# E-TECHIFY

**WAN MUHAMMAD ISMAT BIN WAN AZMY**

**UNIVERSITI TEKNIKAL MALAYSIA MELAKA**

E-TECHIFY

WAN MUHAMMAD ISMAT BIN WAN AZMY

This report is submitted in partial fulfillment of the requirements for the
Bachelor of Computer Science (Software Development) with Honours.

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2021

**DECLARATION**

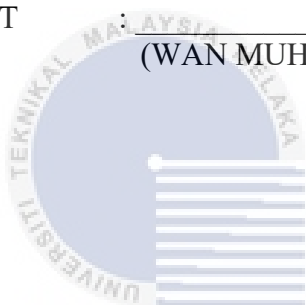I hereby declare that this project report entitled

**E-TECHIFY**

is written by me and is my own effort and that no part has been plagiarized

without citations.

STUDENT        : _____        Date : 12 Sept 2021
                        (WAN MUHAMMAD ISMAT BIN WAN AZMY)

I hereby declare that I have read this project report and found

this project report is sufficient in term of the scope and quality for the award of

Bachelor of Computer Science (Software Development) with Honours.

Ts. MASHANUM BINTI OSMAN
Pensyarah Kanan
Jabatan Kejuruteraan Perisian
Fakulti Teknologi Maklumat dan Komunikasi
Universiti Teknikal Malaysia Melaka (UTeM)

SUPERVISOR      : _____        Date : 12 Sept 2021
                        (TS. MASHANUM BINTI OSMAN)

# DEDICATION

This is dedicated to my beloved parents and my supervisor, who have been our source of inspiration and gave me the strength when we thought of giving up, who continually provide their moral, spiritual, emotional, and financial support.

To my sisters, friends and anyone who shared their words of advice and encouragement to finish this project.

# ACKNOWLEDGEMENTS

# ABSTRACT

e-Techify acts as an e-commerce, inventory management and a tracking system. This is a web-based system, which are the main focus is to develop a simple, and efficient system for end user. It is built with Laravel, jQuery and MySQL as its database. The people who involve in the system are customer, reseller, and distributor. Various functionalities and data are provided with ease of access to all the users involved. Problems such as loss of data due to unsystematic way of organizing, human error itself and stock count is not real-time that could be solve by developing this system. The system will be up and running regardless of time and it is accessible anytime and anywhere. Customer purchasing experience can improve and it is more efficient rather than having to risk themselves of being expose to public as we know that will prone to the deadly virus. The objective of this system is to produce a systematic system that can manage every automated process without any hiccups, open more side jobs opportunity and keep inventory updated in real-time. The methodology used for e-Techify is Waterfall model. In this model, typically, the outcome of one phase acts as the input for the next phase sequentially. After a vast discussion, data and requirements are collected from Xmiryna Technology as they wanted a solution for problems in the computer market from customer to distributor part. Efficiency, proper data management and communication could be the key to success. Each and every process that is automated will ease the burden and require less man power. Cost saving and increase of profit will occur automatically as a special platform is provided to each users of the system. All of the functions and system accessibility is at fingertips. This is one of the core subjects that must not be underestimated.

# ABSTRAK

e-Techify bertindak sebagai e-commerce, pengurusan inventori dan sistem penjejakan. Ini adalah sistem berasaskan web, yang menjadi fokus utama adalah mengembangkan sistem yang mudah dan efisien untuk pengguna akhir. Ia dibina dengan Laravel, jQuery dan MySQL sebagai pangkalan data. Orang yang terlibat dalam sistem ini adalah pelanggan, penjual semula, dan pengedar. Pelbagai fungsi dan data disediakan dengan kemudahan akses kepada semua pengguna yang terlibat. Masalah seperti kehilangan data kerana cara mengatur yang tidak sistematik, kesalahan manusia itu sendiri dan jumlah stok bukan masa nyata yang dapat diselesaikan dengan mengembangkan sistem ini. Sistem ini akan berjalan lancar tanpa mengira masa dan boleh diakses bila-bila masa dan di mana sahaja. Pengalaman membeli pelanggan dapat bertambah baik dan lebih efisien daripada harus mengambil risiko untuk terdedah kepada umum kerana kita tahu bahawa ia akan terdedah kepada virus yang berbahaya. Objektif sistem ini adalah untuk menghasilkan sistem yang sistematik yang dapat menguruskan setiap proses automatik tanpa cegukan, membuka lebih banyak peluang pekerjaan sampingan dan memastikan inventori dikemas kini dalam masa nyata. Metodologi yang digunakan untuk e-Techify adalah model Waterfall. Dalam model ini, biasanya, hasil satu fasa bertindak sebagai input untuk fasa seterusnya secara berurutan. Setelah perbincangan yang luas, data dan keperluan dikumpulkan dari Xmiryna Technology kerana mereka mahukan penyelesaian untuk masalah di pasaran komputer dari pelanggan hingga bahagian pengedar. Kecekapan, pengurusan data dan komunikasi yang tepat dapat menjadi kunci kejayaan. Setiap proses yang automatik akan meringankan beban dan memerlukan tenaga manusia yang lebih sedikit. Penjimatan kos dan peningkatan keuntungan akan berlaku secara automatik kerana platform khas disediakan untuk setiap pengguna sistem. Semua fungsi dan kebolehaksesan sistem berada di hujung jari. Ini adalah salah satu perkara penting yang tidak boleh dipandang rendah.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

## LIST OF ABBREVIATIONS

| | | |
|---|---|---|
| **FYP** | **-** | **Final Year Project** |
| **RMA** | **-** | **Return Merchandise Authorization** |
| **IDE** | **-** | **Integrated Development Environment** |
| **COVID-19** | **-** | **Corona Virus Disease 2019** |
| **TC** | **-** | **Test Case** |
| **UCT** | **-** | **Use Case Testing** |

# LIST OF ATTACHMENTS

**PAGE**

# CHAPTER 1:  INTRODUCTION

## 1.1    Introduction

The e-Techify system will be developed using Laravel and MySQL. The purpose of this idea is to automate all the process and keeping it in a safe database without having to worry about loss of data. Every process is designed to help and ease every user in the system. All of the data or information will be precisely store as this involves important data such as money transaction history, stock information and etc. Possibilities like redundant data, miscount of stock, lost track of current parcel are commonly happened in the current system. In order to avoid these possibilities, validation will be done carefully for every process. Various implementation will be implemented and failure rate due to human errors can be reduce with the assistance of e-Techify. This system is 24 hours and 7 days online, regardless of system user's location and time. They can shop, manage and arrange anywhere, anytime without any restrictions. The main purpose of developing this system is to centralize all process intentionally for customer, reseller and distributor for computer parts. Some of the major additional functionality to make sure this system can give benefits to whom that wants to do a freelance job that could generate side income for themselves. The most important of all, this system existence will help the users of the system to ease their work without having to expose their own self to the outside whereas we know that the condition since Covid-19 pandemic might be very dangerous gives a big impact to everyone that face this situation. To some extent, this will also help the government reduce the current cases. We can no longer keep using the traditional method as it is a totally different condition right now. This system is the combination of e-commerce and inventory type of system exclusively for computer parts. Without having hesitation of going online during this pandemic, people on the internet almost most of the time.

**1.2     Problem Statements**

- Traditional method that will prone to loss of data & un-systematic way of organizing all process.

- No platform that offers job opportunity related to computer services specifically.

- Reseller having hard time to get stock confirmation and it is not real-time.

**1.3     Objective**

1. Produce a systematic system that can manage every automated process without any hiccups.
2. To open more side jobs opportunity, to generate income.
3. Ability to keep inventory updated in real-time.

**1.4     Scope**

**Distributor Module**

In distributor module where it has the ability to manage product, insert serial number in the system to let reseller assign to customer ordered items, manage jobs and manage related Return Merchandise Authorization (RMA) request by customer. Reporting in form of bar and pie charts are also provided to ease the distributor itself to see the simplify report for their performance.

**Customer Module**

In customer module where customer can shop for desire items, send RMA requests, track their own parcel, manage their orders and get job info advertised. All of the payments will be gone through PayPal that offers seamless experience for all customers of e-Techify.

**Reseller Module**

In reseller module it has the ability to manage orders, manage jobs and manage stock from distributor. The dashboard will show reporting in form of bar charts are also provided to ease the reseller itself to see the simplify report for their performance in terms of earnings monthly, earnings annual and specific brand sold per month.

**Tracker Module**

Customer has the access to check their parcel status. Including every detail that is needed to make sure customer is informed for every status updated. Customers may track the progress of their orders from the time they are packed until they are successfully delivered to their doorstep.

**Stock Management Module**

Every stock that is updated by every distributor will be visible for reseller to view all related information. This could give a better view for reseller to do a cross check for their inventory with every distributor.

1.5    **Project Significance**

e-Techify will replace the current traditional system and provide benefits to customer, reseller and distributor that registered with this system. All of the process will be systematically in order to ease users of this system.

1.6    **Expected Output**

e-Techify was built to fulfil the requirements in order to solve the problem statements. I was expecting to get this project done within the time given. This project is resulted from my own observation in this technology industry especially in computer parts. With the help of this system, manual processes can be automated as it can give benefits to everyone that uses this system. Moreover, I hope that by developing this system, positive outputs

would come out for a long-term benefit that could give everyone their own advantages.

## 1.7 Conclusion

This chapter opens with an overview of the role of information technology in the new pandemic norm, as well as the value of this technology to the community. Following that, this chapter discusses the key issues that users of the existing approach are facing, as well as the project's goals. This chapter also discusses the project's importance and contributions, as well as the predicted outcomes when the project is completed. The following chapter, Chapter 2, Literature Review, goes through the information technology used in e-commerce inventory as well as the development approach.

# CHAPTER 2:  LITERATURE REVIEW AND PROJECT METHODOLOGY

## 2.1     Introduction

The facts and findings will be discussed in detail in this chapter of the Literature Review to expose the necessary concepts, theories, and knowledges connected to the system and system domain. The analysis of existing products and systems that provide the same functions as 2.2.2 is discussed in the following part, which consists of comparisons between existing products and this project. Following that, the chosen methodology will be presented, and all activities related to the methodology will be discussed, as in 2.3. Following that, in section 2.4, a list of all the software and hardware requirements is shown. The project schedule will be presented, and milestones will be specified in the final section.

## 2.2     Facts and findings

Web application, information technology application in a mix of e-commerce, inventory, and tracking system are the domains associated to this project. The next parts, from 2.2.1 through 2.2.3, include the explanation.

## 2.2.1    Domain

This section covers the basics of web applications and their prevalent uses in today's society. Then there's a discussion about the advantages of web applications for humans.

### Introduction of Web Application

The evolution of websites or web systems leads to the development of web apps. According to Stuttard (2011), the World Wide Web contained just websites at the start of the internet, which acted as information repositories

containing static documents. Web browsers were developed as a means of extracting and displaying documents.

Web applications use a combination of server-side scripts (PHP and ASP) to store and retrieve data, as well as client-side scripts (JavaScript and HTML) to present data to users. Users can use online forms, content management systems, shopping carts, and other tools to communicate with the organization. Employees may also use the apps to create documents, share information, collaborate on projects, and work on shared documents regardless of where they are or what device they are using.

**Common Web Application Functions**

Web applications are designed to perform almost any useful function or service. In recent years, various web application features have become increasingly popular and widespread. Amazon, for example, offers online shopping, Facebook, a popular social networking site, online banking for both local and foreign banks, Gmail, a web mail service, and Wikipedia, which gives interactive knowledge (Stuttard, 2011).

The current state of information technology is causing an increasing number of web-based applications to be replaced with mobile applications that can be accessed via a smartphone or tablet. The majority of mobile apps connect to the server using a web browser or a specialised client-server application that employs HTTP-based APIs like web services.

Web apps are widely utilised within enterprises to support fundamental business processes, in addition to being widely used on the public Internet. Many applications are in charge of providing access to very sensitive information and functionality. A human resource application, for example, is a platform for retrieving payroll information, writing and receiving performance reviews, and managing recruitment. Second, although being exclusively accessible through a proprietary third-client programme, enterprise resource planning (ERP) software is an example of a business programme that can now

be browsed using a web browser. Another example is cloud services like Microsoft Office Online and Google Apps for Work, which shift traditional desktop office tools like power point slides and spreadsheets to the cloud or web. The examples above, which are generally referred to as "internal" apps, were hosted externally when corporations moved them to outside service providers to save money. These issues are becoming more prevalent presently. As a result, most computer users' only client software will be a web browser, and this is rapidly approaching (Stuttard, 2011).

**Benefits of Web Application**

Web-based applications offer a variety of business benefits as compared to desktop apps. Instead of having to be installed on each computer that you want to access it from, these programs can be accessible via the internet from any computer. Software as a Service (SaaS) refers to the use of web-based software that runs in a virtual, cloud-based environment.

- Web applications run on multiple platforms regardless of OS or device as long as the browser is compatible.
- All users access the same version, eliminating any compatibility issues.
- They are not installed on the hard drive, thus eliminating space limitations.
- Have a chance to be a Progressive Web Apps (PWA), which could be install in mobile devices.

**Information Technology in E-Commerce**

Information Technology (IT) moves in a rapid pace and drives drastic changes to all sectors of today's economy including education, business, manufacturing, etc. One of the fastest evolving is e-commerce. Ecommerce, often known as electronic commerce or online commerce, is the purchasing and selling of goods and services through the internet, as well as the financial and data transfers required to complete these transactions. Ecommerce is frequently used to refer to the online sale of real goods, but it can also refer to

any type of economic transaction that is made possible by the internet. Whereas e-business encompasses all aspects of running an online firm, ecommerce focuses on the exchange of goods and services. There are four basic ecommerce models that can be used to explain practically any transaction between customers and businesses.

- Business to Consumer (B2C)

  When a business sells a good or service to an individual consumer (e.g., You buy a pair of shoes from an online retailer).

- Business to Business (B2B)

  When a business sells a good or service to another business (e.g., A business sells software-as-a-service for other businesses to use)

- Consumer to Consumer (C2C)

  When a consumer sells a good or service to another consumer (e.g., You sell your old furniture on eBay to another consumer).

- Consumer to Business (C2B)

  When a consumer sells their own products or services to a business or organization (e.g., An influencer offers exposure to their online audience in exchange for a fee, or a photographer licenses their photo for a business to use).

**Information Technology in Inventory Management System**

Inventory system is also one of the fastest sectors that follows Information Technology (IT) movement. Perhaps the most significant advantage of employing inventory management software is that it simplifies the process of maintaining your inventory, saving you time, money, and, quite frankly, your sanity. Inventory management software helps to eliminate the risk of human error by automating key business processes. With supply and demand changing throughout the year and stock levels constantly fluctuating, inventory management software helps to avoid the risk of human error by

automating key business processes. Stock-outs and excess inventory are avoided at all costs. When it comes to inventory management, finding the right balance can be tough. After all, having insufficient supply might result in a stock-out, disgruntled customers, and significant revenue loss. Excess goods, on the other hand, can eat up valuable warehouse space and unnecessary cost.

### 2.2.2    Existing System

Traditional method which involves many processes can be the main problem that all people will face. On the top of that since Covid-19 pandemic hit all of the country in this world which are getting worst day by day. Automation is one of the key to help every country by having less physical contact in order to reduce everyday cases. The figure below shows every process from start to end of a transaction. Every transaction involves a customer, reseller & distributor. A long process like this could make it time consuming which could be reduced by having a systematic process through a system.
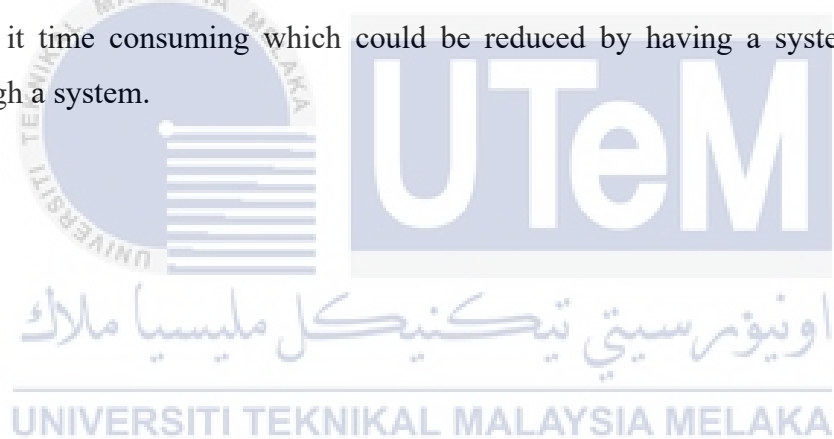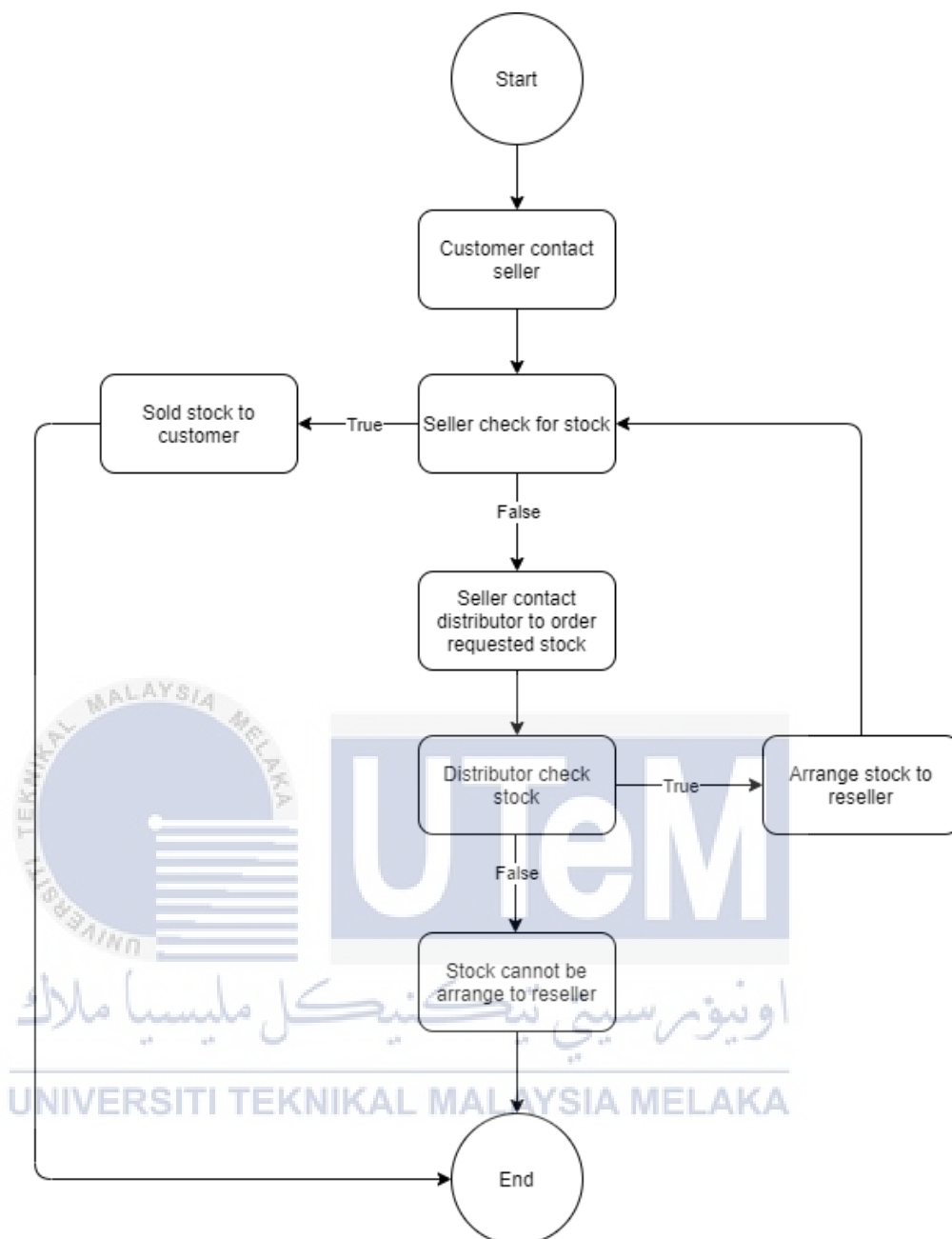
Figure 2.0: Activity Diagram of Current System

**Other Reference for Existing System**
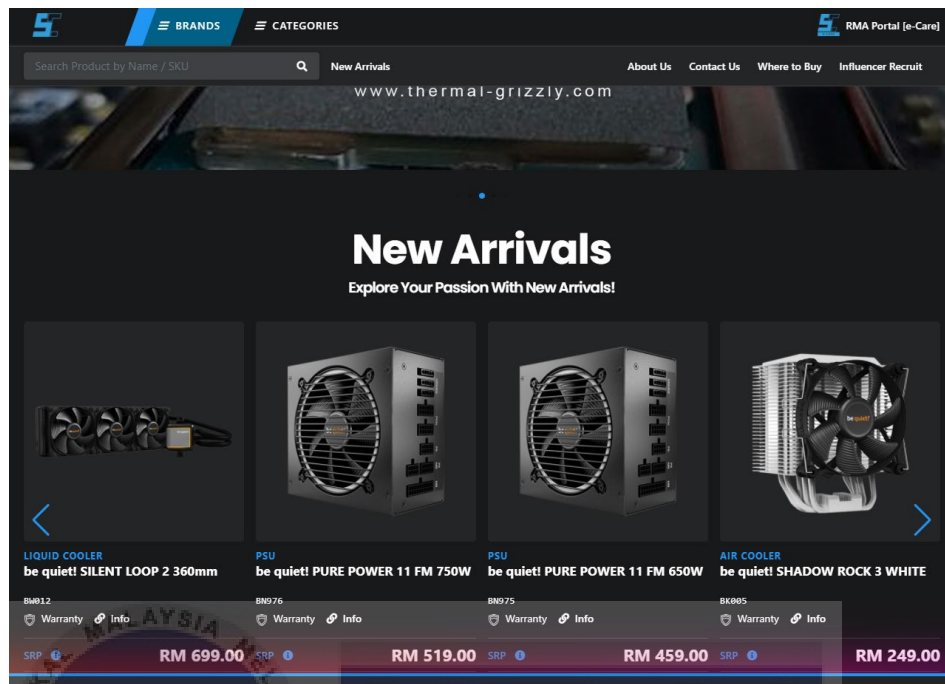


Figure 2.1: Sun Cycle Website

Sun Cycle website only offers viewing for the products that they brought in and RMA portal for end user to send warranty for their distributed products. The features that they lack of are end user cannot purchase directly through their website and limited products offered.

### 2.2.3 Technique

There are several techniques that have been done to gather more information about e-Techify which are:

**Observation**

Observation was carried out to collect and gather more information. As for now I found out that this traditional process has already been going through many years. All of the existing platforms did not offer additional service such as Return Merchandise Authorization (RMA), side-job offerings and stock management integration with their distributor.

**Internet Resources**

Internet resources is one of the simplest ways to find out feedback from other people on how the current system is. The e-commerce of technology items is not frequently known as other platforms that offers various items on their shelf to be sold. My main feedbacks and reviews are taken from a pc gaming community. Truth to be told the community wished that there is a platform solely for technology items that offers different kind of services as stated in the observation. There are some main concerns such as the handling of their ordered goods. I hope by developing this e-Techify could bring benefits to the community and also every people that involve in the process.

## 2.3    Project Methodology
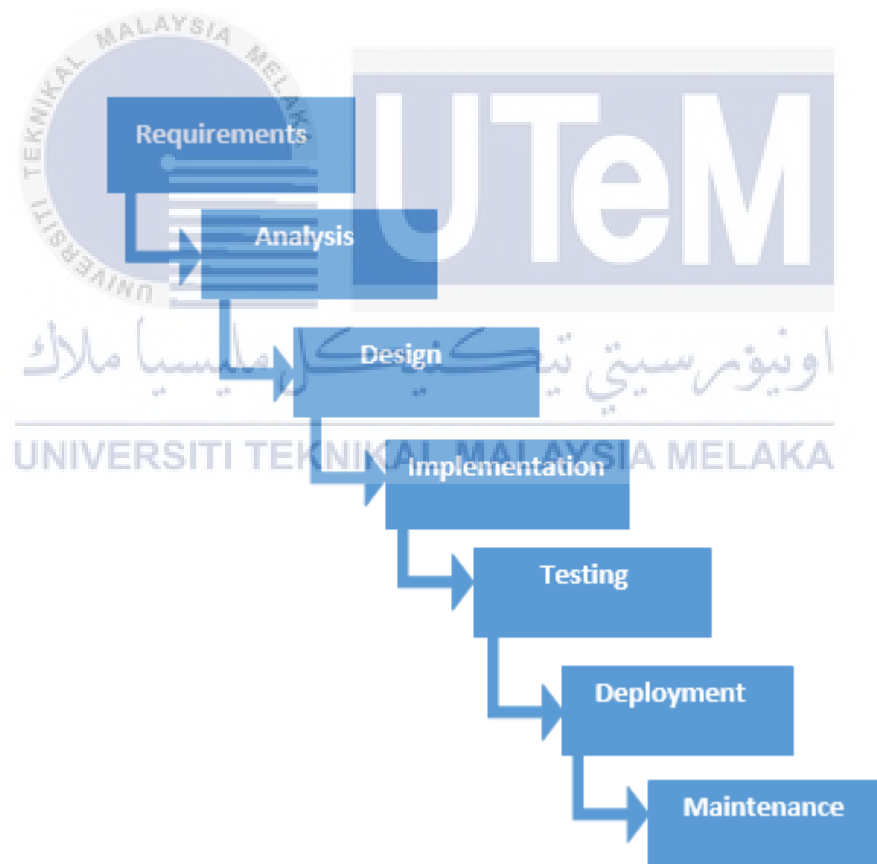


Figure 2.2: Waterfall SDLC Methodology

The Waterfall Model was the first Process Model to be introduced. It is also referred to as a systematic linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed

before the next phase can begin and there is no overlapping in the phases. The Waterfall model is the earliest SDLC approach that was used for software development. The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap. The sequential phases in Waterfall model are –

1. Planning and analysis − All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.

2. System Design − The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.

3. Implementation − With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.

4. Integration and Testing − All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

5. Deployment of system − Once the functional and non-functional testing is done, the product is deployed in the customer environment or released into the market.

6. Maintenance − There are some issues which come up in the client environment. To fix those issues, patches are released and also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model". In this model, phases do not overlap. Some situations where the use of Waterfall model is most appropriate are requirements are very well documented, clear and fixed. There are no ambiguous requirements and ample resources with required expertise are available to support the product.

The advantages of waterfall development are that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one. Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order. Some of the major advantages of the Waterfall Model are 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 and easy to arrange tasks.

The disadvantage of waterfall development is that it does not allow much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-documented or thought upon in the concept stage. The major disadvantages of the Waterfall Model are not suitable for the projects where requirements are at a moderate to high risk of changing. So, risk and uncertainty are high with this process model. High amounts of risk and uncertainty. Adjusting scope during the life cycle can end a project.

**2.3.1    Requirement Gathering and Analysis**

Table 2.0 lists all of the activities involved in acquiring and analyzing requirements. It also comprises the results of the corresponding set of activities.

Table 2.0: Activities and Outcomes in Requirement Gathering and Analysis

| Activities | Outcomes |
|---|---|
| Identify problems faced in current or existing system and gather requirements. | Problems: The problem faced by the current system are all of the process still using the manual method. It will be time consuming for each process and prone to human errors. Moreover, lack of functionality which it offers, and it is more to giving an advantage to some people but not to a whole. |
| Analyses collected requirements and categories into main modules. | Modules: 1. Distributor Module 2. Reseller Module 3. Customer Module |
| Document functional and non-functional requirements, then arrange a meeting to confirm them. | |
| Refine all the requirements and make final confirmation on the latest requirements. | |
| Construct diagram and requirement specification document. | |
| To determine the functional requirements, an analysis of a programming language, tools, and library was conducted. | Language: Laravel UI/UX: HTML5, Bootstrap, CSS, jQuery. Library: All open-source library. |

## 2.3.2 System Design

Table 2.1 presents all the activities that are carried out in system design phase. It also includes the outcomes associated with the related set of activity.

Table 2.1: Activities and Outcomes in System Design

| Activities | Outcomes |
|---|---|
| Develop a user interface for each input and output screen. Hold a discussion about the design that was created. Refine the user interface. All of the screen interface designs' final versions are included in this document. | The final version designs of the input and output screen in all the modules are shown and explained in the Chapter 4.2.2. |
| Update all the system requirement specifications | The final version of the functional and non-functional requirements is listed and explained in the Chapter 3.3.2 and 3.3.3. |
| Discuss the process flow from data entry to final output using the interface design as a temporary prototype. | |
| Design software architecture | The system architecture is discussed in the Chapter 3.2 associated with the diagram. |
| Develop database design, such as Entity Relationship Diagram, Data Flow Diagrams. | All the conceptual, logical, and physical database designs are included in Chapter 3 with explanation and details. Different levels of data flow diagram also can be read in the Chapter 3. |
| Develop algorithm for each functionality in every module | |

### 2.3.3  Implementation

Table 2.2 presents all the activities that are carried out in implementation phase. It also includes the outcomes associated with the related set of activity.

Table 2.2 Activities and Outcomes in Requirement Implementation

| Activities | Outcomes |
|---|---|
| Develop source code for each module based on the design created during the Design phase. | |
| For unit testing, create and construct test cases for each module. Each module should undergo unit testing. | |
| Integrate modules in the system. | |
| Every day, run version control and backup processes. All code changes should be committed to GitHub, and daily backups should be made to the OneDrive folder. | Chapter 5.3 explains steps of practices of this activity. |

### 2.3.4  Testing

Table 2.3 presents all the activities that are carried out in testing phase. It also includes the outcomes associated with the related set of activity.

Table 2.3: Activities and Outcomes in Testing

| Activities | Outcomes |
|---|---|
| For integration testing and system testing, create and construct test cases for each module. | Detailed explanation of testing activities are discussed on the Chapter 6, Testing. |
| Carry out integration testing and keep track of the results. Collect bugs, debug them, and describe the fixes. Rerun the test case that uncovered the flaws. | |

**2.4     Project Requirements**

This section lists out the software and hardware requirements related to the development of this project.

**2.4.1    Software Requirements**

- Development Tools – PHP Storm, GitHub, XAMPP.
- Operating System – Microsoft Windows.
- Database – MySQL

**2.4.2    Hardware Requirements**

- A personal computer (PC) with at least 4GB RAM, a Core i3 processor, and Windows 7 or higher is required.

**2.4.3    Other Requirements**

- Internet Access (at least 1Mbps).

## 2.5 Project Schedule and Milestones

Gantt chart or Program Evaluation Review Technique (PERT) diagram is used to visualize the plan from beginning to finish time.

Table 2.4: Project Schedule

| Activity/Task | Start | End | Week | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| Sending proposal | W1 | W1 | ■ | | | | | | | | | | | | | |
| Approval proposal & gather requirements | W2 | W2 | | ■ | | | | | | | | | | | | |
| System Planning to be developed | W3 | W3 | | | ■ | | | | | | | | | | | |
| Project Identification, System Development & Analysis Planning | W3 | W5 | | | ■ | ■ | ■ | | | | | | | | | |
| System Design and Delivery System Design | W5 | W6 | | | | | ■ | ■ | | | | | | | | |
| System Implementation and Delivery of System Implementation | W6 | W12 | | | | | | ■ | ■ | ■ | ■ | ■ | ■ | ■ | | |
| Final Report Delivery | W13 | W13 | | | | | | | | | | | | | ■ | |
| System Presentation and Assessment | W14 | W14 | | | | | | | | | | | | | | ■ |

## 2.6　Conclusion

This chapter introduces a web-based system that combines different system kinds with web server and client interaction. This chapter also addresses the project's underlying ideas, such as web application, e-commerce information technology, tracking system, and inventory management, as well as the standards that apply to each procedure. Because it is straightforward to understand and implement, the waterfall model was chosen as the development process. It's also been used to foster desirable practices like defining before designing and designing before coding in a number of past projects.

**CHAPTER 3:  ANALYSIS**

## 3.1     Introduction

This chapter outlines how the problems observed in the present system were analyzed, as well as the needs of the proposed system in 3.2. In 3.3.1, 3.3.2, and 3.3.3, the requirement analysis elucidates data needs, functional requirements, and non-functional requirements, respectively. Other project criteria are also discussed, and the chapter concludes with a conclusion.

## 3.2     Problem Analysis

As we can see in the diagram below the process for traditional method is quite long and time consuming. The efficiency of the current system is less which would cause delay to the people involved and prone to human errors. Accessibility is also limited because of existing shortcomings.

Figure 3.0 below illustrates the main process involved in a daily process that will be done by customers. The first begins with customer will have to contact seller and ask for desired items that they wish to buy. Then, the seller will do a stock check for the requested items by the customer. If the requested items are available, then it will be sold to the customer. Otherwise, the seller would have to inform customer that they are out of stock and if the customer willingly to wait for seller to ask from their distributor, then if stock is available from distributor side, the distributor itself will arrange stock delivery to seller. Only then seller could sell the items to customer if stock already arrived at their side. From the flow mentioned above, the process itself already took too much time for customer, seller, and distributor. All of the people involved in this process would not benefit the current system as it would only bring difficult. Sorting process would also be a problem as they are not systematically sorted which could leads to human errors.
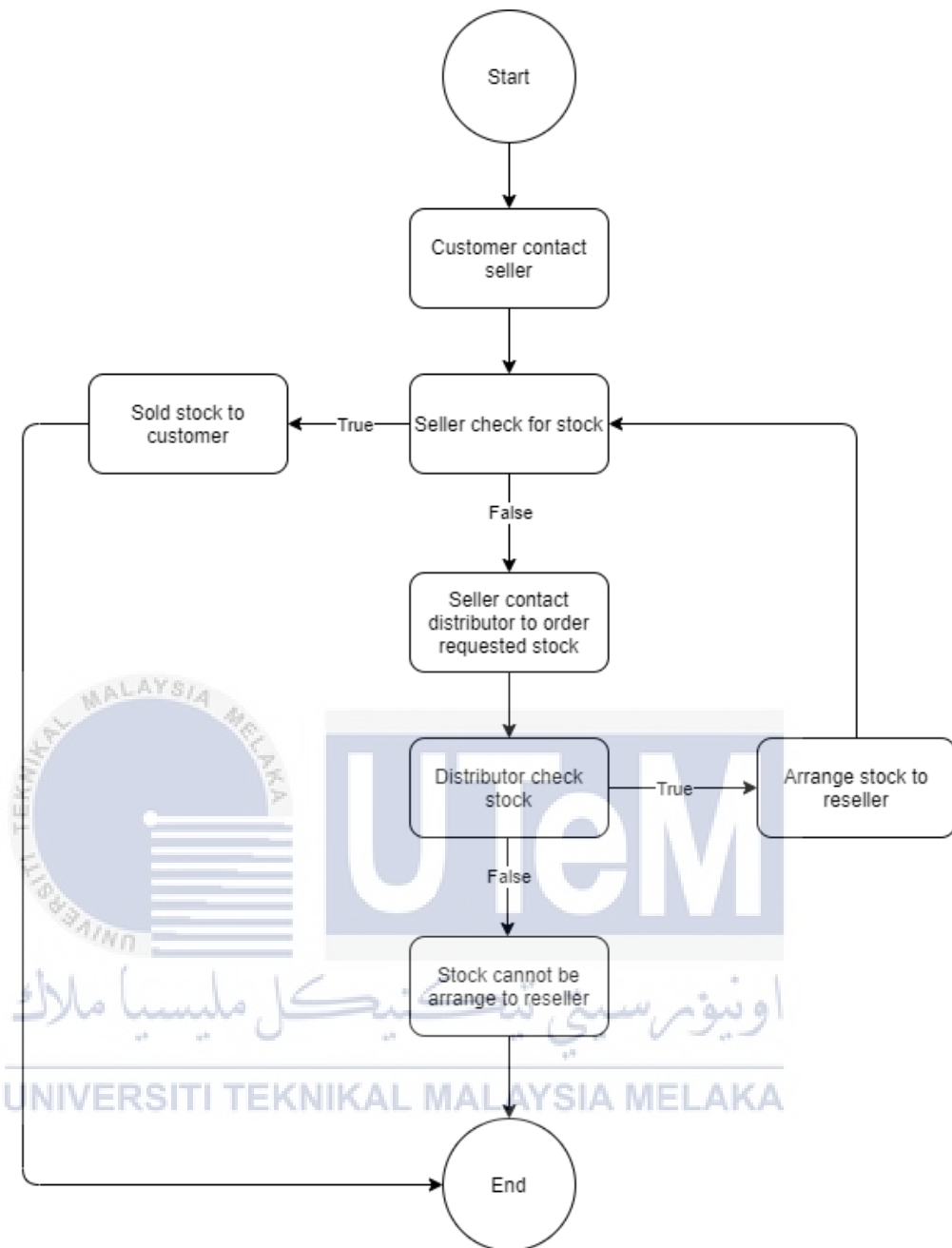
Figure 3.0: Activity Diagram of Current System

## 3.3     Requirement Analysis

This section delves deeper into the analysis of e-Techify's data requirements, functional requirements, and non-functional requirements.

## 3.3.1     Data Requirement

A data dictionary is a database table that records all of the data and information about the table. In every entity, it specifies the field name, type, size, and description of each column. It also includes each entity's primary and foreign keys, as well as color-coded indicators.

a) Addresses

Table 3.1: Addresses Entity

| Field | Type | Size | Primary Key | Foreign Key | Description |
|---|---|---|---|---|---|
| ID | BIGINT | 20 | ✓ | | A unique number ID for addresses. |
| User_id | BIGINT | 20 | | ✓ | FK from user's table. |
| Name | VARCHAR | | | | Name for address. |
| Phone_no | INT | | | | Phone number for address. |
| Address | VARCHAR | | | | Full address. |
| Postcode | INT | | | | Postcode for address |
| Longitude | DOUBLE | | | | Longitude for address. |
| Latitude | DOUBLE | | | | Latitude for address. |
| Default_status | INT | | | | Default status of an address |
| Created_at | TIMESTAMP | | | | When data is created. |
| Updated_at | TIMESTAMP | | | | When data is updated. |

b) Confirm Orders – (confirm_orders)

Table 3.2: Confirm Orders Entity

| Field | Type | Size | Primary Key | Foreign Key | Description |
|---|---|---|---|---|---|
| ID | BIGINT | 20 | ✓ | | A unique number ID for confirm orders. |
| Order_id | BIGINT | 20 | | ✓ | FK from orders table. |
| Addresses_id | BIGINT | 20 | | ✓ | FK from addresses table. |
| Payment_total | INT | | | | Total payment for each confirmed order. |
| Payment_method | VARCHAR | | | | Payment method for each confirmed order. |
| Tracking_num | VARCHAR | | | | Tracking number for each confirmed order. |
| Receipt_no | VARCHAR | | | | Receipt number for each confirmed order. |
| Created_at | TIMESTAMP | | | | When data is created. |
| Updated_at | TIMESTAMP | | | | When data is updated. |

c) Distributor Products – (distributor_products)

Table 3.3: Distributor Products Entity

| Field | Type | Size | Primary Key | Foreign Key | Description |
|---|---|---|---|---|---|
| ID | BIGINT | 20 | ✓ | | A unique number ID for distributor products. |
| Product_id | BIGINT | 20 | | ✓ | FK from products table. |
| User_id | BIGINT | 20 | | ✓ | FK from user's table. |
| Batch_no | INT | | | | Batch number for each distributor product. |
| Serial_number | VARCHAR | | | | Serial number for each distributor product. |
| Status | VARCHAR | | | | Status for each distributor product. |
| Created_at | TIMESTAMP | | | | When data is created. |
| Updated_at | TIMESTAMP | | | | When data is updated. |

d) Jobs

Table 3.4: Jobs Entity

| Field | Type | Size | Primary Key | Foreign Key | Description |
|---|---|---|---|---|---|
| ID | BIGINT | 20 | ✓ | | A unique number ID for jobs. |
| Product_id | BIGINT | 20 | | ✓ | FK from products table. |
| Occupied_by | BIGINT | 20 | | ✓ | FK from user's table. |
| Job_name | VARCHAR | | | | Job name. |
| Job_salary | INT | | | | Job salary. |
| Job_location | VARCHAR | | | | Job location. |
| Job_type | VARCHAR | | | | Job type for each job. |
| Status | VARCHAR | | | | Job status for each job. |
| Created_at | TIMESTAMP | | | | When data is created. |
| Updated_at | TIMESTAMP | | | | When data is updated. |

e) Orders

Table 3.5: Orders Entity

| Field | Type | Size | Primary Key | Foreign Key | Description |
|---|---|---|---|---|---|
| ID | BIGINT | 20 | ✓ | | A unique number ID for orders. |
| User_id | BIGINT | 20 | | ✓ | FK from user's table. |
| Order_status | VARCHAR | | | | Order status for each order. |
| Created_at | TIMESTAMP | | | | When data is created. |
| Updated_at | TIMESTAMP | | | | When data is updated. |
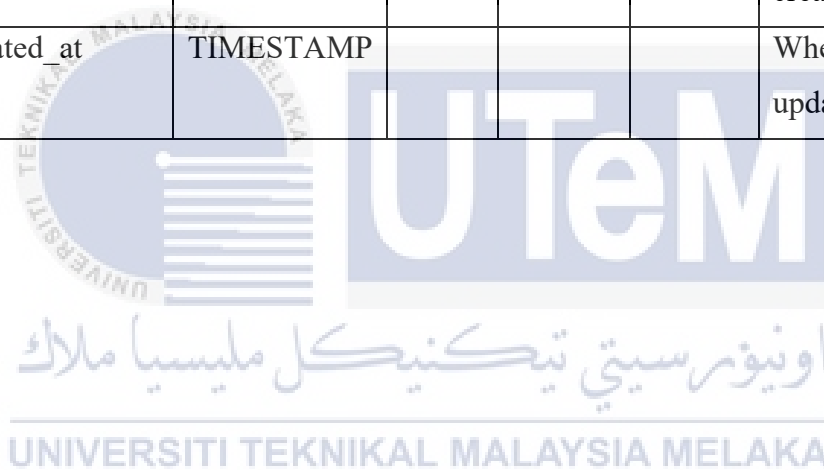
f) Order Details – (order_details)

Table 3.6: Order Details Entity

| Field | Type | Size | Primary Key | Foreign Key | Description |
|---|---|---|---|---|---|
| ID | BIGINT | 20 | ✓ | | A unique number ID for order details. |
| Order_id | BIGINT | 20 | | ✓ | FK from orders table. |
| Product_id | BIGINT | 20 | | ✓ | FK from products table. |
| Product_order _quantity | INT | | | | Ordered products quantity. |
| Serial_number | VARCHAR | | | | Serial number for each product. |
| Created_at | TIMESTAMP | | | | When data is created. |
| Updated_at | TIMESTAMP | | | | When data is updated. |

g) Products

Table 3.7: Products Entity

| Field | Type | Size | Primary Key | Foreign Key | Description |
|---|---|---|---|---|---|
| ID | BIGINT | 20 | ✓ | | A unique number ID for products. |
| User_id | BIGINT | 20 | | ✓ | FK from user's table. |
| Product_name | VARCHAR | | | | Product name. |
| Product_sn | VARCHAR | | | | Product special number. |
| Product_image_path | VARCHAR | | | | Product image path. |
| Product_category | VARCHAR | | | | Product category. |
| Product_brand | VARCHAR | | | | Product brand. |
| Product_warranty _duration | INT | | | | Product warranty duration. |
| Product_price | DECIMAL | | | | Product price. |
| Product_link | VARCHAR | | | | Product link for each product. |
| Product_stock _count | INT | | | | Product stock counter for each product. |
| Created_at | TIMESTAMP | | | | When data is created. |
| Updated_at | TIMESTAMP | | | | When data is updated. |

h) Repairs

Table 3.8: Repairs Entity

| Field | Type | Size | Primary Key | Foreign Key | Description |
|---|---|---|---|---|---|
| ID | BIGINT | 20 | ✓ | | A unique number ID for repairs. |
| Addresses_id | BIGINT | 20 | | ✓ | FK from addresses table. |
| Product_id | BIGINT | 20 | | ✓ | FK from products table. |
| User_id | BIGINT | 20 | | ✓ | FK from user's table. |
| Sn_no | VARCHAR | | | | Product serial number. |
| Date_of_purchase | DATE | | | | Date of purchased products. |
| File_path | VARCHAR | | | | Proof of purchase. |
| Reason | VARCHAR | | | | Reason to send for repair. |
| Status | VARCHAR | | | | Repair status. |
| Tracking_no | VARCHAR | | | | Tracking number. |
| Resolve_solution | VARCHAR | | | | Repair resolution. |
| Receive_at | DATE | | | | Data of received. |
| Created_at | TIMESTAMP | | | | When data is created. |
| Updated_at | TIMESTAMP | | | | When data is updated. |

i) Roles

Table 3.9: Roles Entity

| Field | Type | Size | Primary Key | Foreign Key | Description |
|---|---|---|---|---|---|
| ID | BIGINT | 20 | ✓ | | A unique number ID for roles. |
| Name | VARCHAR | | | | Name of each role. |
| Created_at | TIMESTAMP | | | | When data is created. |
| Updated_at | TIMESTAMP | | | | When data is updated. |

j) Role User – (role_user)

Table 3.10: Role User Entity

| Field | Type | Size | Primary Key | Foreign Key | Description |
|---|---|---|---|---|---|
| ID | BIGINT | 20 | ✓ | | A unique number ID for role user. |
| Role_id | BIGINT | 20 | | ✓ | FK from roles table. |
| User_id | BIGINT | 20 | | ✓ | FK from user's table. |
| Created_at | TIMESTAMP | | | | When data is created. |
| Updated_at | TIMESTAMP | | | | When data is updated. |

k) Shipments

Table 3.11: Shipments Entity

| Field | Type | Size | Primary Key | Foreign Key | Description |
|---|---|---|---|---|---|
| ID | BIGINT | 20 | ✓ | | A unique number ID for shipments. |
| User_id | BIGINT | 20 | | ✓ | FK from user's table. |
| Tracking_no | VARCHAR | | | | Shipment tracking number. |
| Remark | VARCHAR | | | | Remark for each shipment. |
| Receive_at | DATE | | | | Receive date. |
| Status | VARCHAR | | | | Status for each shipment. |
| Proof_of_payment | VARCHAR | | | | Proof of payment. |
| Created_at | TIMESTAMP | | | | When data is created. |
| Updated_at | TIMESTAMP | | | | When data is updated. |

l) Shipment Details – (shipment_details)

Table 3.12: Shipment Details Entity

| Field | Type | Size | Primary Key | Foreign Key | Description |
|-------|------|------|-------------|-------------|-------------|
| ID | BIGINT | 20 | ✓ | | A unique number ID for shipment details. |
| Shipment_id | BIGINT | 20 | | ✓ | FK from shipments table. |
| Product_id | BIGINT | 20 | | ✓ | FK from products table. |
| Product_order _quantity | INT | | | | Ordered products quantity. |
| Created_at | TIMESTAMP | | | | When data is created. |
| Updated_at | TIMESTAMP | | | | When data is updated. |

m) Trackings

Table 3.13: Tracking's Entity

| Field | Type | Size | Primary Key | Foreign Key | Description |
|---|---|---|---|---|---|
| ID | BIGINT | 20 | ✓ | | A unique number ID for tracking's. |
| Order_id | BIGINT | 20 | | ✓ | FK from orders table. |
| Tracking_no | VARCHAR | | | | Tracking number for each order |
| Current_status | VARCHAR | | | | Current status for each order. |
| Created_at | TIMESTAMP | | | | When data is created. |
| Updated_at | TIMESTAMP | | | | When data is updated. |

n) Users

Table 3.14: Users Entity

| Field | Type | Size | Primary Key | Foreign Key | Description |
|---|---|---|---|---|---|
| ID | BIGINT | 20 | ✓ | | A unique number ID for users. |
| Name | VARCHAR | | | | User name. |
| Email | VARCHAR | | | | User email. |
| Email_verified_at | TIMESTAMP | | | | User email verified. |
| Password | VARCHAR | | | | User password. |
| Two_factor_secret | TEXT | | | | 2FA secret. |
| Two_factor _recovery_codes | TEXT | | | | 2FA secret recovery code. |
| Remember_token | VARCHAR | | | | Account remember token. |
| Created_at | TIMESTAMP | | | | When data is created. |
| Updated_at | TIMESTAMP | | | | When data is updated. |

## 3.3.2    Functional Requirement

This section outlines the system's functions, how it records, and the services and functionalities that e-Techify shall deliver to its users. Tables 3.15, 3.16, and 3.17 detail all of the system's functional needs.

Table 3.15: Functional Requirements of e-Techify for reseller

| Support Basic Account Authentication | The user must log in with an existing account and be able to log out at any time. |
|---|---|
| Manage Orders | The system will allow user that have reseller role to manage all customer orders. Information about an order is detailed out which consist of ordered items, when the order is created, tracking number that is assigned, name, phone number, receiver address and current tracking status. Other functionality such as e-receipt, insert serial number, print air waybill, update current parcel status, and cancel order are also provided. |
| Stock Management | The system will allow user to see list of distributors that is exist in the system. In addition, further information is displayed such as stock count and other relevant information for each product. |
| Request Shipment | The system will allow user to request desired items to the distributor of the products. Status for each request will also be displayed which include remark, a receipt and in detailed view of each request information. |
| Receipt Finder | The system will allow user to search and retrieve customer receipt for each order. |

| | |
|---|---|
| Manage Product Serial Number | The system will allow user to insert each product serial number with batch number. Statistic for each batch is recorded and data will be displayed in a bar chart grouped by batch number. A table filled with full information regarding a serial number status will be shown. |
| Manage Job | The system will allow user to insert a job which the inserted jobs will be offered to user that holds customer role. The purpose of this part to let customer generate some side income. |
| Manage Users | The system will allow user to manage all users in the system. Create, update, and delete operation are included in this functionality. |
| Reseller Dashboard | The system provides user in a compact view which offers variety of important functions such as manage all RMA requests, display earnings by monthly and annual, summary of earnings in Q1 till Q4 in bar chart and manage all job offerings. |
| Manage Address | The system will allow user to add new address without any limits and set an address to default address. |

Table 3.16: Functional Requirements of e-Techify for customer

| Support Basic Account Authentication | The user must log in with an existing account and be able to log out at any time. |
| --- | --- |
| Manage Orders | The system will allow user that have customer role to manage their orders. Information about an order is detailed out which consist of ordered items, when the order is created, tracking number that is assigned, name, phone number, receiver address and current tracking status. Other functionality such as e-receipt, check parcel tracking status, and cancel order are provided. |
| Shop | The system will allow user to shop their desired items throughout the list. |
| e-RMA | The system will allow user to submit RMA request for their purchased product if there is a problem they encounter during usage. |
| e-Track Parcel | The system will allow user to track their parcel status in detail. |
| Manage Cart | The system will allow user to add or delete items in cart, alter quantity, change receiver address according to the address that have been saved and make payment by using online payment method. |

| Customer Dashboard | The system provides user in a compact view which offers variety of important functions such as manage all RMA requests, total spending, total orders created, and display all job offerings. |
|---|---|
| Manage Address | The system will allow user to add new address without any limits and set an address to default address. |
| Edit Profile | The system will allow user to alter their account information in terms of name and email address. |

Table 3.17: Functional Requirements of e-Techify for distributor

| Support Basic Account Authentication | The user must log in with an existing account and be able to log out at any time. |
|---|---|
| Manage Products | The system will allow user that have distributor role to manage their products in the system. They will have the ability to insert new product, update and delete existing product. |
| Manage Job | The system will allow user to insert a job which the inserted jobs will be offered to user that holds customer role. The purpose of this part to let customer generate some side income. |

| Manage Shipments | The system will allow user to manage the shipments requested by reseller. All of the requests will be grouped by status. Each request needs to have approval from distributor itself. |
| --- | --- |
| Distributor Dashboard | The system provides user in a compact view which offers variety of important functions such as manage all RMA requests, total RMA, total products, display sale for each brand in bar chart while percentage sale for each brand in pie chart and manage all job offerings. |
| Manage Address | The system will allow user to add new address without any limits and set an address to default address for business purpose. |
| Edit Profile | The system will allow user to alter their account information in terms of name and email address. |

### 3.3.2.1 Data Flow Diagram

With Data Flow Diagrams, this section describes all modules of the e-Techify System (DFD). It includes a context diagram and (Level 0). Each diagram will be briefly explained.
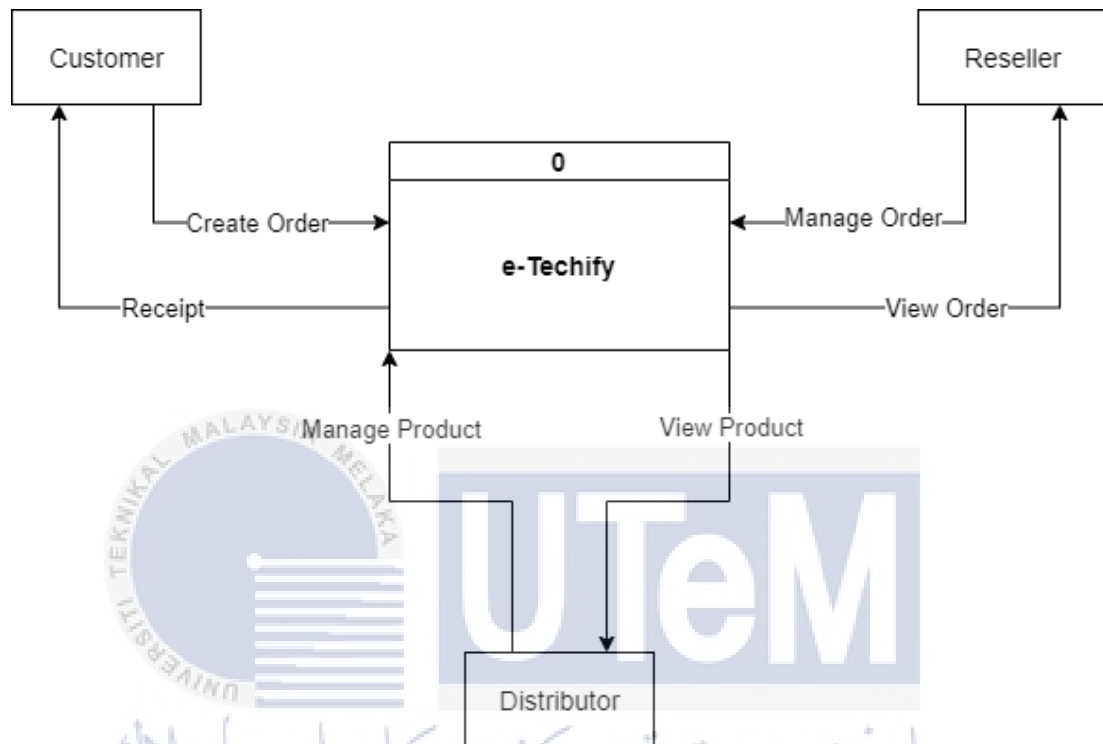


Figure 3.1: Context Diagram
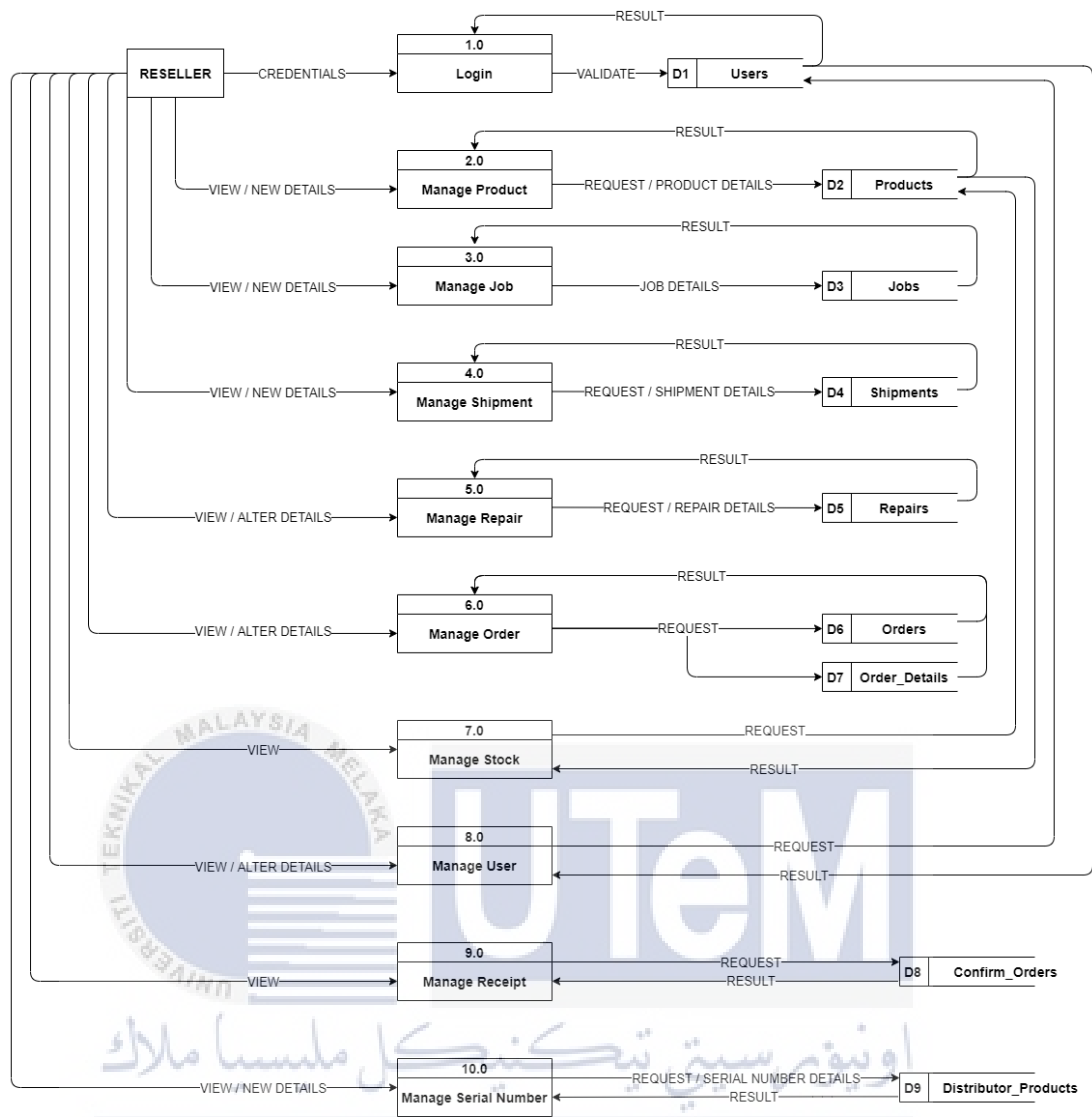
Figure 3.2: Level-0 for Reseller
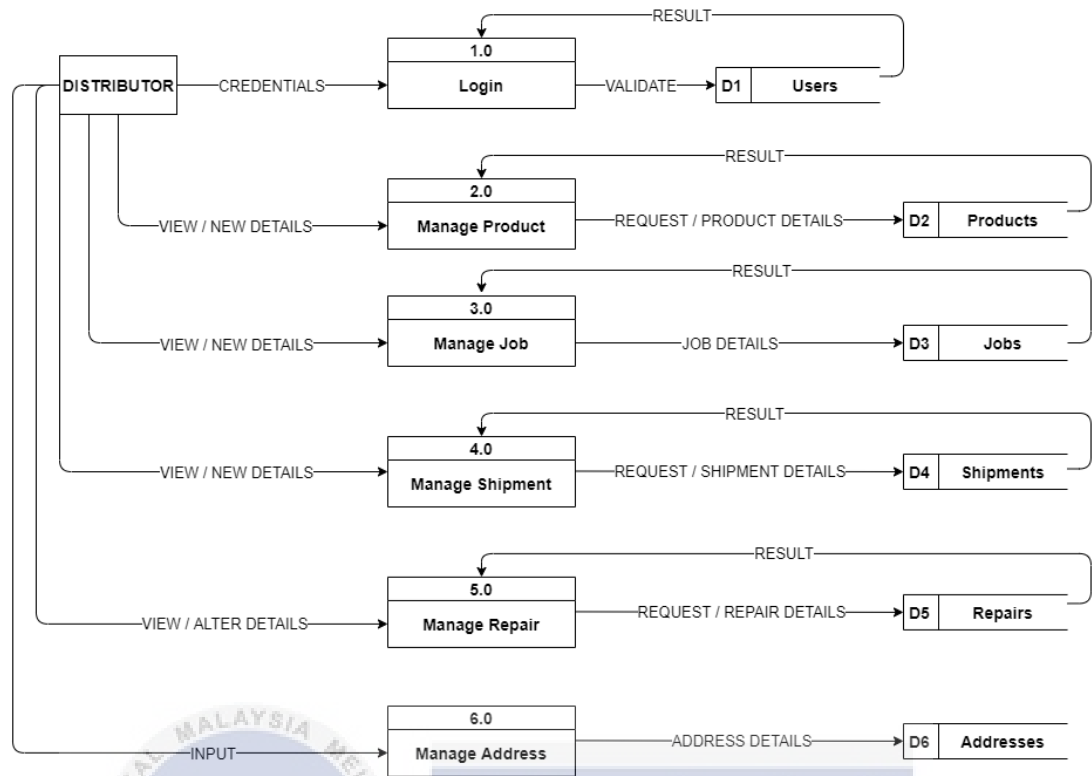
Figure 3.3: Level-0 for Distributor
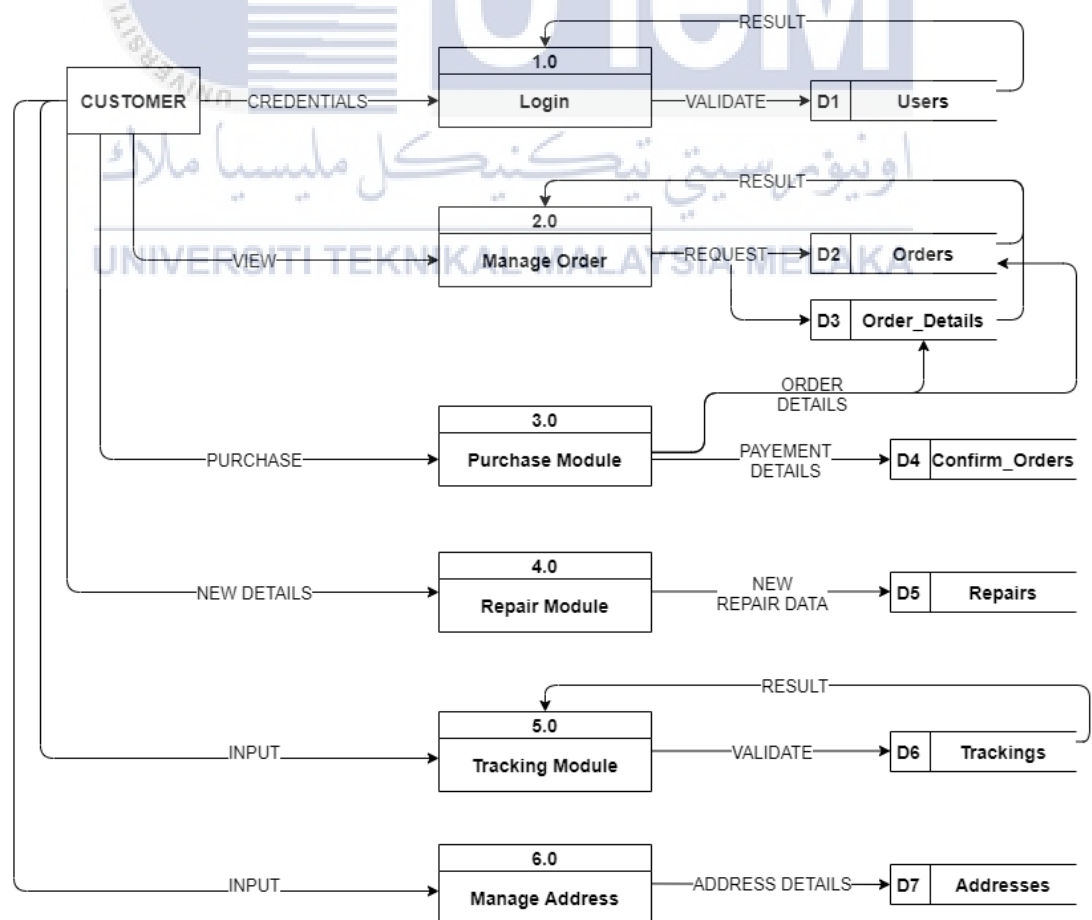


Figure 3.4: Level-0 for Customer

### 3.3.2.2 Flowchart
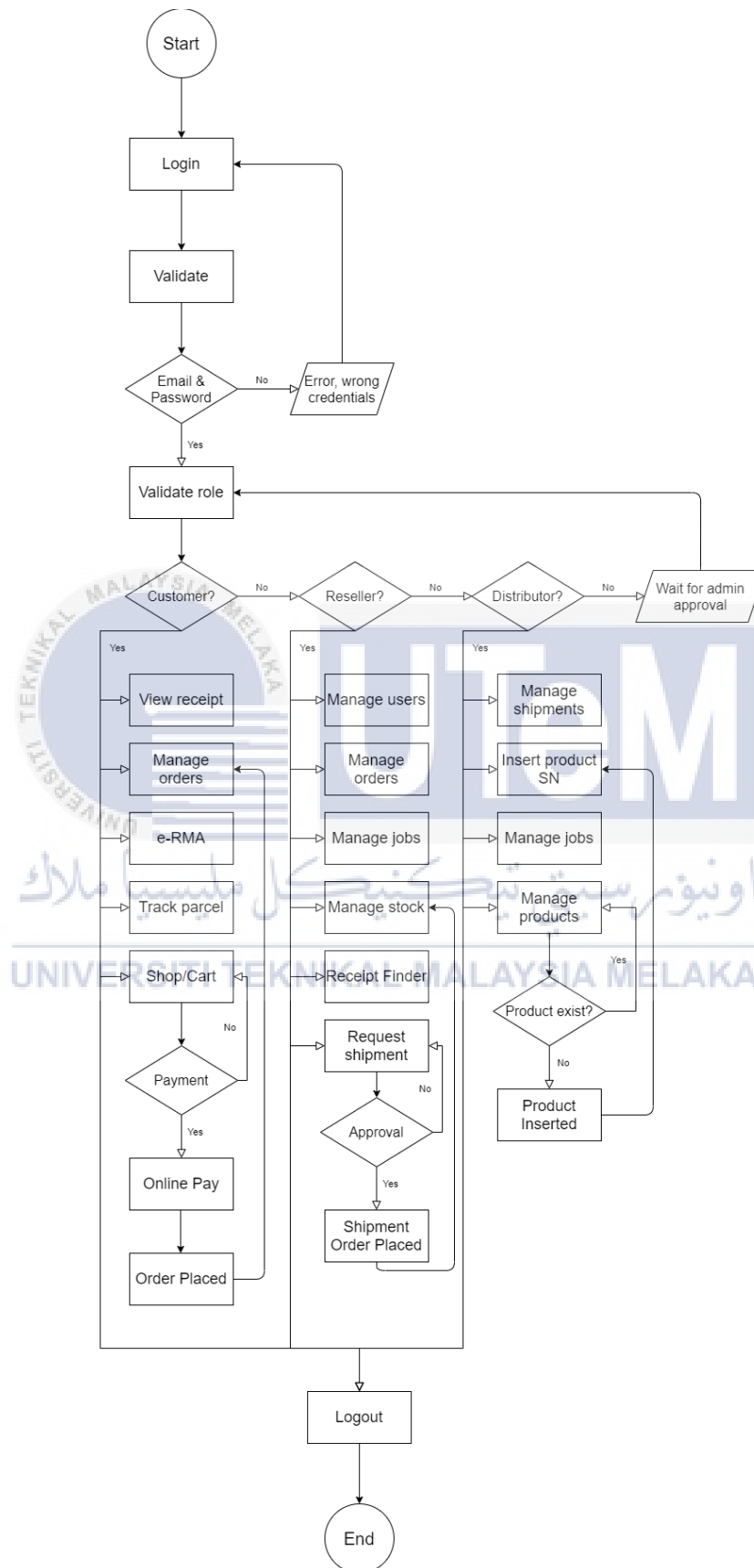


Figure 3.5: Full System Flowchart

### 3.3.3 Non-functional Requirement

Table 3.18: Non-Functional Requirements of e-Techify

| | |
|---|---|
| Extensibility | e-Techify ensures that the system platform is flexible and open to future feasible development. |
| Reliability / Availability | The failure rate of e-Techify must be low. Except for regular maintenance, the system should be available 24 hours a day, seven days a week. In order for the user to have access to the system server, the system also requires an internet connection. |
| Response Time | Each page request should take fewer than 1500 milliseconds to complete. There should be no hiccups in any of the requests. |
| Maintainability | e-Techify must have average time and rapidity in system recovery to make sure user experience during their visit are not interrupted for such time. |
| Usability | e-Techify provides simple and interactive user interface for user ease of access towards the functionality that is offered in the system. |

### 3.3.4   Others Requirement

The system should adhere to the requirements that have been provided by Xmiryna Technology. Each requirements provided are important in order to make the system as a whole. Every real-life scenario will be converted to functions in order to implement in the system.

### 3.4   Conclusion

This chapter discusses how the analytic tools were used to gather all of the requirements. Functional and non-functional requirements, as well as data requirements, have all been thoroughly explored. All of the studied requirements will then be passed on to the design stage, when all of the essential designs, such as database, software, and hardware designs, will be built, as detailed in the following chapter.

**CHAPTER 4: DESIGN**

**4.1     Introduction**

High-level design, which comprises system architecture and user interface designs, is covered in the first section of this chapter. The user interface design area in 4.2.2 covers navigation design, input design, and output design. The high-level design then defines database architecture in terms of conceptual and logical designs, which will be used throughout 4.2.3's development. This chapter's next section covers comprehensive design, which includes both software and physical database design. Each system function's specifics are described in the software design.

**4.2     High-Level Design**

This section delves into the intricacies of high-level design, such as system architecture, user interface design, and database design.

**4.2.1     System Architecture**
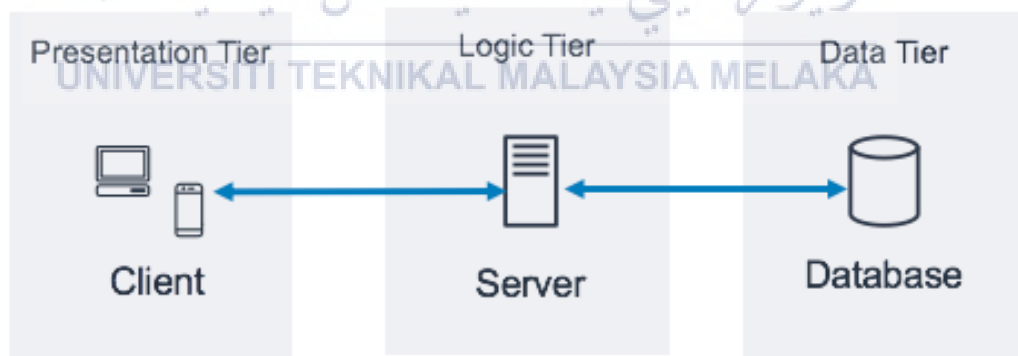


Figure 4.0: Three-tier System Architecture of e-Techify

Figure 4.0 shows the flow of a three-tier system architecture. e-Techify was built by using this architecture. It is a client-server architecture that includes capabilities for presentation, application processing, and data administration. Each function is segregated into three layers and physically segregated.

Three-tier web application development is commonly used in regular websites. The presentation layer is the application's highest level. The user interface is presented in this tier as a means of communication with the user. Its primary responsibility is to analyze user requests and transform the results into a format that the user can comprehend. This tier, for example, is used by the user to acquire the results report for each category of a brand that is sold. This system's presentation tier is the user's web browser, which is used to access it. The front-end platform for displaying the material sent from the server is the browser.

Business logic tier, application layer, and middle layer are all terms used to describe the logic tier. As seen in Figure 4.0, this tier sits between the display and data tiers. The functioning of an application is controlled by this layer, which processes instructions, does calculations, and makes logical decisions and judgments. Between the data tier and the presentation layer, it is also in charge of data access and processing. For example, this layer can read orders data from the data layer and perform computations to determine each customer subtotal price in a specific order. The application server XAMP, which handles content processing on the PHP platform, is the system's intermediate layer.

Data persistence strategies are found in the data tier, which is the lowest layer. This is the location where data is stored and extracted from a database or file system. This layer transfers data from the database to the logic tier for detailed processing, and subsequently shows the user the results. For example, this layer holds all of e-Techify data that has been entered into the system. MySQL is the system's backend database, which comprises data sets as well as database administration tools for managing and accessing the data.

## 4.2.2   User Interface Design

This section covers all the user interface designs of this e-Techify, including navigation design, input design and output design. Each user interface design will be accompanied by its screenshot picture.

### 4.2.2.1 Navigation Design

A navigation diagram, often known as a site map, is a low-level architecture diagram work product that explains how to move across an application's display components. The navigation diagram of e-Techify is shown in the picture below:
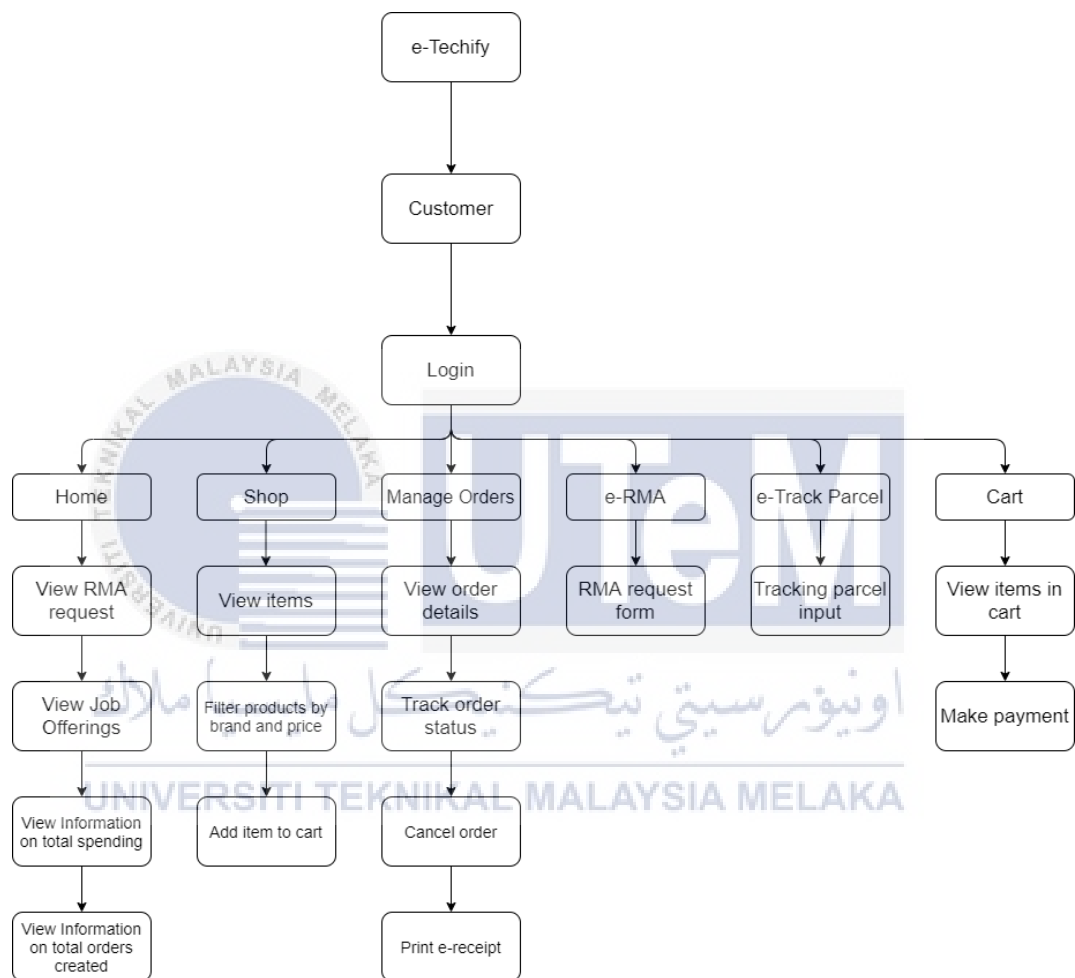


Figure 4.1: Navigation Design for customer
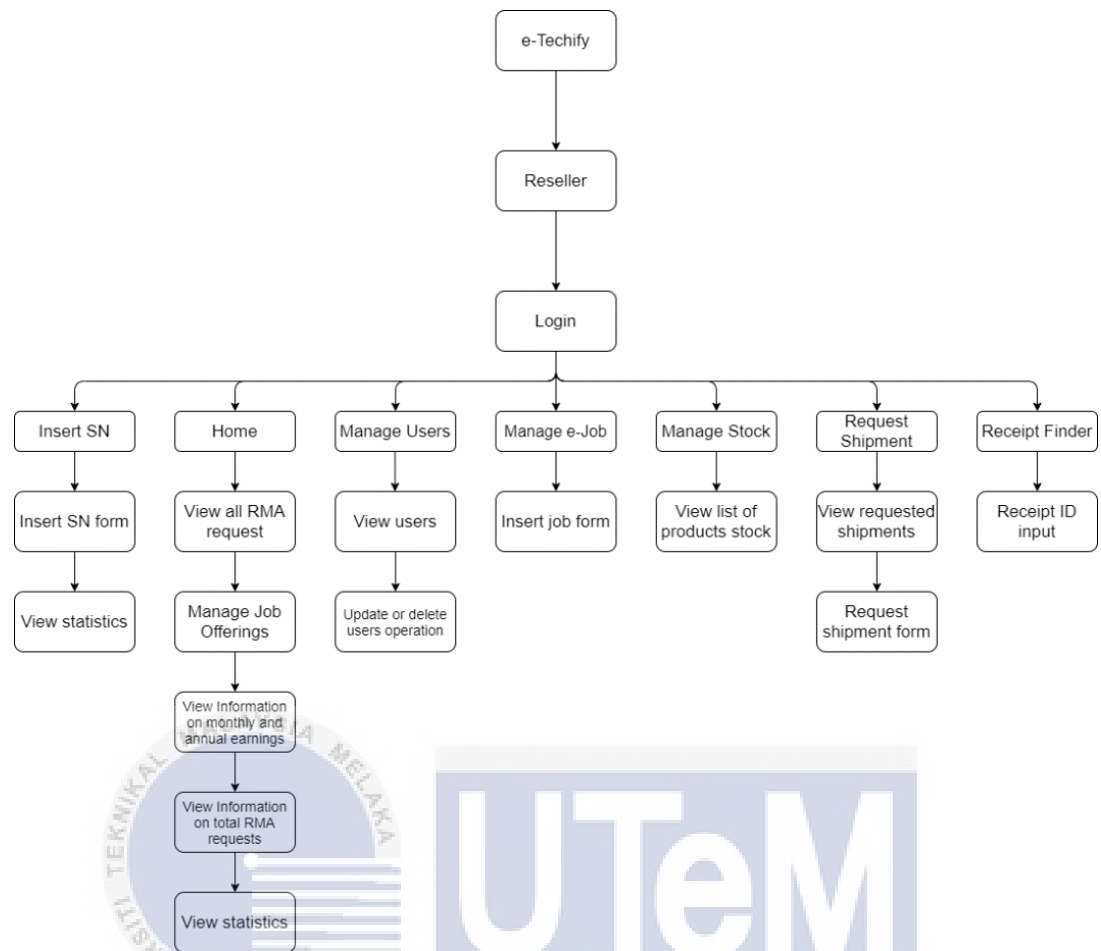
Figure 4.2: Navigation Design for reseller
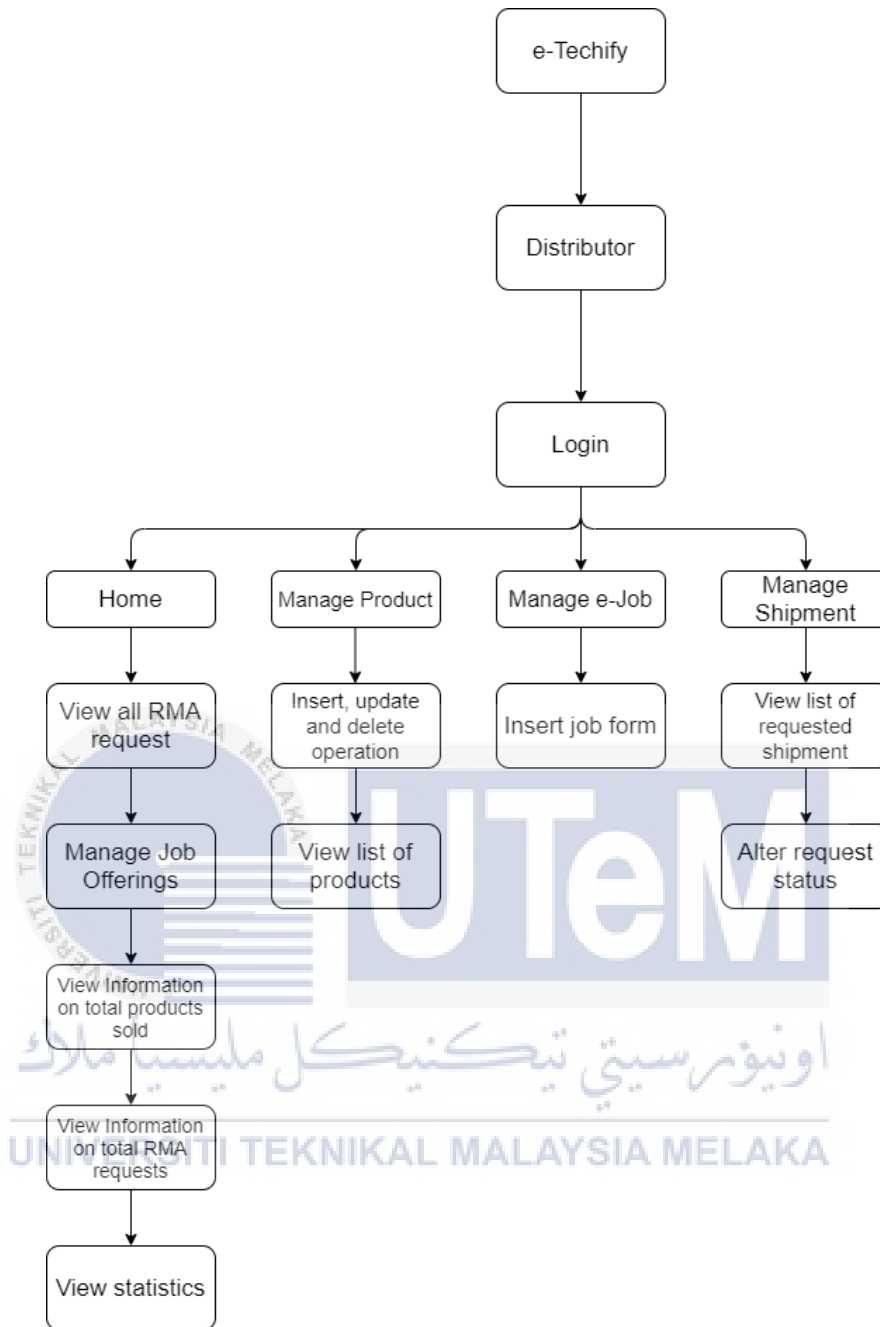
Figure 4.3: Navigation Design for distributor

### 4.2.2.2 Input Design

This section contains all the interface designs of the screen that enable user to input data into this system.



Figure 4.4: Login

Figure 4.4 shows the input screen for users of e-Techify to log into the system. The input for login will be email address and password.



Figure 4.5: Register

Figure 4.5 shows the input screen for registration of e-Techify to create an account in the system. The input for register will be name, email address, password and confirm password.

Figure 4.6: Input Job

Figure 4.6 shows the input screen for a job form. The input for job form is name, salary, type, and location. All of the job inserted will be display in job offerings section at user dashboard. The users that have ability for this functionality reseller and distributor.



Figure 4.7: Input Product

Figure 4.7 shows the input screen for product insertion. The input for insert product will be name, number, price, link, image, warranty duration, category, brand, and total stock. The users that have ability for this functionality are reseller and distributor.

Figure 4.8: Input Product Serial Number

Figure 4.8 shows the input screen for serial number insertion for a product. The input for this part will be product serial number and user have to choose which product from dropdown that belongs to the new serial number. The users that have ability for this functionality are reseller and distributor.

Figure 4.9: Input Serial Number

Figure 4.9 shows the input screen for serial number insertion for each customer orders. The input for this part will be product serial number suggested by the system according to the product ordered. The users that have ability for this functionality are reseller.

Figure 4.10: Request Shipment

Figure 4.10 shows the input screen for request shipment functionality. The input for this part will be brand distributor, brand, product, and product quantity. Once the user has selected the dropdown and add to list button is execute then it will display inserted product at the cart section on the left. On bottom of the screen, requested shipment based on status will be shown. The users that have ability for this functionality are reseller and distributor but the distributor itself can only see the status section.

Figure 4.11: Input Tracking Status

Figure 4.11 shows the input screen for tracking status. The input for this part will be current tracking status. The users that have ability for this functionality are reseller.
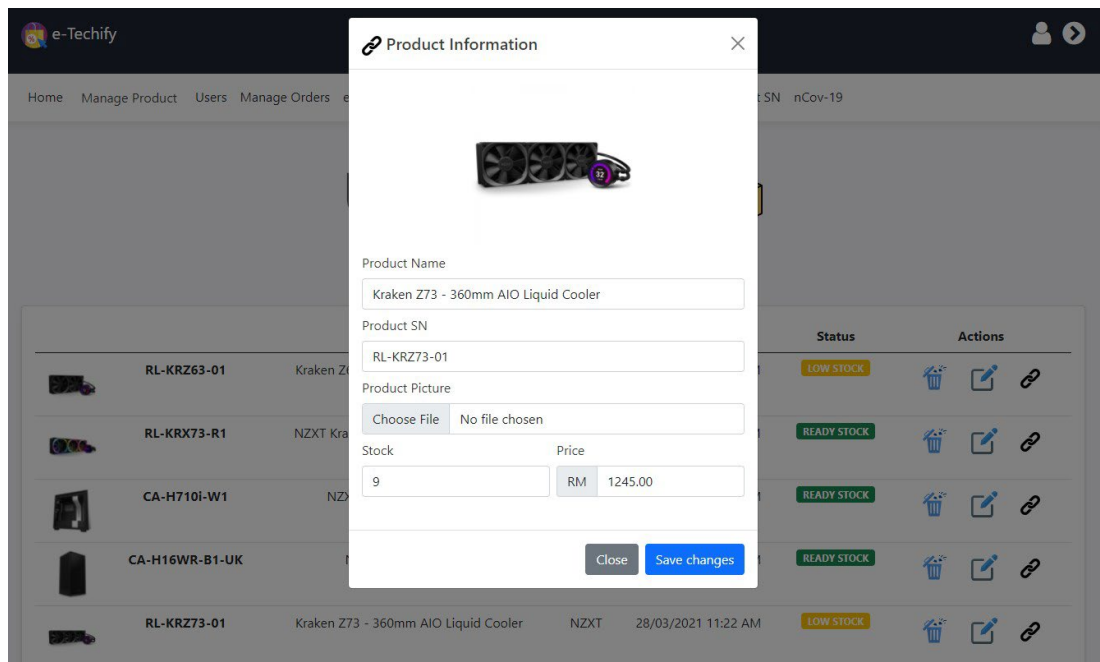
Figure 4.12: Update Product Info

Figure 4.12 shows the input screen for update product info. The input for this part will be product name, number, picture, total stock, and price. The users that have ability for this functionality are reseller and distributor.
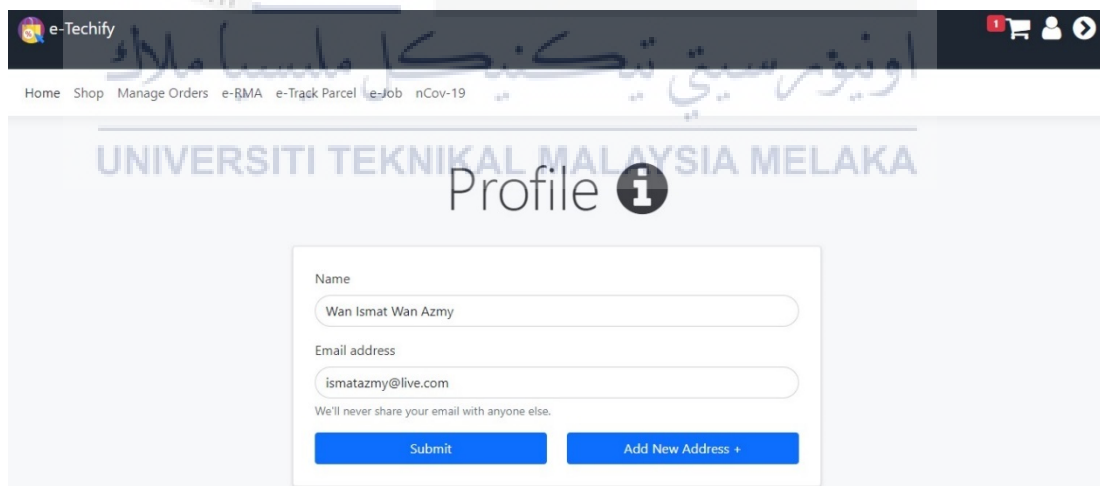


Figure 4.13: Update Profile

Figure 4.13 shows the input for update profile info. The input for this part will be user name and email address. The users that have ability for this functionality are all e-Techify users.

Figure 4.14: Insert Address

Figure 4.14 shows the input screen for address insertion. The input for this part will be address name, full address, phone number. Postcode, longitude, and latitude will be determined by the system itself. The users that have ability for this functionality are all e-Techify users.

Figure 4.15: Update RMA Status

Figure 4.15 shows the input screen for update RMA status. The input for this part will be status, receive at date, tracking number and remark. The users that have ability for this functionality are distributor.

Figure 4.16: Cart

Figure 4.16 shows the input screen for cart functionality. The input for this part will be items in cart, quantity, and payment method by PayPal. The users that have ability for this functionality are customer.

Figure 4.17: Input RMA Request

Figure 4.17 shows the input screen for RMA request form. The input for this part will be address dropdown, date of purchase, proof of purchase, brand name, type of product, serial number, confirm serial number and comments. The users that have ability for this functionality are customer.



Figure 4.18: Input Tracking Number

Figure 4.18 shows the input screen for tracking functionality. The input for this part will be tracking number that is exist in the system.

## 4.2.2.3 Output Design
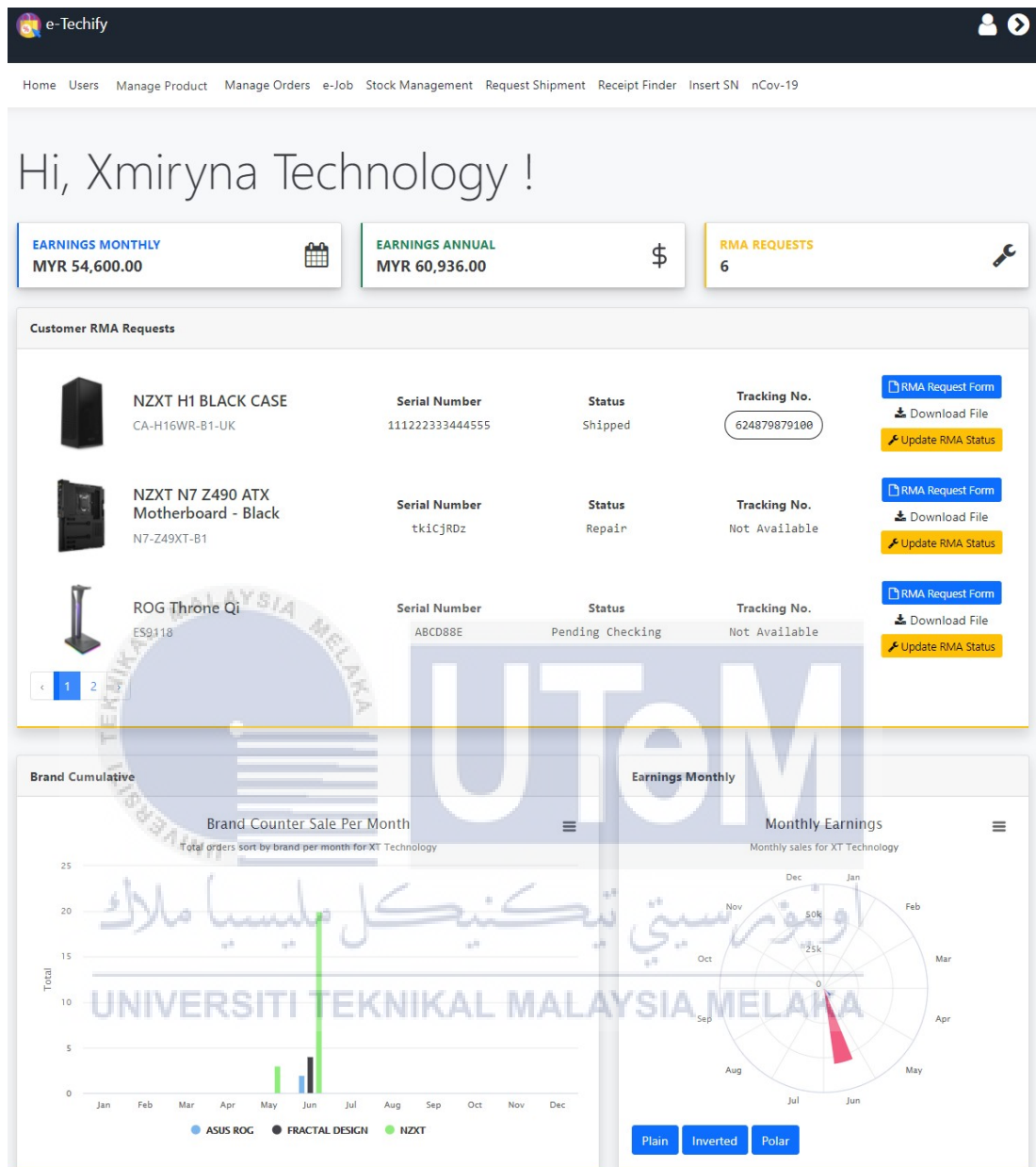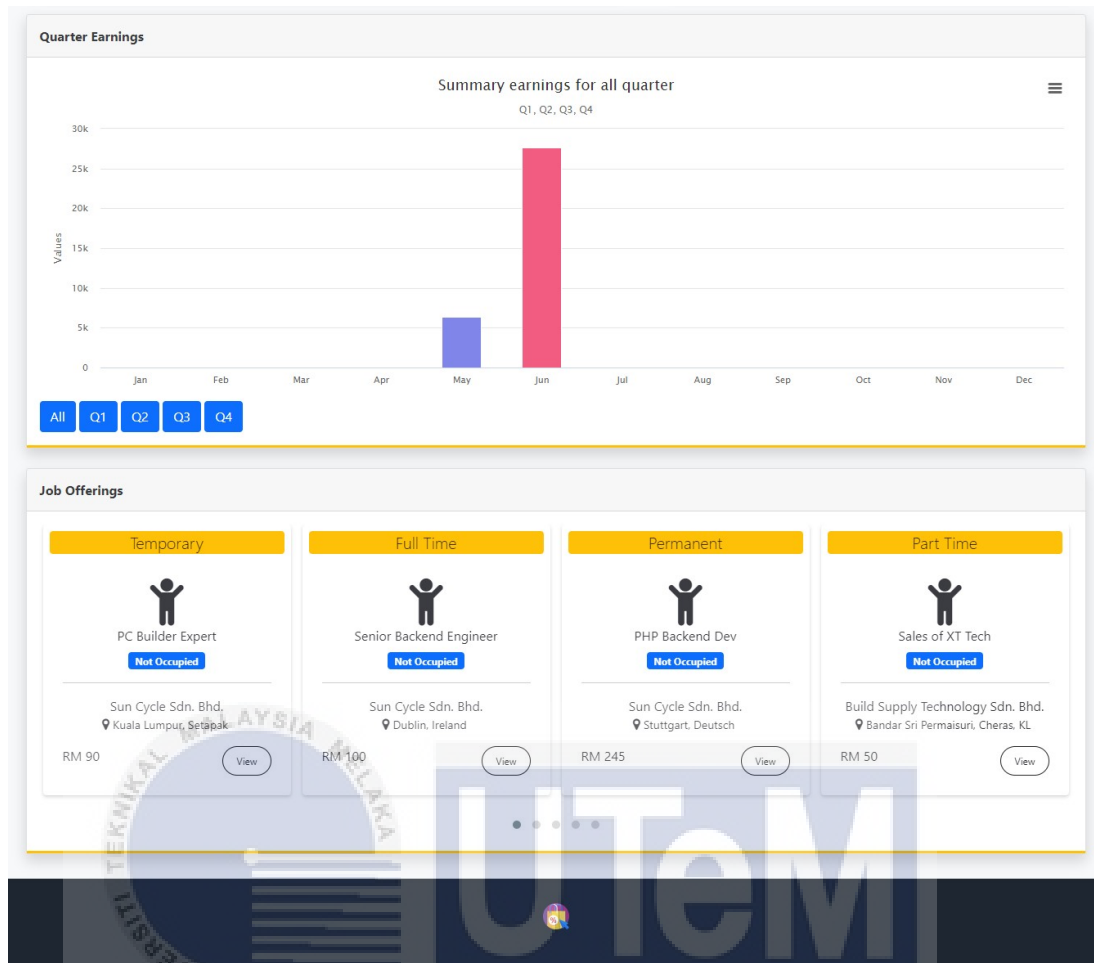
Figure 4.19: Home Dashboard for Reseller

Figure 4.19 shows the output screen for reseller dashboard, it consists of monthly earnings, monthly earnings, display total and all RMA request info. Brand cumulative, earnings monthly are shown in bar chart to ease user access towards data. Job offers are display in a carousel which have a draggable function to view all job offers offered by distributor or reseller itself.

Figure 4.20: Order Details

Figure 4.20 shows the output screen for order details, it consists of list of items that have been ordered by customer with quantity, name and phone number, address, tracking number, timestamp when order is created, invoice number, order current status, payment method and subtotal of the order.

Figure 4.21: Update or Delete Product

Figure 4.21 shows the output screen for update or delete product section. It consists of a table with product details that is exists in the system. Information such as product picture, name, brand, timestamp for last updated product info, status, and actions. The users that have ability for this functionality are reseller and distributor. Pagination is provided in order let user have the ease to view product in a good arrangement of 6 product per page.

Figure 4.22: Receipt Finder

Figure 4.22 shows the output screen for receipt finder. It consists of a simple table with list of receipt that is existed in the system. Each receipt will be created as soon as when customer successfully checkout and payment is approved. Search bar is also provided in order to search receipt by its id to easily find the desired receipt.

Figure 4.23: Stock Management

Figure 4.23 shows the output screen for stock management. This section consists of list of distributors that is registered in the system. All of the registered distributors will be list down in this section and reseller have the ability to preview what kind of product that they bring into the system. The users that have ability for this functionality are reseller.

Figure 4.24: Specific Stock Management

Figure 4.24 shows the output screen for specific stock management. This section is intended to preview specific stock by each distributor that have inserted the product in the system. Details such as product picture, name, brand, timestamp of last updated, stock and status for each product is listed in a table. Pagination is provided in order let user have the ease to view product in a good arrangement of 6 product per page.

Figure 4.25: Manage Users

Figure 4.25 shows the output screen for manage users. This section is intended to manage users by editing their credentials or delete their account. List of users with their id, name and email will be displayed in a table. The users that have ability for this functionality are reseller.



Figure 4.26: Requested Shipment
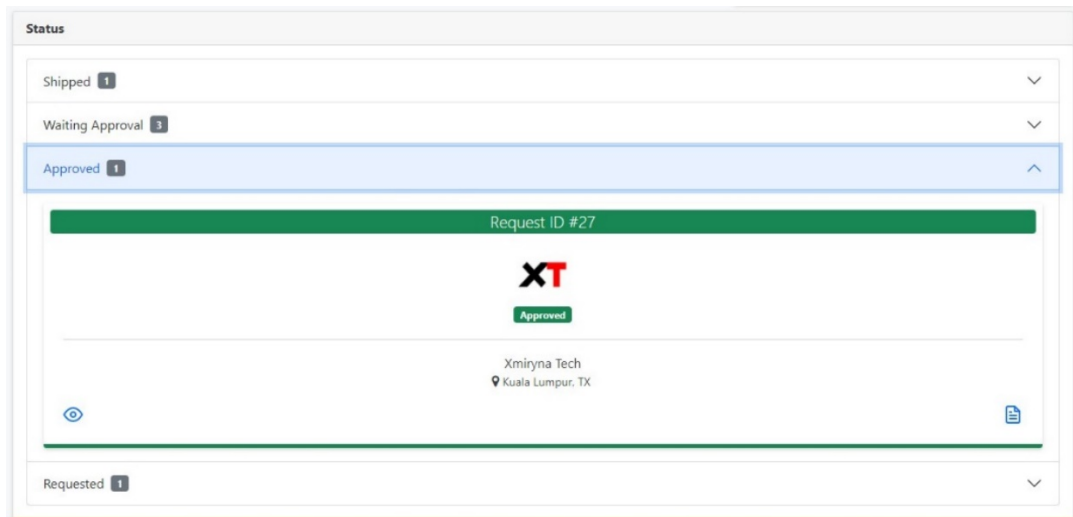
Figure 4.27: Approved Shipment
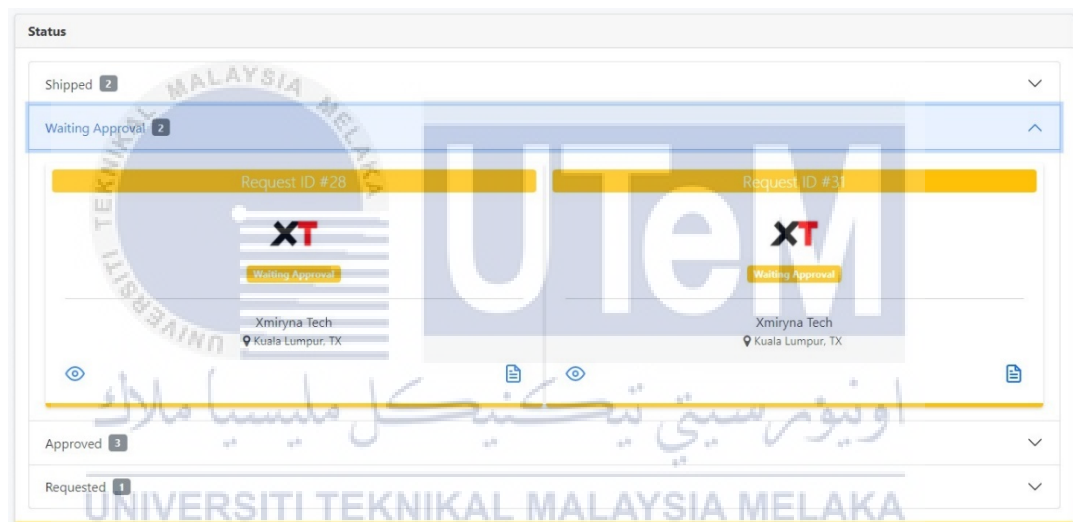


Figure 4.28: Waiting Approval Shipment
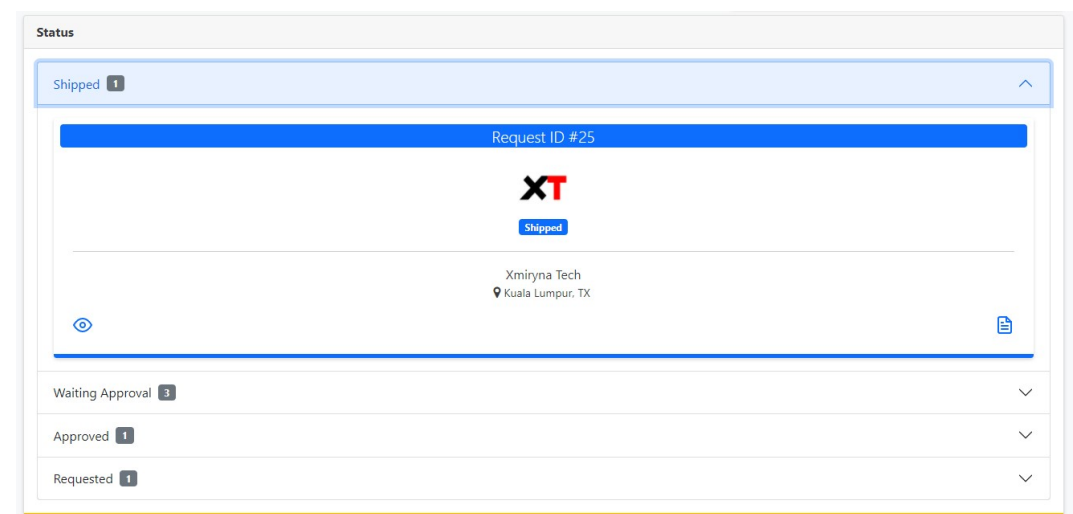


Figure 4.29: Shipped Shipment

Figure 4.26 shows the output screen for requested shipment. While figure 4.27 shows the output screen for approved shipment. Figure 4.28 shows the output screen for waiting approval shipment. Lastly, figure 4.29 shows the output screen for shipped shipment. All of the current shipment status will be displayed in according to their section or groups. It gives reseller the option to view shipment details and have a proof of shipment invoice.
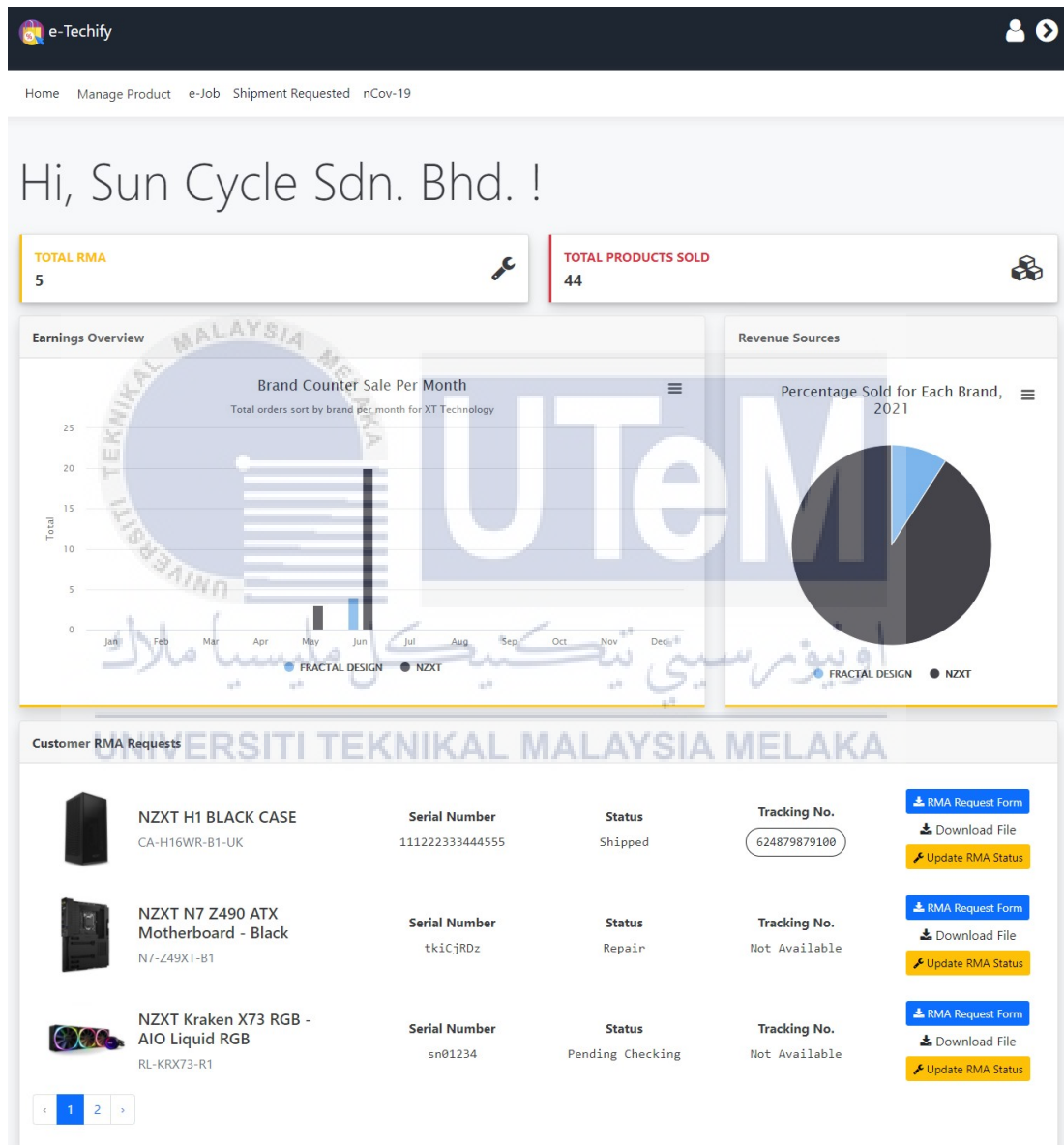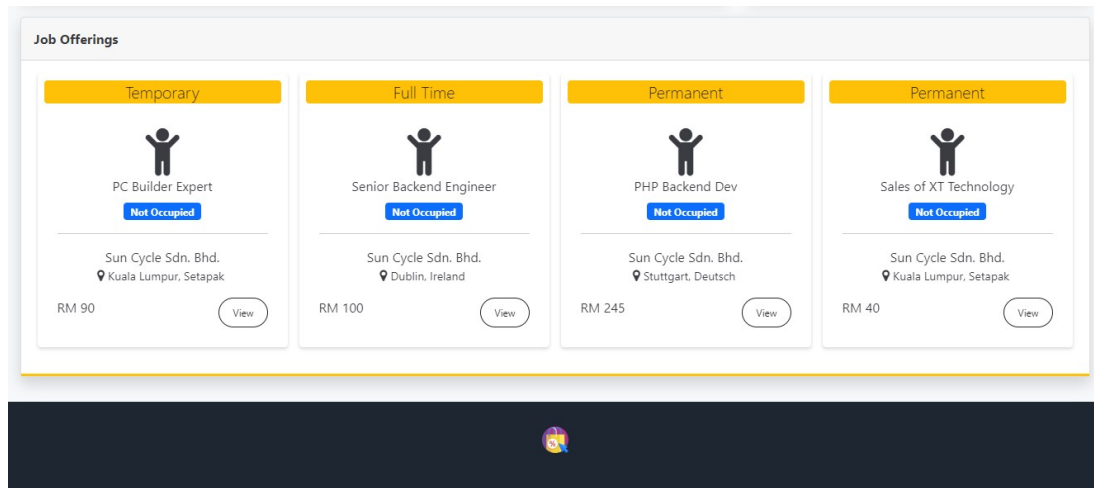
Figure 4.30: Home Dashboard for Distributor

Figure 4.30 shows the output screen for distributor dashboard, it consists of total RMA request created by customer, list of RMA and total products sold. Earnings overviews are shown in bar chart while revenue sources are shown in pie chart to ease user access towards the data collected by the system. Job offers are display in a carousel which have a draggable function to view all job offers offered by distributor or reseller itself.
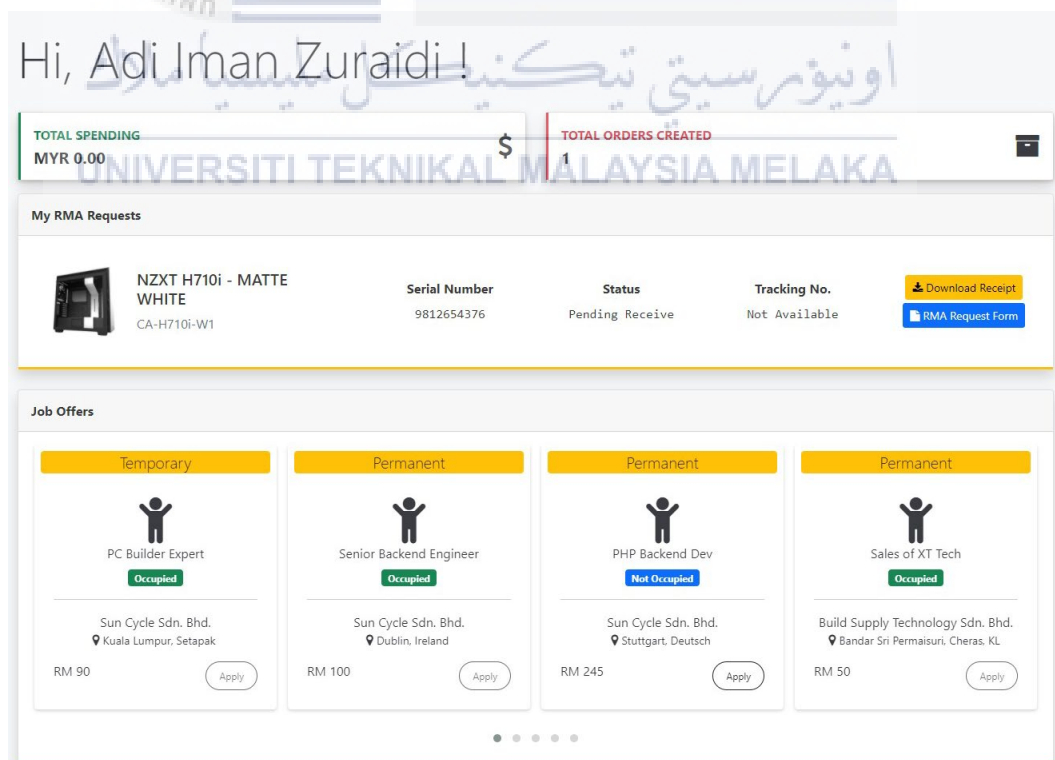


Figure 4.31: Home Dashboard for Customer

Figure 4.31 shows the output screen for customer dashboard, it consists of total spending and total orders created. Job offers are display in a carousel which have a draggable function to view all job offers offered by distributor or reseller. If the user has request a RMA, it will be displayed in the my RMA request section.



Figure 4.32: Tracking Progress

Figure 4.32 shows the output screen for tracking progress. In this section it will show the parcel status before dispatched to official shipping courier. Each progress will be recorded in the system to keep customer updated in all time. A map is shown as a visualization of how far the parcel will travel from the origin to the customer itself.
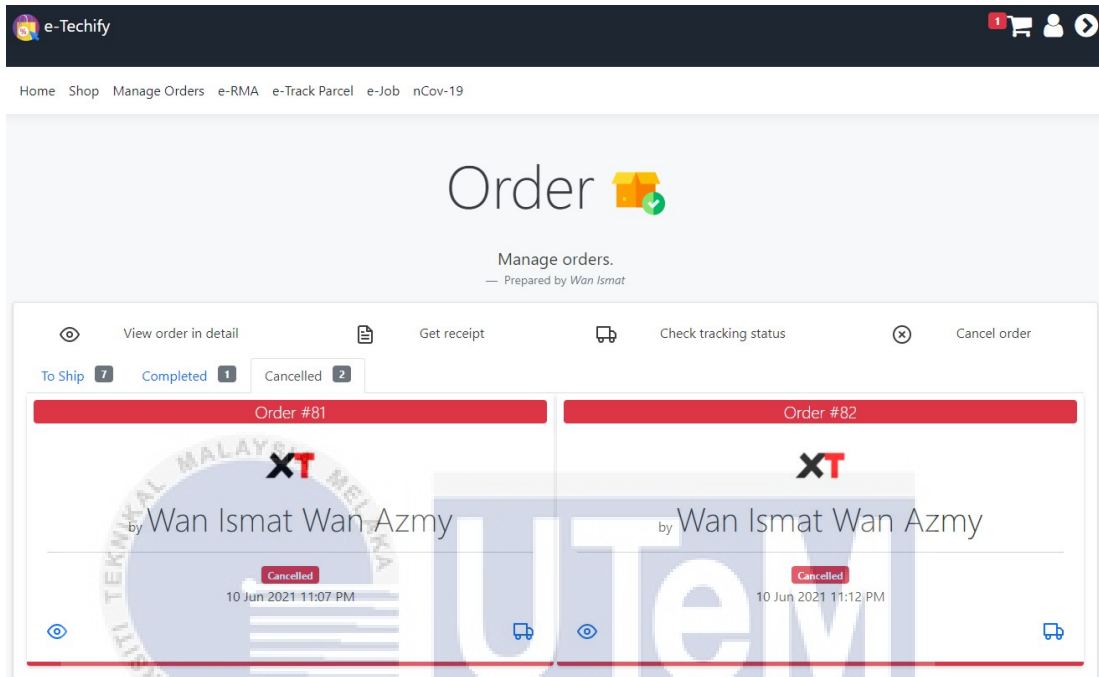


Figure 4.33: Cancelled Order
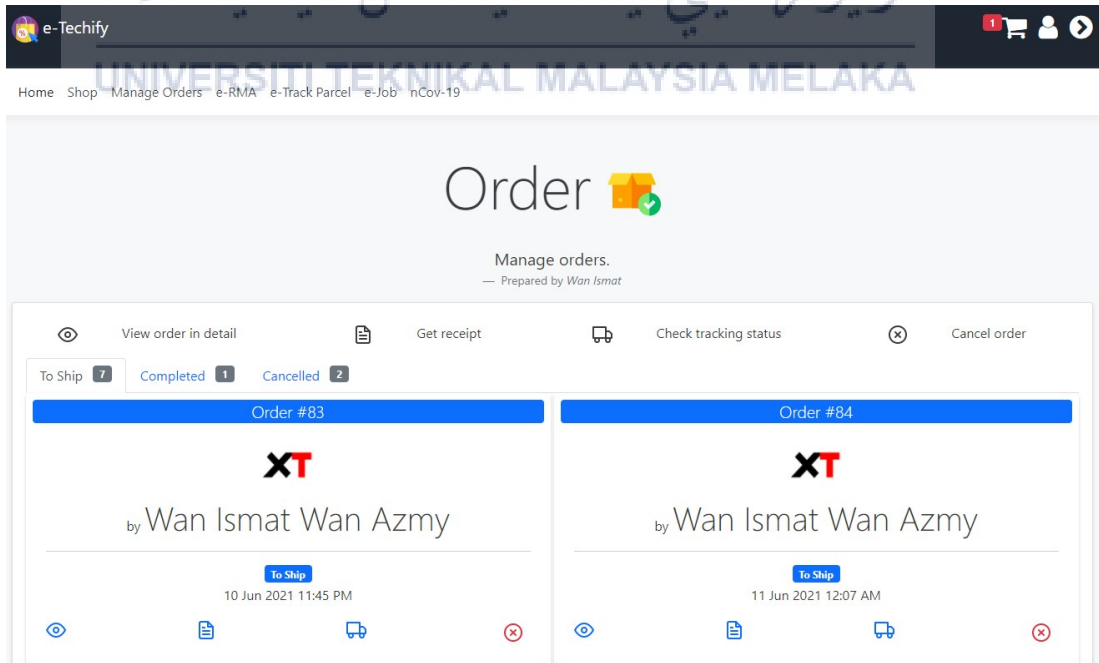


Figure 4.34: To Ship Order
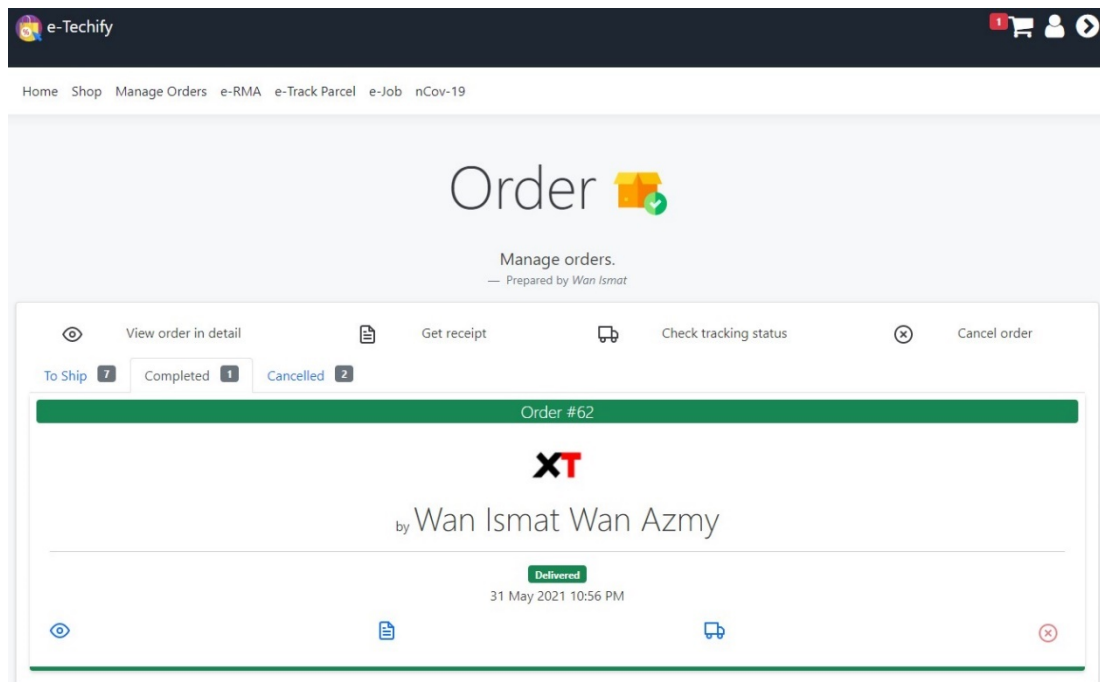
Figure 4.35: Delivered Order

Figure 4.33, 4.34 and 4.35 shows the output screen for three types of order status that have been grouped by according to its status. As we can see in figure 4.33 customer only have the ability to see the order details and last updated tracking status if order is cancelled. While in figure 4.34 we can see all customer to ship order status have the ability to view order details, generate e-receipt, check current tracking status, and cancel the order if the tracking status is not set to "*Processing Order*". Lastly, figure 4.35 shows that customer can do all the function as in to ship order status, but they do not have the ability to cancel the order once the parcel is dispatched to the official courier.

Figure 4.36: Shop

Figure 4.36 shows the output screen for shop section. In this section, customer will have the ability to choose any products that are offered in the system and can add product to cart if they are interested. Customer can filter the shop catalogue by type of brand and price either low to high or vice versa. Pagination is provided in order let user have the ease to view product in a good arrangement of 6 product per page.
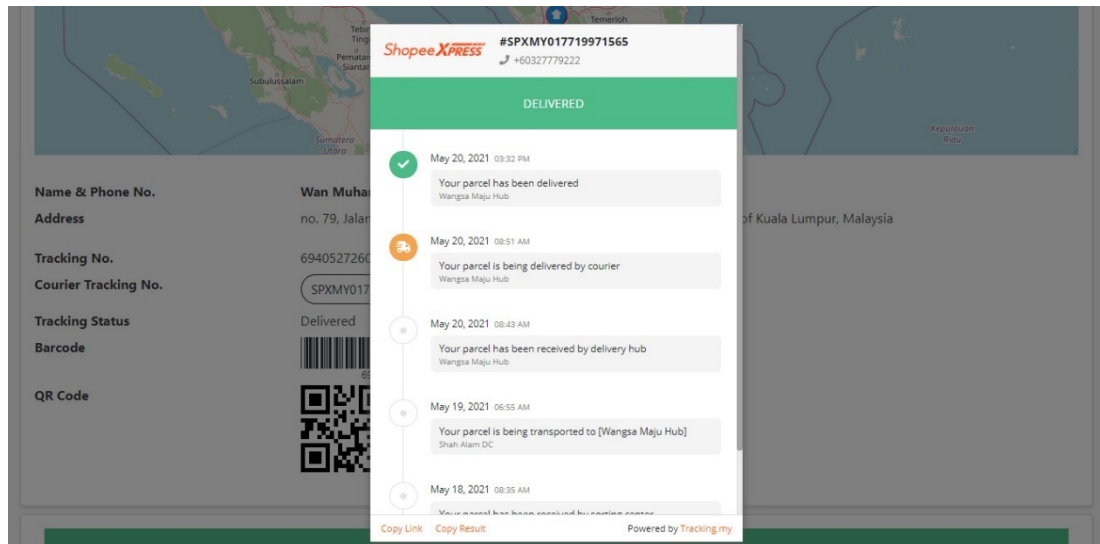
Figure 4.37: Tracking Status Dialog

Figure 4.37 shows the output screen for tracking status dialog. This dialog will be available in RMA request section and in tracking status page. The purpose of using this kind of technique is to ease user in tracking their parcel.



Figure 4.38: Tracking Status e-Techify

Figure 4.38 shows the output screen for tracking status of e-Techify. In this section it will show the current parcel status according to the tracking number that is supplied in the search bar. Each tracking information will be displayed with a suitable icon.

Figure 4.39: Airway Bill

Figure 4.39 shows the output screen for airway bill for each order. Each airway bill will be generated when a purchase has been successfully created. In the airway bill it shows all related information such as order details, sender details, recipient details, a barcode and also a qrcode for the runner to scan in order to contact the customer without having to dial in their phone.

Figure 4.40: Order Confirm Email

Figure 4.40 shows the output screen for order confirmation mail. This email will be sent over to customer once an order purchased have been successfully created. Ordered products or items will be displayed including with its quantity, per unit price, order subtotal and payment method. A button to generate e-receipt is also provided.

**Billed To**

**#XT-722643723054**

**MASHANUM BINTI OSMAN**
Fakulti Teknologi Maklumat dan
Komunikasi (FTMK), UTeM, Durian Tunggal,
Malacca, Malaysia

**23 Jun 2021**

XT Express

**#751104769135**
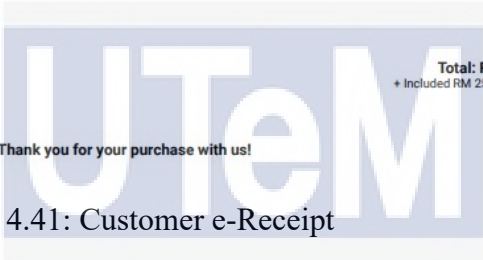
**Xmiryna Technology [SA0546866-M]**

No. 79 Jalan Taman Melati 1,
Taman Melati, Setapak,
53100, Kuala Lumpur
xmiryna.tech@outlook.com / +(60) 17-217 8319
xmiryna.com.my

| Summary | Quantity | Unit Price | Price |
|---|---|---|---|
| **E850 DIGITAL PSU - BLACK**<br>XKoWGawD<br>10 years local distributor warranty. | x1 | RM 649.00 | RM 649.00 |
| **Fractal Design Define 7 Black TG Dark Tint**<br>FD-D7-123<br>2 years local distributor warranty. | x1 | RM 769.00 | RM 769.00 |

**Total: RM 1,443.00**
+ Included RM 25 Shipping Fees

Thank you for your purchase with us!

Figure 4.41: Customer e-Receipt

Figure 4.41 shows the output screen for customer e-receipt. Each receipt that is generated will have their own unique identification which makes it easy for reseller and customer to find their receipt in the system. It consists of order details, address of reseller and customer, system tracking number and list of ordered products with payment information.

**RMA Request**

**#6**

**Muhammad Afiq Iskandar Bin Mohamad Azri**
No.1, Taman Air Tawar Indah, Taman Air
Tawar Indah, 13050 Butterworth, Pulau
Pinang, Malaysia

**Sun Cycle Sdn. Bhd.**
Sun Cycle Sdn Bhd (HQ), Kawasan
Perusahaan Menglembu, Ipoh, Perak,
Malaysia

+(60) 329359598

**+(60)125001524**

Requested On

**26 May 2021**

exclusively by
**Xmiryna Technology**

| Summary | Brand | Product SKU | Serial Number | Date Of Purchase |
|---|---|---|---|---|
| **NZXT H1 BLACK CASE**<br>10 years local distributor warranty. | NZXT | CA-H16WR-B1-UK | 111222333444555 | 2021-05-25 |

Reason
**My PCIE bracket did not detect VGA.**

**Please print this and include inside your parcel. Thank You!**

Figure 4.42: RMA Request Receipt

Figure 4.42 shows the output screen for RMA request receipt. Each RMA request will be provided a receipt as a proof of request have been made in the system. Customer must include this receipt inside the parcel when they are sending over for a repair. Details such as address of distributor and customer are shown with a date of requested on repair including RMA details.

Figure 4.43: Job Application Success Mail

Figure 4.43 shows the output screen for job application success mail. This email will be sent over to customer once a job application that is applied by customer have been approved by the employer. Job details will be displayed and for further information about the job they have to contact the respective employer for further instructions.
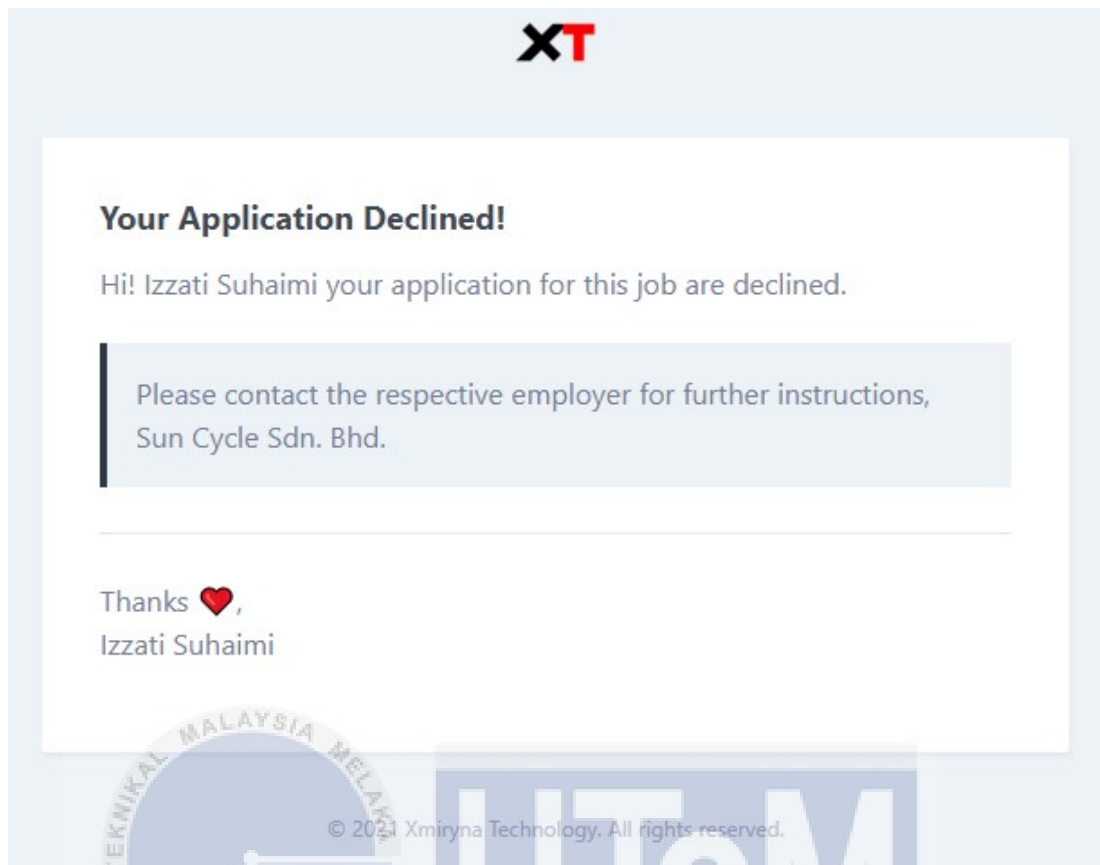
Figure 4.44: Job Application Decline Mail

Figure 4.44 shows the output screen for job application decline mail. This email will be sent over to customer once a job application that is applied by customer have been declined by the employer.
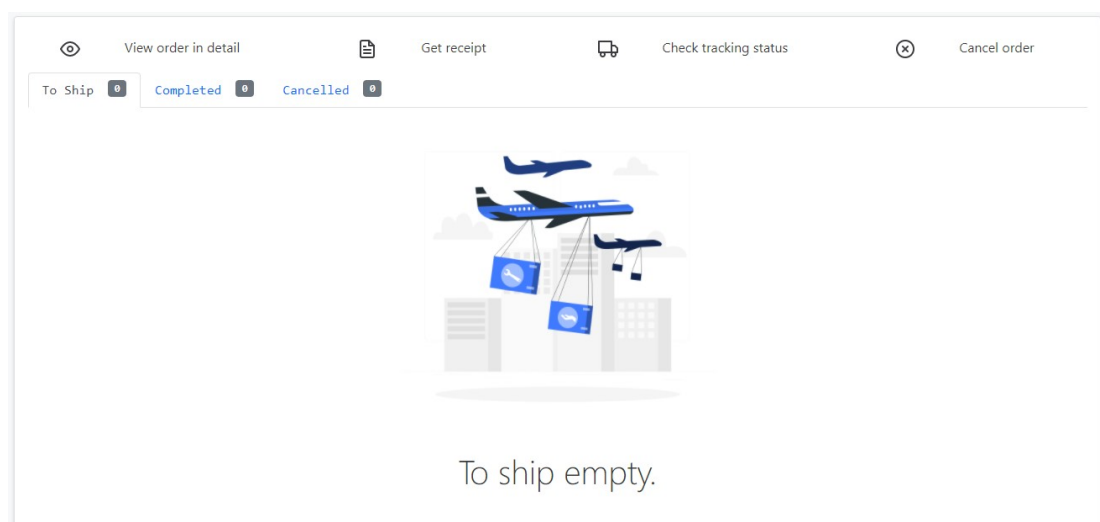


Figure 4.45: Empty to Ship Status
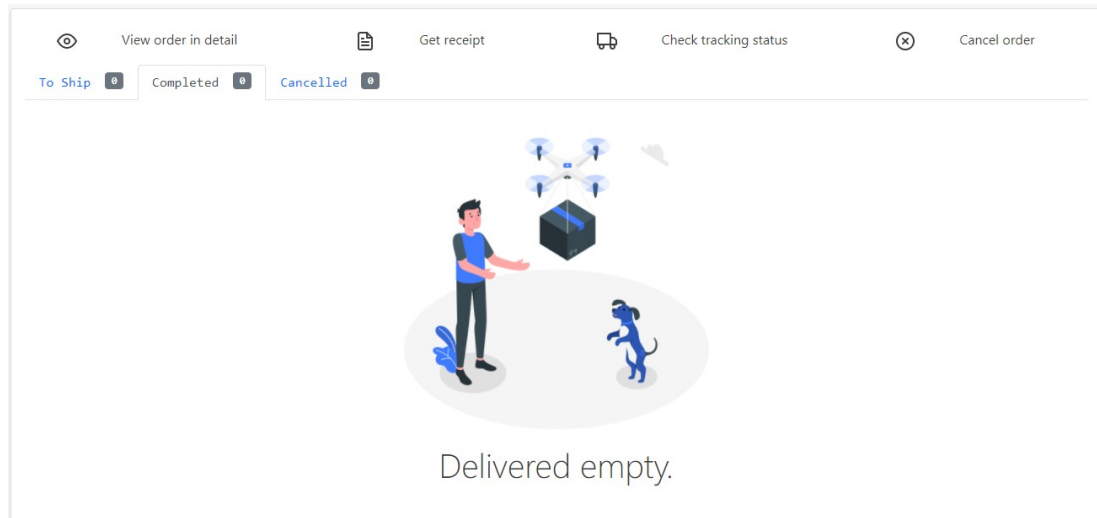
Figure 4.46: Empty Delivered Status
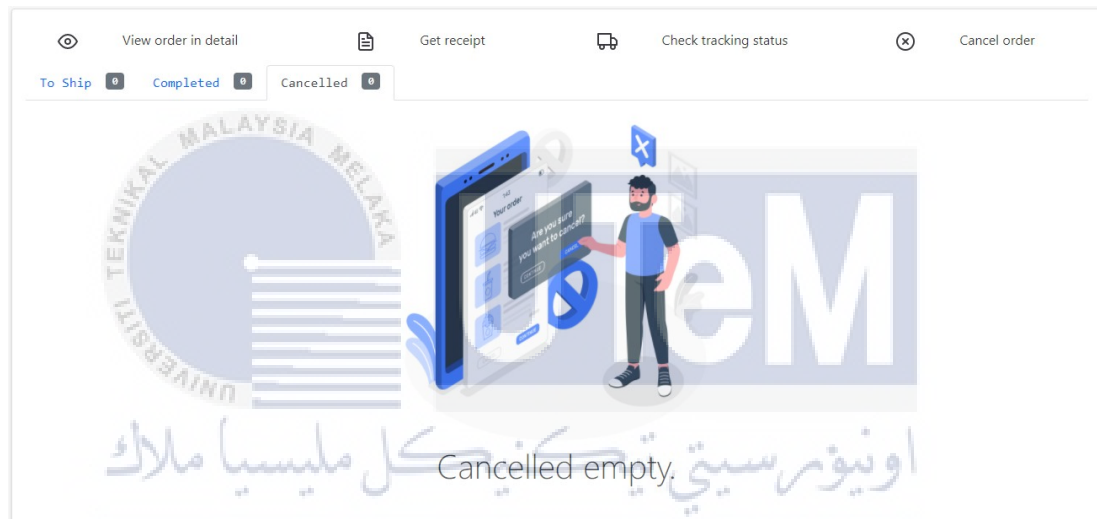


Figure 4.47: Empty Cancelled Status

Figure 4.45, 4.46 and 4.47 shows the output screen for three types of order status that have been grouped by according to its status. As we can see in figure 4.45 customer can view order with "To Ship" status. While in figure 4.46 customer can view order with "Completed" status. Lastly, figure 4.47 gives ability to customer can view order with "Cancelled" status. If there is no order in those three status, it will show a message, a picture and indicator to show it is empty.

### 4.2.3    Database Design

This section contains the conceptual, logical database designs and entity relationship diagram (ERD) of e-Techify System.

### 4.2.3.1  Conceptual and Logical Database Design

This section contains the conceptual and logical database of e-Techify. Conceptual database design is displayed in the Figure 4.24. The database design of e-Techify consists of fourteen entities. They are confirm_orders, trackings, orders, order_detail, addresses, repairs, roles, role_user, users, products, jobs, shipments, shipment_details and distributor_products. Logical database design has the same entity as the conceptual database design, but it has attributes with its data types.

## 4.2.3.2 Conceptual Database Design
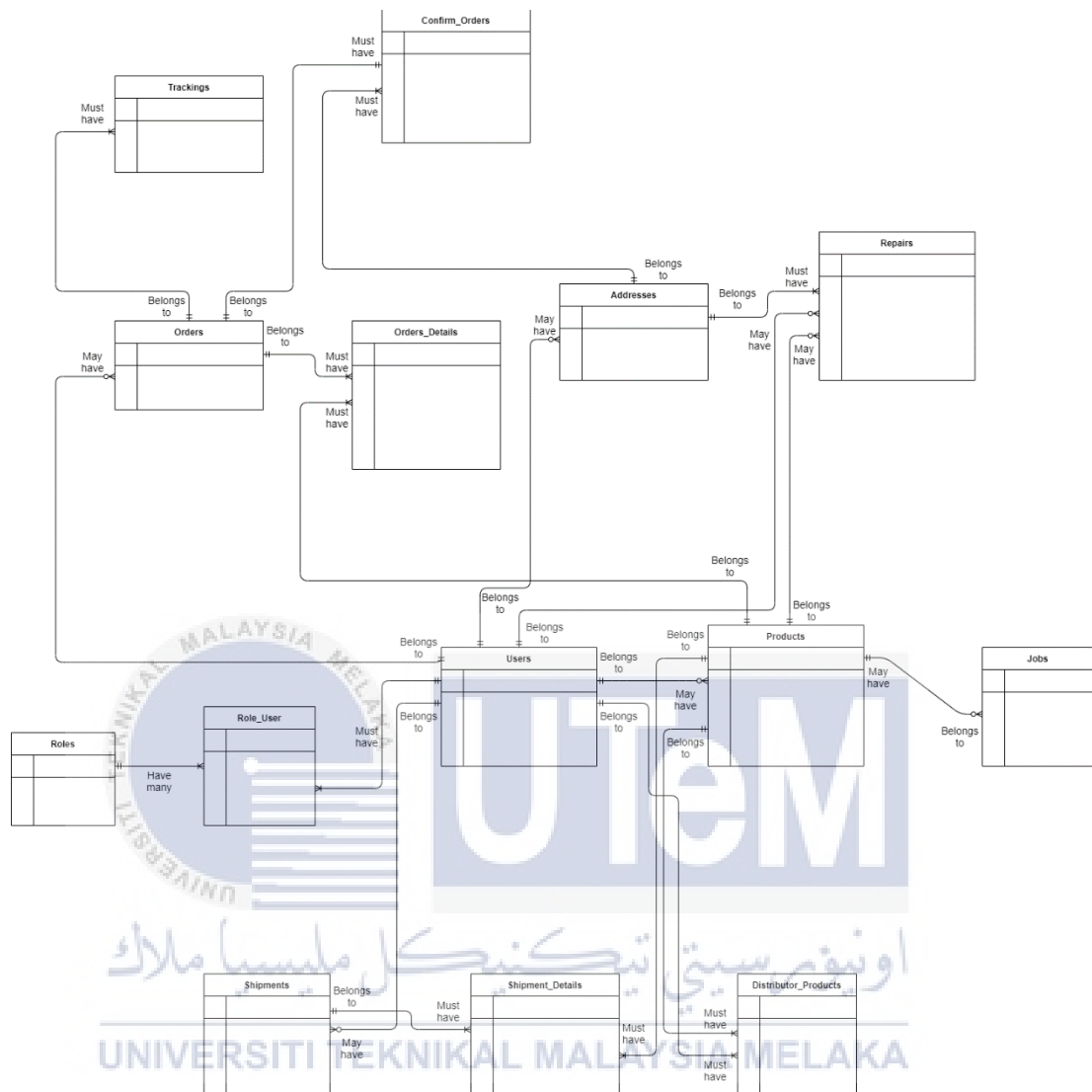


Figure 4.48: Conceptual Design for e-Techify

**4.2.3.3 Logical Database Design**
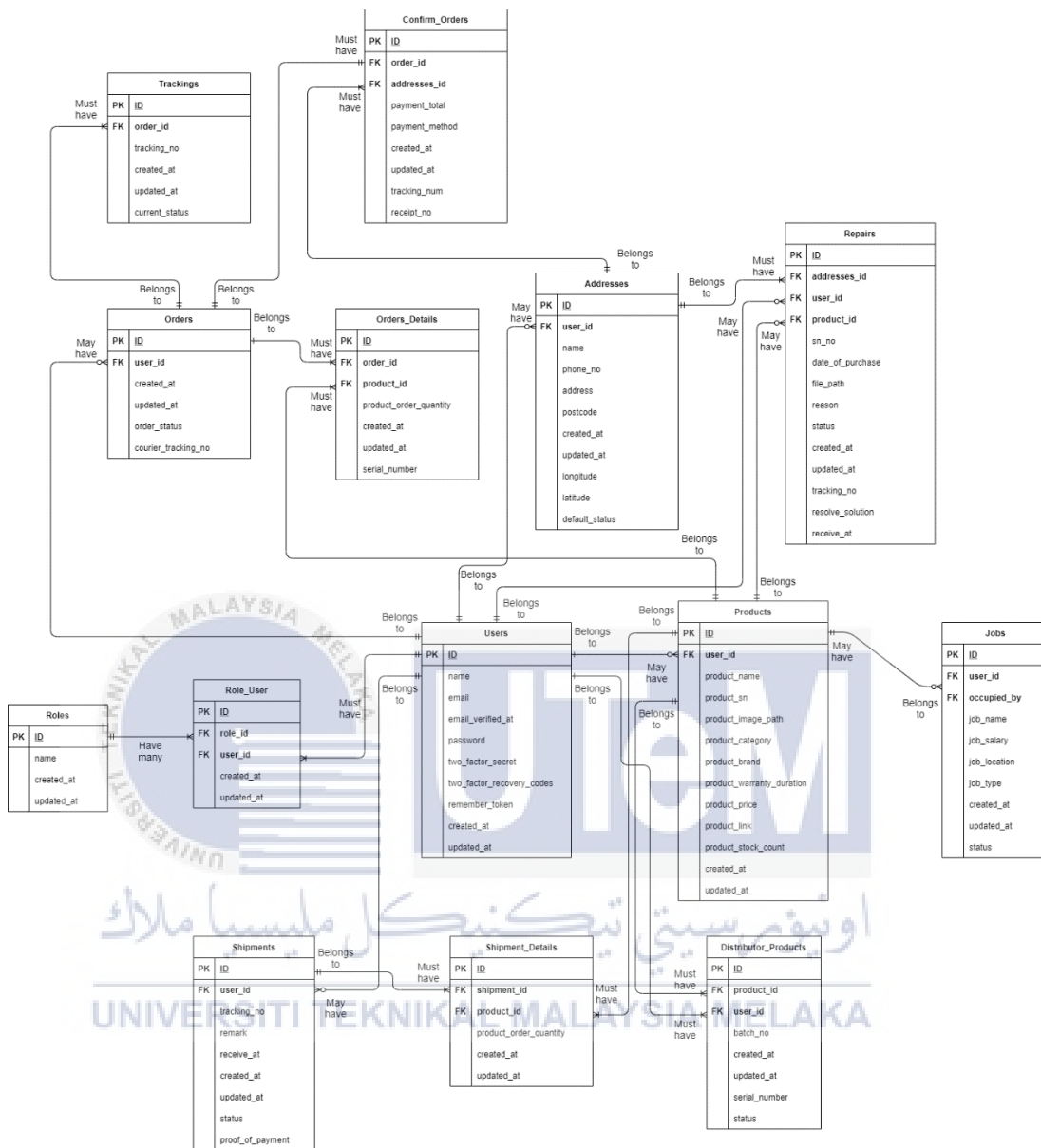


Figure 4.49: Logical Design for e-Techify

**4.2.3.4  Entity Relationship Diagram**



Figure 4.50: ERD of e-Techify

Entity Relationship Diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is a component of data. In other words, ER diagrams illustrate the logical structure of databases.

**4.2.3.5  Business Rules**

A business rule is a component of the business rules flow that is used to assert business structure or impact business behavior. The database architecture relies heavily on identifying and documenting business rules. The designer can use business

rules to build relationship participation rules and limitations, as well as a correct data model.

Below are the business rules for the system:

➢ One and only one address belongs to a user, while a user may have many or none address.

➢ One and only one confirm_orders belong to an order, while an order must have one and only one confirm_orders.

➢ One and only one confirm_orders belong to an address, while an address must have one or many confirm_orders.

➢ One and only one distributor_products belong to a product, while a product must have one or many distributor_products.

➢ One and only one distributor_products belong to a user, while a user must have one or many distributor_products.

➢ One and only one job belongs to a user, while a user may have many or no job.

➢ One and only one order belongs to a user, while a user may have many or no order.

➢ One and only one order_details belong to an order, while an order must have one or many order_details.

➢ One and only one order_details belong to a product, while a product must have one or many order_details.

➢ One and only one product belongs to a user, while a user may have many or no products.

➢ One and only one repairs belong to an address, while an address must have one or many repairs.

➢ One and only one repair belongs to a user, while a user may have many or none repair.

➢ One and only one repair belongs to a product, while a product may have many or none repair.

➢ A user can have many roles and a role can have many users.

➢ One and only one shipment belongs to a user, while a user may have many or no shipment.

➢ One and only one shipment_details belong to a shipment, while a shipment must have one or many shipment_details.

➢ One and only one shipment_details belong to a product, while a product must have one or many shipment_details.

➢ One and only one tracking belong to an order, while a order must have one or many tracking.

## 4.3     Detailed Design

This section elaborates project detailed designs, including physical database design and software design.

## 4.3.1    Software Design
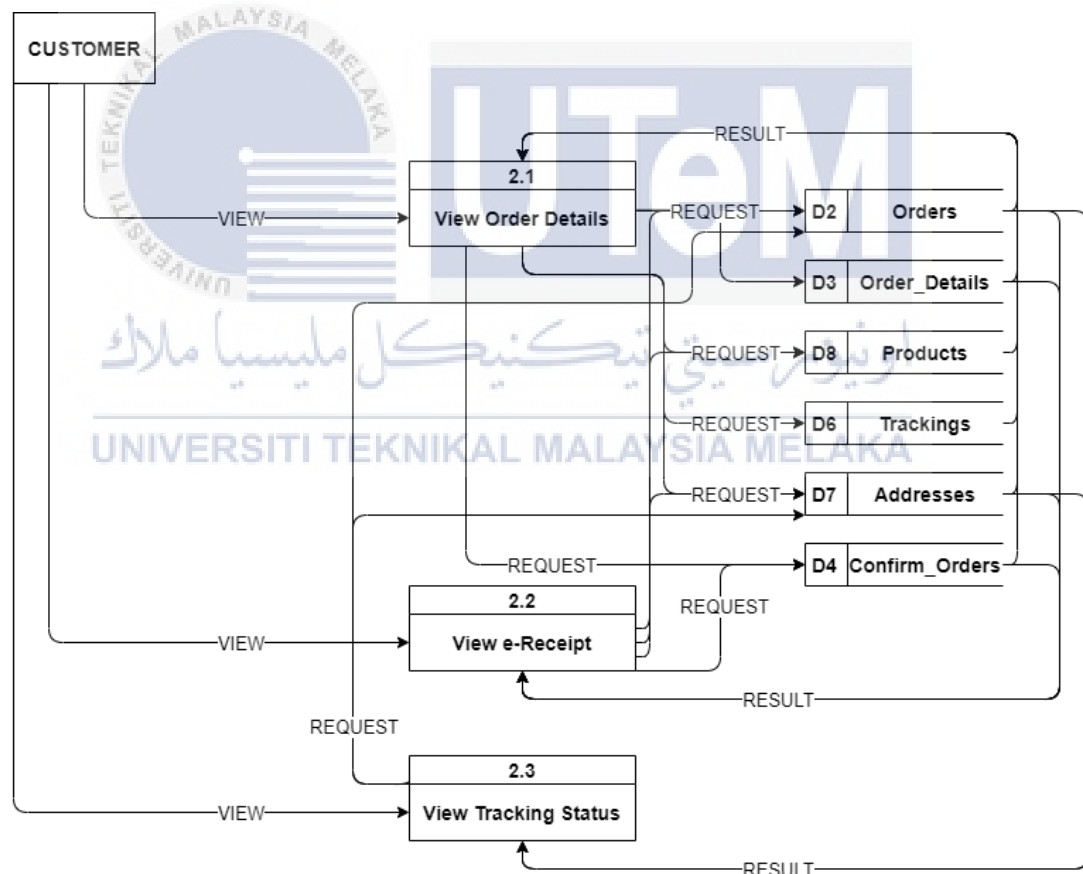


Figure 4.51: Level-1 for Customer Manage Orders Process

### 4.3.2 Physical Database Design
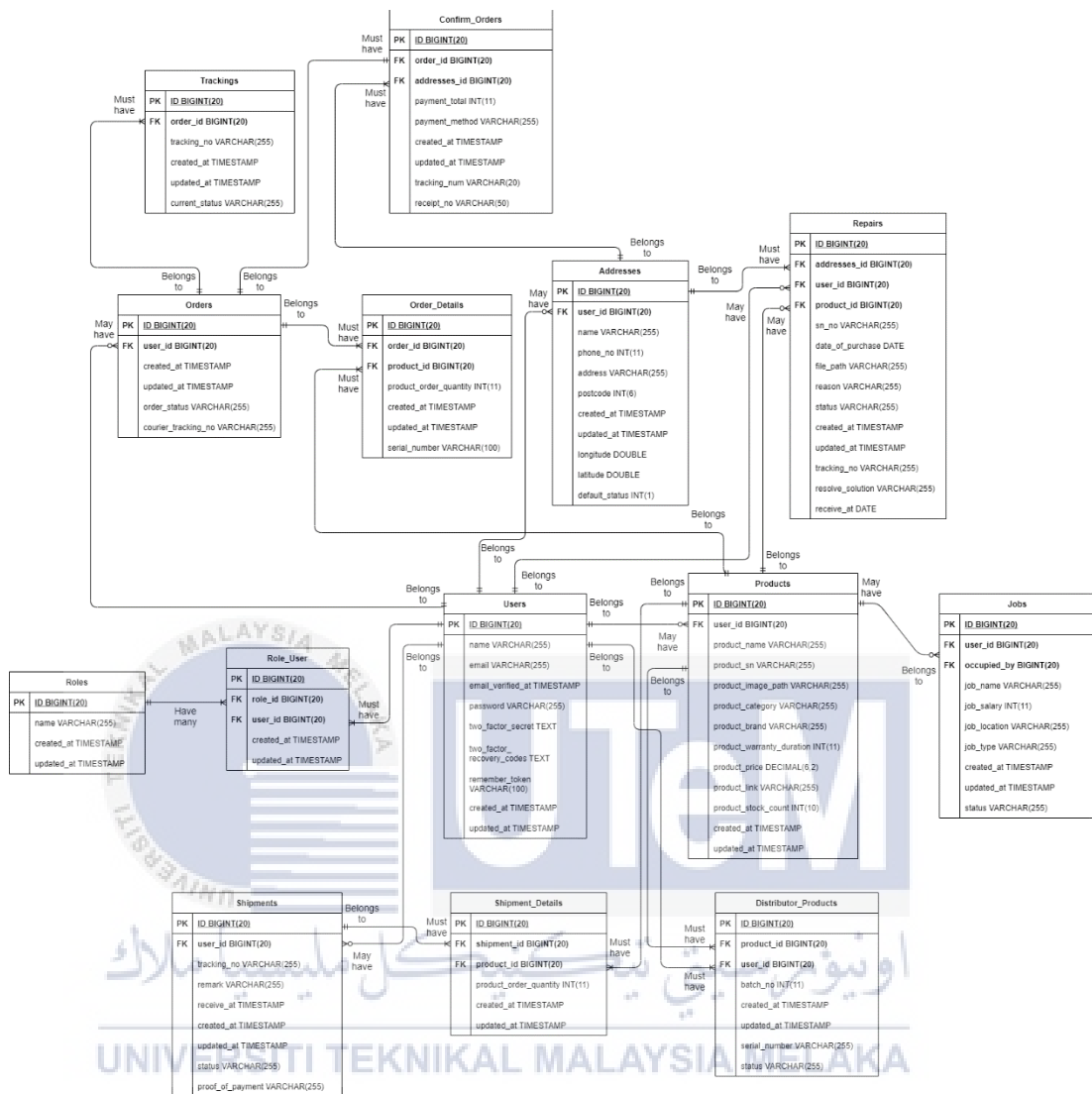


Figure 4.52: Physical Design of e-Techify

The physical database design of e-Techify consists of fourteen entities with its data types and length. They are confirm_orders, trackings, orders, order_detail, addresses, repairs, roles, role_user, users, products, jobs, shipments, shipment_details and distributor_products.

**4.4     Conclusion**

The various designs that were used to help develop this system are outlined and described in this chapter. All of the diagrams in the Waterfall model are relics from the design stage. The design diagrams serve as a template for the project's next stage, implementation, and contain system architecture, user interface, database, and software design. During the implementation stage, the data represented on the design works is transferred and converted into a usable component, module, or prototype.

# CHAPTER 5: IMPLEMENTATION

## 5.1 Introduction

The implementation of the e-Techify is discussed in this chapter. It involves the creation of the software development environment, software configuration management, and the status of each module's implementation.

## 5.2 Software Development Environment Setup
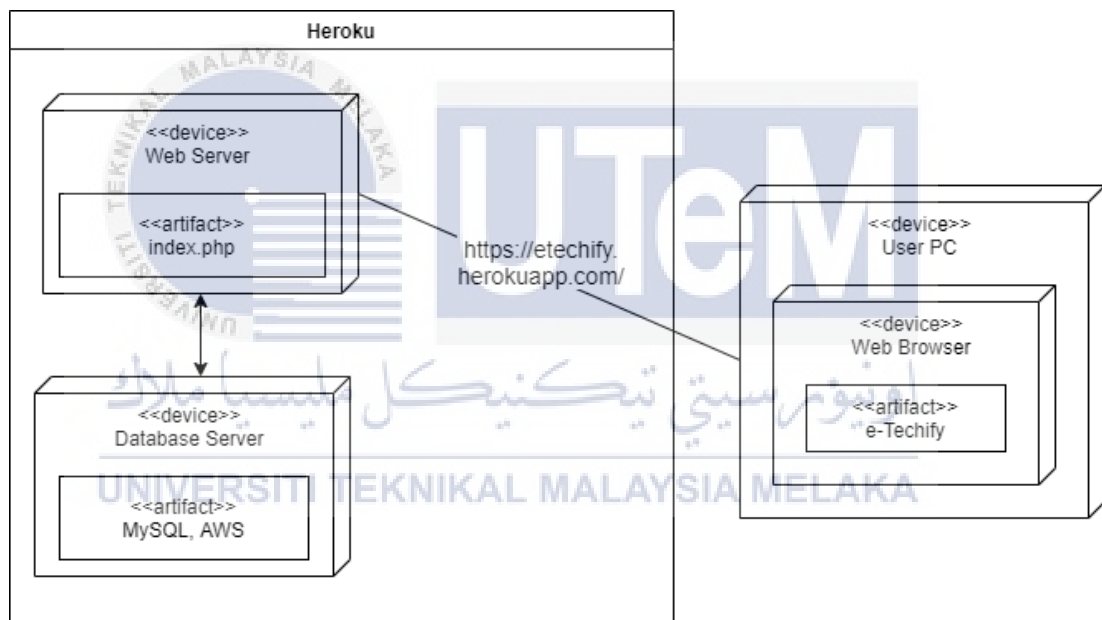


Figure 5.1: Deployment Diagram

Figure 5.1 shows the deployment diagram of e-Techify. According to the figure, the Heroku contains a cloud server, database and the client machine will be connected to the Heroku server via public Uniform Resource Locator (URL) with encrypted SSL. Client user must install at least one internet browser. It is suggested that every user should be using Google Chrome or Microsoft Edge as their primary browser in order to get the best experience during their browse time.

## 5.3    Software Configuration Management

This section explains the configuration environment setup and version control procedures.

### 5.3.1    Configuration Environment Setup

The configuration environment of this system requires Heroku cloud server. It is a cloud platform as a service that supports php programming language and manages all system request made by client to server. The following part covers the deployment of e-Techify to Heroku cloud. Then, the explanation will be followed by setting up MySQL database.

#### 5.3.1.1  Installation for Heroku and JawsDB

Step 1: Create a new app in Heroku.



Figure 5.2: Heroku Create New App Landing Page

Step 2: Input the proper app name to create a new app in Heroku and choose Europe as the region for our application.

Figure 5.3: Heroku Create New App Form

Step 3: After successfully create an app in Heroku, we are going to proceed with the deployment method. In this demonstration we will be using GitHub. Users have options to choose other branches as well. Heroku will fetch all project files according to the selected branch. If all the process run smoothly, it will show up a status says "Connected".
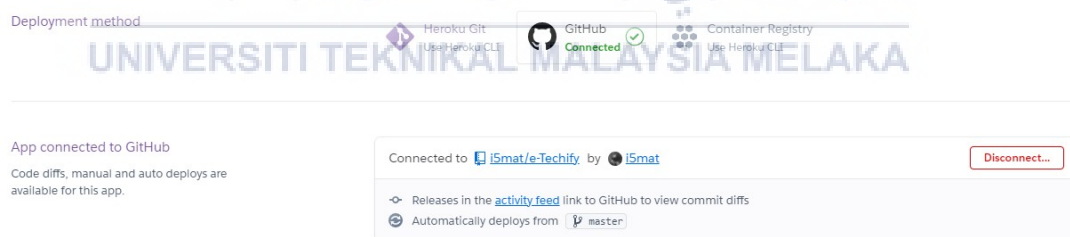


Figure 5.4: Heroku Deployment Method

Step 4: After all deployment method have succeeded, user have to proceed with add-ons. In this step we will add JawsDB MySQL as our database as we have added it from Resources tab and in add-ons section. JawsDB will provide credentials for our database setup such as DB_DATABASE, DB_HOST, DB_PASSWORD, DB_PORT and DB_USERNAME in order to access the database provided.
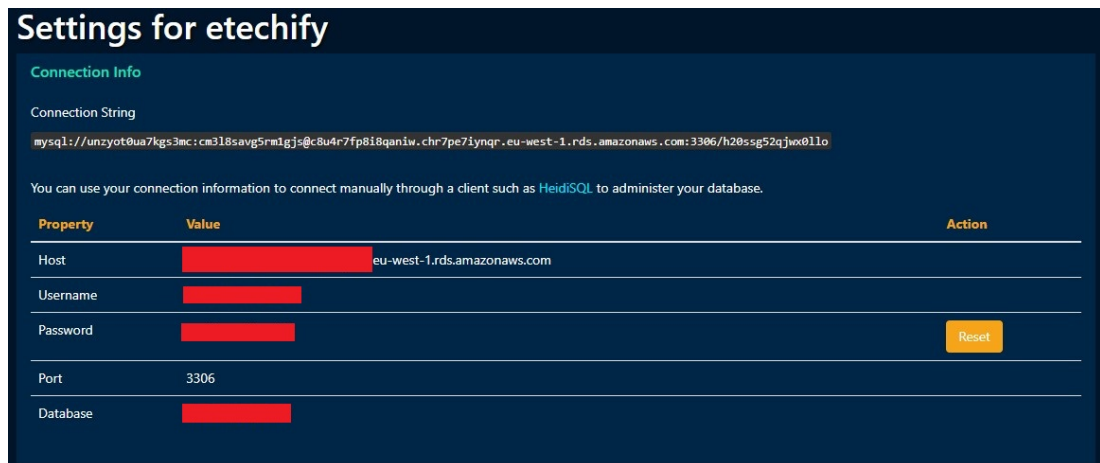
Figure 5.5: JawsDB Credentials
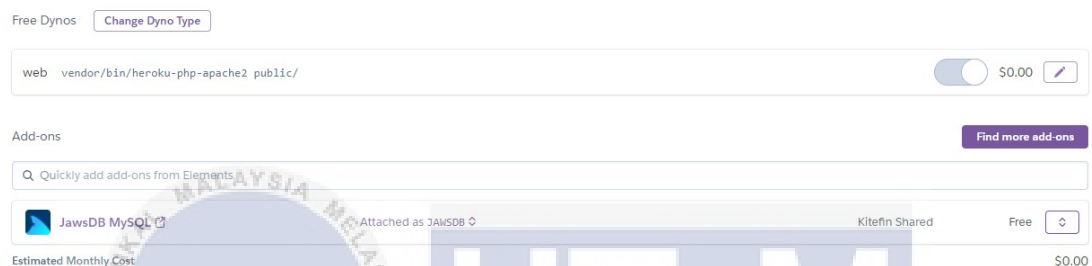


Figure 5.6: Heroku Add-Ons

Step 5: When user have done set-up the add-ons. User can freely use any Integrated Development Environment (IDE) to view and have full access to the database provided by JawsDB. The next step will proceed with config vars which means user have to add all the environment variables inside this section in order for Heroku to recognize our application.
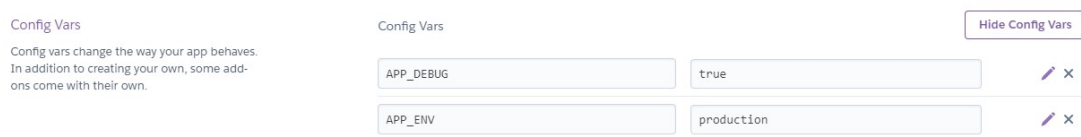


Figure 5.7: Heroku Environment Variables

Step 6: After user have configured all related environment variables, user will be provided with a dashboard to view elements like installed add-ons, dyno formation, collaborator activity and latest activity update of our system deployment. All of the changes will be detected immediately by Heroku in order to make sure our application is up to date. Differentiation in each line of codes will be shown which make it easy for a user to recognize small or big changes. This is also with the help of GitHub.

Further explanation for GitHub will be explained in version control procedure, 5.3.2 section.



Figure 5.8: Heroku User Dashboard

Step 7: After all steps are correctly executed, our application should be available and live in Heroku cloud server. https://**appname**.herokuapp.com/

### 5.3.1.2  Installation for Amazon Web Service S3 Bucket

Step 1: Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/.

Step 2: Choose Create bucket. Then, the Create bucket page opens.



Figure 5.9: Create Bucket Setup

Step 3: In Bucket name, enter a DNS-compliant name for your bucket. We must carefully choose the right name because we can't change its name once it is created. The bucket name must:

- Be unique across all of Amazon S3.
- Be between 3 and 63 characters long.
- Not contain uppercase characters.
- Start with a lowercase letter or number.

Step 4: Select the AWS Region where you want the bucket to be stored in region. Choose a Region that is geographically close to you to reduce latency and expenses while also meeting regulatory standards. Objects saved in a Region never leave it unless you transfer them to another Region directly.

Step 5: Keep the remaining settings set to the defaults. Then, click "Create Bucket".
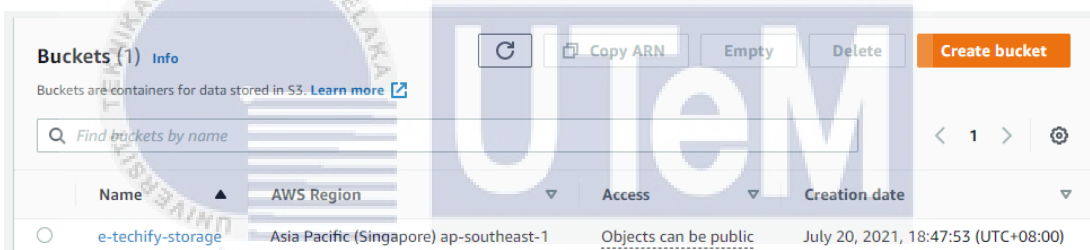


Figure 5.10: Bucket in S3

Step 6: We have created a bucket in Amazon S3. List of buckets will be display in S3 Management Console. All of the settings can be configured by custom in the access section.

### 5.3.2 Version Control Procedure

GitHub is used to manage the system's version control. GitHub is a website that hosts Git repositories on the web. With its enhanced features, GitHub delivers Git functionality such as version control and source code management. It is also an effective means of storing file changes and maintaining file integrity, in addition to offering a version control method for code development.

Step 1: Create a Repository



Figure 5.11: Create New Repository in GitHub

Figure 5.11 shows a form that is provided by GitHub to fill in repository name, description, choose either public or private, and some checkbox to add a readme, .gitignore or choose a license.

Step 2: View project files in GitHub.
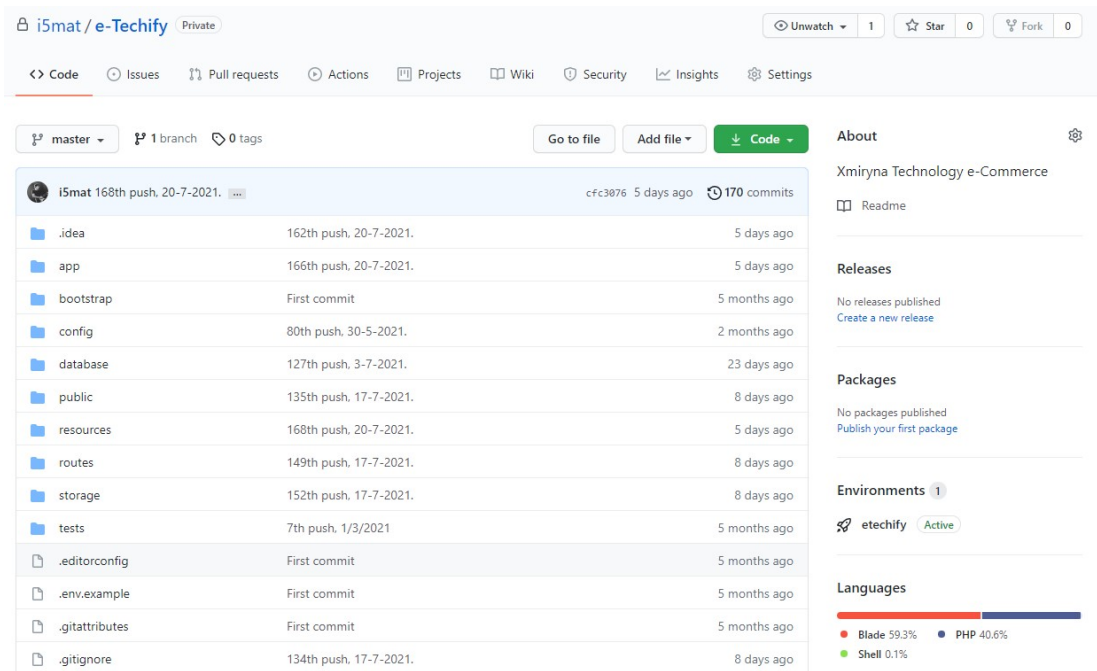
Figure 5.12: e-Techify Repository

Figure 5.12 illustrates a GitHub repository named e-Techify that was created to administer the project. This repository can hold all of the project's files, data, and directories. As a result, this repository will hold all of the code files.

Step 3: Commit and push project files to GitHub.
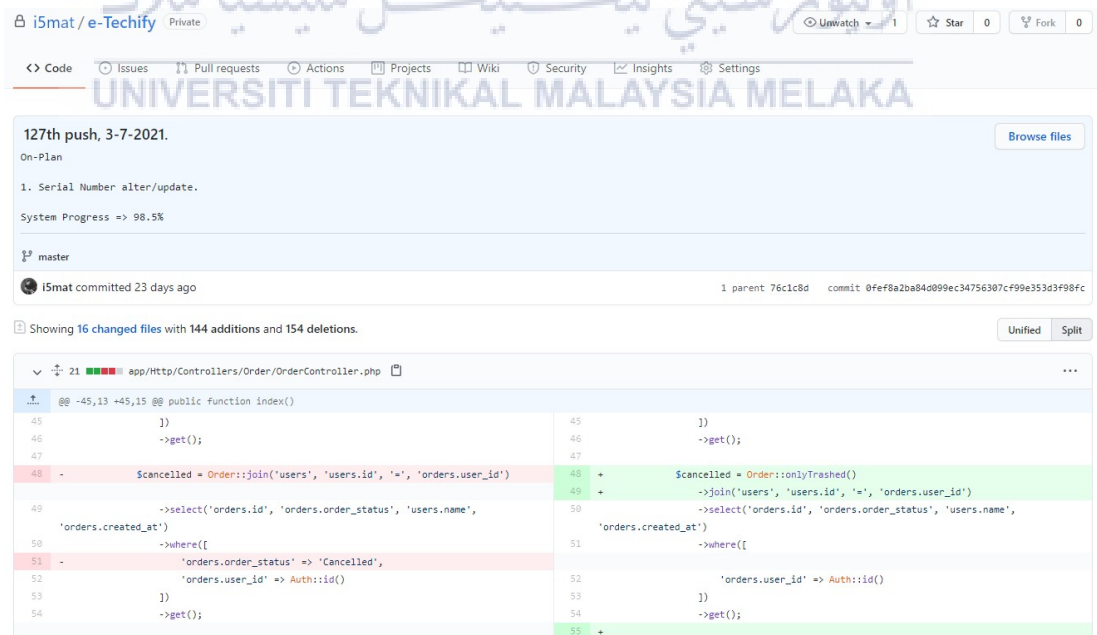


Figure 5.13: Push and Commit Changes to GitHub Repository

Every modification is saved to the GitHub repository and is referred to as a commit on GitHub. Each commit is accompanied by a commit message, which explains why the change was made. As a result, commit messages serve to document the history of the recorded modifications while also assisting other visitors or contributors in understanding what was done and why. Any updates to the e-Techify source code are committed to the e-Techify GitHub repository. "127th push, 3-7-2021," according to the screen shown in Figure 5.13, is the commit message connected with the commit. As can be seen, the area marked in red denotes the old version of the code, whilst the part highlighted in green denotes the current version.

## 5.4    Implementation Status

The current stage of development for each of the system's components is summarized in this section. The current status of each major module and sub module's implementation is shown in Table 5.1.

Table 5.1: Implementation Status of Main Modules & Sub Modules

| Module Name | Description | Duration | Date Completed |
|---|---|---|---|
| Login | The user must log in with an existing account and be able to log out at any time. | 1 Days | March – June 2021 |
| Register | Users have the ability to register either as a or a distributor. | 4 Days | |
| Manage Orders | The system will allow user that have reseller role to manage all customer orders. Information about an order is detailed out which consist of ordered items, when the order is created, tracking number that is assigned, name, phone number, receiver address and current tracking status. Other functionality such as e-receipt, insert serial number, | 5 Days | |

| | print air waybill, update current parcel status, and cancel order are also provided. (Reseller)<br><br>The system will allow user that have customer role to manage their orders. Information about an order is detailed out which consist of ordered items, when the order is created, tracking number that is assigned, name, phone number, receiver address and current tracking status. Other functionality such as e-receipt, check parcel tracking status, and cancel order are provided. (Customer) | | |
|---|---|---|---|
| Manage Products | The system will allow user that have distributor role to manage their products in the system. They will have the ability to insert new product, update and delete existing product. | 1 Week | |
| Manage Job | The system will allow user to insert a job which the inserted jobs will be offered to user that holds customer role. The purpose of this part to let customer generate some side income. | 2 Days | |
| Manage Shipments | The system will allow user to manage the shipments requested by reseller. All of the requests will be grouped by status. Each | 3 Days | |

| | | | |
|---|---|---|---|
| | request needs to have approval from distributor itself. | | |
| Shop | The system will allow user to shop their desired items throughout the list. | 5 Days | |
| e-RMA | The system will allow user to submit RMA request for their purchased product if there is a problem they encounter during usage. | 2 Days | |
| Manage Repair | The system will allow distributor and reseller to manage RMA request by customers. | 5 days | |
| e-Track Parcel | The system will allow user to track their parcel status in detail. | 1 Day | |
| Manage Cart | The system will allow user to add or delete items in cart, alter quantity, change receiver address according to the address that have been saved and make payment by using online payment method. | 1 Week | |
| Stock Management | The system will allow user to see list of distributors that is exist in the system. In addition, further information is displayed such as stock count and other relevant information for each product. | 2 Days | |
| Request Shipment | The system will allow user to request desired items to the distributor of the products. Status for each request will also be displayed which include remark, | 5 Days | |

| | | | |
|---|---|---|---|
| | a receipt and in detailed view of each request information. | | |
| Receipt Finder | The system will allow user to search and retrieve customer receipt for each order. | 1 Days | |
| Manage Product Serial Number | The system will allow user to insert each product serial number with batch number. Statistic for each batch is recorded and data will be displayed in a bar chart grouped by batch number. A table filled with full information regarding a serial number status will be shown. | 3 Weeks | |
| Manage Users | The system will allow user to manage all users in the system. Create, update, and delete operation are included in this functionality. | 1 Days | |
| User Dashboard | The system provides user in a compact view which offers variety of important functions such as manage all RMA requests, display earnings by monthly and annual, summary of earnings in Q1 till Q4 in bar chart and manage all job offerings. (Reseller)<br><br>The system provides user in a compact view which offers variety of important functions such as manage all RMA | 2 Weeks | |

| | requests, total spending, total orders created, and display all job offerings. (Customer)  The system provides user in a compact view which offers variety of important functions such as manage all RMA requests, total RMA, total products, display sale for each brand in bar chart while percentage sale for each brand in pie chart and manage all job offerings. (Distributor) | | |
|---|---|---|---|
| Manage Address | The system will allow user to add new address without any limits and set an address to default address. | 3 Days | |
| Manage Profile | The system will allow user to alter their account information in terms of name and email address. | 1 Days | |

## 5.5    Conclusion

The first half of this chapter covers software development environment setup, while the second section, which goes into configuration environment setup and version control mechanisms in greater depth, covers software configuration management. It also gives an overview of how each module is being implemented in e-Techify.

# CHAPTER 6: TESTING

## 6.1 Introduction

The details of the test plan are revealed in this chapter, which includes a description of the test organization, test environment, and test timetable. In terms of test classes, it also describes the test strategy. The test design, which includes a test description and test data, follows the chapter. The test findings and analysis based on the testing activities are then presented in the following chapter.

## 6.2 Test Plan

This section explains the personnel involved in the test organization. Then, test environment and test schedule are also documented in the following sections.

### 6.2.1 Test Organization
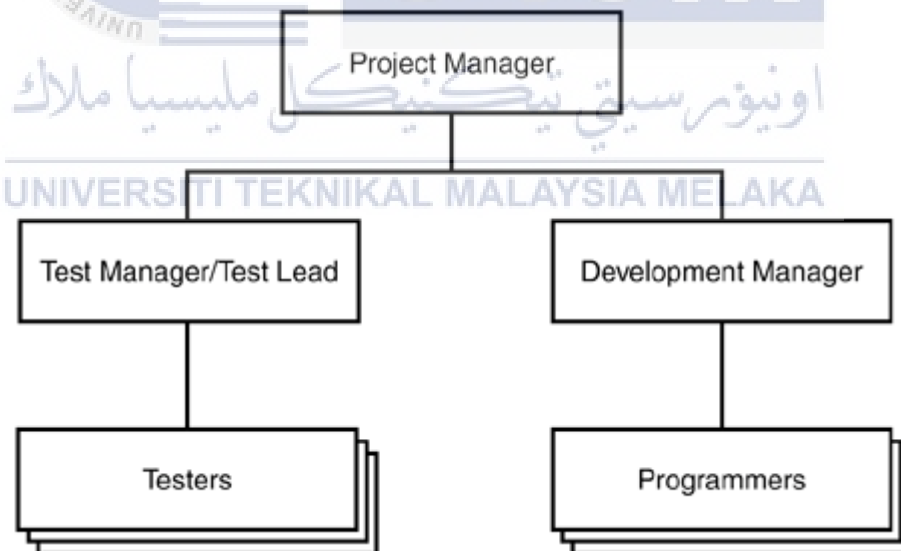


Figure 6.1: Test Organization Chart

During the project's testing phase, the supervisor serves as the testing management, while the developer serves as the testes, preparing test cases, test data, and the test environment. The testing manager oversees all of the testing in the functionality test to ensure that the output is correct. During the user acceptance test,

the supervisor and external industry personnel play the role of the user, doing all essential tests in accordance with the test cases, as well as a few proposed test cases to find additional faults or errors.

### 6.2.2    Test Environment

All testing is done in the development environment, including unit testing, integration testing, and system testing. As a result, the hardware specifications and firmware settings of personal computers are identical to those used in development environments. It is explained in Chapter 2.4.2. Before deploying to Heroku's cloud server, all testing is done in a LAN-based environment with access to the internet.

### 6.2.3    Test Schedule

All testing activities are scheduled to be finished between 19th of July until 27th of August 2021, according to the test plan. Table 6.1 illustrates the test schedule, which shows when each primary module and sub-module of the system will be tested. All of the testing activities on the first day consist of running all of the test cases at least once, with some test cases being run at least twice to clarify the detected bugs. Once the patch is complete, all of the test cases that failed on the first day should be performed on the second testing day, which is also known as regression testing.

Table 6.1: Test Schedule

| No. | Module | Duration | Schedule |
|-----|--------|----------|----------|
| 1. | Reseller Module | 13 days | 19/7 – 1/8 |
| 2. | Customer Module | 12 days | 1/8 – 13/8 |
| 3. | Stock Management Module | 2 days | 13/8 – 15/8 |
| 4. | Tracker Module | 6 days | 15/8 – 21/8 |
| 5. | Distributor Module | 6 days | 21/8 – 27/8 |

### 6.3    Test Strategy

The test strategy used during the testing phase is discussed in this section. A test strategy combines many types of test case design methodologies to build a well-

structured series of stages that ensure software testing success. Testing experts, such as testing specialists, software engineers, and project managers, are often in charge of developing test strategies.

A **Bottom-up** method with **Black-Box Test** classes and a **Big Bang Approach** was chosen as the test technique. Bottom-up testing begins with the most fundamental component testing, such as unintegrated submodule testing, and advances to main module testing. The testing phase next comprises integration and system testing. Unit testing is the most fundamental form of testing. It examines the smallest unit of the system, which could be a module or component. Because unit testing can contribute to system testing and is required by test strategy, the smallest module, which can be the lowest collection of lines of code, is the unit of each unit testing. A **White-Box Test** is a term used to describe unit testing.

Integration testing is carried out after more than one tested unit has been combined to form a larger structure. Integration testing is usually focused on the interfaces between components and modules. If the quality attribute of components cannot be assessed effectively, integration testing is performed on the broader structure that is being combined by more than one component or module. System testing aims to do an end-to-end quality check of the entire system. This testing is done in accordance with the system's functional and required specifications. Non-functional quality attributes, such as performance, reliability, and usability, are also assessed during system testing (Chauhan & Singh, 2014).

Black-box testing is a technique used by testers to check the functionality of a component without knowing how the component's internal workings work (Khan & Khan, 2014). The primary purpose of black-box testing is to determine how well a system meets its given requirements. Black box testing does not demand complete understanding or comprehension of the system's core logical structure because testers just test the relevant features of the system. The black-box testing determines whether all inputs and outputs are accepted and produced correctly (Chauhan & Singh, 2014).

### 6.3.1 Classes of Tests

Following the acceptance of a set of suitable input, a functionality test is run to see if the described system features are working as intended and producing correct output. All of the functionality-related test cases are explained in the next section, Chapter 6.4. Web service sessions are subjected to security testing and identifying an id for a procedure that involves critical and sensitive data will be encrypted to ensure its integrity, including one-way encryption of account passwords.

## 6.4 Test Design

Every module's test case is described in this portion of the test description. It also includes the test data that was utilized to execute the test cases.

### 6.4.1 Test Description

This section designs and documents test case identification, use case testing, test cases, and expected results for each module.

### 6.4.2 Use Case Testing (Black Box Testing)

Use Case Testing is a software testing technique that aids in the identification of test cases that span the full system from beginning to end on a transaction-by-transaction basis. Interactions between users and software applications are called test cases. Use case testing identifies holes in software applications that would otherwise go undetected if individual software components were tested. This testing consists of the following modules: login, register, manage products (insert new product), manage job (insert new job), request shipment (insert shipment cart), manage product serial number (insert new serial number), shop (filter product), e-track parcel, manage profile, manage address (insert new address), manage cart, e-RMA and manage repair.

### 6.4.2.1 Use Case Testing: Login Module

Table 6.2: UCT: Login Module

| Main Success Scenario | Step | Description |
|---|---|---|
| A: Actor | 1 | A: Enter Email & Password |
| S: System | 2 | S: Validate Email |
| | 3 | S: Validate Password |
| | 4 | S: Allow Account Access |
| Extensions | 2a | **Email not valid** <br> S: These credentials do not match our records. |
| | 3a | **Password not valid** <br> S: These credentials do not match our records. |

### 6.4.2.2 Use Case Testing: Register Module

Table 6.3: UCT: Register Module

| Main Success Scenario | Step | Description |
|---|---|---|
| A: Actor | 1 | A: Enter Name, Email, Password & Confirm Password |
| S: System | 2 | S: Validate Name |
| | 3 | S: Validate Email |
| | 4 | S: Validate Password |
| | 5 | S: Validate Confirm Password |
| | 6 | S: Allow Account Access & Creation |
| Extensions | 2a | **Name not valid** <br> S: The name field is required. |
| | 3a | **Email not valid** <br> S: The name field is required. |
| | 3b | **Email incorrect format** <br> S: Please include an @ in the email address. |
| | 3c | **Email not complete** <br> S: Please enter a part following @. Email is incomplete. |

| | 4a | **Password not valid** |
| | | S: The name field is required. |
| | 4b | **Password length less than 8** |
| | | S: The password must be at least 8 characters. |
| | 5a | **Confirm password does not tele with password** |
| | | S: The password confirmation does not match. |

### 6.4.2.3 Use Case Testing: Manage Products (Insert New Product) Module

Table 6.4: UCT: Manage Products (Insert New Product) Module

| Main Success Scenario | Step | Description |
|---|---|---|
| A: Actor <br> S: System | 1 | A: Enter Product Name, Product No, Product Price, Product Link, Image File, Brand Distributor (**Reseller**), Warranty Duration, Product Category, Product Brand & Product Stock. |
| | 2 | S: Validate Product Name |
| | 3 | S: Validate Product No |
| | 4 | S: Validate Product Price |
| | 5 | S: Validate Product Link |
| | 6 | S: Validate Image File |
| | 7 | S: Validate Brand Distributor (Reseller) |
| | 8 | S: Validate Warranty Duration |
| | 9 | S: Validate Product Category |
| | 10 | S: Validate Product Brand |
| | 11 | S: Validate Product Stock |
| | 12 | S: Allow New Product Creation |
| Extensions | 2a | **Product Name not valid** |
| | | S: Please enter product name. |
| | 3a | **Product No. not valid** |
| | | S: Please enter serial number. |

| | 4a | **Product Price not valid** |
|---|---|---|
| | | S: This field is required. |
| | 5a | **Product Link not valid** |
| | | S: This field is required. |
| | 6a | **Product Image File not valid** |
| | | S: This field is required. |
| | 7a | **Brand Distributor (Reseller) not valid** |
| | | S: This field is required. |
| | 8a | **Warranty Duration not valid** |
| | | S: This field is required. |
| | 9a | **Product Category not valid** |
| | | S: This field is required. |
| | 10a | **Product Brand not valid** |
| | | S: This field is required. |
| | 11a | **Product Stock not valid** |
| | | S: This field is required. |

### 6.4.2.4 Use Case Testing: Manage Job (Insert New Job) Module

Table 6.5: UCT: Manage Job (Insert New Job) Module

| Main Success Scenario | Step | Description |
|---|---|---|
| A: Actor<br>S: System | 1 | A: Enter Job Name, Job Salary, Job Type & Job Location |
| | 2 | S: Validate Job Name |
| | 3 | S: Validate Job Salary |
| | 4 | S: Validate Job Type |
| | 5 | S: Validate Job Location |
| | 6 | S: Allow New Job Creation |
| Extensions | 2a | **Job Name not valid**<br>S: Please input all fields with relevant information. |
| | 3a | **Job Salary not valid** |

| | | S: Please input all fields with relevant information. |
|---|---|---|
| | 4a | **Job Type not valid**<br>S: Please input all fields with relevant information. |
| | 5a | **Job Location not valid**<br>S: Please input all fields with relevant information. |

### 6.4.2.5 Use Case Testing: Request Shipment (Insert in Shipment Cart) Module

Table 6.6: UCT: Request Shipment (Insert in Shipment Cart) Module

| Main Success Scenario | Step | Description |
|---|---|---|
| A: Actor<br>S: System | 1 | A: Enter Brand Distributor, Product Brand, Product & Product Quantity. |
| | 2 | S: Validate Brand Distributor |
| | 3 | S: Validate Brand |
| | 4 | S: Validate Product |
| | 5 | S: Validate Product Quantity |
| | 6 | S: Allow Insert Product in Shipment Cart |
| Extensions | 2a | **Brand Distributor not valid**<br>S: Please input all fields with relevant information. |
| | 3a | **Product Brand not valid**<br>S: Please input all fields with relevant information. |
| | 4a | **Product not valid**<br>S: Please input all fields with relevant information. |
| | 5a | **Product Quantity**<br>S: Please input all fields with relevant information. |

**6.4.2.6 Use Case Testing: Manage Product Serial Number (Insert New Serial Number) Module**

Table 6.7: UCT: Manage Product Serial Number (Insert New Serial Number) Module

| Main Success Scenario | Step | Description |
|---|---|---|
| A: Actor<br>S: System | 1 | A: Enter Product, Batch & Product Serial Number |
| | 2 | S: Validate Product |
| | 3 | S: Validate Batch |
| | 4 | S: Validate Product Serial Number |
| | 5 | S: Allow Insert New Product Serial Number |
| Extensions | 2a | **Product not valid**<br>S: Please input all fields with relevant information. |
| | 3a | **Batch not valid**<br>S: Please input all fields with relevant information. |
| | 4a | **Product Serial Number not valid**<br>S: Please input all fields with relevant information. |

**6.4.2.7 Use Case Testing: Shop (Filter Product) Module**

Table 6.8: UCT: Shop (Filter Product) Module

| Main Success Scenario | Step | Description |
|---|---|---|
| A: Actor<br>S: System | 1 | A: Enter Brand, Category & Price |
| | 2 | S: Validate Brand |
| | 3 | S: Validate Category |
| | 4 | S: Validate Price |
| | 5 | S: Allow Product Filtration |
| Extensions | 2a | **Brand not valid**<br>S: Please input brand or category or price. |

| | 3a | **Category not valid** |
|---|---|---|
| | | S: Please input brand or category or price. |
| | 4a | **Price not valid** |
| | | S: Please input brand or category or price. |

### 6.4.2.8  Use Case Testing: e-Track Parcel Module

Table 6.9: UCT: e-Track Parcel Module

| Main Success Scenario | Step | Description |
|---|---|---|
| A: Actor | 1 | A: Enter Tracking Number |
| S: System | 2 | S: Validate Tracking Number |
| | 3 | S: Allow Searching for Current Tracking Status |
| Extensions | 2a | **Tracking Number not valid** |
| | | S: Please input tracking number. |

### 6.4.2.9  Use Case Testing: Manage Profile Module

Table 6.10: UCT: Manage Profile Module

| Main Success Scenario | Step | Description |
|---|---|---|
| A: Actor | 1 | A: Enter Name & Email |
| S: System | 2 | S: Validate Name |
| | 3 | S: Validate Email |
| | 4 | S: Allow Update Profile Information |
| Extensions | 2a | **Name not valid or same** |
| | | S: Please make some changes to the input to make changes. |
| | 3a | **Email not valid or same** |
| | | S: Please make some changes to the input to make changes. |
| | 3b | **Email incorrect format** |
| | | S: Please include an @ in the email address. |

| | 3c | **Email not complete** |
| | | S: Please enter a part following @. Email is incomplete. |

### 6.4.2.10 Use Case Testing: Manage Address (Insert New Address) Module

Table 6.11: UCT: Manage Address (Insert New Address) Module

| Main Success Scenario | Step | Description |
|---|---|---|
| A: Actor<br>S: System | 1 | A: Enter Name, Address, Phone No., Postcode, Longitude & Latitude |
| | 2 | S: Validate Name |
| | 3 | S: Validate Address |
| | 4 | S: Validate Phone No. |
| | 5 | S: Validate Postcode |
| | 6 | S: Validate Longitude |
| | 7 | S: Validate Latitude |
| | 8 | S: Allow Insert New Address |
| Extensions | 2a | **Name not valid**<br>S: Please enter name. |
| | 3a | **Address not valid**<br>S: Enter address. |
| | 4a | **Phone No. not valid**<br>S: Please enter phone number. |
| | 4b | **Phone No. length more than 11**<br>S: Please enter no more than 11 characters. |
| | 5a | **Postcode not valid**<br>S: This field is required / Please enter a valid number. |
| | 5b | **Postcode length more than 6**<br>S: Please enter no more than 6 characters. |
| | 6a | **Longitude not valid**<br>S: This field is required. |

| | 7a | **Latitude not valid** |
| | | S: This field is required. |

### 6.4.2.11 Use Case Testing: Manage Cart Module

Table 6.12: UCT: Manage Cart Module

| Main Success Scenario | Step | Description |
|---|---|---|
| A: Actor | 1 | A: Enter Quantity |
| S: System | 2 | S: Validate Quantity |
| | 3 | S: Allow User to Place Order |
| Extensions | 2a | **Quantity not valid** |
| | | S: Stock is around {total stock} only. |

### 6.4.2.12 Use Case Testing: e-RMA Module

Table 6.13: UCT: e-RMA Module

| Main Success Scenario | Step | Description |
|---|---|---|
| A: Actor | 1 | A: Enter Address, Date of Purchase, Proof of Purchase, Brand, Product, Serial Number, Confirm Serial Number & Comments |
| S: System | 2 | S: Validate Address |
| | 3 | S: Validate Date of Purchase |
| | 4 | S: Validate Proof of Purchase |
| | 5 | S: Validate Brand |
| | 6 | S: Validate Product |
| | 7 | S: Validate Serial Number |
| | 8 | S: Validate Confirm Serial Number |
| | 9 | S: Validate Comments |
| | 10 | S: Allow User to Submit RMA Request |
| Extensions | 2a | **Address not valid** |
| | | S: Please input all fields with relevant information. |

| | 3a | **Date of Purchase not valid** |
|---|---|---|
| | | S: Please fill out this field. |
| | 4a | **Proof of Purchase not valid** |
| | | S: Please attach proof of purchase in PDF form. |
| | 5a | **Brand not valid** |
| | | S: Please input all fields with relevant information. |
| | 6a | **Product not valid** |
| | | S: Please input all fields with relevant information. |
| | 7a | **Serial Number not valid** |
| | | S: Please enter serial number. |
| | 8a | **Confirm Serial Number not valid** |
| | | S: Please enter serial number confirmation. |
| | 8b | **Confirm Serial Number not tele with Serial Number** |
| | | S: SN does not match ! |
| | 9a | **Comments not valid** |
| | | S: Please enter reason. |

### 6.4.2.13 Use Case Testing: Manage Repair Module

Table 6.14: UCT: Manage Repair Module

| Main Success Scenario | Step | Description |
|---|---|---|
| A: Actor<br>S: System | 1 | A: Enter Update Status, Receive At, Tracking No. & Remark |
| | 2 | S: Validate Tracking No. |
| | 3 | S: Validate Remark |
| | 4 | S: Allow to Alter Repair Information |
| Extensions | 2a | **Tracking No, not valid**<br>S: Please input all fields with relevant information for Tracking No. and Remark. |

| | 3a | **Remark not valid** |
| | | S: Please input all fields with relevant information for Tracking No. and Remark. |

### 6.4.3    Test Case

Each test case is shown in this section. References are provided as each test case will be referenced to an appendix.

Table 6.15: Test Case

| Test Case (Module) | Reference | Page |
|---|---|---|
| Login | **Appendix A** | **1** |
| Register | **Appendix B** | **2 – 4** |
| Manage Products | **Appendix C** | **4 – 8** |
| Manage Job | **Appendix D** | **8 – 9** |
| Request Shipment | **Appendix E** | **9 – 11** |
| Manage Product Serial Number | **Appendix F** | **12 – 13** |
| Shop | **Appendix G** | **13 – 14** |
| e-Track Parcel | **Appendix H** | **14 – 15** |
| Manage Profile | **Appendix I** | **15 – 16** |
| Manage Address | **Appendix J** | **16 – 19** |
| Manage Cart | **Appendix K** | **19 – 21** |
| e-RMA | **Appendix L** | **21 – 23** |
| Manage Repair | **Appendix M** | **24 – 25** |
| Manage Users | **Appendix N** | **26** |
| Manage Orders | **Appendix O** | **27 – 29** |

### 6.4.4 Test Data

Each test case's test data is shown in this section. Tick signifies that the obtained real result of a test case matches the predicted outcome in the following sections.

Table 6.16: Test Data

| Test Case (Module) | Reference | Page |
|---|---|---|
| Login | **Appendix A** | **1** |
| Register | **Appendix B** | **2 – 4** |
| Manage Products | **Appendix C** | **4 – 8** |
| Manage Job | **Appendix D** | **8 – 9** |
| Request Shipment | **Appendix E** | **9 – 11** |
| Manage Product Serial Number | **Appendix F** | **12 – 13** |
| Shop | **Appendix G** | **13 – 14** |
| e-Track Parcel | **Appendix H** | **14 – 15** |
| Manage Profile | **Appendix I** | **16** |
| Manage Address | **Appendix J** | **17 – 19** |
| Manage Cart | **Appendix K** | **20 – 21** |
| e-RMA | **Appendix L** | **22 – 24** |
| Manage Repair | **Appendix M** | **25 – 26** |
| Manage Users | **Appendix N** | **27** |
| Manage Orders | **Appendix O** | **28 – 30** |

### 6.4.5 Integration Testing (Big Bang Approach)

Integration testing is a sort of testing in which software elements are logically linked and tested as a unit. A typical software project is made up of several software modules written by various programmers. The goal of this level of testing is to find flaws in how these software modules interact when they are put together. This testing consists of the following modules: login, register, manage products, manage job, request shipment, manage product serial number, shop, e-track parcel, manage cart, e-RMA and manage orders.

### 6.4.5.1 Integration Testing: Login Module

Table 6.17: Integration Testing: Login Module

| Test Case ID | Test Case Objective | Test Case Description | Expected Result |
|---|---|---|---|
| LOGINT1 | Check the interface link between Login and User Dashboard module. | Enter login credentials and click on login button. | To be redirected to the user dashboard. |
| LOGINT2 | Check the interface link between Login and Forgot Password module. | Click on "Reset Here" link. | To be redirected to the forgot password page. |

### 6.4.5.2 Integration Testing: Register Module

Table 6.18: Integration Testing: Register Module

| Test Case ID | Test Case Objective | Test Case Description | Expected Result |
|---|---|---|---|
| REGINT1 | Check the interface link between Register and User Dashboard module. | Enter all fields and click on submit button. | To be redirected to the user dashboard and registered as a new user. |

### 6.4.5.3 Integration Testing: Manage Products Module

Table 6.19: Integration Testing: Manage Products Module

| Test Case ID | Test Case Objective | Test Case Description | Expected Result |
|---|---|---|---|
| MPINT1 | Check the interface link between Manage Products and Manage Product (Insert New Product) module. | Enter all fields and click on submit button. | To be redirected to the manage product page and new product should appear in list of products. |

#### 6.4.5.4 Integration Testing: Manage Job Module

Table 6.20: Integration Testing: Manage Job Module

| Test Case ID | Test Case Objective | Test Case Description | Expected Result |
|---|---|---|---|
| MJINT1 | Check the interface link between Manage Job and Manage Job (Insert Job) module. | Enter all fields and click on submit button. | To be redirected to the manage job page and new job should appear in list of jobs available. |

#### 6.4.5.5 Integration Testing: Request Shipment Module

Table 6.21: Integration Testing: Request Shipment Module

| Test Case ID | Test Case Objective | Test Case Description | Expected Result |
|---|---|---|---|
| MRSINT1 | Check the interface link between Request Shipment and Manage Shipment module. | Enter all fields and click on add to list button. | To be redirected to the manage shipment page and new shipment is created according to its current status. |

#### 6.4.5.6 Integration Testing: Manage Product Serial Number Module

Table 6.22: Integration Testing: Manage Product Serial Number Module

| Test Case ID | Test Case Objective | Test Case Description | Expected Result |
|---|---|---|---|
| MPSNINT1 | Check the interface link between Manage Product Serial Number and Manage Product Serial Number (Insert New Product Serial Number) module. | Enter all fields and click on insert button. | To be redirected to the manage product serial number page and new product serial number is created. |

### 6.4.5.7 Integration Testing: Shop Module

Table 6.23: Integration Testing: Shop Module

| Test Case ID | Test Case Objective | Test Case Description | Expected Result |
|---|---|---|---|
| SINT1 | Check the interface link between Shop and Manage Cart module. | Click on add to cart button at the desired product. | To be redirected to the manage cart page and selected product is added to cart. |

### 6.4.5.8 Integration Testing: e-Track Parcel Module

Table 6.24: Integration Testing: e-Track Parcel Module

| Test Case ID | Test Case Objective | Test Case Description | Expected Result |
|---|---|---|---|
| eTPINT1 | Check the interface link between e-Track Parcel and Input Tracking Number module. | Fill in tracking number and click on search button. | To be redirected to the result of current tracking status page. |

### 6.4.5.9 Integration Testing: Manage Cart Module

Table 6.25: Integration Testing: Manage Cart Module

| Test Case ID | Test Case Objective | Test Case Description | Expected Result |
|---|---|---|---|
| MCRTINT1 | Check the interface link between Manage Cart and Manage Cart (Payment) module. | Click on PayPal button. | To be redirected to the thank you page after payment is success and order is created. |

**6.4.5.10 Integration Testing: e-RMA Module**

Table 6.26: Integration Testing: e-RMA Module

| Test Case ID | Test Case Objective | Test Case Description | Expected Result |
|---|---|---|---|
| eRMAINT1 | Check the interface link between e-RMA and Manage Repair module. | Enter all fields and click on submit button. | Should appear at user dashboard page in "My RMA Requests" section. |

**6.4.5.11 Integration Testing: Manage Orders Module**

Table 6.27: Integration Testing: Manage Orders Module

| Test Case ID | Test Case Objective | Test Case Description | Expected Result |
|---|---|---|---|
| MORDINT1 | Check the interface link between Manage Orders and e-Track Parcel module. | Click on track icon. | To be redirected to the parcel tracking view. |
| MORDINT2 | Check the interface link between Manage Orders and Order Details module. | Click on eye icon. | To be redirected to the order detail's view. |
| MORDINT3 | Check the interface link between Manage Orders and e-Receipt module. | Click on receipt icon. | To be redirected to the e-Receipt view. |
| MORDINT4 | Check the interface link between Manage Orders and Manage Orders (Insert Product SN) module. | Click on arrow up icon. | To be redirected to the insert product serial number for an order view. |
| MORDINT5 | Check the interface link between Manage Orders and Manage | Click on printer icon. | To be redirected to the new |

| | Orders (Generate Airway Bill) module. | | generated airway bill view. |
|---|---|---|---|
| MORDINT6 | Check the interface link between Manage Orders and Manage Orders (Cancel Order) module. | Click on red cross icon. | Selected order should appear in the Cancelled section. |

## 6.5    Test Results and Analysis

According to the results of the test cases in Chapter 6.4, more than 90% of the test cases passed with the intended output being produced appropriately in testing phase 1. While, in testing phase 2 all of the test cases passed 100% with expected output. Users were actively involved in the testing process by working with testers to complete all test cases. Overall, the user is pleased with the completed project because the majority of the test cases return good results, indicating that the defined requirements have been effectively implemented and met in this project.

### 6.5.1    Testing Result Phase 1 for Test Case

Table 6.28: Testing Result Phase 1

| Test Case ID | Pass/Fail | Remark | Tested By |
|---|---|---|---|
| **Login Module** | | | |
| LOGIN1 | PASS | System working as intended with expected output. | Zulazri Zulkarnain |
| LOGIN2 | PASS | | |
| **Register Module** | | | |
| REG1 | PASS | System working as intended with expected output. | Akmal Khairi |
| REG2 | PASS | | |
| REG3 | PASS | | |
| REG4 | PASS | | |
| **Manage Products Module** | | | |
| PROD1 | PASS | System working as intended with expected output. | Akmal Sabri |
| PROD2 | PASS | | |

| PROD3 | PASS | | |
|---|---|---|---|
| PROD4 | PASS | | |
| **Manage Job Module** | | | |
| JOB1 | PASS | System working as intended with expected output. | Zulazri Zulkarnain |
| JOB2 | PASS | | |
| **Request Shipment Module** | | | |
| SHIP1 | PASS | System working as intended with expected output. | Akmal Sabri |
| SHIP2 | PASS | | |
| SHIP3 | PASS | | |
| SHIP4 | PASS | | |
| **Manage Product Serial Number Module** | | | |
| SN1 | PASS | System working as intended with expected output. | Adi Iman |
| SN2 | PASS | | |
| **Shop Module** | | | |
| SHOP1 | PASS | System working as intended with expected output. | Akmal Khairi |
| SHOP2 | PASS | | |
| SHOP3 | PASS | | |
| **e-Track Parcel Module** | | | |
| TRACK1 | PASS | System working as intended with expected output. | Akmal Sabri |
| TRACK2 | PASS | | |
| TRACK3 | PASS | | |
| **Manage Profile Module** | | | |
| UP1 | PASS | System working as intended with expected output. | Zulazri Zulkarnain |
| UP2 | PASS | | |
| **Manage Address** | | | |
| ADDRESS1 | PASS | System working as intended with expected output. | Akmal Khairi |
| ADDRESS2 | PASS | | |
| ADDRESS3 | PASS | | |
| ADDRESS4 | PASS | | |
| **Manage Cart Module** | | | |
| CART1 | FAIL | Product is not deleted; the URL is added with PayPal parameters | Adi Iman |

| | | which leads to invalid product deletion. | |
|---|---|---|---|
| CART2 | PASS | System working as intended with expected output. | |
| CART3 | PASS | | |
| CART4 | FAIL | PayPal does not capture recipient address, subtotal, and all of the products for a current order. | |
| **e-RMA Module** | | | |
| RMA1 | PASS | System working as intended with expected output. | Zulazri Zulkarnain |
| RMA2 | PASS | | |
| **Manage Repair Module** | | | |
| REPAIR1 | PASS | System working as intended with expected output. | Adi Iman |
| REPAIR2 | PASS | | |
| REPAIR3 | PASS | | |
| **Manage Users Module** | | | |
| USR1 | PASS | System working as intended with expected output. | Akmal Khairi |
| USR2 | PASS | | |
| USR3 | PASS | | |
| **Manage Orders Module** | | | |
| ORDER1 | FAIL | System does not give the expected result output. Tracking number is not updated. | Akmal Sabri |
| ORDER2 | FAIL | System still updates even though data is null. | |
| ORDER3 | PASS | System working as intended with expected output. | |
| ORDER4 | PASS | | |
| ORDER5 | PASS | | |
| ORDER6 | PASS | | |
| ORDER7 | PASS | | |

## PASS OR FAIL PERCENTAGE TESTING PHASE 1
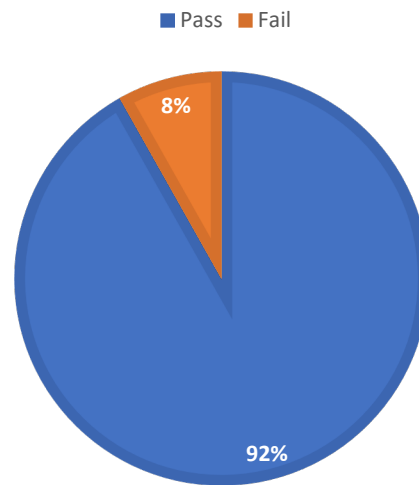
Pass ■ Fail



Figure 6.2: Pie-Chart for Testing Phase 1
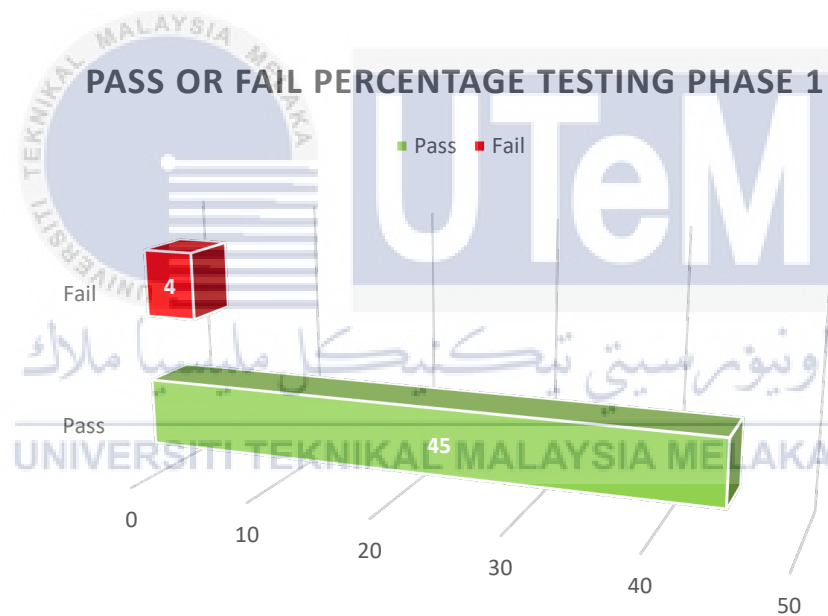


Figure 6.3: Bar-Chart for Testing Phase 1

### 6.5.2 Testing Result Phase 2 for Test Case

Table 6.29: Testing Result Phase 2

| Test Case ID | Pass/Fail | Remark | Tested By |
|---|---|---|---|
| **Manage Cart Module** | | | |
| CART1 | ~~FAIL~~ PASS | System working as intended with | Adi Iman |
| CART4 | ~~FAIL~~ PASS | expected output. | |

| Manage Orders Module | | | |
|---|---|---|---|
| ORDER1 | ~~FAIL~~ PASS | System working as intended with | Akmal Sabri |
| ORDER2 | ~~FAIL~~ PASS | expected output. | |

**PASS OR FAIL PERCENTAGE TESTING PHASE 2**
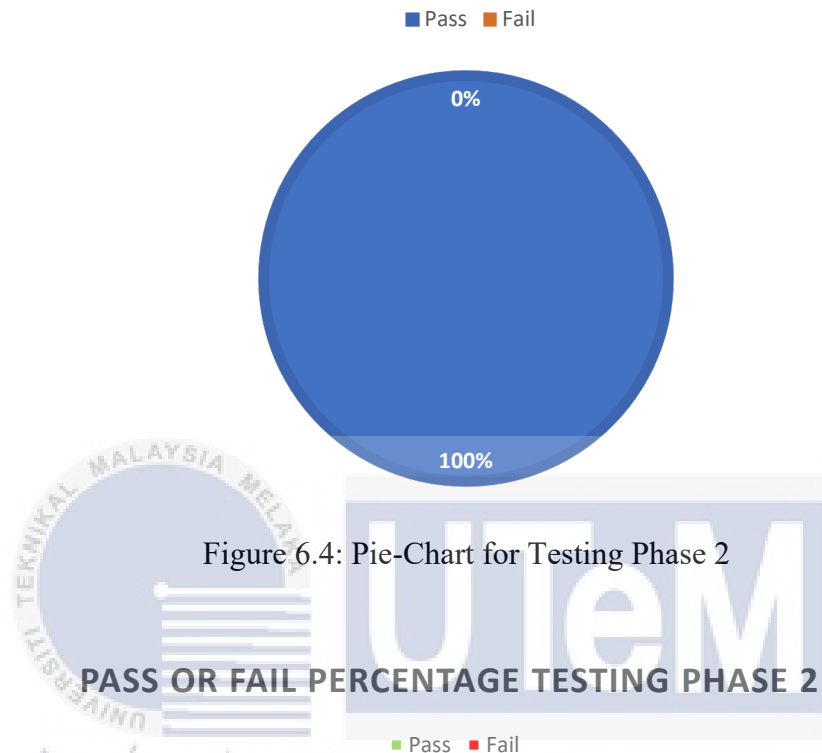


Figure 6.4: Pie-Chart for Testing Phase 2
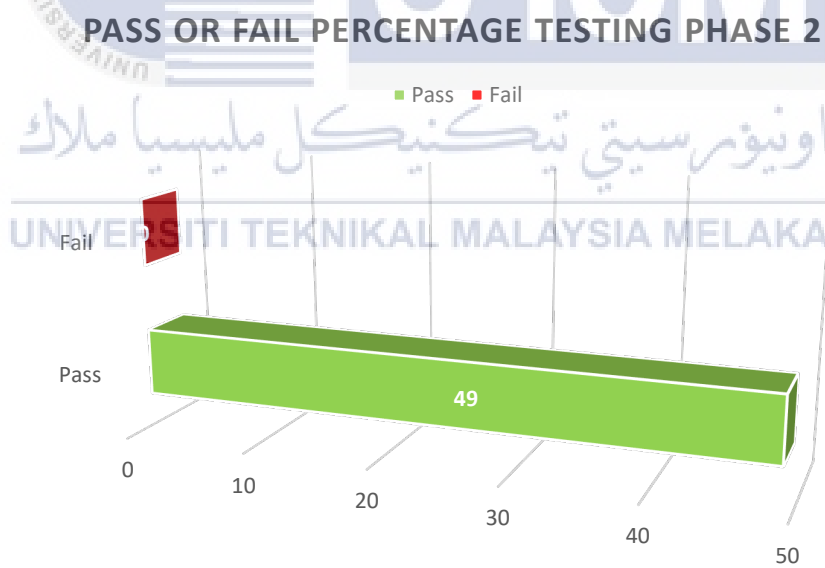


Figure 6.5: Bar-Chart for Testing Phase 2

### 6.5.3 Integration Testing Result

Table 6.30: Integration Testing Results

| Test Case ID | Pass/Fail | Remark | Tested By |
|:---:|:---:|:---|:---:|
| LOGINT1 | PASS | | Ismat Azmy |
| LOGINT2 | PASS | | |
| REGINT1 | PASS | | |
| MPINT1 | PASS | | |
| MJINT1 | PASS | | |
| MRSINT1 | PASS | | |
| MPSNINT1 | PASS | System working as intended with expected result. All of the process is smoothly executed without any hiccups. | |
| SINT1 | PASS | | Izzati Suhaimi |
| eTPINT1 | PASS | | |
| MCRTINT1 | PASS | | |
| eRMAINT1 | PASS | | |
| MORDINT1 | PASS | | |
| MORDINT2 | PASS | | |
| MORDINT3 | PASS | | |
| MORDINT4 | PASS | | |
| MORDINT5 | PASS | | |
| MORDINT6 | PASS | | |

### 6.6 Conclusion

This chapter outlines and discusses the various testing phases and tests that are carried out to evaluate and validate the system's e-Techify module upgrades. The next chapter presents the project's conclusion, as well as the project's prospective improvements and contributions.

# CHAPTER 7:  CONCLUSION

## 7.1     Observation on Weaknesses and Strengths

One of the advantages of the e-Techify system is that customers may pay with confidence. All payments will be processed through a secure transaction managed by PayPal. PayPal's payment flexibility makes it simple for anyone with a Visa, MasterCard, or American Express card to make a payment. Regardless matter whether you use a credit or debit card, I believe money should be easy to obtain and spend. That's why PayPal's services are tailored to fit into our daily routines in order to make things as simple as possible. The e-Techify system provides these services to customers in order to attract more customers, increase revenue, and make every transaction as smooth as butter.

Second, this project's development includes additional features such as the ability to view all of the data in various forms of interactive graphs. This will allow resellers and distributors to see or see data in a more organized manner. Every process that runs through a function that uses graphs will immediately update the graph to reflect the most recent data and summarize a large data set in visual form without any hitches. Each data is distributed according to their category and frequency distribution.

Third, the graphical user interface has been modified to reflect modern web technologies, including responsive design and compatibility for many types of screen displays. Before being displayed to the user, the designs, photos, and phrases will be changed to fit the screen resolution. Aside from allowing display on various screens, the website's entire appearance has been updated, with the font and backdrop modified to make them more readable and pleasant to look at. Every element is designed to be exactly where it should be, and it is also user friendly.

This enhancement also includes a few minor enhancements to the existing functions. When a reseller or distributor adds new data, for example, input validation and user input checking are introduced. When a user attempts to add existing data to the system, a validation check is performed to ensure that the data request is valid. All of these validation capabilities help users avoid making mistakes while inputting data into the system, avoiding data redundancy and data management errors.

Despite the development introduces few strength, it also consists of some weaknesses. One of the major weakness is, e-Techify system is not designed for mobile. The user interface is not responsive for a small resolution screen which could give a hard time for user during their surfing time to the system.

Finally, once public access to the system is granted, the system's security must be considerably enhanced and maintained. The existing security measures in place in the e-Techify system are insufficient to protect it against hacking or hostile intruder attacks. More techniques should be used to validate each user's identification, with 2-Step Verification, often known as two-factor authentication, being a good example. This method adds an additional degree of security to the user account, allowing attackers to be kept out even if they know the user password. The security aspect, which ensures the integrity, availability, and confidentiality of all data, must be given top attention in future development.

The development of this system should not cease with the completion of this project; rather, it should continue in the future to guarantee that the system's strengths are preserved, and its shortcomings are enhanced or at the very least mitigated.

## 7.2    Propositions for Improvement

I have a few ideas for improvements that I'd like to enlighten here. First and foremost, because this is a multi-platform system with multiple sorts of systems, it requires its own mobile application. In order to make it easier for users to utilize either the mobile or online platform, both the mobile platform and the web platform must have the same capabilities.

In addition, one of the flaws stated in the previous section is security, which must be addressed and enhanced in order for the system to run properly and accurately while also protecting data from unwanted access.

## 7.3    Project Contribution

The project's main contribution is to Xmiryna Technology by supplying all of the knowledge about all of the processes that are required to automate the present system from start to finish. Each procedure is described in detail, providing a clearer picture of how to construct e-Techify. However, with the new implementation of e-Techify it can provide lots of benefits for each users of the system.

This project also benefits the environment by reducing the reliance on paper through the use of digital forms. Receipts and order information can be created as softcopy, such as photos or webpages, and delivered across multiple platforms and devices. As a result, this initiative can not only assist save money but also conserve the environment by reducing paper usage.

## 7.4    Conclusion

This project complied with all of Xmiryna Technology's criteria, including the addition of modules to enhance the current system. This project was a success, and it should be further extended and enhanced in the future. I am grateful because this project has the potential to advance the automation of the current system to the point where every individual or user involved would benefit from the system's functions.

# REFERENCES

- Ramadan, M. (n.d.). The web application hacker's handbook. Web Application Hackers Handbook. Retrieved June 24, 2021, from https://www.academia.edu/40654302/The_web_application_hackers_handbook

- Chauhan, R. K., & Singh, I. (n.d.). Latest Research and Development on Software Testing Techniques and Tools - Inpressco. Latest Research and Development on Software Testing Techniques and Tools. Retrieved July 28, 2021, from http://inpressco.com/latest-research-and-development-on-software-testing-techniques-and-tools/

- Khan, M., & Khan, F. (n.d.). Importance of Software Testing in Software Development Life Cycle. Retrieved July 28, 2021, from http://ijcsi.org/papers/IJCSI-11-2-2-120-123.pdf

- Stress Testing | Software Testing. (2019, April 18). GeeksforGeeks. https://www.geeksforgeeks.org/stress-testing-software-testing/

- Software Testing | Security Testing - GeeksforGeeks. (2019, May 10). GeeksforGeeks. https://www.geeksforgeeks.org/software-testing-security-testing/

- Step 1: Create your first S3 bucket - Amazon Simple Storage Service. (n.d.). Create Your First S3 Bucket. Retrieved August 4, 2021, from https://docs.aws.amazon.com/AmazonS3/latest/userguide/creating-bucket.html

**APPENDICES**

**APPENDIX A:** TC: Login Module

| Test Case ID | Test Case | Pre-Requisite | Test Step | Test Data | Expected Result | Status (P/F) |
|---|---|---|---|---|---|---|
| LOGIN1 | Users' login with valid credentials. | User must be registered. | 1. Go to site https://etechify.herokuapp.com/login 2. Enter email and password. 3. Click login button | Email address: **ismatazmy@live.com** Password: **12345678** | Users should be able to login into system and redirect to user dashboard. | P |
| LOGIN2 | Users' login with invalid credentials. | User is not registered. | 1. Go to site https://etechify.herokuapp.com/login 2. Enter email and password. 3. Click login button | Email address: **ismat@live.com** Password: **123456789abcd123** | Users should not be able to login into system and an error "These credentials do not match our records." | P |

**APPENDIX B:** TC: Register Module

| Test Case ID | Test Case | Pre-Requisite | Test Step | Test Data | Expected Result | Status (P/F) |
|---|---|---|---|---|---|---|
| REG1 | All fields are filled with correct format. | User must fill in all the blanks or fields with correct format. | 1. Go to site https://etechify.herokuapp.com/register <br> 2. Enter name, email, password and confirm password. <br> 3. Click submit button | Name: **Ismat Azmy** <br> Email address: **ismatazmy@live.com** <br> Password: **12345678** <br> Confirm Password: **12345678** | User account will be successfully registered and will redirected to user dashboard. | P |
| REG2 | Not all fields are filled or not in correct format. | User does not fill in all the blanks or fields with correct format. | 1. Go to site https://etechify.herokuapp.com/ register <br> 2. Fields are left empty or not in correct format such | Name: **-** <br> Email address: **ismatazmy@live.com** <br> Password: | Account registration will be rejected, and errors will appear "The name/email/passwor | P |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | as name, email, password and confirm password. <br> 3. Click submit button | **12345678** <br> Confirm Password: <br> **12345** | d/confirm password is required." | |
| REG3 | Email is existed in the system. | Email have already been registered previously with an account in the system. | 1. Go to site https://etechify.herokuapp.com/ register <br> 2. Enter name, email, password and confirm password. <br> 3. Click submit button | Name: <br> **Wan Ismat** <br> Email address: <br> **admin@etechify.com** <br> Password: <br> **12345678** <br> Confirm Password: <br> **12345678** | An error will appear "The email has already been taken." and account registration is rejected. | P |
| REG4 | Incorrect format for email and password length. | User input email in a wrongly format. | 1. Go to site https://etechify.herokuapp.com/register <br> 2. Enter email without @ or with @ but without any | Name: <br> **Ismat Azmy** <br> Email address: <br> **ismatazmy@live.com** | Errors for email will appear "Please include an @ in the email address." and "Please enter a part | P |

| | | | | Password: **12345** Confirm Password: **12345** | following @". While errors for password and confirm password will appear "The password must be at least 8 characters." | |

*(continued)*: connection, password, confirm password length less than 8 and name. 3. Click submit button.

**APPENDIX C:** TC: Manage Products Module

| Test Case ID | Test Case | Pre-Requisite | Test Step | Test Data | Expected Result | Status (P/F) |
|---|---|---|---|---|---|---|
| PROD1 | All fields are filled with correct format for insert new product. | User must fill in all the blanks or fields with correct format. | 1. Go to site https://etechify.herokuapp.com/product/insertProduct | Product Name: **Any name** Product No.: **Any product no.** Product Price: **Any product prices** | A new product will be added into the system. | P |

| | | | 2. Enter all fields with correct information and format.<br><br>3. Click submit button | Product Link:<br>**Any product links**<br>Image File:<br>**Any image files**<br>Brand Distributor (only for reseller):<br>**Any brand distributor**<br>Warranty Duration:<br>**Any warranty duration**<br>Product Category:<br>**Any category**<br>Product Brand:<br>**Any brand**<br>Product Stock:<br>**Any number** | | |
| PROD2 | Not all fields are filled with correct | User does not fill in all the blanks or | 1. Go to site https://etechify.herokuap | Product Name:<br>**Any name** | System validation will reject the new | P |

| | format for insert new product. | fields with correct format. | p.com/product/insertProduct<br><br>2. Fields are left empty or not in correct format.<br><br>3. Click submit button | Product No.:<br>**Any product no.**<br>Product Price:<br>**Any product prices**<br>Product Link:<br>**Any product links**<br>Image File:<br>**Any image files**<br>Brand Distributor (only for reseller):<br>**Any brand distributor**<br>Warranty Duration:<br>-<br>Product Category:<br>-<br>Product Brand:<br>-<br>Product Stock: | request and errors such as "This field is required" will appear. | |

| | | | | - | | |
|---|---|---|---|---|---|---|
| PROD3 | Delete existing product in system. | The product must be existed in the system. | 1. Go to site https://etechify.herokuapp.com/product/manageProduct 2. Click bin icon in actions column for each row of product. | Click bin icon in actions column. | Selected product will be deleted, and a success message will appear. | P |
| PROD4 | Alter product information for a specific product. | The product must be existed in the system. | 1. Go to site https://etechify.herokuapp.com/product/manageProduct 2. Click pencil icon in actions column for each row of product and a modal will pop-up retrieving the current | Product Name: **NZXT Kraken Z63** Product SN: **RL-KRZ63-01** Product Picture: **Any picture** Stock: **3** Price: RM **1115.00** | Product information is altered, and a success message will appear. | P |

| | | | selected product information. | Dealer Price: RM **1005.00** | | |
|---|---|---|---|---|---|---|
| | | | 3. Enter the desired fields with correct information and format. 4. Click save changes button. | | | |

**APPENDIX D:** TC: Manage Job Module

| Test Case ID | Test Case | Pre-Requisite | Test Step | Test Data | Expected Result | Status (P/F) |
|---|---|---|---|---|---|---|
| JOB1 | All fields are filled with correct format for insert new job. | User must fill in all the blanks or fields with correct format. | 1. Go to site https://etechify.herokuap p.com/job/new-job 2. Enter all fields with correct information and format. | Job Name: **Any job names** Job Salary: **Any job salary** Job Type: **Any job types** | A new job will be inserted, and it will be listed in a collection of job that is existed in the system. | P |

| | | | 3. Click submit button | Job Location: **Any job location** | | |
|---|---|---|---|---|---|---|
| JOB2 | Not all fields are filled with correct format for insert new job. | User does not fill in all the blanks or fields with correct format. | 1. Go to site https://etechify.herokuapp.com/job/new-job 2. Fields are left empty or not in correct format. 3. Click submit button | Job Name: - Job Salary: **Any job salary** Job Type: **Any job types** Job Location: - | System validation will reject the new request and an error "Please input all fields with relevant information" will appear. | P |

**APPENDIX E:** TC: Request Shipment Module

| Test Case ID | Test Case | Pre-Requisite | Test Step | Test Data | Expected Result | Status (P/F) |
|---|---|---|---|---|---|---|
| SHIP1 | All fields are filled with correct format for insert | User must fill in all the blanks or fields with correct format. | 1. Go to site https://etechify.herokuapp.com/shipment/new-shipment | Brand Distributor: **Any brand distributor** | A new shipment request will be created in the system and product will be | P |

| | | | | Brand: | added in cart section. | |
|---|---|---|---|---|---|---|
| | new item in shipment cart. | | 2. Enter all fields with correct information and format.<br>3. Click add to list button | **Any brand**<br>Select Product:<br>**Any product**<br>Select Product Quantity:<br>**Any product quantity** | Shipment will be changed to "Requested" status. | |
| SHIP2 | Not all fields are filled with correct format for insert new item in shipment cart. | User does not fill in all the blanks or fields with correct format. | 1. Go to site https://etechify.herokuapp.com/shipment/new-shipment<br>2. Fields are left empty or not in correct format.<br>3. Click add to list button | Brand Distributor:<br>-<br>Brand:<br>-<br>Select Product:<br>-<br>Select Product Quantity:<br>**Any product quantity** | System validation will reject the new request and an error "Please select input brand and product" will appear. | P |

| SHIP3 | Delete product in shipment cart. | The product must be existed in the system. | 1. Go to site https://etechify.herokuap p.com/shipment/new-shipment<br><br>2. Click bin icon for which product we want to remove. | Click bin icon in actions column. | Selected product will be deleted, and a success message will appear. | P |
|---|---|---|---|---|---|---|
| SHIP4 | Request new shipment with product in cart. | Selected items must be inside the cart. | 1. Go to site https://etechify.herokuap p.com/shipment/new-shipment<br><br>2. Click request shipment button and respond to the alert by clicking "Yes, submit it!". | Click "Yes, submit it!" button in a confirmation pop-up. | A shipment request will be submitted, and a success message will appear. Current shipment will be changed to "Waiting Approval" status. | P |

**APPENDIX F:** TC: Manage Product Serial Number Module

| Test Case ID | Test Case | Pre-Requisite | Test Step | Test Data | Expected Result | Status (P/F) |
|---|---|---|---|---|---|---|
| SN1 | All fields are filled with correct format for insert new product serial number. | User must fill in all the blanks or fields with correct format. | 1. Go to site https://etechify.herokuapp.com/distributor/insert-product 2. Enter all fields with correct information and format. 3. Click insert button | Select Product: **Any product** Select Batch: **Any batch** Insert SN Product: **Any product serial number** | A new product serial number will be inserted, graph will be updated, and it will be listed in a collection of serial number that is existed in the system. | P |
| SN2 | Not all fields are filled with correct format for insert new product serial number. | User does not fill in all the blanks or fields with correct format. | 1. Go to site https://etechify.herokuapp.com/distributor/insert-product 2. Fields are left empty or not in correct format. | Select Product: **Any product** Select Batch: **Any batch** Insert SN Product: **-** | System validation will reject the new request and an error "Please input all fields with relevant information" will appear. | P |

| | | | 3. Click insert button | | | |
|---|---|---|---|---|---|---|

**APPENDIX G:** TC: Shop Module

| Test Case ID | Test Case | Pre-Requisite | Test Step | Test Data | Expected Result | Status (P/F) |
|---|---|---|---|---|---|---|
| SHOP1 | Add product to cart. | The product must be existed in the system. | 1. Go to site https://etechify.herokuapp.com/product/items 2. Click add to cart button on a product that we want to purchase. | Click "Add to cart" button. | The selected product will be added to user cart and a success message "You have added this product to your cart!" will appear. | P |
| SHOP2 | Add product that is out of stock into cart. | Product must be existed in the system but no stock available. | 1. Go to site https://etechify.herokuapp.com/product/items | Click "Add to cart" button. | System will disable the "Add to cart" button for a product that is out of stock. | P |

| SHOP3 | Filter product based on brand, brand category and price. | The product must be existed in the system. | 1. Go to site https://etechify.herokuapp.com/product/items <br> 2. Choose which data that we want to filter in the dropdown. <br> 3. Click apply filters button | Select Brand: **Any brand** <br> Select Brand Category: **Any brand category** <br> Select Price: **Any price range** | System will filter the products that is existed in the system to match the keywords that have been chosen. | P |

**APPENDIX H:** TC: e-Track Parcel Module

| Test Case ID | Test Case | Pre-Requisite | Test Step | Test Data | Expected Result | Status (P/F) |
|---|---|---|---|---|---|---|
| TRACK1 | Search for existing tracking number in system. | Tracking number must be existed in the system. | 1. Go to site https://etechify.herokuapp.com/tracking/parcel <br> 2. Fill in tracking number. <br> 3. Click search button | Tracking No.: **32550903038** | System will show the current parcel status. | P |

| TRACK2 | Search for a non-existing tracking number in system. | Tracking number is not existed in the system. | 1. Go to site https://etechify.herokuapp.com/tracking/parcel 2. Fill in a non-existing tracking number. 3. Click search button | Tracking No.: **143093211531** | An error will appear "Tracking is not exist in the system." and no result found. | P |
|--------|------|------|------|------|------|---|
| TRACK3 | Empty value in tracking search bar. | User does not enter any value in the field. | 1. Go to site https://etechify.herokuapp.com/tracking/parcel 2. Click search button | Tracking No.: - | A pop-up of an error will appear "Please input tracking field". | P |

**APPENDIX I:** TC: Manage Profile Module

| Test Case ID | Test Case | Pre-Requisite | Test Step | Test Data | Expected Result | Status (P/F) |
|---|---|---|---|---|---|---|
| UP1 | Alter data either user name or email address. | Account is exists in the system. | 1. Go to site https://etechify.herokuapp.com/user/profile 2. Alter data either user name or email address with correct format. 3. Click submit button | Name: **Any names** Email address: **Any email address** | System will update to latest data and a success message "Profile is updated." will appear. | P |
| UP2 | Submit the same data to update current data. | User does not change any existing value. | 1. Go to site https://etechify.herokuapp.com/user/profile 2. Fields are filled with the same current data. 3. Click submit button | Name: **Ismat Azmy** Email address: **ismatazmy@live.com** | System validation will reject the new request and an error "Please make some changes to the input to make changes." will appear. | P |

**APPENDIX J:** TC: Manage Address Module

| Test Case ID | Test Case | Pre-Requisite | Test Step | Test Data | Expected Result | Status (P/F) |
|---|---|---|---|---|---|---|
| ADDRESS1 | All fields are filled with correct format for insert new address. | User must fill in all the blanks or fields with correct format. | 1. Go to site https://etechify.herokuapp.com/user/address 2. Enter all fields with correct information and format. 3. Click submit button | Name: **Any names** Address: **Any address** Phone No.: **Any phone number** Postcode: **Any postcode or system generated.** Longitude: **System generated.** Latitude: | A new address will be inserted, and it will be listed in a collection of address that is existed in the system for a specific user. | P |

| | | | | System generated. | | |
|---|---|---|---|---|---|---|
| ADDRE SS2 | Not all fields are filled with correct format for insert new address. | User does not fill in all the blanks or fields with correct format. | 1. Go to site https://etechify.herokuapp.com/user/address 2. Fields are left empty or not in correct format. 3. Click submit button | Name: - Address: - Phone No.: - Postcode: **Any postcode or system generated.** Longitude: **System generated.** Latitude: **System generated.** | System validation will reject the new request and an error "Please make some changes to the input to make changes." will appear. | P |

| ADDRESS3 | Delete address in the list. | The address must exist in the system. | 1. Go to site https://etechify.herokuapp.com/user/address 2. Click cross icon for which address we want to remove. | Click cross icon in actions column. | The selected address will be removed, and a success message will appear to indicate a success deletion. | P |
|---|---|---|---|---|---|---|
| ADDRESS4 | Set an existing address as a default address. | The address must exist in the system. | 1. Go to site https://etechify.herokuapp.com/user/address 2. Click update button for which address we want to make it as a default shipping address. | Click "Submit" button in a confirmation pop-up and click "Okay". | A success message "Your default status has been updated" will appear and default address will be set to the chosen one. | P |

**APPENDIX K:** TC: Manage Cart Module

| Test Case ID | Test Case | Pre-Requisite | Test Step | Test Data | Expected Result | Status (P/F) |
|---|---|---|---|---|---|---|
| CART1 | Delete product in cart. | The product must be in cart and existed in the system. | 1. Go to site https://etechify.herokuapp.com/product/cart 2. Click bin icon for which product we want to remove. | Click bin icon in actions column. | Selected product will be deleted, and a success message will appear. | P |
| CART2 | Alter product quantity. | The product must be existed in the system. | 1. Go to site https://etechify.herokuapp.com/product/cart 2. Increase and decrease quantity by using spinner. | Increase and decrease quantity using spinner. | System will automatically update merchandise subtotal according to the total quantity in spinner. | P |
| CART3 | Input product quantity more than total stock. | User input more quantity than total stock quantity. | 1. Go to site https://etechify.herokuapp.com/product/cart | Quantity (total stock - 3): **10** | A pop-up error "Stock is around **X** only" will appear, | P |

| | | | 2. Increase and decrease quantity by using spinner or by inputting more than total stock quantity. | | PayPal button will be hidden, and quantity reset to 1. | |
|---|---|---|---|---|---|---|
| CART4 | Complete order by doing payment using PayPal. | At least one item is in cart. | 1. Go to site https://etechify.herokuapp.com/product/cart <br> 2. Click PayPal button. | Click "PayPal" button. | System will open a small pop-up to let user to complete the payment and once payment is succeeded, system will redirect user to a thank you page and order will be created. | P |

**APPENDIX L:** TC: e-RMA Module

| Test Case ID | Test Case | Pre-Requisite | Test Step | Test Data | Expected Result | Status (P/F) |
|---|---|---|---|---|---|---|
| RMA1 | All fields are filled with correct format. | User must fill in all the blanks or fields with correct format. | 1. Go to site https://etechify.herokuapp.com/rma/new-request<br>2. Enter all fields with correct information and format.<br>3. Click submit button | Select Address:<br>**Any address**<br>Date of Purchase:<br>**Any date of purchase**<br>Proof of Purchase:<br>**Any proof of purchase in form of pdf.**<br>Brand:<br>**Any brand**<br>Product:<br>**Any product**<br>Serial Number: | A new RMA request will be added into the system. | P |

| | | | | - | | |
|---|---|---|---|---|---|---|
| | | | | Serial Number: | | |
| | | | | - | | |
| | | | | Confirm Serial Number: | | |
| | | | | - | | |
| | | | | Comments: | | |
| | | | | - | | |

**APPENDIX M:** TC: Manage Repair Module

| Test Case ID | Test Case | Pre-Requisite | Test Step | Test Data | Expected Result | Status (P/F) |
|---|---|---|---|---|---|---|
| REPAIR 1 | All fields are filled with correct format. | User must fill in all the blanks or fields with correct format. | 1. Go to site https://etechify.herokuapp.com/user/userdash 2. Click on wrench icon in RMA request section and a modal will pop-up. 3. Enter all fields with correct information and format. 4. Click update button | Update Status: **Any status** Receive At: **Any date but not the day after.** Tracking No.: **Any tracking numbers** Remark: **Any remark** | The selected RMA request will be updated by the system according to the latest data. | P |
| REPAIR 2 | Not all required fields are filled with correct format. | User does not fill in all the blanks or fields with correct format. | 1. Go to site https://etechify.herokuapp.com/user/userdash | Update Status: **Any status** Receive At: **Any date but not the day after.** | System validation will reject the update request and errors such as a pop-up "Please input all | P |

| | | | 2. Click on wrench icon in RMA request section and a modal will pop-up.<br>3. Required fields are left empty or not in correct format.<br>4. Click update button | Tracking No.:<br>-<br>Remark:<br>- | required fields" will appear. | |
|---|---|---|---|---|---|---|
| REPAIR 3 | Check tracking status. | Tracking number must exist in the system. | 1. Go to site https://etechify.herokuapp.com/user/userdash<br>2. Click at "Tracking No." column for each row of RMA case in RMA requests section. | Click a button with a tracking number under "Tracking No." column. | System will pop-up a small window and fetch all tracking data for the specific tracking number by using Tracking.my API. | P |

**APPENDIX N:** TC: Manage Users Module

| Test Case ID | Test Case | Pre-Requisite | Test Step | Test Data | Expected Result | Status (P/F) |
|---|---|---|---|---|---|---|
| USR1 | Edit user information. | APPENDIX I: MANAGE PROFILE MODULE | | | | P |
| USR2 | Delete user. | User is exist in the system. | 1. Go to site https://etechify.herokuapp.com/admin/users 2. Click delete button | Click "Delete" button under actions column. | Selected user will be deleted, and a success message will appear. | P |
| USR3 | Alter an account role. | User is exist in the system. | 1. Go to site https://etechify.herokuapp.com/admin/users/{id}/edit 2. Select one or more of the roles listed at the checkboxes. | Choose 1 or more from the 3 roles that is existed in the system. | System will update current user role according to the selection at the checkboxes. | P |

**APPENDIX O:** TC: Manage Orders Module

| Test Case ID | Test Case | Pre-Requisite | Test Step | Test Data | Expected Result | Status (P/F) |
|---|---|---|---|---|---|---|
| ORDER1 | Update parcel tracking status. | Tracking number is exist in the system and not in delivered status. | 1. Go to site https://etechify.herokuapp.com/ tracking/track-parcel/{id}<br>2. Enter all fields with correct information and format.<br>3. Click submit button | Tracking Status: **Processing Order** | A new tracking status will be added into the system. | P |
| ORDER2 | Update parcel tracking status with incorrect format or null value. | User does not fill in all the blanks or fields with correct format. | 1. Go to site https://etechify.herokuapp.com/tracking/track-parcel/{id}<br>2. Fields are left empty or not in correct format. | Tracking Status: - | System validation will reject the new request and errors such as a pop-up "Please input all fields with relevant | P |

| | | | | 3. Click submit button | | information" will appear. | |
|---|---|---|---|---|---|---|---|
| ORDER3 | View order in detail. | Customer orders must exist in system. | | 1. Go to site https://etechify.herokuapp.com/ order/purchase/{id} | Click eye icon button. | Order details of a specific order will show. | P |
| ORDER4 | View receipt for an order. | | | 1. Go to site https://etechify.herokuapp.com/ order/receipt/{id} | Click paper icon button. | e-Receipt will be generated for the specific order. | P |
| ORDER5 | Insert serial number for products in a specific order. | | | 1. Go to site https://etechify.herokuapp.com/order/insert-prod-sn/{id} 2. Drag the related serial number from the suggested SN and drop to the respective product | Insert SN: **A6TX0IC2K0146H27** | Serial number will be assigned to each of the respective product. | P |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | according to total quantity for each product. | | | |
| ORDER6 | Print airway bill (AWB) consignment. | | 1. Go to site https://etechify.herokuapp.com/ order/purchase/awb/{id} | Click arrow up icon button. | System will generate an airway bill consignment for a parcel that belongs to an order. | P |
| ORDER7 | Cancel an order. | | 1. Click on the wrong cross button. | Click wrong cross icon button. | An order will be canceled unless the order status have reached processing order. | P |

**APPENDIX P:** Sample Source Code

```js
onApprove: function (data, actions) {
    return actions.order.capture().then(function (details) {
        //alert('Transaction completed by ' + details.payer.name.given_name);
        console.log(details)

        var pay_total = details.purchase_units[0].amount.value;
        var pay_method = details.purchase_units[0].soft_descriptor;
        var order_Id = $("input[name=get_ord_id]").val();
        var name = '{{ env('APP_URL') }}';

        Swal.fire({
            icon: 'success',
            title: 'Processing your request, waiting for PayPal and arranging your orders!',
            didOpen: function () {
                Swal.showLoading()
                $.ajax({
                    type: 'GET',
                    url: name+"/order/purchase/success/" + order_Id,
                    contentType: "application/x-www-form-urlencoded; charset=UTF-8",
                    data: $("#form_cart").serialize() + '&pay_method=' + pay_method + '&pay_total=' + pay_total,
                    success: function (data) {
                        console.log(data)
                        if (data['success']) {

                            var msg = 'Dear, ' + recipient[0].name.toUpperCase() + '. Thank you for shopping with us at e-Techify! Your order ID is ' + myA[0].order_id
                            var phone = recipient[0].phone_no;
                            fetch("https://terminal.adasms.com/api/v1/send?_token=Zk24gMmtAqWl1QJbIBwp3biEtZPp10bo&phone=60"+ phone +"&message=" + msg, {
                                "method": "POST"
                            })
                                .then(response => console.log(response))
                                .catch(err => console.error(err));
```

**Attachment 1:** PayPal JS code

```
@foreach($orderInfo as $i)
    <form method="POST" action="{{ route('order.purchase.updatesn', $i->order_id) }}" id="sn_insert_form">
        @csrf
    <tr>
        <th scope="row">{{ $loop->iteration }}</th>
        <td>
            <img src="{{ \Storage::disk('s3')->url('product/'.$i->product_image_path) }}" style="...">
        </td>
        <td>{{ $i->product_name }} <p class="lead">[x{{ $i->product_order_quantity }}]</p></td>
        <td>
            <input type="text" id="getproduct_qty" name="getproduct_qty" value="{{ $i->product_order_quantity }}" hidden>
            @for ($x = 0; $x < $i->product_order_quantity; $x++)
                <input type="text" class="form-control" id="droppable{{ $x }}{{ $i->product_id }}" name="{{ $i->product_id }}[]" style="..."
            @endfor
            <input type="text" id="getproduct_sn" name="getproduct_sn" value="{{ $i->product_id }}" hidden>

        </td>
        <td>
            {{ date('Y-m-d H:i A', strtotime($i->updated_at)) }}
        </td>
    </tr>
@endforeach
```

**Attachment 2:** Laravel Blade code

## Soft-Delete

Laravel soft delete,
without deleting column.

```
// Soft Delete.
Order::find($id)->delete();

return redirect()->back()->with('danger', 'Order has been cancelled');
```

## Soft-Delete Retrieval

Laravel – retrieve all soft-deleted
orders in "Cancelled" section.

```
$cancelled = Order::onlyTrashed()
    ->join( table: 'users', first: 'users.id', operator: '=', second: 'orders.user_id')
    ->select( columns: 'orders.id', 'orders.order_status', 'users.name', 'orders.created_at')
    ->where([
        'orders.user_id' => Auth::id()
    ])
    ->get();
```

## Order Cancelled

Every soft-delete order will be
categorized in "Cancelled" section.

**Attachment 3:** Soft delete Laravel code