final section elaborates on the details of software and hardware to be applied in the system.

3.1 Requirements Planning

There are three tasks in this phase. The following are the tasks in Requirements Planning:

i. Research Current Situations

There are three current systems under investigations of advantages and disadvantages. The systems are SAP ERP, and Oracle ERP which involves complex processes, however provides a more efficient of working environment. ERP is a short term for Enterprise Resource Planning. It is business management software which is suite of integrated applications that store and manage information from sales, and production, The proposed system, RPMS, will be develop as an integrated applications of sales, and production for the operations of finished good road premix.

ii. Define Requirements

The targeted users are Sales supervisor, and Production supervisor that responsible for maintaining (insert, update and delete) the information of sales, and for road premix production respectively in the Seri Barat Mixed Sdn Bhd, and its respective branches Kuari Dinar Sdn Bhd at Gua Musang, Damai Kuari Sdn Bhd at

Gua Musang, and Dimensi Timal Sdn Bhd at Tanah Merah. Currently, there are only 4 premix plants in these companies, and there will be new premix plant in the futures. Only manger is allow to gain access for viewing a full report of all branches and its branch. The reports composed sales report, and production report from daily to yearly or according to the date range specified.

iii. **Finalize Requirements**

a. **Administrator: Manager and Branch Manager**

Correctness

- The system will produce two types of automated reports; Sales report, and Production reports from all branches. The report is in the format of tabular form and graphical chart and is produced according to the selection; date range, daily, monthly, and yearly.
- The required accuracy of the reports containing 100%.
- The completeness of the sales, and production is 100%.
- The documentation of the reports standards and guidelines comply by the client is provided.

Integrity

- Only Administrator has the privileges access for the reporting from all branches through the internet. The software requirements include the possibility of viewing, and copying but no changes could be made.

Usability

- At least 4 records of data entering per day.
- Training a new supervisor will take no more than two days (16 training hours), and at least five records of data entering per day.

3.1.1 Software requirement

Software Interface	Software Descriptions
Window 7 Home Premium	A personal computer operating system
Sublime Text 2.0	A text editor for code, markup and prose
Adobe Photoshop CS6	A graphic editing program
phpMyAdmin and Apache	A Web server application
Rational Software Architect	An modeling and development program
Microsoft Office Word 2013	A word processor
Microsoft PowerPoint 2013	A slide show presentation program

3.1.2 Hardware requirement

- **ASUS U31SD**
  Support:
  Windows 7 Ultimate
  Windows 7 Professional
  Windows 7 Home Premium
  Windows 7 Home Basic
  Chipset:
  Intel® HM65 Express Chipset
  Memory:
  DDR3 1333 MHz SDRAM, 2 x SO-DIMM socket for expansion up to 8 GB SDRAM
  Storage:
  640GB 5400RPM
- **Printer CANON IP2770**
  Type:
  Printer only
- **Thumb Drive Kingston**
  Storage size:
  16GB

3.1.3 Use Case Diagram

The RPMS is required to provide the most value to the user of Seri Barat Mixed Sdn Bhd and its respective branches followed by the Sales Department, and Production Department those that process, administrate and maintain the sales, and production.
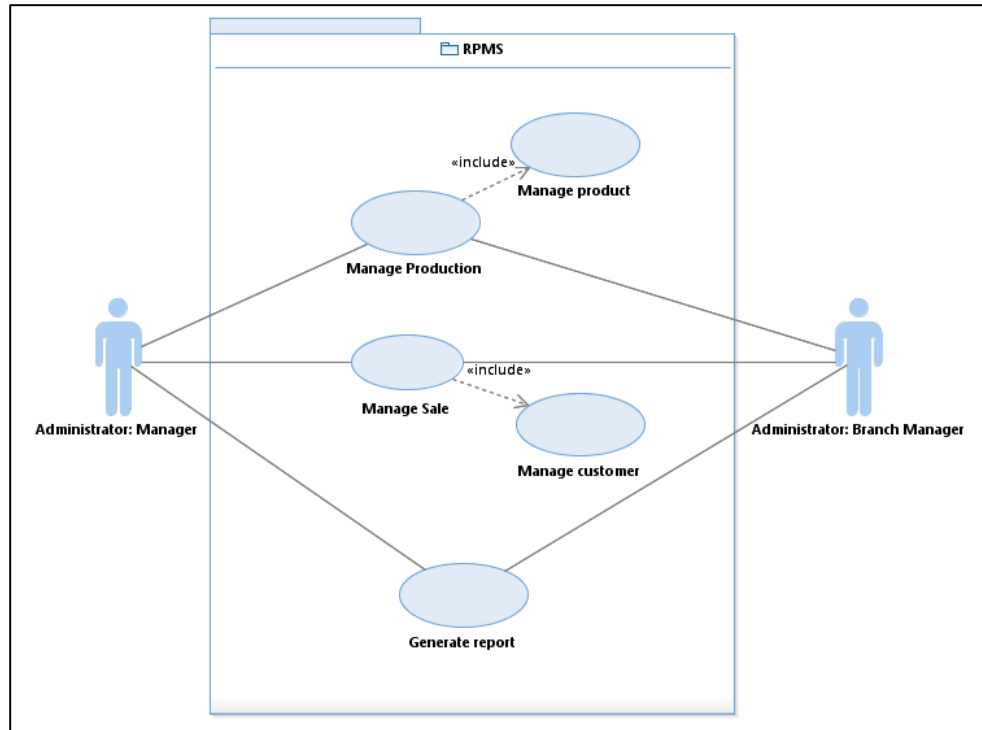

Figure 3. 1 Use case diagram for RPMS

Figure 3.1 shows the interaction on manage sale and manage production, manage product, manage customer and generate report by administrator in RPMS.

3.4.4 Use Case Description

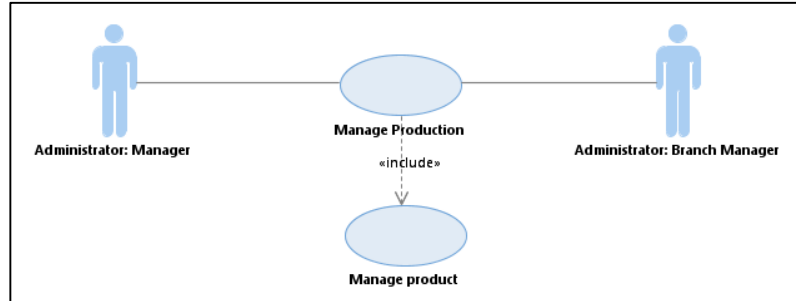**Use Case Diagram for Managing Production**

Figure 3.2 Use case diagram for managing production

Figure 3.2 shows the use case diagram that describes the interaction of Administrator and manage production and manage product.

Table 3.1 shows the use case description that describe the manage production in the RPMS.

Table 3. 1 Use case description for manage production

Use Case ID	SRS-RPMS001
Use Case Name	Manage Production
Brief Description	This use case describe about production management functionality available in the RPMS.
Actor	Administrator: Manager, Administrator: Branch Manager
Pre-Conditions	Administrator is logged in the system

Basic Flow	1. Access Manage Production
	    The administrator accesses the Manage Production main menu
	2. Access Create New Production
	    The administrator accesses the Create New Production and entered the required data. (2A, 2B, 2C)

	3. Return to Production of Today
	The page displays the data of today production entered.
Alternative Flow	2A Access All Productions of Today
	The administrator view on the today productions
	2B Access Change Productions
	The administrator make a change on the today productions by delete or edit the data.
	2B Access Deleted Productions
	The administrator view deleted sales on the today productions by delete or edit.
Exception Flow	None
Post-Conditions	A new production successfully created in the database and return to the Production of Today page.
Rules	None
Constraints	None
Sequence Diagram	Refer Appendix
	A-3.1 : Sequence Diagram for Manage Production

Table 3.2 shows the use case description that describe the manage production in the RPMS.

Table 3.2 Use case description for manage product

Use Case ID	SRS-RPMS002
Use Case Name	Manage Product
Brief Description	This use case describe about product management functionality available in the RPMS.
Actor	Administrator: Manager, Administrator: Branch Manager

| Pre-Conditions	Administrator is logged in the system

Basic Flow	1. Access Manage Product
	
	    The administrator accesses the Manage Product main menu
	
	2. Access Create New Product
	
	    The administrator accesses the Create New Product and entered the required data. (2A)
	
	3. Return to All Product
	
	    The page displays all of the products. The changes of product, delete or edit can be made at this page.

Alternative Flow	2A Access All Products The administrator view on the products
Exception Flow	None
Post-Conditions	A new product successfully created in the database and return to the All Product page
Rules	None
Constraints	None
Sequence Diagram	Refer Appendix A-3.2 : Sequence Diagram for Manage Product

Use Case Diagram for Managing Sale
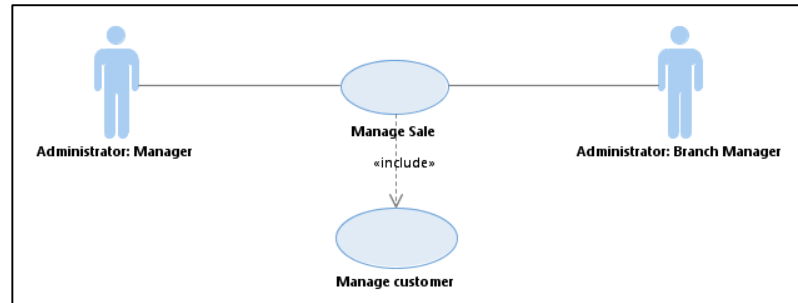
Figure 3.3 Use case diagram for manage sale

Figure 3.3 shows the use case diagram that describes the interaction of Administrator and manage sale and manage customer.

Table 3.3 shows the use case description that describe the manage sale in the RPMS.

Table 3.3 Use case description for manage sale

Use Case ID	SRS-RPMS003
Use Case Name	Manage sale
Brief Description	This use case describe about sale management functionality available in the RPMS.
Actor	Administrator: Manager, Administrator: Branch Manager
Pre-Conditions	Administrator is logged in the system

Basic Flow	1. Access Manage Sale
	    The administrator accesses the Manage Sale main menu
	2. Access Create New Sale
	    The administrator accesses the Create New Sale and entered the required data. (2A, 2B, 2C)
	3. Return to Sale of Today
	    The page displays the data of today production entered.

Alternative Flow	2A Access All Sales of Today
	The administrator view on the today sale
	2B Access Change Sales
	The administrator make a change on the today sales by delete or edit the data.
	2B Access Deleted Sales
	The administrator view deleted sales on the today sales by delete or edit.
Exception Flow	None
Post-Conditions	A new sale successfully created in the database and return to the Sale of Today page.
Rules	None
Constraints	None
Sequence Diagram	Refer Appendix A-3.3 : Sequence Diagram for Manage Sale

Table 3.4 shows the use case description that describe the manage sale in the RPMS.

Table 3.4 Use case description for manage sale

Use Case ID	SRS-RPMS004
Use Case Name	Manage Customer
Brief Description	This use case describe about customer management functionality available in the RPMS.
Actor	Administrator: Manager, Administrator: Branch Manager
Pre-Conditions	Administrator is logged in the system
Basic Flow	1. Access Manage Customer

	The administrator accesses the Manage Customer main menu
	2. Access Create New Customer
The administrator accesses the Create New Customer and entered the required data.
3. Return to All Customers
The page displays all of the customers. The changes of product, delete or edit can be made at this page.
Alternative Flow	None
Exception Flow	None
Post-Conditions	A new customer successfully created in the database and return to the All Customer page
Rules	None
Constraints	None
Sequence Diagram	Refer Appendix
A-3.4 : Sequence Diagram for Manage Customer

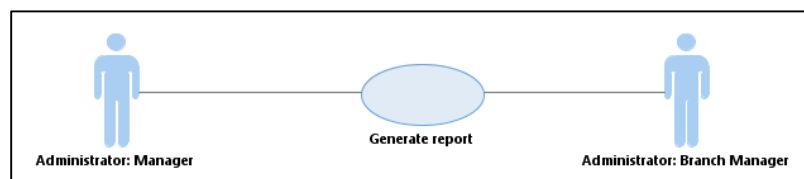**Use Case Diagram for Generate Report**

Figure 3.4 Use case diagram for generate report

Figure 3.4 shows the use case diagram that describes the interaction of Administrator and generate report.

Table 3.5 shows the use case description that describe the manage sale in the RPMS.

Table 3. 5 Use case description for generate report

Use Case Name	Generate report
Use Case ID	SRS-RPMS005
Brief Description	This use case describes about generate report functionality available in the RPMS. It involves the retrieval of data from database and the display of report.
Actor	Administrator: Manager, Administrator: Branch Manager
Pre-Conditions	Administrator is logged into the system.

Basic Flow	1. Access Reports
	   The administrator accesses to Date Range Report. (1A, 1B, 1C)
	2. Return Date Range Report
	   The administrator entered date.

Alternative Flow	1A Access Daily Report
	   The administrator accesses to Date Range Report.
	1B Access Monthly Report
	   The administrator accesses to Date Range Report.
	1C Access Yearly Report
	   The administrator accesses to Date Range Report.

Exception Flow	None
Post-Conditions	Report is displayed with total of sales and total of productions
Rules	None
Constraints	None
Sequence Diagram	Refer Appendix A-3.5 : Sequence Diagram for Generate Report

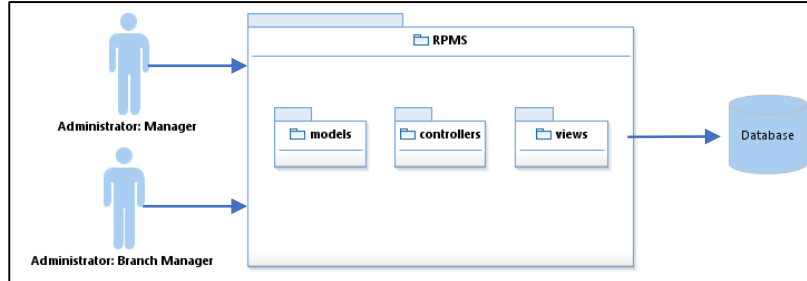## 3.2 User Design

3.2.1 System Overview

Figure 3.5 System design overview

Figure 3.5 shows the system overview of RPMS with interaction of administrator, system and database. The actors consists of manager and branch manager. RPMS has three subsystems, models, controllers, and views. The RPMS connects to the database for data management in the system.
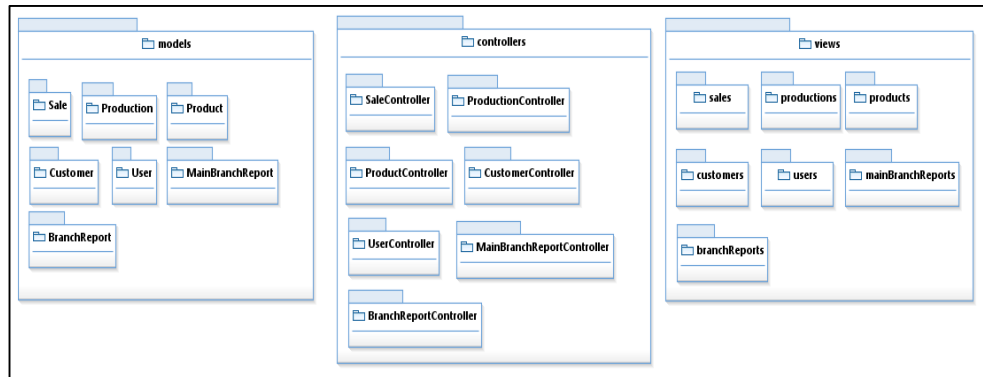
3.2.2 System Architecture

Figure 3.6 Static organization

Figure 3.6 shows the system architecture describes the internal organizational structure of the system. There are sale, production, product, customer, branch report and main branch report.
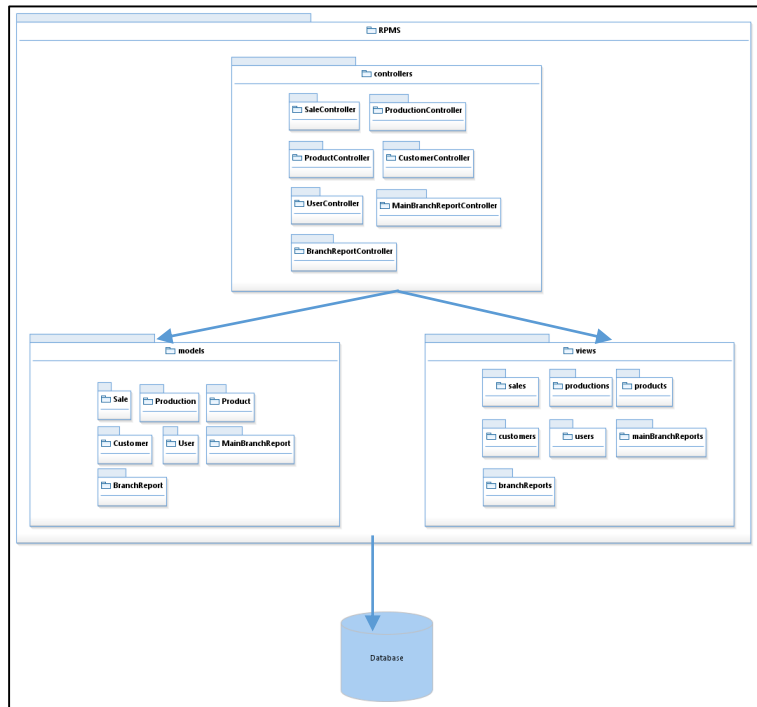
Figure 3.7 Subsystem interfaces

Figure 3.7 shows the subsystem interfaces of model, view and controller in RPMS.

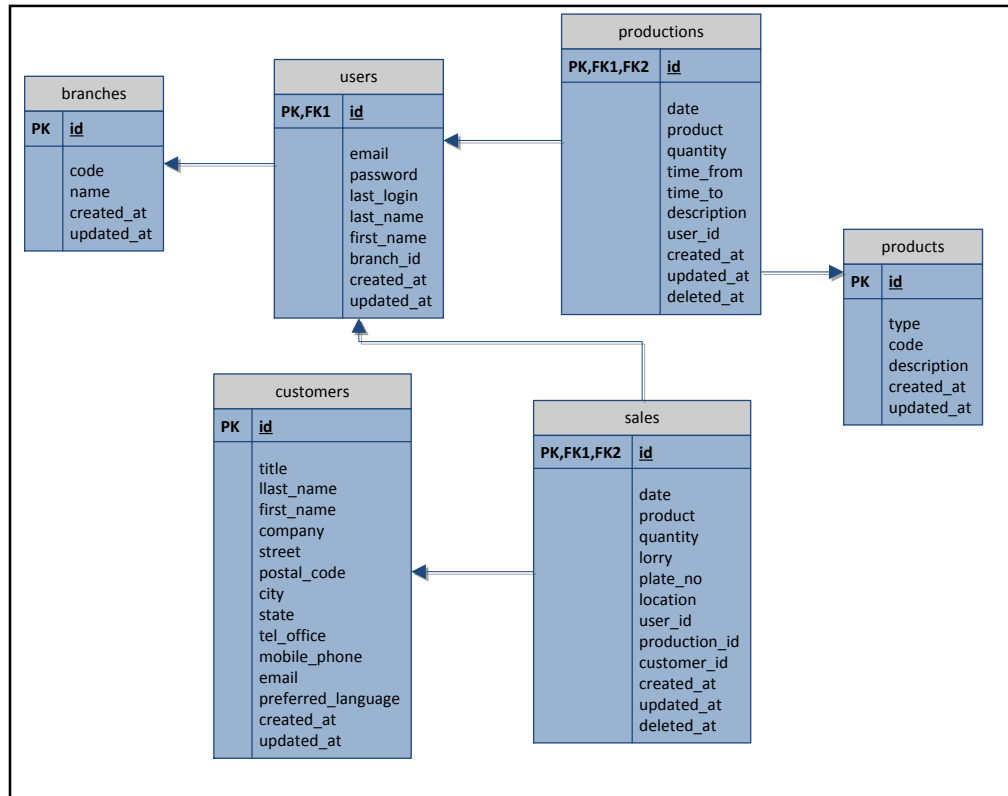**Entity Relationship Diagram**

Figure 3.8 Entity Relationship Diagram

Figure 3.8 shows the entity relationship diagram of RPMS. There are branches, users, productions, sales, customers, and products.

Entity of ERD

Table 3.6 shows the structure of RPMS. There are customer, product, production, sale, and user tables.

Table 3.6 Structure of table

No.	Entity	Description
1	branches	Contains details of a branches
2	customers	Contains details of a customer
3	productions	Contains details of production
4	products	Contains details of each product
5	sales	Contains details of sales from customer

No.	Name	Definition
5	user	Contains details of user of the system

Branches Table

Table 3.7 shows the branches table with name and its definition.

Table 3.7 Branches table

No.	Name	Data Type	Definition
1	id (PK)	int(10)	Defines branches ID
2	code	varchar(5)	Defines branches code
3	name	varchar(255)	Defines branches name
4	created_at	timestamp	Defines the created at
5	updated_at	timestamp	Defines the updated at

Customers Table

Table 3.8 shows the customers table with name and its definition.

Table 3.8 Customer table

No.	Name	Data Type	Definition
1	id (PK)	int(10)	Defines customer ID
2	title	varchar(5)	Defines customer title
3	last_name	varchar(255)	Defines customer last name
4	first_name	varchar(255)	Defines customer first name
5	company	varchar(255)	Defines customer company
6	street	varchar(255)	Defines customer street
7	postal_code	varchar(255)	Defines customer postal code
8	city	varchar(255)	Defines customer city
9	state	varchar(255)	Defines customer state
10	tel_office	varchar(255)	Defines customer tel_office
11	mobile_phone	varchar(255)	Defines customer mobile_phone
12	email	varchar(255)	Defines customer email address
13	preferred_language	varchar(255)	Defines customer preffered language

No.	Name	Data Type	Definition
14	created_at	timestamp	Defines the created at
15	updated_at	timestamp	Defines the updated at

Productions Table

Table 3.9 shows the productions table with name and its definition.

Table 3.9 Production table

No.	Name	Data Type	Definition
1	id (PK)	int(10)	Defines production id
2	date	varchar(255)	Defines production date
3	product	varchar(255)	Defines product
4	quantity	varchar(255)	Defines production quantity
5	time_from	varchar(255)	Defines production time from
6	time_to	varchar(255)	Defines production time to
7	description	text	Defines production description
8	user_id	int(11)	Defines user id from user table
9	created_at	timestamp	Defines the created at
10	updated_at	timestamp	Defines the updated at
11	deleted_at	timestamp	Defines the deleted at

Product Table

Table 3.10 shows the products table with name and its definition.

Table 3.10 Product table

No.	Name	Data Type	Definition
1	id (PK)	int(11)	Defines product id
2	type	varchar(255)	Defines product type
3	code	varchar(255)	Defines product code
4	description	varchar(255)	Defines product description
5	created_at	timestamp	Defines the created at
6	updated_at	timestamp	Defines the updated at

Sales Table

Table 3.11 shows the sales table with name and its definition.

Table 3.11 Sales table

No.	Name	Data Type	Definition
1	id (PK)	int(10)	Defines sales id
2	date	varchar(255)	Defines production date
3	product	varchar(255)	Defines product
4	quantity	varchar(255)	Defines sales quantity
5	lorry	varchar(255)	Defines sales lorry
6	plate_no	varchar(255)	Defines lorry plate number
7	location	varchar(255)	Defines sales location
8	user_id	int(11)	Defines user id
9	production_id	int(11)	Defines production id
10	customer_id	int(11)	Defines customer id
11	created_at	timestamp	Defines the created at
12	updated_at	timestamp	Defines the updated at
13	deleted_at	timestamp	Defines the deleted at

Users Table

Table 3.12 shows the users table with name and its definition.

Table 3.12 Users table

No.	Name	Data Type	Definition
1	id (PK)	int(10)	Defines user ID
2	email	varchar(255)	Defines user email
3	password	varchar(255)	Defines user password
4	permissions	text	Defines user permissions
5	activated	tinyint	Defines user activated
6	activation_code	varchar(255)	Defines user activation code
7	activated_at	timestamp	Defines user activated at

8	last_login	timestamp	Defines user last login
9	persists_code	varchar(255)	Defines user persists code
10	reset_password_code	varchar(255)	Defines user reset password code
11	branch_name	varchar(255)	Defines branch name
12	created_at	timestamp	Defines the created at
13	updated_at	timestamp	Defines the updated at

3.3 Construction

3.3.1 Database structure

The structure of database, rpms is design for user. The structure of this database is shown below:

Figure 3.9 Database Structure

Figure 3.8 shows the rpms database of customers, products, productions, sales, and users tables.

Figure 3.10 Customers table

Figure 3.9 shows customers table in rpms database.

Figure 3. 11 Productions table

Figure 3.10 shows productions table in rpms database.

Figure 3.12 Products table

Figure 3.11 shows products table in rpms database.

Figure 3.13 Sales table

Figure 3.12 shows sales table in rpms database.

Figure 3.14 Users table

Figure 3.13 shows users table in rpms database.

3.6.2   System Interface

3.6.2.1 Preliminary Design

The user interface of RPMS is design accordingly to the user requirement. The layout of the interface is shown below:
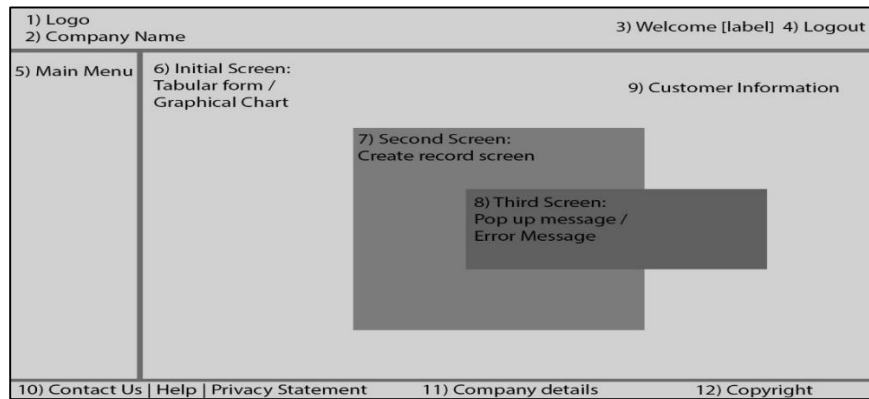
Figure 3.15 Main Interface

Figure 3.14 shows the main interface of RPMS for admin that consists of content of sales, production and etc.

3.6.2.2 Detailed Design
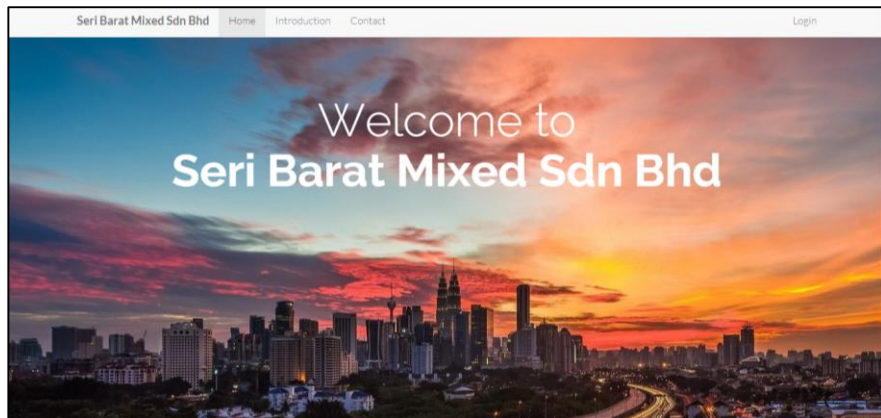
Figure 316 Home page

Figure 3.15 shows the home page of RPMS. The layout of the interface is shown above.
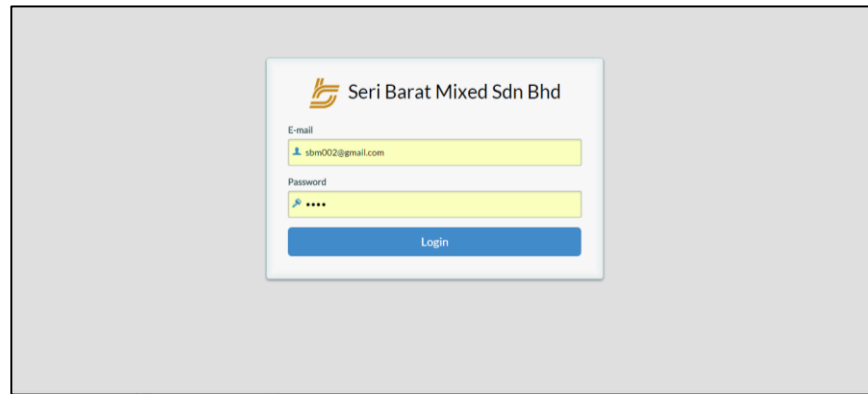
Figure 3.17 Login page

Figure 3.16 shows the login interface of RPMS is designed accordingly to the user requirement.
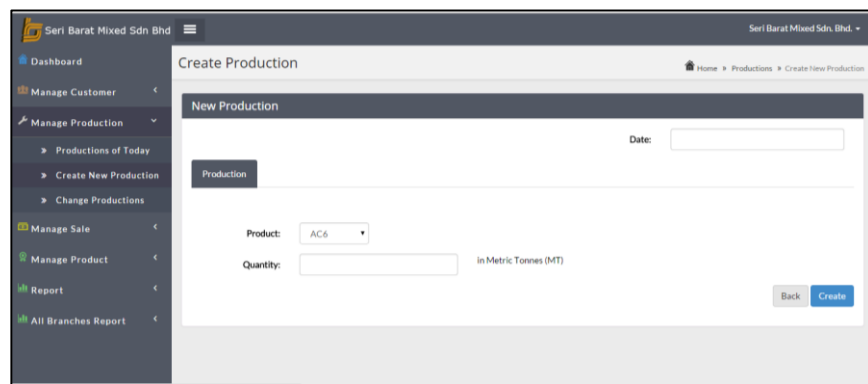
Figure 3.18 Production page

Figure 3.17 shows the production interface of RPMS that is designed accordingly to the user
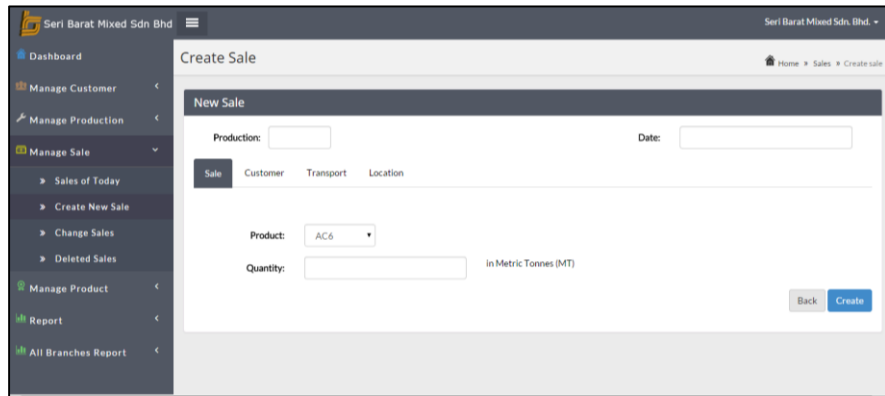
Figure 3.19 Sale page

Figure 3.18 shows the sale interface of RPMS that is designed accordingly to the user requirement.

3.6.3 Coding Implementation

3.6.3.1 Production Coding

This is the coding for creating production in RPMS based on the design. The diagram below shows part of the coding in production interface only. The production coding is shown below:

Figure 3.20 Coding of productions

Figure 3.19 shows the coding for creating production in RPMS based on the design. The diagram below shows part of the coding in production interface only. The production coding is shown below:

Figure 3.21 Coding of sales

Figure 3.20 shows the coding for creating sale in RPMS based on the design. The diagram below shows part of the coding in production interface only. The sale coding is shown below:

3.4   Cutover

i.   Construct User Training

Training sessions are conducted to instruct future users of the new system on how it operates. This training is completed before the application is placed into production. The training sessions are based on materials developed during the RC stage and follow the training plan that was developed at that time. The training may include a train-the-trainers program if a large number of users have to be trained or permanent training is required.

ii.   Perform Data Conversion

Information necessary for the operation of the new system is converted from existing data sources into a format accessible by the new system. The converted data is then loaded into the data structures associated with the system.

iii.   Install RPMS System

Steps necessary to commence operation of the system in the production environment are performed in this task. Necessary adjustments to the hardware and system software configuration are completed, instructions given to the operations personnel who will be operating the system, and software libraries loaded with the production versions of the application software.

iv.   **Accept System Installation**

The new system installation can be regarded as successful and will be accepted when it operates for a specified period within defined tolerances for performance, error rate and usability. This acceptance is based upon agreements among users, production/operations staff, support staff, and the information management organization about schedules, procedures, pricing policies, warranty agreements, software, and systems documentation.

3.4.1   Testing plan

Testing only take process in the administrator system. The home page of this system is not the main focus on this project. The feedback form is not functioning and its just a dummy. The home page only shows the profile of the company and there is a login button to the administrator. This part will be run for testing purposes and several testings which are unit testing, integration testing and system testing will be operated.

3.4.1.1 Testing

Table 3.12 shows the products table with name and its definition.

Table 3.13 Testing cases

Sub-system	File	Functional being tested	Test data	Expected result
Login	login	Email * Password * (each branches)	mainbranch@gmail.com 1234	Passed
		Email *	mainbranch	Failed
Manage Customer	Create	Title * Last name * First name * Company	Mr/Mrs Ahmad Muhamad Lubuk Kawah	Passed

		Street	Tanah merah	
		Postal code	16250	
		City	Kota Bharu	
		State	Kelantan	
		Telephone	09-7192828	
		Office	017-9822828	
		Mobile Phone	ahmad@gmail.com	
		Email	English	
		Preffered language		
		Title *	Null	Failed
		Last name *	Null	
		First name *	Null	
		Company	(Not required)	
		Street	(Not required)	
		Postal code	abc	
		City	123	
		State	(Not required)	
		Telephone	abc/09-719	
		Office	abc/017-982	
		Mobile Phone	ahmad	
		Email	(Not required)	
		Preffered language		
Manage Production	Production of today	Chart Tabular Calculation	Date format Product Value (any d.p)	Passed
	Create	Date * Product * Quantity *	2014-12-12 AC14 122.12(any d.p)	Passed

Manage Sale	Sale of today	Chart Tabular Calculation	Date format Product Value (any d.p)	Passed
	Create	Date * Product * Quantity * Customer * Production * Transport Location *	2014-12-12 AC-14 122.12(any d.p) 12 12 Insider Lorry Kota Bharu	Passed
Manage Product	Create	Type of product * Code * Description	Valid input	Passed
Report	Daily	Date	2014-12-12	Passed
			abc	Failed
	Date Range	Date from and to	2014-12-12	Passed
			abc	Failed (No validation)
	Monhtly	Date	2014-12-12	Passed
			abc	Failed (No validation)
	Yearly	Date	2014-12-12	Passed
			abc	Failed (No validation)
All Branhes Report	Daily Report	Date	2014-12-12	Passed

			abc	Failed (No validation)
	Date Range	Date from and to	2014-12-12	Passed
			abc	Failed (No validation)
	Monhtly	Date	2014-12-12	Passed
			abc	Failed (No validation)
	Yearly	Date	2014-12-12	Passed
			abc	Failed (No validation)

CHAPTER 4

CONCLUSION

4.1   Conclusion

RPMS is developed for managing the sales department, and production department in SBMSD and its branches based on different modules of sales, and production and encompassed of reporting functionalities. The system will be developed in Web-based management system for maintaining the data.

4.2   Results

i.   Successfully developed a Web-based management system for data entry and management of the information for manager and branch manager

ii. Web-based management system for collecting on of daily reports that dynamically extracting information from different districts into a customize form of tabular and graphical chart is developed for manager and branch manager.

iii. Successfully calculate the difference of sales and production

4.3   Future Works

To design MySQL Cluster, an in-memory clustered distributed database management system in the RPMS. This MySQL Cluster is built on a shared-nothing architecture and includes advanced features such as failover, node recovery, synchronous data replication and no single-point-of failure.

REFERENCES

Leff, A. (2001). *Web-Application Development Using the ModelNiewlController Design Pattern.*

Ren, Z. (2008). *Building a Business Intelligence Application with.* Shanghai.

Rapid application developement. (2010, 12). Retrieved from StudyMode.com: http://www.studymode.com/essays/Rapid-Application-Developement-526159.html

Introduction. (2011). Retrieved from Seri Barat Mixed Sdn Bhd: http://www.seribarat.com/introduction.html

SDLC Waterfall Model. (2014). Retrieved from tutorialspoint: http://www.tutorialspoint.com/

Ansell, S. (2010, April 8). *ERP Software Clash of the Titans: SAP vs. Oracle.* Retrieved from backbonemag: http://www.backbonemag.com/Backblog/erp-software-clash-of-the-titans-sap-vs-oracle.aspx

istqb. (2012, January 11). *What is Waterfall model- advantages, disadvantages and when to use it?* Retrieved from istqbexamcertification: http://istqbexamcertification.com/

Oracle. (2014). *Oracle E-Business Suite.* Retrieved from Oracle: http://www.oracle.com/us/products/applications/ebusiness/overview/index.html

SAP. (2014). *Leverage world-class ERP software.* Retrieved from SAP: http://www.sap.com/pc/bp/erp.html

APPENDIX A WEB DEVELOPMENT QUESTIONNAIRE

Part I - Basic Website Information

The following questionnaire will help us estimate the timeline and cost of your project. The success of this undertaking directly depends on clearly established project vision and goals. It is important that you answer all applicable questions thoroughly.

1. CONTACT INFORMATION			
Company Name:	SERI BARAT MIXED SDN BHD	City:	Kota Bharu
Address:	602,2 Jln Hamzah	State/Province:	Kelantan
		Zip/Postal Code:	15050
Contact Person:	09-7482979	Country:	Malaysia
Position:	Manager	Telephone:	
Email:		Mobile Phone:	

2. DOMAIN NAME INFORMATION							
2A.	Do you already have a domain name?	No	http://				
2B.	If you have multiple domains, please list them all here, and specify which one will be the default:	No					
2C.	If no, please list the top three choices for your domain name:						
a.	rpms		.com	~~.net~~	~~.org~~	~~.biz~~	~~.us~~

Part II - Functionality

4. CONTENT MANAGEMENT	
Content Management System (CMS) enables you to add / edit / delete and organize pages on your website. If your website has e-commerce functionality, the display and management of sales, production, and others. Please let us know which of the following content management features you would like on your website.	
7A. Will you need the CMS functionality?	Yes/~~No~~
7B. Are there any specifics we need to know pertaining to CMS for your site? Please describe. Sales, and production managements	

8. USER MANAGEMENT
User management usually means visitors can create an account connected to the site, which gives them access to "restricted" content, store their preferences, see wholesale prices, use certain features reserved only for registered users, etc. If you have a need for such a system, please describe it in detail in the area below. Describe which information will be needed to create an account, which credentials will be required to log into an account, will the administrator intervention be required before an account becomes active, what benefits will a registered user have, are there multiple user levels / groups, etc.

8A.	I don't need. Currently, only 4 branches.

9. SEARCH ENGINE OPTIMIZATIONS
At this time the most relevant search engine and industry standard setter is Google. This is why your website will be optimized according to current Google guidelines. Please list below up to five key phrases / keywords for which you think your website will be relevant. In other words, which words or phrases should a person type in a search engine to find your website.

9A.	Not applied

PART III – Project Goal and Expectations

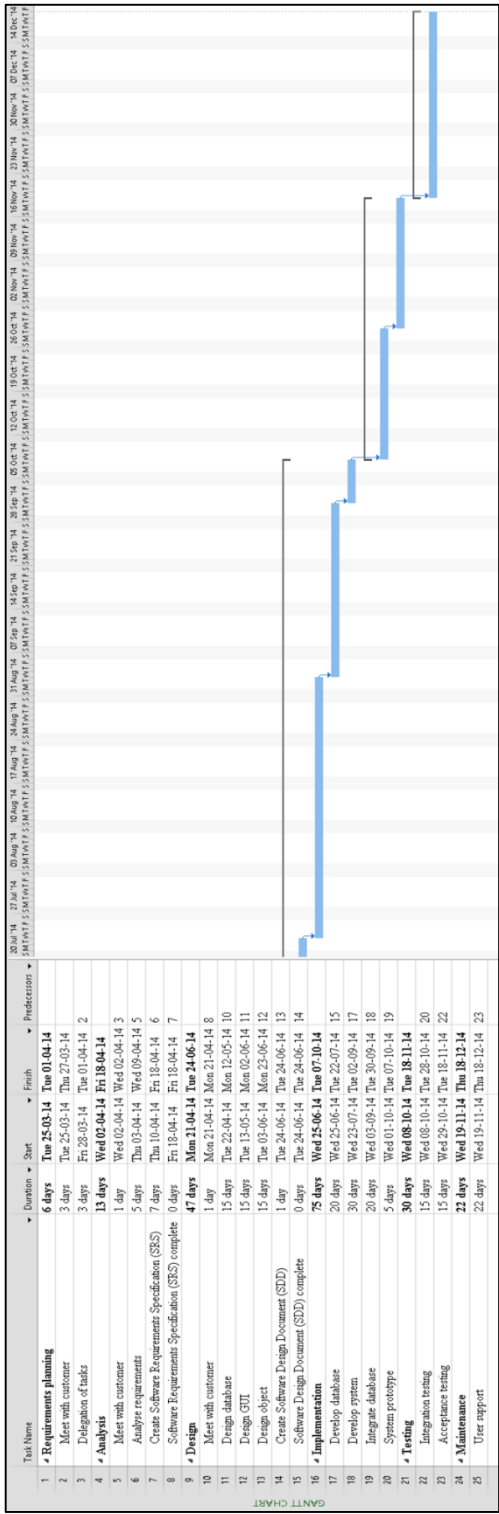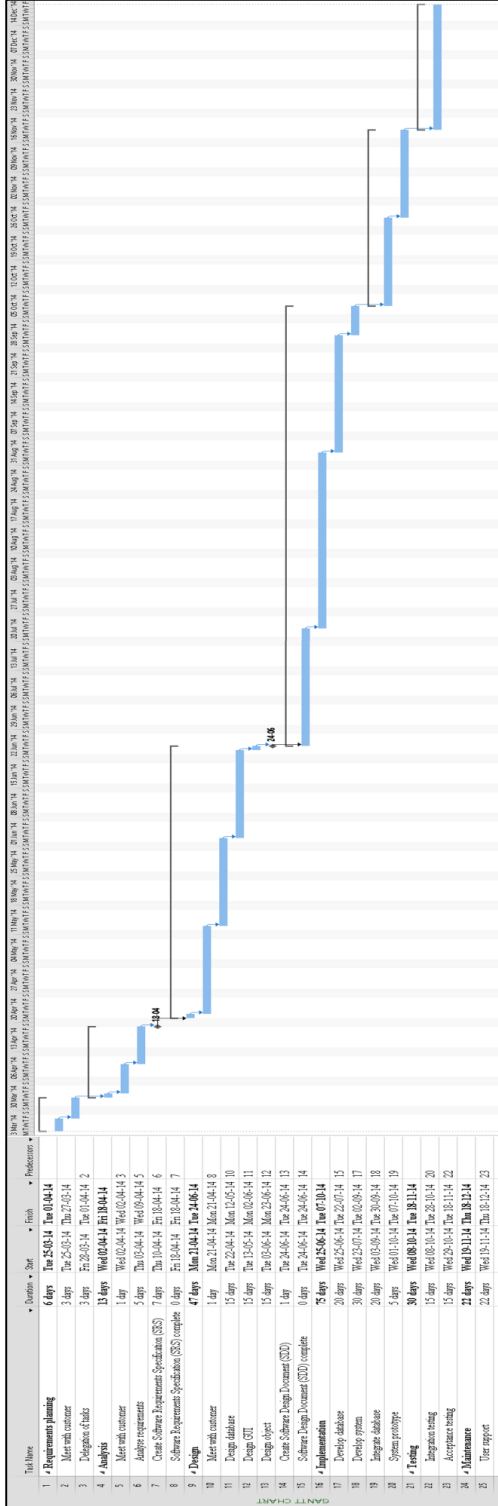| 11. PROJECT GOALS AND EXPECTATIONS |
|---|
| Please use the area below to clearly explain:
1. Why do you want a system?
2. What would you identify as the single, most important purpose of your system? |

11A.	I want this system can handle 4 branches. I want my company to collect data from other branches in sales and productions department. I want this website to manage data and produce chart and table format.

APPENDIX B GANTT CHART

#	Task Name	Duration	Start	Finish	Predecessors
1	◢ **Requirements planning**	**6 days**	**Tue 25-03-14**	**Tue 01-04-14**	
2	Meet with customer	3 days	Tue 25-03-14	Thu 27-03-14	
3	Delegation of tasks	3 days	Fri 28-03-14	Tue 01-04-14	2
4	◢ **Analysis**	**13 days**	**Wed 02-04-14**	**Fri 18-04-14**	
5	Meet with customer	1 day	Wed 02-04-14	Wed 02-04-14	3
6	Analyse requirements	5 days	Thu 03-04-14	Wed 09-04-14	5
7	Create Software Requirements Specification (SRS)	7 days	Thu 10-04-14	Fri 18-04-14	6
8	Software Requirements Specification (SRS) complete	0 days	Fri 18-04-14	Fri 18-04-14	7
9	◢ **Design**	**47 days**	**Mon 21-04-14**	**Tue 24-06-14**	
10	Meet with customer	1 day	Mon 21-04-14	Mon 21-04-14	8
11	Design database	15 days	Tue 22-04-14	Mon 12-05-14	10
12	Design GUI	15 days	Tue 13-05-14	Mon 02-06-14	11
13	Design object	15 days	Tue 03-06-14	Mon 23-06-14	12
14	Create Software Design Document (SDD)	1 day	Tue 24-06-14	Tue 24-06-14	13
15	Software Design Document (SDD) complete	0 days	Tue 24-06-14	Tue 24-06-14	14
16	◢ **Implementation**	**75 days**	**Wed 25-06-14**	**Tue 07-10-14**	
17	Develop database	20 days	Wed 25-06-14	Tue 22-07-14	15
18	Develop system	30 days	Wed 23-07-14	Tue 02-09-14	17
19	Integrate database	20 days	Wed 03-09-14	Tue 30-09-14	18
20	System prototype	5 days	Wed 01-10-14	Tue 07-10-14	19
21	◢ **Testing**	**30 days**	**Wed 08-10-14**	**Tue 18-11-14**	
22	Integration testing	15 days	Wed 08-10-14	Tue 28-10-14	20
23	Acceptance testing	15 days	Wed 29-10-14	Tue 18-11-14	22
24	◢ **Maintenance**	**22 days**	**Wed 19-11-14**	**Thu 18-12-14**	
25	User support	22 days	Wed 19-11-14	Thu 18-12-14	23

APPENDIX C CLIENT DOCUMENTATION

APPENDIX D USER ACCEPTANCE TESTING

SERI BARAT MIXED SDN BHD

Functions	Descriptions	Criteria
Login	Login into RPMS system	
Dashboard	Today's Production calculate the total of today production	√
	Last week calculate the total production of last week	√
	Today's Sale calculate the total of today sale	√
	Last week calculate the total sale of last week	√
	Difference calculate the difference of production and sale	√
	Percentage difference calculate percentage of difference	√
	Customers counts total of customer	√
	Yearly chart shows all the monlhly sales and productions	√
	Company Perfomance Yearly shows chart of the branch	√
	Company Performance of All Branches Daily shows the all branches	√
Manage Customer	All customers shows all of the customer from all branches	√
	Create new customer store a new customer	√
	Change customer can show, edit, and delete customer	√
Manage Production	Production's of today shows chart and tabular table	√
	Notifications shows yesterday and today production	√
	Create new production store new production	√
	Change production can show, edit and delete production	√
	Deleted production show the deleted production	√
	Searching for all production records	√
Manage Sale	Sale's of today shows chart and tabular table	√
	Notifications shows yesterday and today sale	√
	Create new sale store new sale	√
	Change sale can show, edit and delete sale	√
	Deleted sales show the deleted sale	√

	Searching for all sale records	√
Manage Product	All product shows all of the products	√
	Create new product store new product	
Report	Daily report will shows daily report of branch	√
	Date range report shows the date range report of branch	√
	Montly report shows the monthly report of branch	√
	Yearly report shows the yearly report of branch	√
All Branches Report	Daily report will shows daily report of other branch	√
	Date range report shows the date range report of other branch	√
	Montly report shows the monthly report of other branch	√
	Yearly report shows the yearly report of other branch	√
Logout	Logout of the RPMS system	√

No.	Comments
	Improve your two of sales and production Sales and production display correctly.

Signature: ……………………………………….

Name: Henry Tan

Date: 12 Dec 2014

KUARI DINAR SDN BHD

Functions	Descriptions	Criteria
Login	Login into RPMS system	
Dashboard	Today's Production calculate the total of today production	√
	Last week calculate the total production of last week	√
	Today's Sale calculate the total of today sale	√
	Last week calculate the total sale of last week	√
	Difference calculate the difference of production and sale	√
	Percentage difference calculate percentage of difference	√
	Customers counts total of customer	√
	Yearly chart shows all the monlhly sales and productions	√
	Company Perfomance Yearly shows chart of the branch	√
	Company Performance of All Branches Daily shows the all branches	√
Manage Customer	All customers shows all of the customer from all branches	√
	Create new customer store a new customer	√
	Change customer can show, edit, and delete customer	√
Manage Production	Production's of today shows chart and tabular table	√
	Notifications shows yesterday and today production	√
	Create new production store new production	√
	Change production can show, edit and delete production	√
	Deleted production show the deleted production	√
	Searching for all production records	√
Manage Sale	Sale's of today shows chart and tabular table	√
	Notifications shows yesterday and today sale	√
	Create new sale store new sale	√
	Change sale can show, edit and delete sale	√
	Deleted sales show the deleted sale	√
	Searching for all sale records	√

3Manage Product	All product shows all of the products	√
	Create new product store new product	
Report	Daily report will shows daily report of branch	√
	Date range report shows the date range report of branch	√
	Montly report shows the monthly report of branch	√
	Yearly report shows the yearly report of branch	√
All Branches Report	Daily report will shows daily report of other branch	√
	Date range report shows the date range report of other branch	√
	Montly report shows the monthly report of other branch	√
	Yearly report shows the yearly report of other branch	√
Logout	Logout of the RPMS system	√

No.	Comments
	Generate chart slow Design oL.

SERI BARAT MIXED SDN BHD
602,2 Jalan Hamzah, 15050 Kota Bharu.
Kelantan. Malaysia.
Tel: +609 7482979 Fax: +609 7485606
Email: enquiries@seribarat.com

Signature: ……………………………………….

Name: Henry Tan

Date: 12 Dec 2014

APPENDIX E IMPLEMENTATION CODE

```php
<?php

class ProductionController extends BaseController {

        public function index()
        {
                $user           = Sentry::getUser();
                $user_id        = $user->id;

                $production = DB::table('productions')
                                        ->where('date', '=', DB::raw('CURDATE()'))
                                ->where('productions.user_id', '=', $user_id)
                                        ->get();

                return View::make('admin.productions.index')->with('production',
$production);
        }
        public function create()
        {
                return View::make('admin.productions.create');
        }
        public function store()
        {
                $validation = Validator::make(Input::all(), Production::$rules);

                if($validation->fails()) {

                        return Redirect::route('admin.productions.create')-
>withErrors($validation);

                }else {

                        $user                   = Sentry::getUser();
                        $user_id                = $user->id;

                        Production::create(array(
                                'date'          => Input::get('date'),
                                'product'       => Input::get('product'),
                                'quantity'      => Input::get('quantity'),
                                'from'          => Input::get('from'),
                                'to'            => Input::get('to'),
                                'user_id'       => $user_id,
                        ));
```

```php
                Flash::success('You had successfully created a new production!');
                return Redirect::to('admin/productions');
        }
    }
    public function show($id)
    {
        $production = Production::find($id);

        return View::make('admin.productions.show')->with('production',
$production);
    }
    public function edit($id)
    {
        $production = Production::find($id);

        return View::make('admin.productions.edit')->with('production',
$production);
    }
    public function update($id)
    {
        $validation = Validator::make(Input::all(), Production::$rules);

        if ($validation->fails()) {

                return Redirect::to('admin/productions/' . $id . '/edit')-
>withErrors($validation);

        } else {

                $production                 = Production::find($id);
                $production->date       = Input::get('date');
                $production->product    = Input::get('product');
                $production->from       = Input::get('form');
                $production->to                  = Input::get('to');
                $production->quantity   = Input::get('quantity');
                $production->save();

                Flash::success('You had successfully updated the production ' . $id.
'!');
                return Redirect::to('admin/productions');
        }
    }
    public function destroy($id)
    {
        $production = Production::find($id);
```

APPENDIX F TURNITIN