E-DocPh

ABDUL HADI MAZBAH

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

E-DocPh

ABDUL HADI MAZBAH

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-DocPh

is written by me and is my own effort and that no part has been plagiarized without citations.

STUDENT : ______________________________ Date : 9/9/2021

ABDUL HADI MAZBAH

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.

Tandatangan dan Cop
Signature & Stamp
KASTURI KANCHYMALAY
Pensyarah
Jabatan Kejuruteraan Perisian
Fakulti Teknologi Maklumat Dan Komunikasi
Universiti Teknikal Malaysia Melaka

SUPERVISOR : ______________________________ Date : 12/9/21

Dr. Kasturi Kanchymalay

DEDICATION

In the name of almighty Allah, the most merciful and most gracious. I thank my lord for all the strength, protection, guidance and health.

I would like to thank from depth of my heart to my beloved parents who are my strength, my inspiration my heroes and every other person who helped on this journey.

Thanks to my brothers, Chowdhury Mohammad Tawsif Khan, Abdullah Al Mamun and Aminul Islam for their motivation, advice and support.

ACKNOWLEDGEMENTS

In the name of almighty Allah, the most merciful and most gracious. Thanks to almighty Allah that I have completed my project and report for PSM.

It is a great pleasure for me to work under supervision of my advisor as well as supervisor Dr. Kasturi Kanchymalay. She guided me throughout the semester and always appreciated my ideas. She taught me and guided me while I was having trouble regarding the project. A big thank you from my heart to Dr. Kasturi Kanchymalay for always supporting me and guide me.

Thanks to my family for their moral support and love. I am really grateful to them for giving such a wonderful life and opportunities.

Thank you to my friends and seniors who helped with ideas and resources. Motivation that I got from them during this project really encouraged me to move forward. It is a nice journey.

ABSTRACT

Online pharmacy industry is increasing day by day. Pharmacy is a place where we can find medicines and other medical components. As a matter of fact, online pharmacy is a part of telemedicine. Whether it is pandemic or normal days we need medication. By uprising e-pharmacy industry, now we have a service that will take care of daily-to-daily medication related needs. In this project I am working to build a online pharmacy web application named E-DocPh. This web application can be used by three different users who are doctors, admins and customers (patient). In past year we saw corona pandemic. Millions of people were affected and still getting affected. In these hard situations we could not go outside even if we were in emergency. This web application will be a solution for that kind of emergency problems. People can buy medication online and medicines will be delivered on their doorstep. This application will solve the problem of taking medicine without prescription as well as using them as drugs. This is because, every time patients wants to buy medicines, they will have upload a prescription image which will be authorized by a doctor. Therefore, problems like won't be happen by using this web application. This project will be designed with couple of programming languages. Front-end will be developed by HTML, CSS and JavaScript whereas back-end programming will be done by Django3 which is a framework of python and JavaScript. SQLite3 will be used as database for the project. The system will be designed as three different modules. Admin module is to monitor and maintain the system. Doctor module is for doctor to approve orders that were ordered by patients. Lastly, patient module which is e-commerce site where patient can buy medical related stuffs.

ABSTRAK

Industri farmasi dalam talian semakin meningkat dari hari ke hari. Farmasi adalah tempat di mana kita dapat mencari ubat-ubatan dan komponen perubatan lain. Sebenarnya farmasi dalam talian adalah sebahagian daripada telemedicine. Sama ada pandemik atau hari biasa kita memerlukan ubat. Dengan membangkitkan industri e-farmasi, sekarang kami memiliki layanan yang akan mengurus keperluan yang berkaitan dengan ubat setiap hari. Dalam projek ini saya berusaha untuk membina aplikasi web farmasi dalam talian bernama E-DocPh. Aplikasi web ini dapat digunakan oleh tiga pengguna berbeza iaitu doktor, pentadbir dan pelanggan (pesakit). Pada tahun lalu kami melihat wabak korona. Berjuta-juta orang terjejas dan masih terjejas. Dalam keadaan sukar ini kita tidak boleh keluar walaupun kita berada dalam keadaan kecemasan. Aplikasi web ini akan menjadi penyelesaian untuk masalah kecemasan seperti itu. Orang ramai boleh membeli ubat secara dalam talian dan ubat-ubatan akan dihantar di depan pintu rumah mereka. Aplikasi ini akan menyelesaikan masalah pengambilan ubat tanpa resep serta menggunakannya sebagai ubat. Ini kerana, setiap kali pesakit ingin membeli ubat, mereka akan memuat naik gambar preskripsi yang akan mendapat kebenaran daripada doktor. Oleh itu, masalah seperti tidak akan berlaku dengan menggunakan aplikasi web ini. Projek ini akan dirancang dengan beberapa bahasa pengaturcaraan. Front-end akan dikembangkan oleh HTML, CSS dan JavaScript sedangkan pengaturcaraan back-end akan dilakukan oleh Django3 yang merupakan kerangka python dan JavaScript. SQLite3 akan digunakan sebagai pangkalan data untuk projek tersebut. Sistem ini akan dirancang sebagai tiga modul yang berbeza. Modul pentadbir adalah untuk memantau dan menyelenggara sistem. Modul doktor adalah untuk doktor meluluskan pesanan yang dipesan oleh pesakit. Terakhir, modul pesakit yang merupakan laman web e-commerce di mana pesakit boleh membeli barang berkaitan perubatan.

Table Of Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

FYP	--------------------------	Final Year Project
E-DocPh	--------------------------	Electronic Doctor & Pharmacy
SDLC	--------------------------	Software Development Life Cycle
IEEE	--------------------------	Institute of Electrical and Electronics Engineers
T.D	--------------------------	Test Data
DFD	--------------------------	Data Flow Diagram
ERD	--------------------------	Entity Relationship Diagram
P.D.D	--------------------------	Physical Database Design

CHAPTER 1. INTRODUCTION

1. Introduction

Today, organizations have a broad data-supported knowledge base on the desires, expectations, routines and even emotional states of their consumers (D. H. Lee, 2018). In order to create in value opportunities and organizational innovation the main drivers are advance digital technology(S. M. Lee & Lee, 2020) . Pharmaceutical marketing is a broad term that covers a wide range of topics. It’s been a huge platform for consumers to get services regarding all type of medical tonics and related objects.

General and specific awareness of marketing and business practices related to the product/service in the fields of pharmacy, counseling, and health enhancement Patients', and purchasers' wellbeing, with profits going to the society and organizations that offer the product/service of economy and healthcare (Ignjatović & Stanić, 2019).

With respect to patient-related outcomes, satisfaction, perceived information exchange, interpersonal relationship building, and perceived shared decision making showed no significant differences between face-to-face and screen-to-screen consultations (Tates, Antheunis, Kanters, Nieboer, & Gerritse, 2017). This system will be used by people who are in need of medicine related items. However, in this process of getting this service, consumers need to get a doctor’s authorization for security purposes.

In a research of Interactive Telemedicine Flodgren G. et al found no difference in mortality between participants with heart failure receiving care through telemedicine, compared to those receiving health care without telemedicine. The results of the studies differed for admissions to hospital, from a relative decrease of 64% to an increase of 60% (Flodgren, Rachas, Farmer, Inzitari, & Shepperd, 2015).

Disease-specific quality of life was slightly improved for heart failure participants receiving telemedicine as compared to those receiving usual care only. Thus, online consultancy is not worse or far better than normal way. It’s almost same. However, it improves time efficiency, it is more flexible and according to some research better works than normal way. Although this system is not going to make any arrangement to provide consultation to the consumers. Even though system is going to arrange a process in order to contact doctors with consumer. This way consumers can talk about their problems and get consultation about which tonics or related objects they can order from the system on their health basis.

The goal of E-DocPh is to overcome such problems like distance, pandemic situation and bad quality. This will also handle security matters. Sometime people misuse such systems to get drugs and use it anyway whether they need it or not. There are some drug issues as well out there. People has a intention to buy drugs and use it in order to get high. These are extremely dangerous for the society. Remembering that, E-DocPh has made it so secure as no one can buy any tonics that can be harmful to consumers.

On the other hand, we can think of this recent pandemic. It was horrible to get outside to go to a doctor or pharmacy for medicine. This is because, it was risky at time same time breaking rules. This system can handle everything during such pandemic. Patient doesn’t have to get out from their home but still they will get best service and advice they can possibly get. This system will be easy to use, flexible, comfortable and time efficient.

1.2 Problem Statement

➢ **Time Consuming:** It is time consuming to get outside and buy medicines. In addition it’s not flexible to get outside and buy medicine from pharmacy store.

➢ **Safety Issue:** It is not safe to go outside for medicine in the event of any pandemic. Even if cause can be critical and emergency. One thing is always matters that when we are going outside, we are actually letting our guard down to corona virus. Therefore, there is always a risk.

➢ **Security Issue:** It is not secure to buy some of medicines that can be used as drugs. Sometimes people just buy drugs and get high. There is another type of consumers who just buy tonics because they think they need it to play home doctor. This is really a serious issue. By doing this every year a lot of people faces serious troubles. As Mahmood Karimy et al (Karimy, Rezaee-Momtaz, Tavousi, Montazeri, & Araban, 2019) found that in Iran maximum of women does self-medication in their family which leads to serious issues.

➢ **Quality issue:** Previously there was many studies on how consumers get bad quality medicines which causes them a lot of troubles. Roger Bate et al. (Bate, Ginger, Jin, Mathur, & Attaran, 2016) found that there are manufacturing companies that produce bad quality medicines which is really health concerning for consumers. Some pharmacies do not check the quality of medicines which leads consumers getting poor quality medicines. These phenomena can create casualties.

1.3 Objectives

- To build a system that will help patient to get facilities easily by professionals.
- To assess doctors to provide health service.
- To ensure patient good service satisfaction.
- To secure all the purchases and make sure not causing society bleed.

1.4 Scope

Modules to be developed:

- **Registration :** allows admins, patients and doctors to register in the system. They will be able to use the system with their own account by doing this.

- **Authentication :** to authenticate three different types of users.

- **Cart :** This will allow patients to save their desired items to be stored as targeted to purchase later.

- **Payment :** to allow patients pay for their purchased items bill.

- **Pharmacy window :** all the items to be sale added by the admins will show here. This window will allow patients to choose and search for their respective medicine or related stuffs.

- **Doctor Module:** this will allow doctors to accept or reject order that patients made. Doctor can see patient’s uploaded prescription image or can call them directly in order to accept or reject the order.

- **Email automation:** By this module users can get their order confirmation or rejection messages from the system. This module will be also responsible for reset password if user forgot their password.

- **Admin module:** admin can monitor the system from here. Admins can see the database and attributes of it in here.

- **Create, update and delete module:** admins can create, update and delete items in the database.

Target users:

- Pharmacy authority as admins
- Doctors
- People as Patients

1.5 Project significance

- ➢ This will help pharmacy authority to be in business even in pandemic time. This will be a automatic system so in order to maintain the system labor is not necessary. Only few developers can maintain the system fine. This will help pharmacy authority to maintain the data and manage it.
- ➢ Helps doctors to provide healthcare from anywhere.
- ➢ Helps consumers to get quality-full medication and related stuffs. Consumers can also get doctor’s authorization before taking any medicine.

1.6 Expected Output

In presence of modern technology, it is our bonanza that we are making our possible needs automatic which is making our life easier. By using this proposed system people will be able to get their needs.

It will be a risk free, hopefully a bug-free web application that will help all the people who needs good pharmaceutical products. This system will be easy to use flexible and comfortable.

All the purchases will be authorized by doctors. Admins will be able to monitor the whole system from this web application. Pharmaceutical business will be running running high with help of this website even though pandemic has stopped everything.

1.7 Conclusion

In short, E-DocPh is a web application that will provide a lot of services to different types of users. Admins can maintain their business by this application. People as Patients can get their needed items through this website's store module. Patients can get doctor authorized items. It will be secure and quality-full items.

Doctors can provide healthcare anywhere anytime using telemedicine technique. This system provides a module for doctors.

The reason of developing this web application is to provide healthcare to people in any situation with quality-full work.

CHAPTER 2. LITERATURE REVIEW AND PROJECT METHODOLOGY

2.1 Introduction

This chapter contains information and necessary components to describe the work of previous systems and publications that took place before initialization of this project. In addition project methodology and other related topics will be included in chapter.

Previously, there were systems built for the same objectives or less where experts fulfilled their objectives however there were some lacking. In this chapter I will discuss further regarding mentioned projects and their lacking. This chapter will also include the findings and techniques used by other personnel in the similar type of system or research.

Onward methodology will be discussed such as processes to make the system, flow of the total project work, requirements and analysis and time management for the whole project.

2.2 Fact and findings (based on topic)

2.2.1 Domain

Domain of E-DocPh is information, communication and technology (ICT) in Web application. This web application will be a example of e-commerce in medical services. Advancing technology in e-commerce and web application requires services that are easier and comfortable. This challenge has been taken while developing E-DocPh to fulfill such criteria.

2.2.2 Existing System

E-pharmacy industry will provide a huge contribution for the growth of online medicine services. p. Kumari et al. (Nandal & Research Scholar, 2017) a system developed with xampp and other frameworks to develop a website server test. By using PHP and xampp the system had to make a server friendly system that in future it would not be dificult to modify the code.

According to Atisha S. Patil et al.(Nandal & Research Scholar, 2017) A system that can authenticate medicines if the consumer is taking it in need or not before purchasing. The reason is there are some medicine which can be dangerous to use without consulting doctor. Worst scenario, some of the medicines can be used as drugs. In order to prevent this type of phenomena it is very appreciable to come up with such method.

To make a server and user-friendly website they started by using CSS, HTML, PHP, JavaScript to make web Pages. As database they used mysql and xampp. Every web page had their own unique title and description, tags and so on. By using this method, they found it easy and fast response from server. This way it appears that coding can be quality and much standard.

Using a form that contain a place to upload prescription for the medicines consumer want to buy is the method to overcome the authentication problem. By these prescription doctors who are associated with the system can check the items user want to buy and match the items with the prescription they uploaded to the system. After checking doctors can either accept the order or reject it. Once doctor decide and execute their decision then the problem of order authentication can be solved.

In the research by Kumari et al.(Nandal & Research Scholar, 2017) they found their system is working fine and coding was quality-full and also it was fast responsive. However, the system can be more efficient if used any framework. The result of the research was as expected.

Trying to overcome the problem of order authentication was successful and as expected. The medicines that are sensitive cannot buy by consumer without getting acceptance from a doctor. Using this method is such user friendly and faster. It will take only couple of second to upload the prescription. So, the result was as expected.

2.2.3 Technique

Although the system of Kumari et al. (Nandal & Research Scholar, 2017) has reached their expectation, there are some room for advancement. They used php and MySQL to build their system. Instead of that if they used python and its web-based framework Django, the scenario can be more efficient. Django already has an integrated database named sqlite3 database. By using this database developers can update their work very easily. It comes with an integrated admin system where developer can test any data directly just like a user. Whereas, by using MySQL it can’t be possible unless developer make a method for login authentication. Using xampp is another backdated thing. Using this sometimes developer can face issue to configure.

On the other hand, Django by using Django and SQLite3, there is no need for a third-party software in order to connect the system. This method is going to be used in this E-DocPh project.

However, online e-pharmcy team made an authentication progress but it can be more optimized. There is no room for emergency in their system. Let's say a consumer need some medicine urgently but they don't have any prescription for that medicine they want to buy. What can they do now. Without upload a prescription they cannot order anything. A method will be added in E-DocPh to overcome this problem. Once, user select all the medicine they will be redirected to order authentication page. In the page they will have asked to upload a prescription to authenticate their order. Once, doctor open their order, doctor will search for prescription first. If doctor think that order should be accepted then consumer can proceed with their order else cancel.

2.3 Project Methodology

In this project, I used waterfall model as our system development methodology. Waterfall model is the earliest SDLC approach that was used for software development and also referred to as a linear-sequential life cycle model. 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 water waterfall model, the phase do not overlap.

The waterfall approach, the whole process of software development is divided into separate phases. In this model, typically, the outcome of one phase acts as the input for the next phase sequentially. The sequential phases in waterfall model are:

- ➢ Requirement Gathering and Analysis – Modules that to be developed for the system will be documented as a specification requirement document in this phase.

- System Design – System design should be prepared in this phase on basis of first phase. This system design aids in designing the architecture of the system as well as describing hardware and system requirements.

- Implementation − The system is first built as discrete program, called units, which are then merged in the next phase, using inputs from the system design. Unit testing is the process of developing and testing each unit for its functioning.

- Integration and Testing − After each unit has been tested, all of the units built during the implementation phase are merged into a system. The entire system is then tested for any errors or failures after it has been integrated.

- Deployment of system − The product is deployed in the client environment or released into the market once functional and non-functional testing is completed.

- Maintenance − In the client environment, there are a few challenges that arise. Patches are published to address these vulnerabilities. Additionally, better versions of the product have been launched in order to improve the product. Maintenance is carried out in order to bring about these modifications in the customer's environment.

The following illustration is a representation of the different phases of the Waterfall Model:

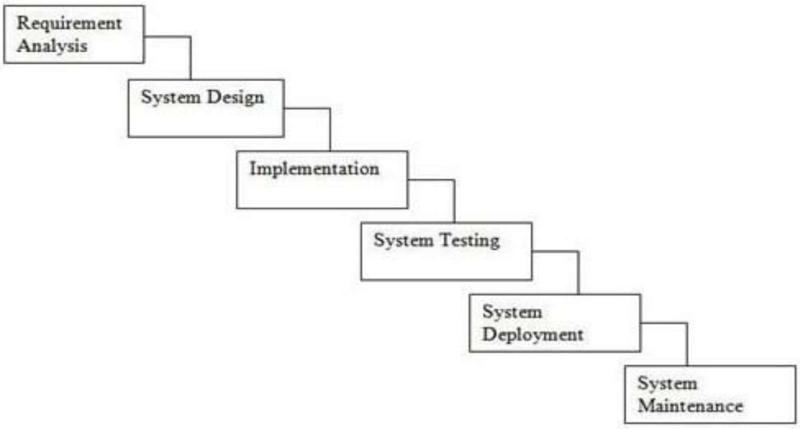


Figure 1Waterfall model

2.4 Project Requirement

2.4.1 Software Requirement

Table 1: Software Requirements

Software name	Version	Description
Ubuntu	20.04 LTS	This will be the operating system for the project.
VS Code	1.55.2	This will be the editor to write codes.
Python	3.8.5	This will be used as back-end programming.
Django	3.1.8	This is a web-based framework of python.
HTML	5	This is standard markup language for documents designed to be displayed in a web browser.
CSS	2.1	This is a programming language that will be responsible for design and coloring of the system's interface.
JavaScript	0	This will be the language responsible for system behavior.
LibreOffice Writer	6.4.7.2	This is a software, where report will be written.
Mendeley	1.19.8	This is a software that will help to cite references.
Google Chrome	0	This is where system will be tested during and afterwards of coding.

2.4.2 Hardware Requirement

Table 2: Hardware Requirement

Hardware Name	Description
Laptop Model	Asus VivoBook s15
Processor	Intel core i5 8th gen
Installed RAM	8 GB
Graphics card	Intel UHD Graphics 620 and Nvidia Geforce MX150
Operating System	Ubuntu 20.04 LTS
Hard Drive	128 GB SSD, 1 TB HDD

2.4.3 Other Requirement

There is no other requirement to describe for the system. All the necessary requirements have been mentioned in previous phases.

2.5 Project Schedule and Milestones

Grantt Chart:

In this chart I am showing the process and time management for each of the phases of the waterfall model that has been chosen for this project.

From week 1 until 3 will be for requirement analysis which includes searching and analysis the current system and also taking inquiries and data related component in order to plan the system. Week 3 and 4 are to design the system which includes system front-end design such as UI design and design for to place back-end component. From week 5 until 10 is reserved for implementing code in order to build the system. Week 11 is reserved for system testing. This phase is to test the system to check if everything is working as it were planned or not. Week 12 and 13 are for to review the system. If any bug found in testing phase, this is the time to fix that or those bug/bugs and recompile. Week 14 is reserved for presentation. This week university supervisor and evaluators will see and check the system. Final report submission will be also take place in this week.

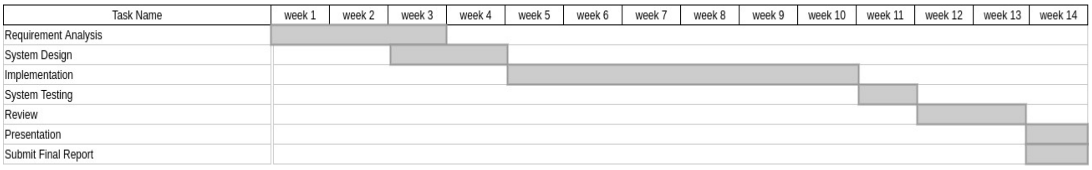


Figure 2: Grantt Chart

2.6 Conclusion

This chapter was for to discuss and describe about literature and methodology. Above I have informed about the related work for the project which should be included in this chapter. We will see next part of system which is analysis in the next chapter.

In this section I have described about analysis of current system. Flow of the current system and their techniques have been discussed and visualized. Functional and non-functional requirements have been stated as well as structured chart and grantt chart for timetable management discussion of the system.

CHAPTER 3. ANALYSIS

3.1 Introduction

This chapter is to analyze the system data structure and behavior. This chapter will illustrate some designs that describes how the data will flow, what would be the structure and also a data-set illustrator. The requirements needed to build the system will be shown in this chapter.

3.2 Problem Analysis

Previously a team worked on a similar project like E-DocPh. Ashita S. Patil et al.(Patil, Patil, More, Sankpal, & Student, 2019) researched on a project similar to E-DocPh named EasyMeds. Some of their work details are going to use here for research purposes.

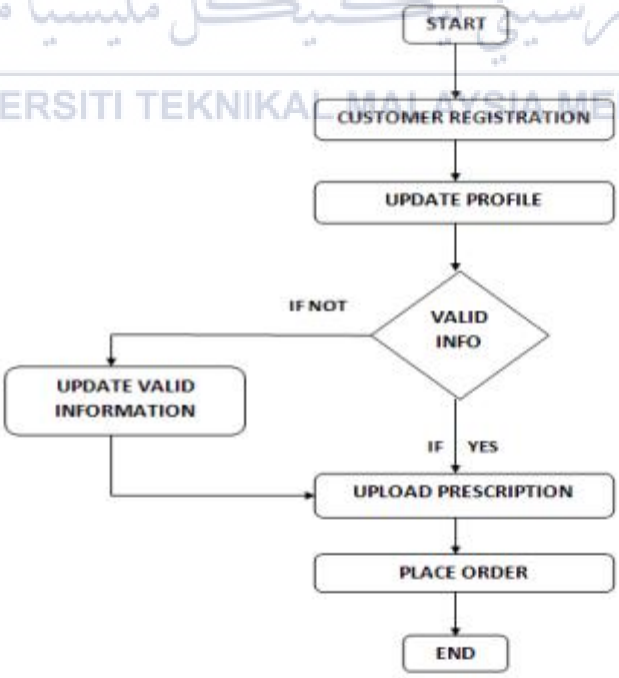


Figure 3: Prescription Upload

This Flowchart is for users in EasyMeds system. As we can see this process is to verify the order drugs. In here system is asking user to give a prescription to proof that they need the drug.

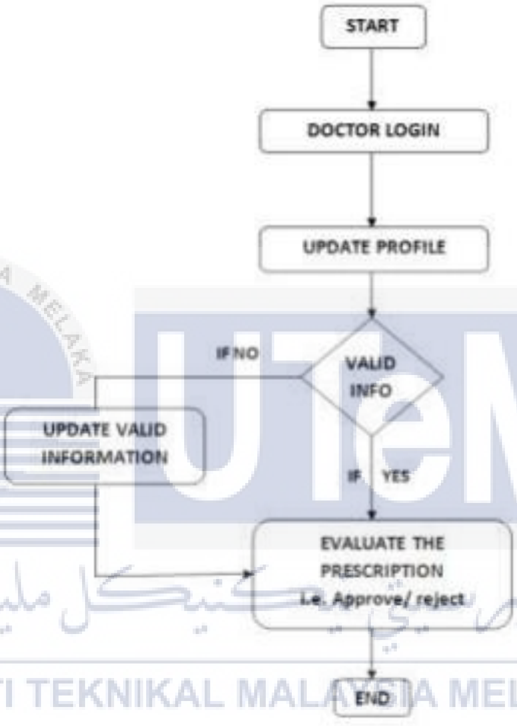


Figure 4: Doctor approval

Here in figure 4 we can see that this is a DFD where doctor is approving or rejecting the order of respective customers.

In Figure 3 we can see that a customer is uploading a prescription while they are signing up in the system. In chapter we discussed problem statement. This feature in EasyMeds is one of that problem. Uploading a prescription during signup is a one time problem solution. A customer does not have only one prescription for their whole life. While EasyMeds is taking prescription during sign up, they can only one time verify that particular customers order with that prescription. By doing that they will not able locate that customers order again for security purposes. Let’s put that aside and take a look from a different way. Let’s that particular customer need to order a medicine on emergency circumstances. Now, in order to order any drugs that the customer actually need and let’s say they also have a prescription for that. Still that customer cannot order it and expect that he/she will receive the order because there is chance that his/her order get rejected because that customer ordered a drug that is not included in his/her last uploaded prescription. Now, system also cannot help in this matter because in order to get the latest prescription, customer has to register again with a new account and need to upload the latest prescription there which is a time consuming especially when the customer is in emergency.

In figure 4 we can see that doctor is approving the order on basis of uploaded prescription by customer. Now, what if doctor is confused to make a decision. Such as what if doctor need to contact with the customer for some more details. That facilities had not included in EasyMeds system. On the other hand, what if customer does not have any prescription but need some basic medicine. Customer cannot order anything unless they have any prescription. This is a serious drawback for the system. Any alternative way should be there in order to overcome these circumstances.

3.3 Requirement analysis

3.3.1 Data Requirement

Table 3: Product Table

Column Name	Data Type	Field Length	Constraint	FK Reference	Description
m_productID	Integer	-	Primary key	-	Primary and unique for the table. Each product will have their own unique primary key.
Name	Varchar	200	Not Null	-	Name of the product.
Brand	Varchar	200	Null = true.	-	Brand of the product.
quantity	Integer	-	Default = 0. null =true, blank =true.	-	This will be the number of the product are available to sale.
Price	Decimal	7,2	-	-	This will be the price of the product.
Digital	Boolean	-	Default = false, null = true, blank = true.	-	This is meant for products that are digital thing not physical. These products cannot be shipped.
Image	Image	-	Null = true. Blank = true.	-	This is for to upload saving path for product images.
Description	varchar	2000	-	-	This is to write description for product.
Tags	Varchar	20	-	-	This is originally meant to be a key for recommendation algorithm.

Table 4: Patient Table

Column Name	Data Type	Field Length	Constraint	FK Reference	Description
PatientID	Integer	-	Primary key	-	Primary and unique for the table. Each patient will have their own unique primary key.
user	Varchar	200	Null = true, blank = true, on_delete = cascade.	-	If this patient is deleted than all of the data connected with this patient will be deleted.
Name	Varchar	200	Not Null	-	Name of the patient.
Email	Varchar	200	Not null, unique = true	-	This will be the email of the patient.
Password	Varchar	50	-	-	This will be the password of the patient.
IC_no	Varchar	20	-	-	This is IC no of patient.
Contact_no	Varchar	13	-	-	This is contact number of patient.

Table 5: Prescription Table

Column Name	Data Type	Field Length	Constraint	FK Reference	Description
PrescriptonID	Integer	-	Primary key	-	Primary and unique for the table. Each prescription will have their own unique primary key.
patient	-	200	Null = true, blank = true, on_delete = set null.	Patient	This is a foreign key.
date_ordered	Date	-	Auto add.	-	This will automatically record date and time for all the product ordered.
Complete	Boolean	-	Default = false.	-	This will check if delivery is complete or not.
transaction_id	Varchar	100	Null = true	-	This will be the transaction of the order.
authorize	Varchar	200	Default = "Pending"	-	This is the decision of doctor to the respective prescription.
prescripImage	Image	-	Null = true	-	This is for to upload prescription image file by patient.

Table 6: Order Item Table

Column Name	Data Type	Field Length	Constraint	FK Reference	Description
OrderitemID	Integer	-	Primary key	-	Primary and unique for the table. Each orderitem will have their own unique primary key.
product	-	-	Null = true, on _delete = set null.	m_product	This is a foreign key.
Order	-	-	Null = true, on _delete = set null.	Prescription	This is a foreign key.
quantity	Integer	-	Default = 0. null =true, blank =true.	-	This will be the number of the product that are added in to cart for the product.
date_added	Date	-	Auto add.	-	This will automatically record date and time for the product ordered.

Table 7: Shipping Address Table

Column Name	Data Type	Field Length	Constraint	FK Reference	Description
shippingAddressID	Integer	-	Primary key	-	Primary and unique for the table. Each shippingAddress will have their own unique primary key.
customer	-	-	Null = true, on _delete = set null.	Patient	This is a foreign key.
Order	-	-	Null = true, on _delete = set null.	Prescription	This is a foreign key.
address	Varchar	200	Not Null	-	Address of shipping.
date_added	Date	-	Auto add.	-	This will automatically record date and time.
city	Varchar	200	Not Null	-	city of shipping.
state	Varchar	200	Not Null	-	state of shipping.
zipcode	Varchar	200	Not Null	-	zipcode of shipping.
Contact no	Varchar	20	-	-	Phone number of shipping.

3.3.2 Functional Requirement

Context Diagram

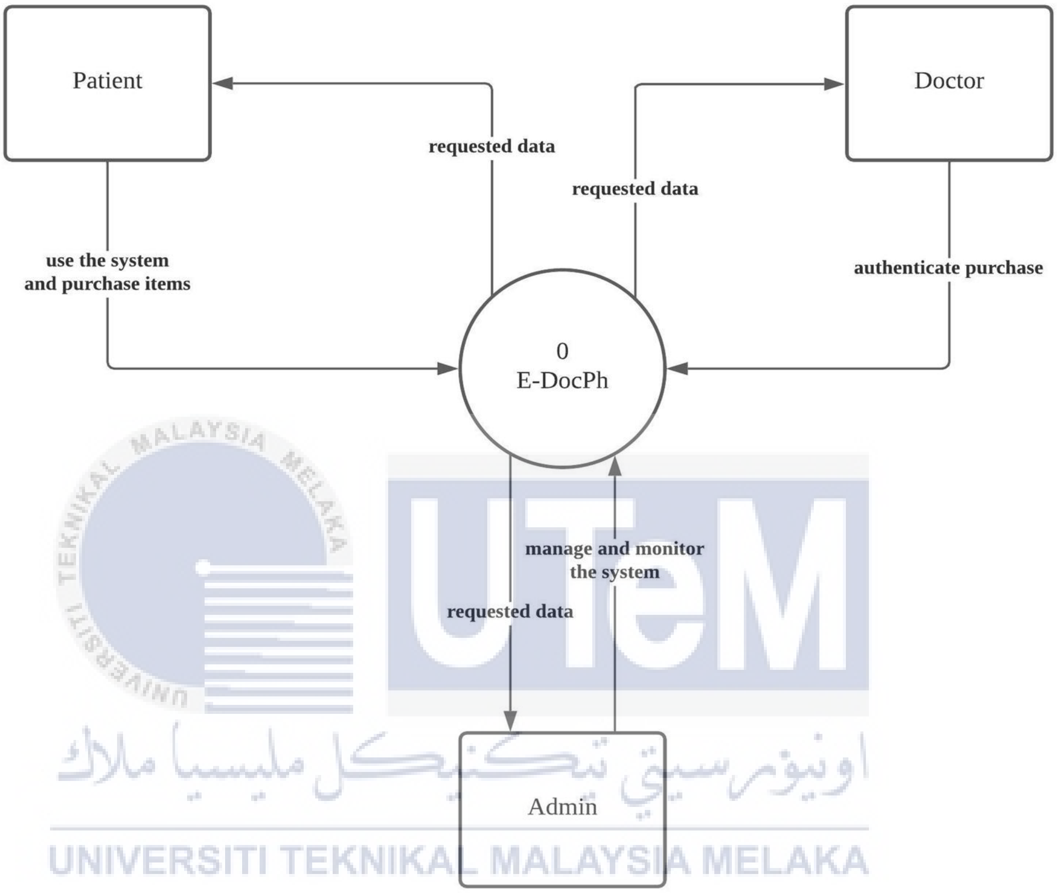


Figure 5: Context Diagram

Data-flow-diagram:

Level 0:

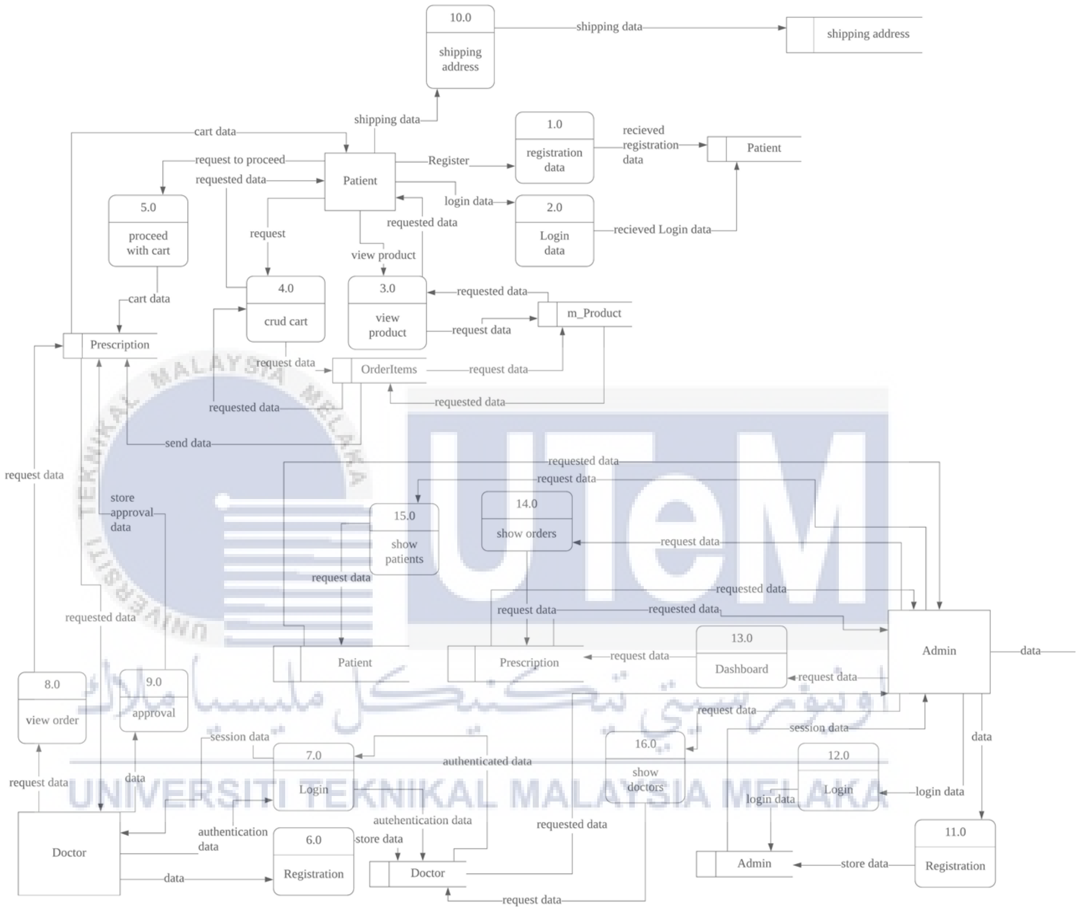


Figure 6: Data Flow Diagram level 0

Level 1:

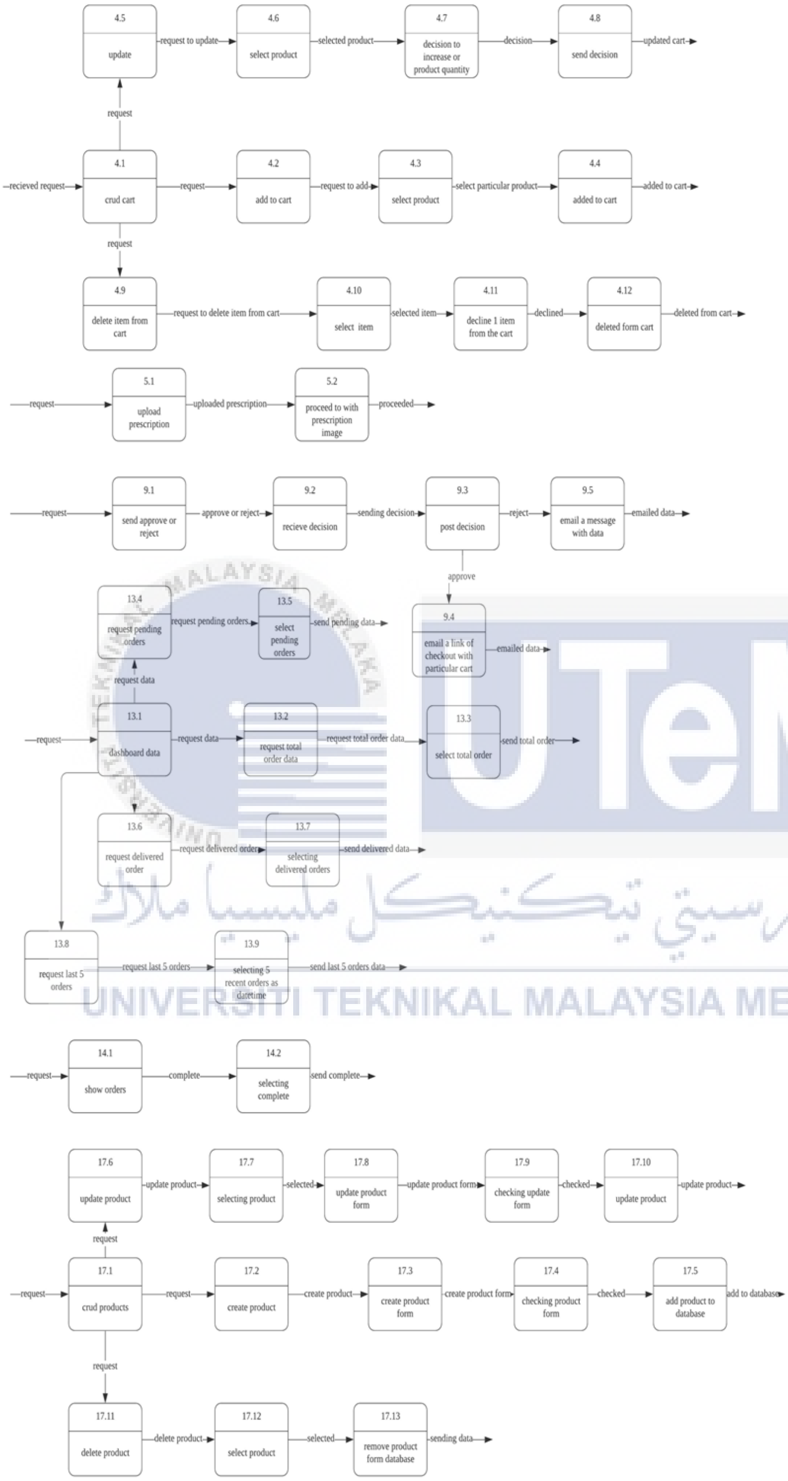


Figure 7: DFD level 1

3.3.3 Non-functional Requirement

Non-functional requirements are constraints on the services or functions offered by the system. They include timing constraints, constraints on the development process and constraints imposed by standards. This section describes the general non-functional requirements for E-DocPh system.

- ➢ Any operation on data in the system shall not take longer than 1s.
- ➢ Users will not be able to get access on their profile unless they are logged in.
- ➢ anyone will be able to see the system and purchase items without login the system as guest user.
- ➢ Any medicine type items need to be granted by a doctor before purchase.
- ➢ Any notification shall no longer take more than 1 seconds to reach target device.

3.4 Conclusion

In conclusion, this chapter describes about current system and their constraint. Where and how the lack of features is creating problems is well described in there. On the other hand, data dictionary took place in this chapter which illustrates how they should be stored in the system as well as their structures and their purposes. Mentioned of data, DFD (Data Flow Diagram) took place where the way of data and their functionality, transmit, records and behavior has been discussed in this chapter. Some non-fictional requirements are also mentioned. By completing this chapter now we are able to understand the data behavior of E-DocPh, system constraints of current system and functional, non-functional requirements of E-DocPh.

CHAPTER 4 : DESIGN

4.1 Introduction

In this section we will observe the designs regarding to the system. This chapter is going to include possible database designs and details. Mentioned database will describe how database is working in the system. The structure of data as well as relationship between different table.

Nevertheless, This chapter will contain system architecture where diagrams will illustrate how system will work from front-end and some explanation about back-end too. However more than one diagrams could be used in such matter.

Moreover, Interface design will be provided to understand the front-end easily. This section will contain some screenshot directly taken from the system. This way it will be easier to understand the flowchart as well as the whole system.

4.2 High-level Design

4.2.1 System Architecture

Flowchart:

Registration:

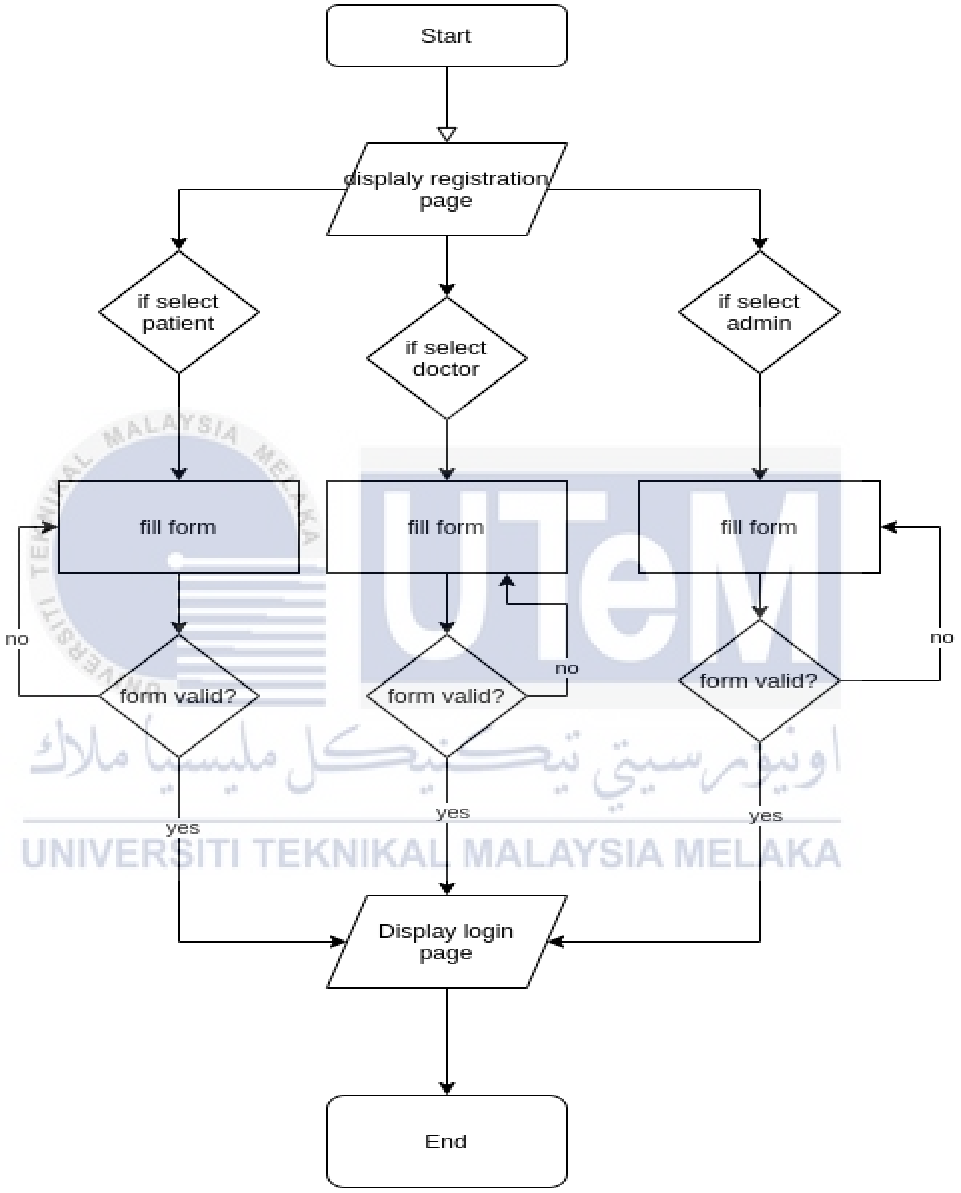


Figure 8: Registration

Login:

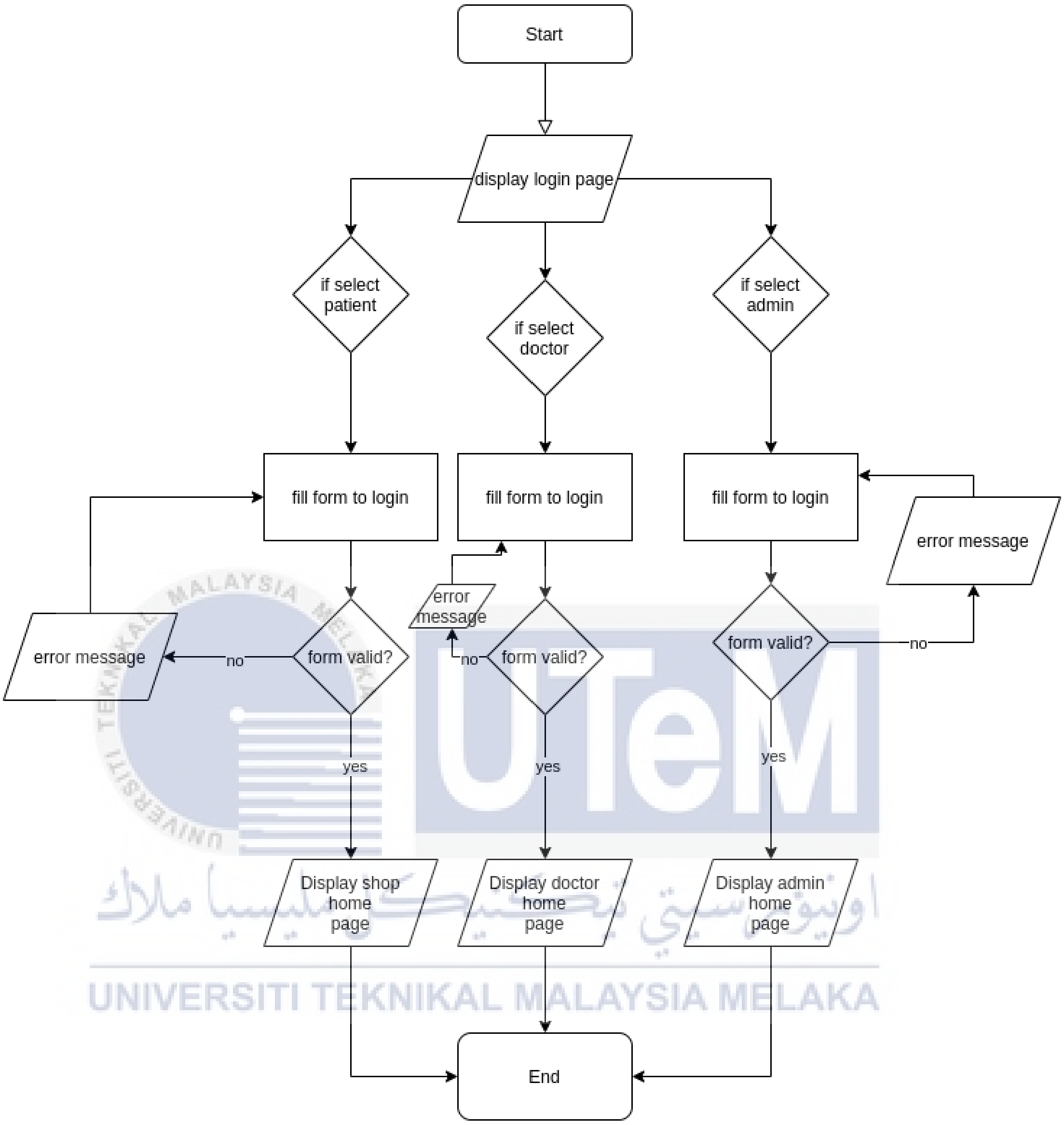


Figure 9: Login

Home:

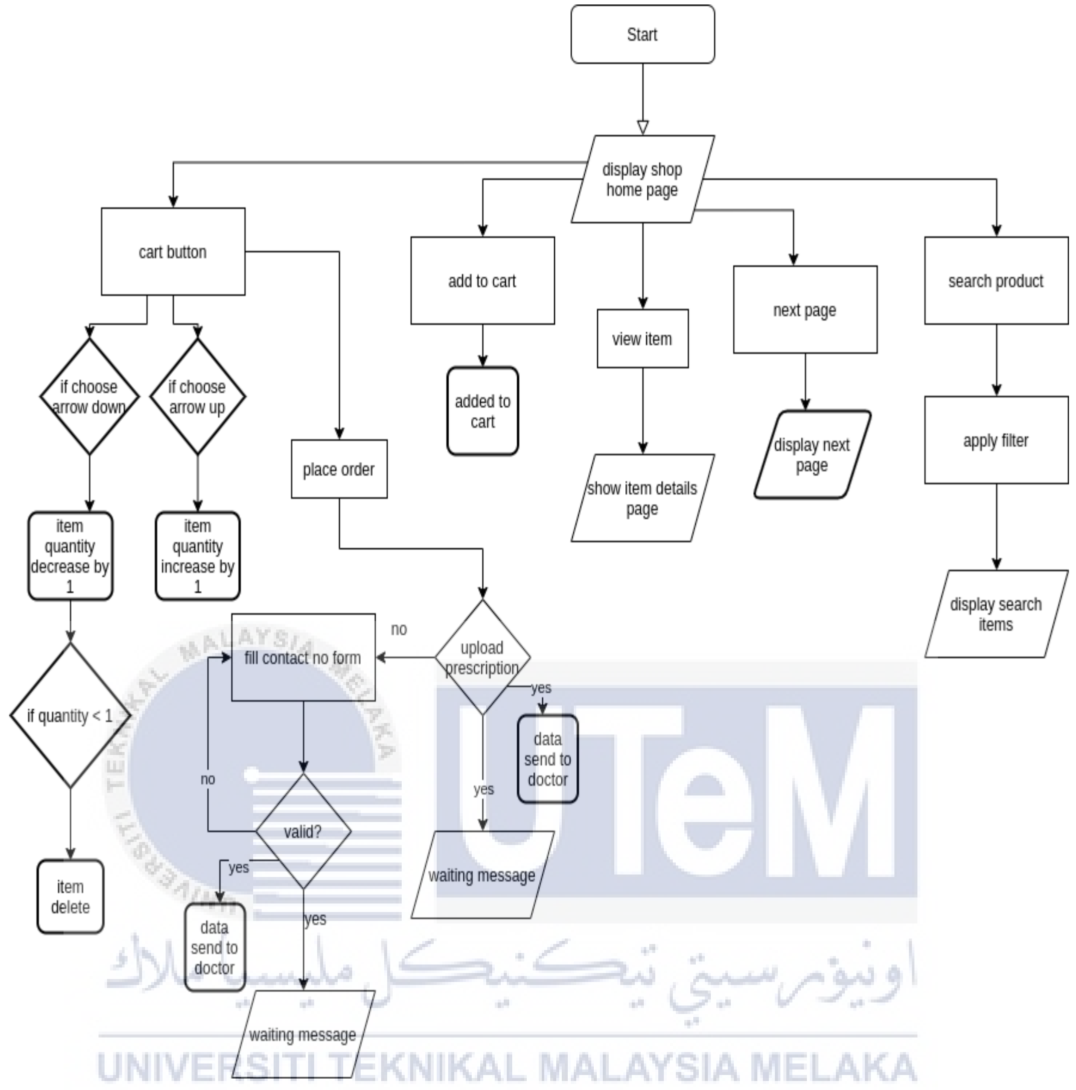


Figure 10: Home

Checkout:

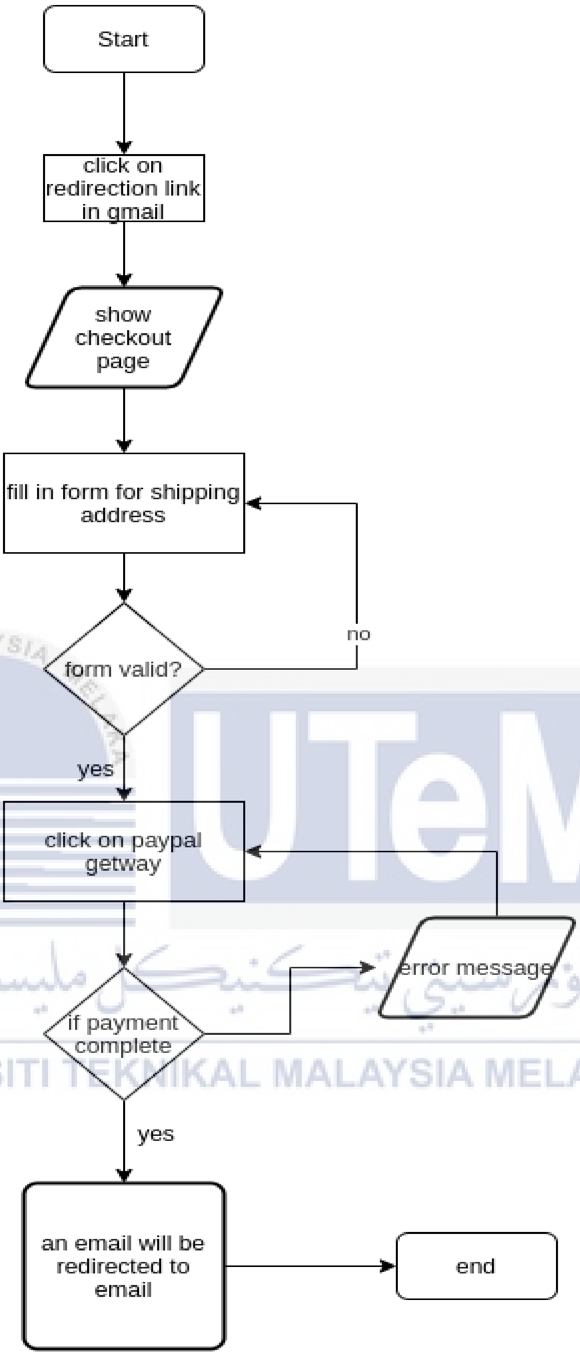

Start
click on redirection link in gmail
show checkout page
fill in form for shipping address
form valid?
no
yes
click on paypal getway
error message
if payment complete
yes
an email will be redirected to email
end

Figure 11: Checkout

Doctor:

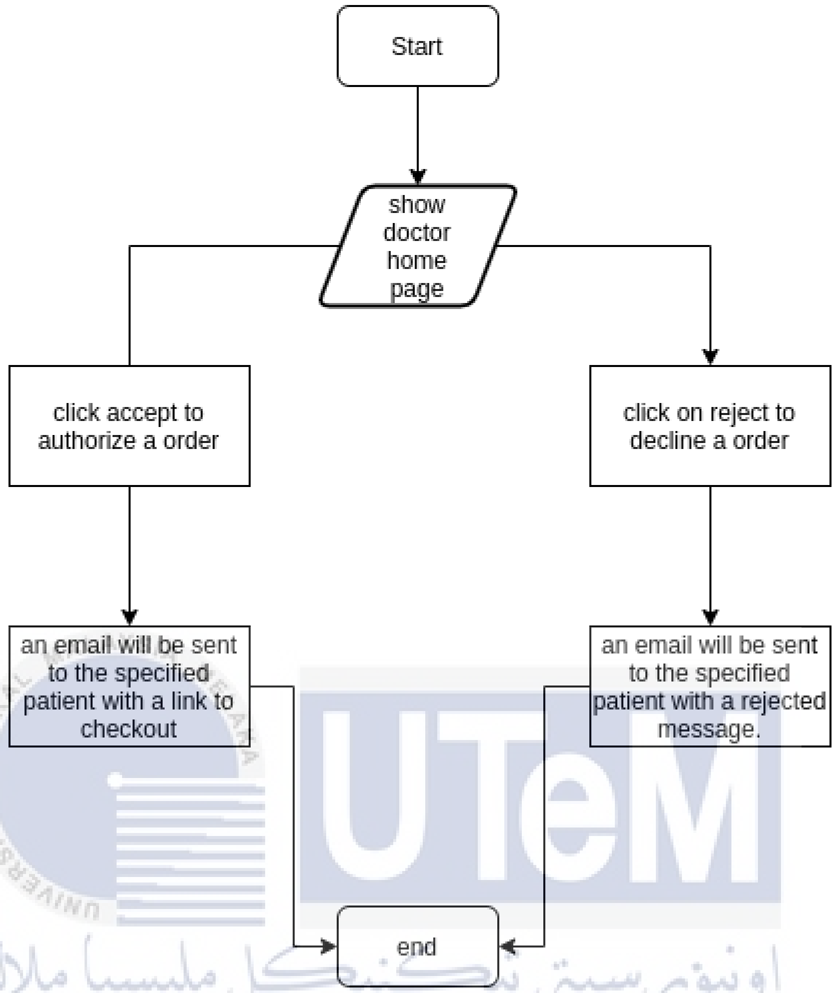

Start
show doctor home page
click accept to authorize a order
click on reject to decline a order
an email will be sent to the specified patient with a link to checkout
an email will be sent to the specified patient with a rejected message.
end

Figure 12: Doctor

Admin

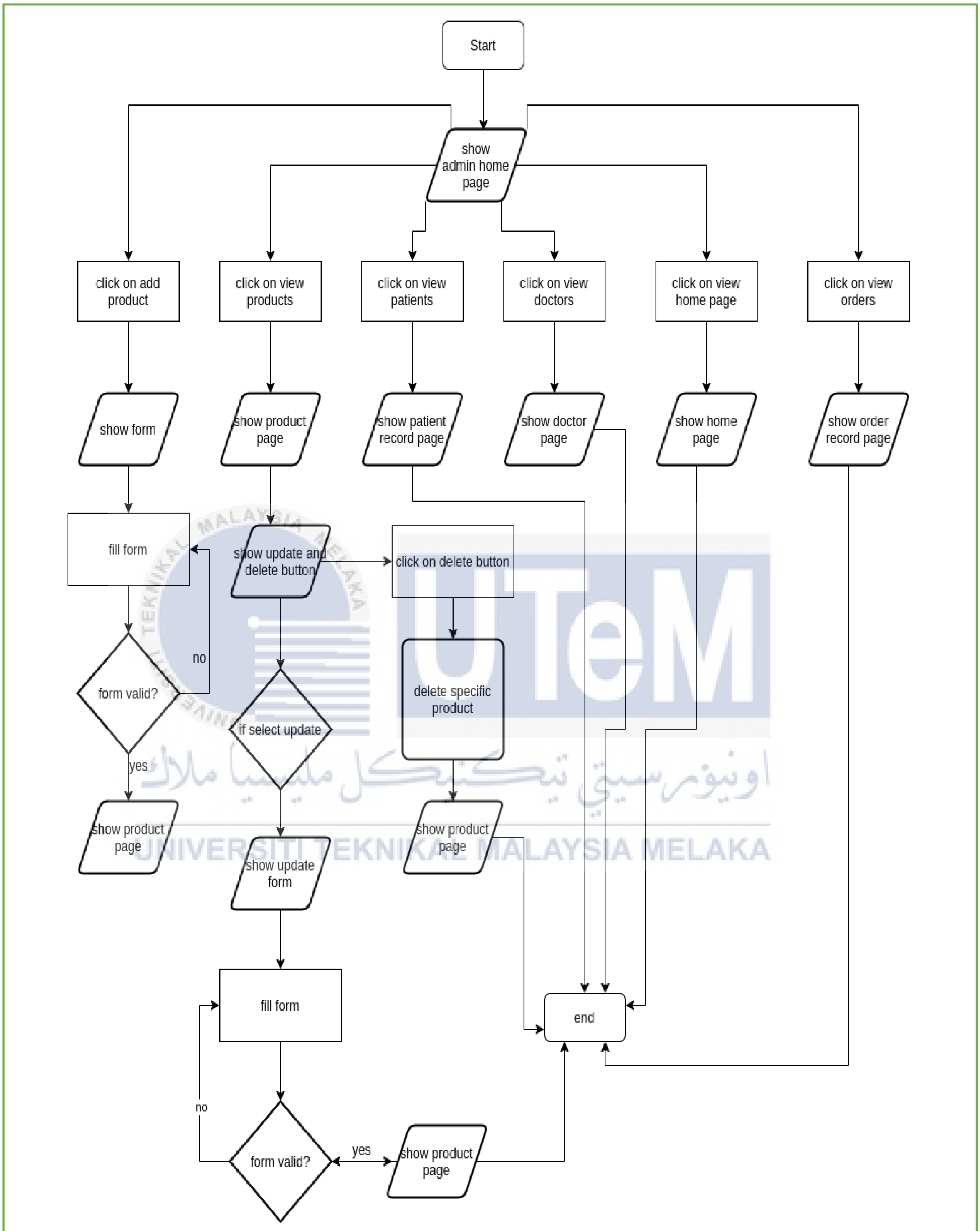


Figure 13: Admin

4.2.2 User Interface Design

Navigation Design:

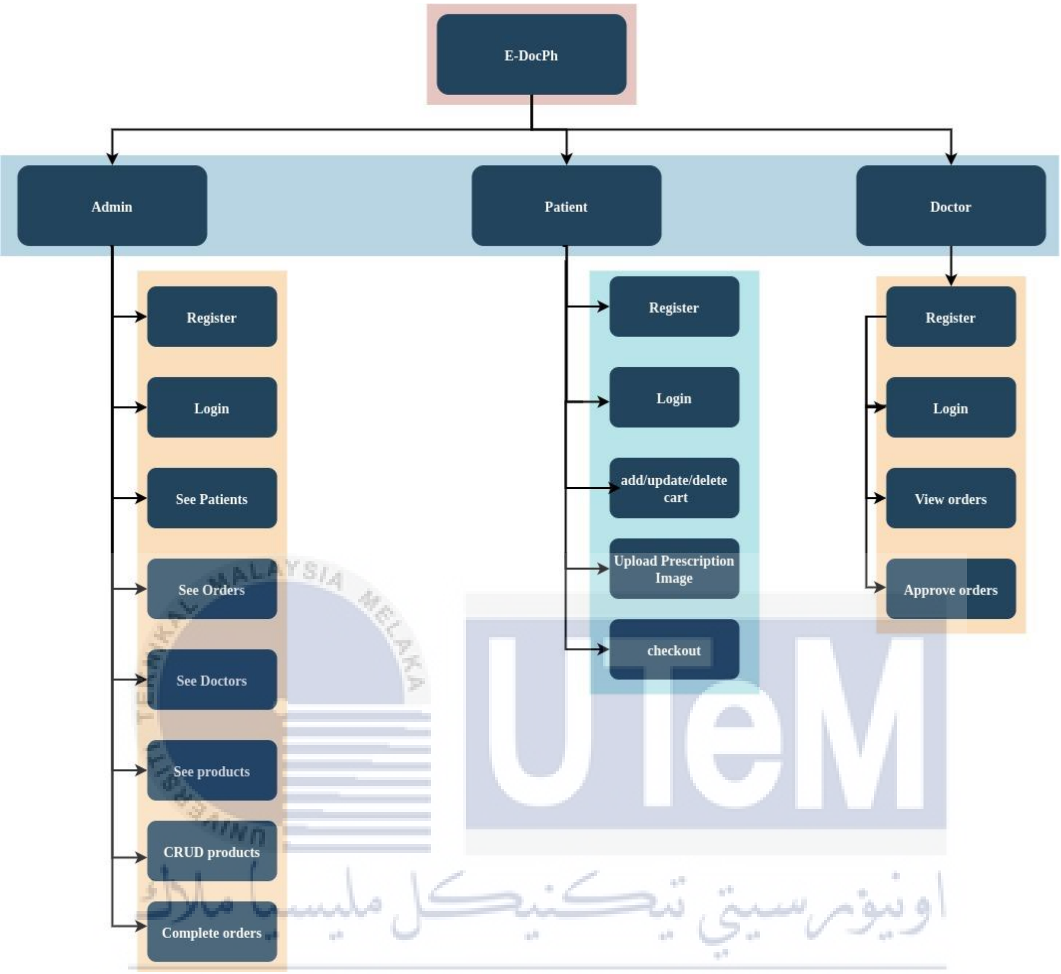
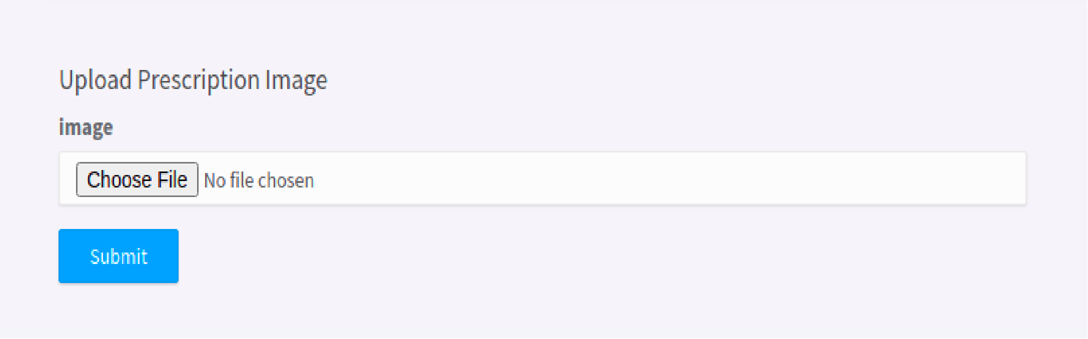


Figure 14: Navigational design

Input Design

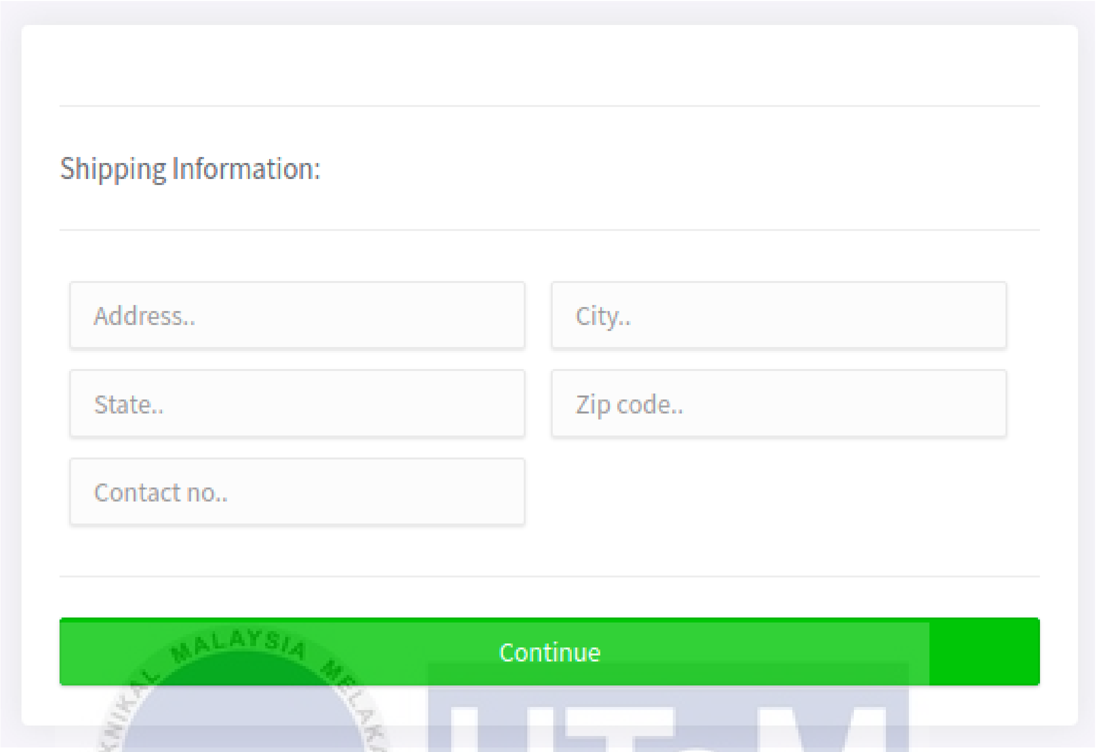


Upload Prescription Image

image

Figure 15: Prescription Upload

Here in figure 15 we can see a field where we can upload a image. This image would be the prescription image and patient need to choose the image file and submit it.



Shipping Information:

Address..

City..

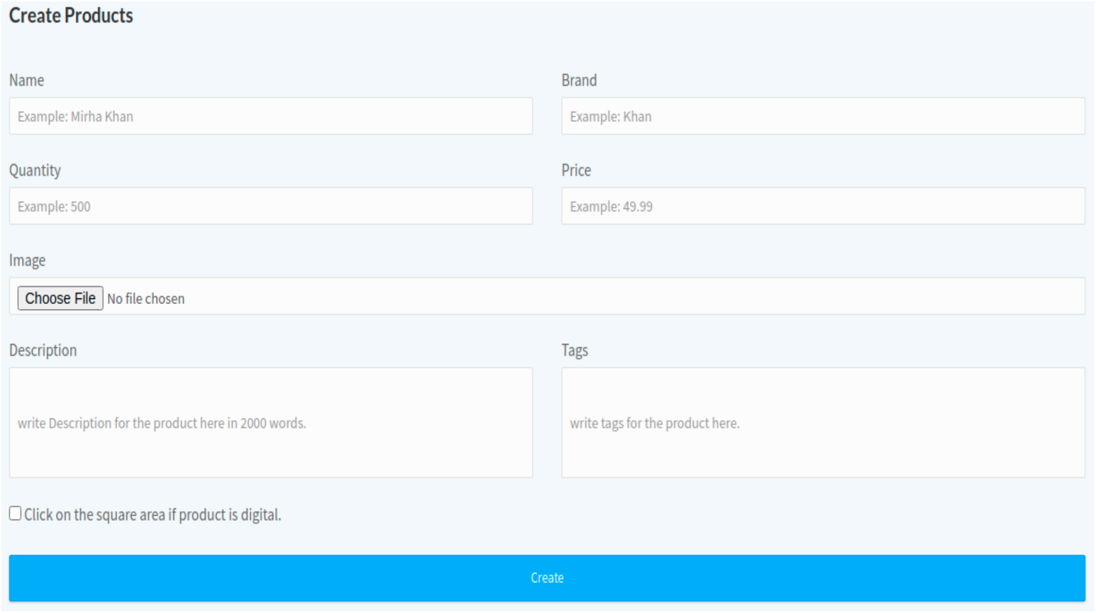
State..

Zip code..

Contact no..

Figure 16: shipping form for authenticated user

In figure 16 we can see the shipping form for authenticated patients. In the shipping information section patient need to type their address in respective field as datatype. Address, city, state and contact should be varchar and zip code should be integer.



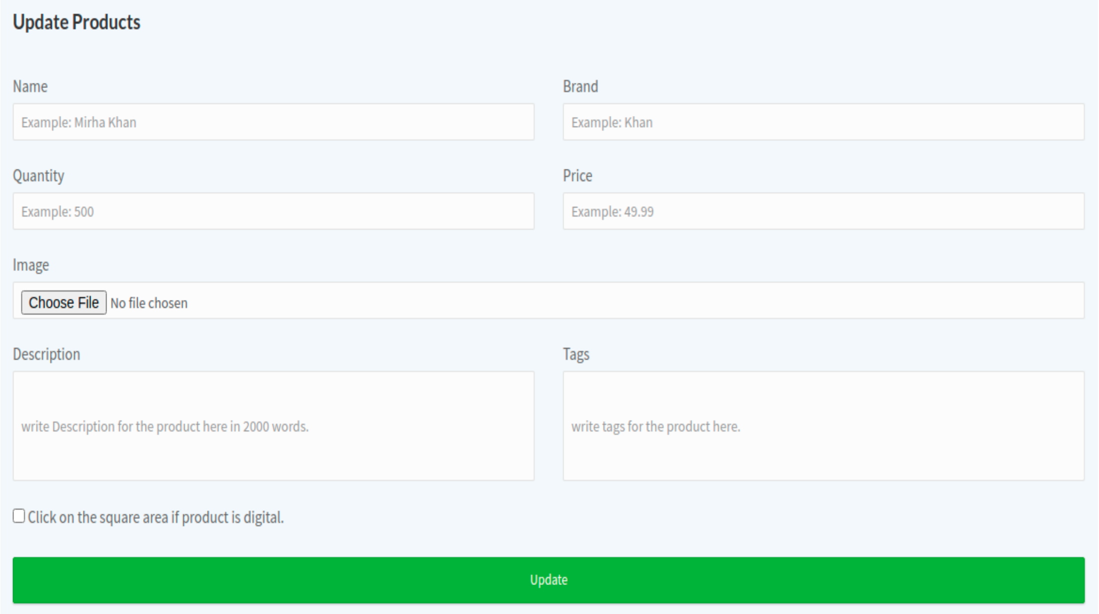
Create Products

Name Example: Mirha Khan	Brand Example: Khan
Quantity Example: 500	Price Example: 49.99
Image Choose File No file chosen	
Description write Description for the product here in 2000 words.	Tags write tags for the product here.

□ Click on the square area if product is digital.

Figure 17: add product

Here in figure 17 a form took place which is used to add products into the database. The name field is varchar, and it is used for product name attribute. Brand is varchar and used for brand name of the product. Quantity is integer field, Price is a decimal field, Image used for to upload the image of the product; admins need to upload a image file here. Description is basically some paragraph or any sort of texting method that describes about the product. Tags is varchar, and digital is a Boolean field. Digital is checkbox so once patient check it the value will be true.



Update Products

Name

Example: Mirha Khan

Brand

Example: Khan

Quantity

Example: 500

Price

Example: 49.99

Image

Choose File No file chosen

Description

write Description for the product here in 2000 words.

Tags

write tags for the product here.

□ Click on the square area if product is digital.

Update

Figure 18: update product

Here in figure 18 a form took place which is used to update products into the database. The name field is varchar, and it is used for product name attribute. Brand is varchar and used for brand name of the product. Quantity is integer field, Price is a decimal field, Image used for to upload the image of the product; admins need to upload a image file here. Description is basically some paragraph or any sort of texting method that describes about the product. Tags is varchar, and digital is a Boolean field. Digital is checkbox so once patient check it the value will be true.

4.2.3 Database Design

4.2.3.1 Conceptual and logical design

- ➢ Entity-Relationship Diagram

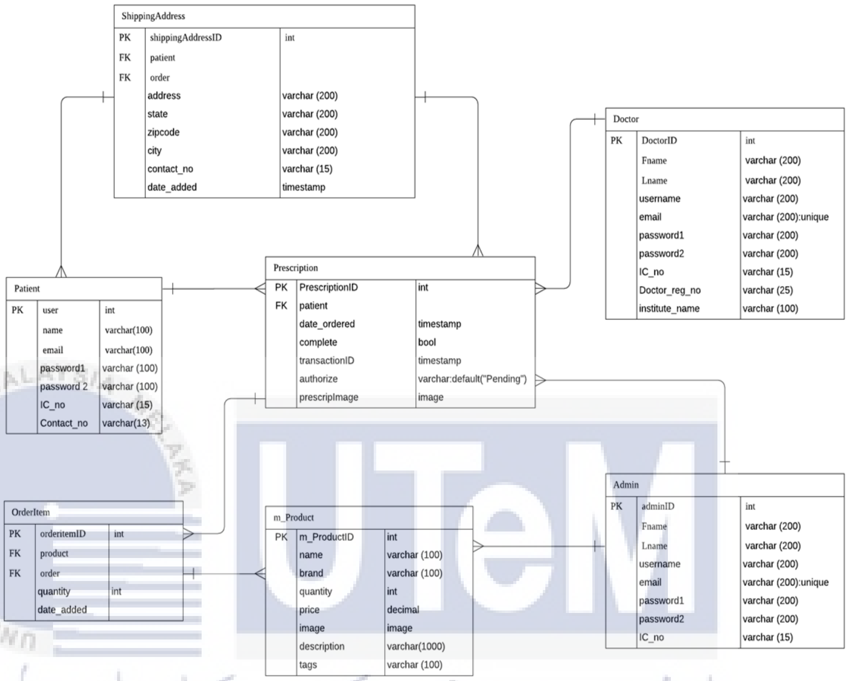


Figure 19: ERD Model

Above in figure 19 Entity-Relationship Diagram (ERD) we have seven different tables that are going to be used in order to develop E-DocPh. First of all we have patient table which has been made for patients or customer. In this table patient's data will be stored. As of diagram we can see that patient table is connected with two different tables named Prescription and ShippingAddress. Prescription table is used for to store all the data coming from another table OrderItem which is also a child table of Prescription. While customers will select any product add to their cart, those products will be stored in OrderItem table first under a prescriptionID which is a primary key for Prescription table. We can simply say about these two table that a customer can choose more than one product to buy. While they are choosing more than one product, each product will be stored in OrderItem table with different primary key and all these orderitem component will be stored under one PrescriptionID.

Speaking of products, m_product table is a table to store all the data of a product. This table is also a foreign key to OrderItem table. Going back to Prescription table, patient table is a foreign key to prescription table. In simple words prescriptionID is a component that owned by a patient and its data is coming from OrderItem table as list of products. Patients will be registered and logged in on basis of patient table.

Furthermore, a doctor can be registered and logged in through doctor table components. Doctor store all the data relevent to a doctor. While patients are done adding items in cart they can upload prescription in order to send the data to a doctor for authorization of their order. Again, in prescription table we have a attribute name authorize. This attribute is a varchar type attribute, and its default value is ‘pending’. While doctors want to see the data of a order, doctor can see all the products that has been ordered by a fellow patient; doctor can also see the prescription image that was uploaded by patient. On basis of prescription image, doctor can either accept or reject the order. Upon that decision of doctor patient will either get a link to redirect to checkout page where patient can pay for the products or get a rejection message.

As patient get a checkout link, it directs us to the shippingAddress table. This table contains all the data needed for the delivery of the products. Patients must fill up the form and submit it in order to get the payment gateway request. Once payment is done a transactionID will be set to the Prescription table.

Next, Admin table is a table to store the data of a admin. Admins are the users who monitor the system and maintain it. Admins can register and login through this table. Once a patient done with payment for their order, admins can see the transaction and can complete in order to save the order as done.

Data Dictionary and Normalization:

Table 8: Data Dictionary ShippingAddress

ShippingAddress					
Column Name	Data Type	Field Length	Constraint	FK Reference	Description
shippingAddressID	Integer	-	Primary key	-	Primary and unique for the table. Each shippingAddress will have their own unique primary key.
patient	-	-	Null = true, on _delete = set null.	Patient	This is a foreign key.
Order	-	-	Null = true, on _delete = set null.	Prescription	This is a foreign key.
address	Varchar	200	Not Null	-	Address of shipping.
date_added	Date	-	Auto add.	-	This will automatically record date and time.
city	Varchar	200	Not Null	-	city of shipping.
state	Varchar	200	Not Null	-	state of shipping.
zipcode	Varchar	200	Not Null	-	zipcode of shipping.
Contact no	Varchar	20	-	-	Phone number of shipping.

Table 9: Data Dictionary OrderItem

OrderItem					
Column Name	Data Type	Field Length	Constraint	FK Reference	Description
orderitemID	Int	-	Primary key	-	Primary and unique for the table.
product	-	-	Null = true, on _delete = set null.	m_Product	This is a foreign key.
Order	-	-	Null = true, on _delete = set null.	Prescription	This is a foreign key.
Quantity	Int	-	Not Null	-	Quantity of ordered item
date_added	Date	-	Auto add.	-	This will automatically record date and time.

Table 10: Data Dictionary Prescription

Prescription					
Column Name	Data Type	Field Length	Constraint	FK Reference	Description
prescriptionID	Int	-	Primary key	-	Primary and unique for the table.
patient	-	-	Null = true, on _delete = set null.	Patient	This is a foreign key.
Complete	bool	-	Not null	-	Default = False
TransactionID	Timestamp	-	Not Null	-	Transaction for order
Authorize	Varchar	-	Not Null	-	Default = Pending
date_ordered	Date	-	Auto add.	-	This will automatically record date and time.
PrescripImage	Image	-	null	-	This is for patients to upload prescription image.

Table 11: Data Dictionary Patient

Patient					
Column Name	Data Type	Field Length	Constraint	FK Reference	Description
user	Int	-	Primary key	-	Primary and unique for the table. Each patient will have their own unique primary key.
Name	Varchar	200	Not Null	-	Name of the patient.
Email	Varchar	200	Not null, unique = true	-	This will be the email of the patient.
Password	Varchar	50	-	-	This will be the password of the patient.
IC_no	Varchar	20	-	-	This is IC no of patient.
Contact_no	Varchar	13	-	-	This is contact number of patient.

Table 12: Data Dictionary m_Product

m_Product					
Column Name	Data Type	Field Length	Constraint	FK Reference	Description
m_productID	Int	-	Primary key	-	Primary and unique for the table. Each product will have their own unique primary key.
Name	Varchar	200	Not Null	-	Name of the product.
Brand	Varchar	200	Null = true.	-	Brand of the product.
quantity	Integer	-	Default = 0. null =true, blank =true.	-	This will be the number of the product are available to sale.
Price	Decimal	7,2	-	-	This will be the price of the product.
Digital	Boolean	-	Default = false, null = true, blank = true.	-	This is meant for products that are digital thing not physical. These products cannot be shipped.
Image	Image	-	Null = true. Blank = true.	-	This is for to upload saving path for product images.
Description	varchar	1000	-	-	This is to write description for product.
Tags	Varchar	100	-	-	This is originally meant to be a key for recommendation algorithm.

Table 13: Data Dictionary Doctor

Doctor					
Column Name	Data Type	Field Length	Constraint	FK Reference	Description
DoctorID	Int	-	Primary key	-	Primary and unique for the table.
Fname	Varchar	200	Not Null	-	First name of the doctor.
Lname	Varchar	200	Not Null	-	Last name of the doctor.
Username	Varchar	200	Not null	-	Username for the account.
email	varchar	200	Unique	-	Email of the doctor.
password1	varchar	200	-	-	password
password2	varchar	200	-	-	password
IC_no	Varchar	15	-	-	This is IC no of doctor.
Doctor_reg_no	varchar	25	-	-	This is the registration number of the doctor
institute_name	Varchar	100	-	-	Institution name where doctor practice.

Table 14: Data Dictionary Admin

Admin					
Column Name	Data Type	Field Length	Constraint	FK Reference	Description
AdminID	Int	-	Primary key	-	Primary and unique for the table.
Fname	Varchar	200	Not Null	-	First name of the admin.
Lname	Varchar	200	Not Null	-	Last name of the admin.
Username	Varchar	200	Not null	-	Username for the account.
email	varchar	200	Unique	-	Email of the admin.
password1	varchar	200	-	-	password
password2	varchar	200	-	-	password
IC_no	Varchar	15	-	-	This is IC no of admin.

4.3 Detailed Design

4.3.1 Software Design

Table 15: Register patient

Program Specification 1.0 for registration data	
Name	Register patients
Purpose	Registering patients into the system.
Programmer	Abdul Hadi Mazbah
Languages	HTML, CSS JAVASCRIPT, PYTHON3
Events	patients need to fill the form and click on register button.
Input name	**Type**
Email	email
Username	Username
Password1	Password1
Password2	Password2
Output name	**Type**
Email	email
Username	username
Password1	password1
Password2	password2
Pseudocode	

```
(registrationPage)
if user is authenticated == true
return redirect (store)
else
if form is valid == true
return redirect(loginpatient)
```

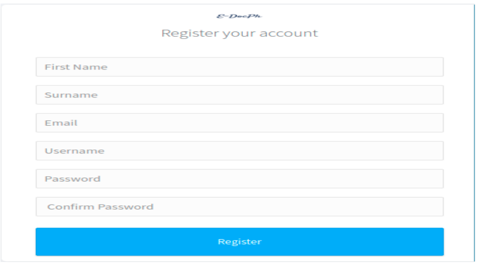
Screen	 E-DocPh Register your account First Name Surname Email Username Password Confirm Password Register

Table 16: Register Doctor

Program Specification 6.0 for registration	
Name	Register doctors
Purpose	Registering doctors into the system.
Programmer	Abdul Hadi Mazbah
Languages	HTML, CSS JAVASCRIPT, PYTHON3
Events	doctors need to fill the form and click on register button.
Input name	**Type**
Email	email
Username	Username
Password1	Password1
Password2	Password2
Output name	**Type**
Email	email
Username	username
Password1	password1
Password2	password2
Pseudocode	

```
(registrationPage)
if user is authenticated == true
return redirect (doctor)
else
if form is valid == true
return redirect(logindoctor)
```

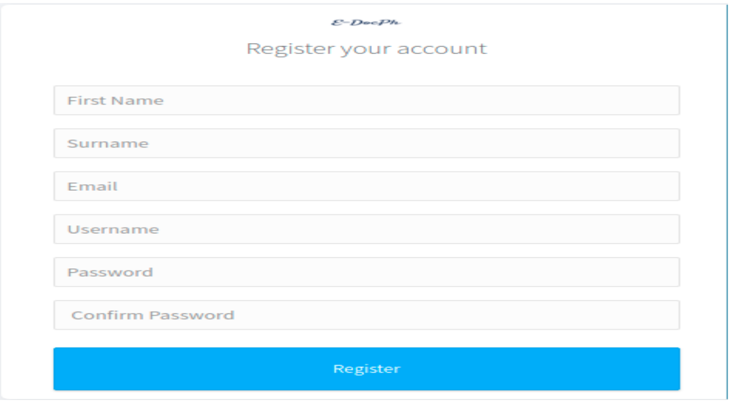
Screen	 E-DocPh Register your account First Name Surname Email Username Password Confirm Password Register

Table 17: Register Admins

Program Specification 11.0 for registration	
Name	Register admins
Purpose	Registering admins into the system.
Programmer	Abdul Hadi Mazbah
Languages	HTML, CSS JAVASCRIPT, PYTHON3
Events	admins need to fill the form and click on register button.
Input name	**Type**
Email	email
Username	Username
Password1	Password1
Password2	Password2
Output name	**Type**
Email	email
Username	username
Password1	password1
Password2	password2
Pseudocode	
(registrationPage) if user is authenticated == true	

```
return redirect (dashboard)
else
if form is valid == true
return redirect(login)
```

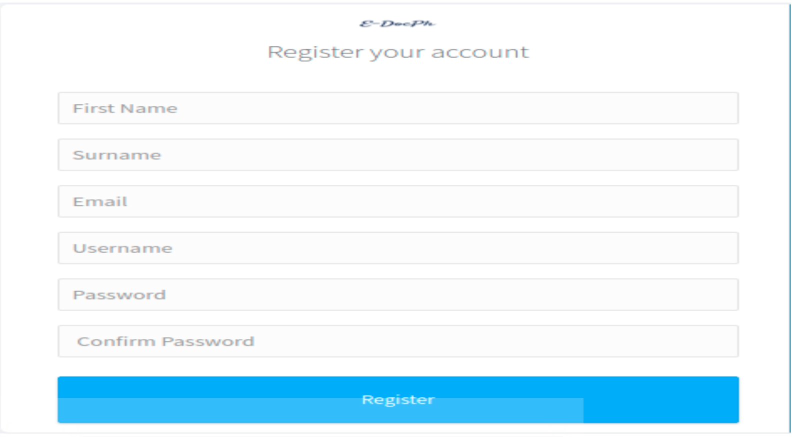
Screen	 E-DocPh Register your account First Name Surname Email Username Password Confirm Password Register

Table 18: login data

Program Specification 2.0 for login data		
Name	login patient	
Purpose	login patient into the system.	
Programmer	Abdul Hadi Mazbah	
Languages	HTML, CSS JAVASCRIPT, PYTHON3	
Events	patient need to fill the form and click on login button.	
Input name		**Type**
Username		username
Password1		password
Output name		**Type**
Username		username
Password1		password
Pseudocode		

```
(LoginPage)
if user is authenticated == true
```

return redirect (store)
else
if username = username && password1 = password1
login(user)
return redirect(store)	

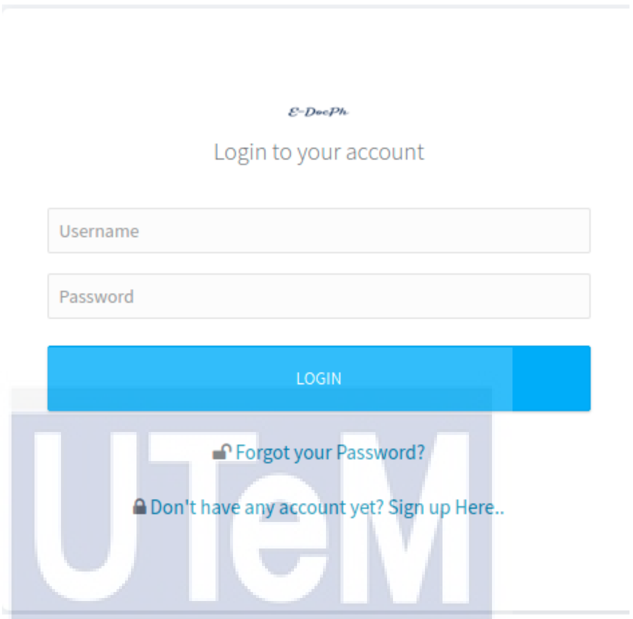
Screen	 E-DocPh Login to your account Username Password LOGIN Forgot your Password? Don't have any account yet? Sign up Here..

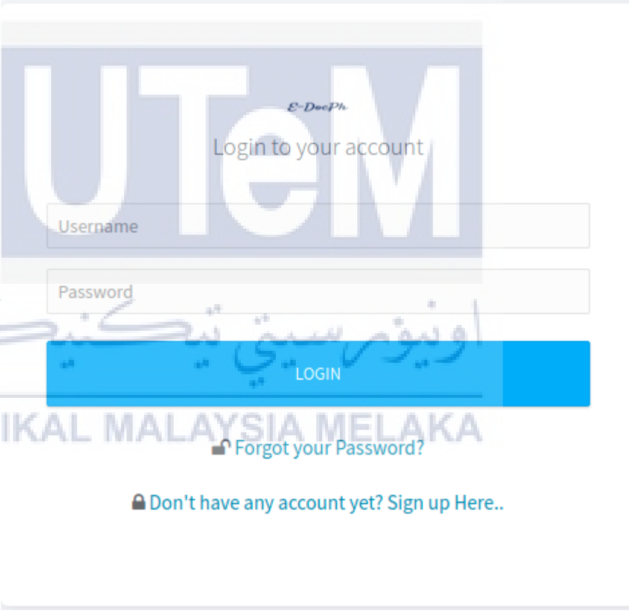
Table 19: login doctor

Program Specification 7.0 for login		
Name	login doctor	
Purpose	login doctor into the system.	
Programmer	Abdul Hadi Mazbah	
Languages	HTML, CSS JAVASCRIPT, PYTHON3	
Events	doctor need to fill the form and click on login button.	
Input name		**Type**
Email		email
Password1		password

Output name	**Type**
Email	email
Password1	password

Pseudocode

```
(LoginPage)
if user is authenticated == true
return redirect (doctor)
else
if email = email && password1 = password1
login(user)
return redirect(doctor)
```

Screen	

E-DocPh

Login to your account

Username

Password

LOGIN

Forgot your Password?

Don't have any account yet? Sign up Here..

Table 20: login admins

Program Specification 12.0 for login	
Name	login admins
Purpose	login admins into the system.
Programmer	Abdul Hadi Mazbah

Languages	HTML, CSS JAVASCRIPT, PYTHON3
Events	admins need to fill the form and click on login button.
Input name	**Type**
Email	email
Password1	password
Output name	**Type**
Email	email
Password1	password

Pseudocode

```
(LoginPage)
if user is authenticated == true
return redirect (dashboard)
else
if email = email && password1 = password1
login(user)
return redirect(dashboard)
```

Screen

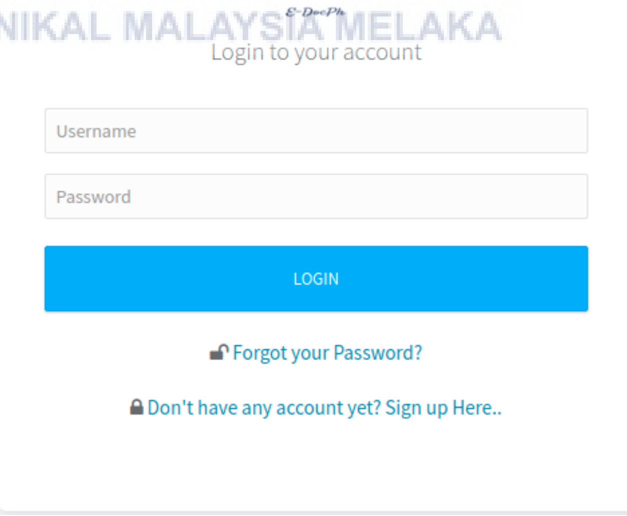


Table 21: view product

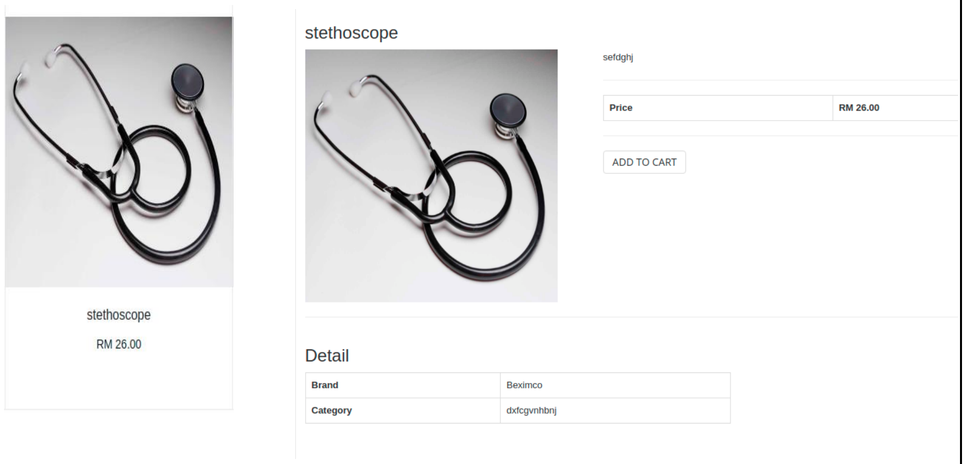
Program Specification 3.0 for view product	
Name	View product
Purpose	For patients to see product detail page
Programmer	Abdul Hadi Mazbah
Languages	HTML, CSS JAVASCRIPT, PYTHON3
Events	Patients need to click on view button of a product card.
Input name	**Type**
Click on view button	click
Output name	**Type**
Product detail page	Data return as array
Pseudocode	
(Product detail page) product = product.id for product in products product.data	
Screen	 stethoscope RM 26.00 stethoscope sefdghj Price RM 26.00 ADD TO CART Detail Brand Beximco Category dxfcgvnhbnj

Table 22: cart

Program Specification 4.0 for crud cart	
Name	Crud cart
Purpose	For patients to crud their cart
Programmer	Abdul Hadi Mazbah
Languages	HTML, CSS JAVASCRIPT, PYTHON3
Events	Patients need to click on add to cart button of a product card. Click on cart button. Patient need to click on up or down arrow to increase or reduce quantity of the product.
Input name	**Type**
Click on add to cart button	click
Click on up or down arrow	click
Click on cart button	click
Output name	**Type**
Added to cart	Data return as array
Update cart	Data return as array
Delete cart	Data remove from cart
Pseudocode	

```
(cart)
product = product.id
item = []
for product in products
product.data
product.append(item)
if down
item -= item
if up
```

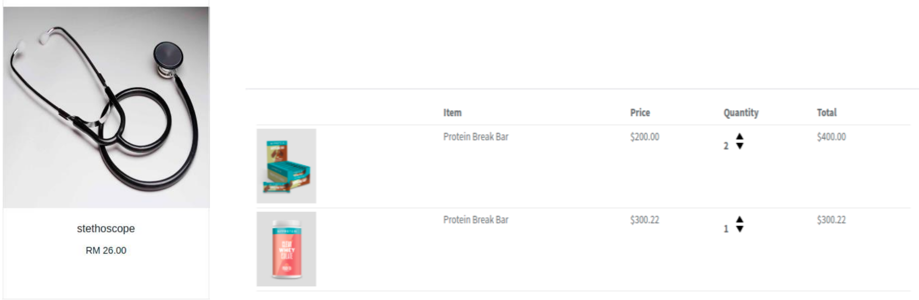
item += item	
Screen	 stethoscope RM 26.00 Item | Price | Quantity | Total Protein Break Bar | $200.00 | 2 | $400.00 Protein Break Bar | $300.22 | 1 | $300.22

Table 23: proceed with order

Program Specification 5.0 for proceed with cart		
Name	proceed with cart	
Purpose	Upload prescription image file form	
Programmer	Abdul Hadi Mazbah	
Languages	HTML, CSS JAVASCRIPT, PYTHON3	
Events	patient need to fill the form and click on submit button.	
Input name		**Type**
prescipImage		image
Output name		**Type**
Uploaded image		Image
Pseudocode		
(pres_up) if user is authenticated == true if form is valid = true form.save return redirect(store)		

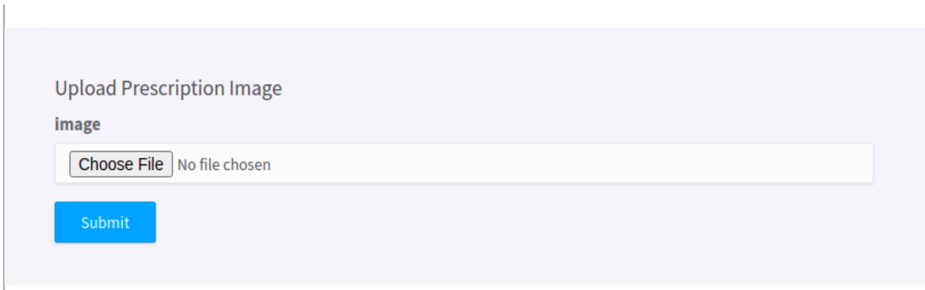
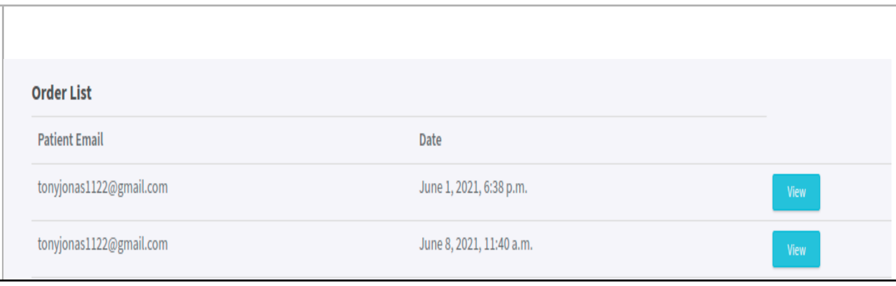
Screen	 Upload Prescription Image image Choose File No file chosen Submit

Table 24: view order

Program Specification 8.0 for view order	
Name	View order
Purpose	For doctor to see order detail page
Programmer	Abdul Hadi Mazbah
Languages	HTML, CSS JAVASCRIPT, PYTHON3
Events	doctors need to click on view button.
Input name	**Type**
Click on view button	click
Output name	**Type**
Order detail page	Data return as array
Pseudocode	

```
(Order detail page)
orders = order.id
for order in orders
order.data
```

Screen	

Order List

Patient Email	Date	
tonyjonas1122@gmail.com	June 1, 2021, 6:38 p.m.	View
tonyjonas1122@gmail.com	June 8, 2021, 11:40 a.m.	View

Table 25: order approval

Program Specification 9.0 for approval	
Name	Approval
Purpose	For doctors to approve or reject a order
Programmer	Abdul Hadi Mazbah
Languages	HTML, CSS JAVASCRIPT, PYTHON3
Events	Doctor need to click on accept button. Click on reject button.
Input name	**Type**
Click on accept button	String (Accepted)
Click on reject button	String (Rejected)
Output name	**Type**
Order accepted	Accepted
Order rejected	Rejected
Pseudocode	

(approval)
data = {}
product = prescription.id
decision = product.get(authorize)
 	
if data == accepted
product.decision = Accepted
return product.decision
elif data == rejected
product.decision == Rejected
return product.decision	

Screen	

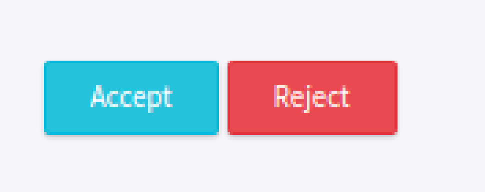
	

Table 26: shipping address

Program Specification 10.0 for shipping address	
Name	Shipping address
Purpose	Patients to fill up shipping address form
Programmer	Abdul Hadi Mazbah
Languages	HTML, CSS JAVASCRIPT, PYTHON3
Events	patient need to fill the form and click on continue button.
Input name	**Type**
Address	String (200)
City	String
State	String
Zipcode	String
Contact no	String
Output name	**Type**
Address	String (30)
City	String (9)
State	String (6)
Zipcode	String (5)
Contact no	String (13)
Pseudocode	

```
(shippingAddress)
if user is authenticated == true
if form is valid = true
form.save
```

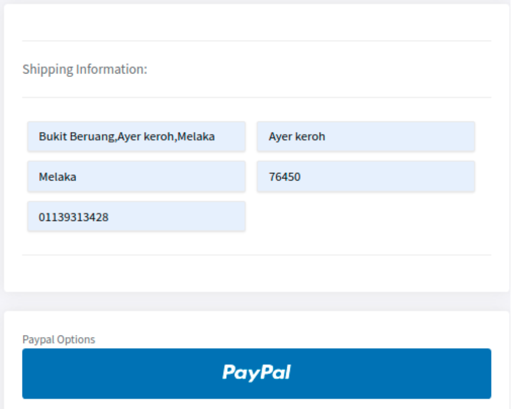
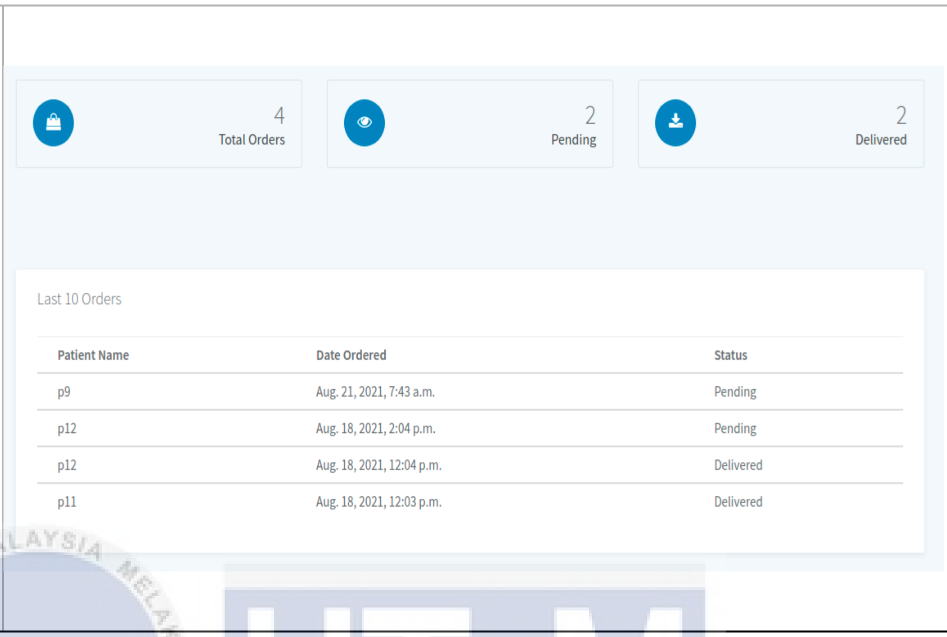
return redirect(store)	
Screen	 Shipping Information: Bukit Beruang,Ayer keroh,Melaka Ayer keroh Melaka 76450 01139313428 Paypal Options PayPal

Table 27: Dashboard

Program Specification 13.0 for dashboard	
Name	View dashboard
Purpose	For admins to see dashboard
Programmer	Abdul Hadi Mazbah
Languages	HTML, CSS JAVASCRIPT, PYTHON3
Events	admins need to click on dashboard button.
Input name	**Type**
Click on dashboard button	click
Output name	**Type**
Dashboard page	Data return as array
Pseudocode	

```
(dashboard)
orders = prescription.id
lastfive = order[:5]
total = orders.count()
pending = orders.get(complete=false).count()
delivered = orders.get(complete = true).count()
```

for order in orders order.data	
Screen	

4 Total Orders

2 Pending

2 Delivered

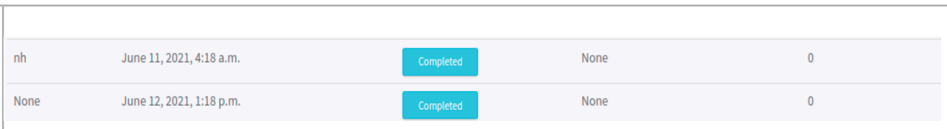
Last 10 Orders

Patient Name	Date Ordered	Status
p9	Aug. 21, 2021, 7:43 a.m.	Pending
p12	Aug. 18, 2021, 2:04 p.m.	Pending
p12	Aug. 18, 2021, 12:04 p.m.	Delivered
p11	Aug. 18, 2021, 12:03 p.m.	Delivered

Table 28: show orders

Program Specification 13.0 for show orders	
Name	View orders
Purpose	For admins to see all the orders
Programmer	Abdul Hadi Mazbah
Languages	HTML, CSS JAVASCRIPT, PYTHON3
Events	admins need to click on show orders button. Admins need to click on complete if order is done delivering.
Input name	**Type**
Click on orders button	click
Click on complete button	False
Output name	**Type**
order page	Data return as array
completed	True
Pseudocode	

```
(show_orders)
orders = prescription.id
for order in orders
order.data
if order.complete == false
order.complete = true
return order.complete
```

Screen	

nh	June 11, 2021, 4:18 a.m.	Completed	None	0
None	June 12, 2021, 1:18 p.m.	Completed	None	0

Table 29: show patients

Program Specification 13.0 for show patients	
Name	View patients
Purpose	For admins to see all the patients
Programmer	Abdul Hadi Mazbah
Languages	HTML, CSS JAVASCRIPT, PYTHON3
Events	admins need to click on show patients button.
Input name	**Type**
Click on patients button	click
Output name	**Type**
patients page	Data return as array
Pseudocode	

```
(show_orders)
orders = patient.all()
for order in orders
order.data
```

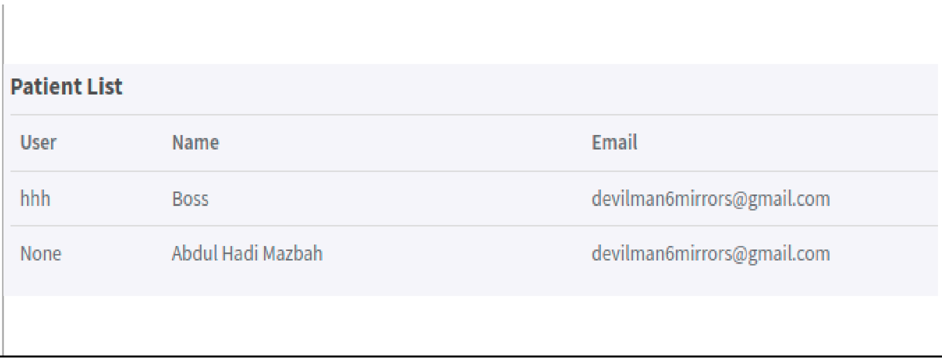
Screen	 Patient List User / Name / Email hhh / Boss / devilman6mirrors@gmail.com None / Abdul Hadi Mazbah / devilman6mirrors@gmail.com

Table 30: crud products

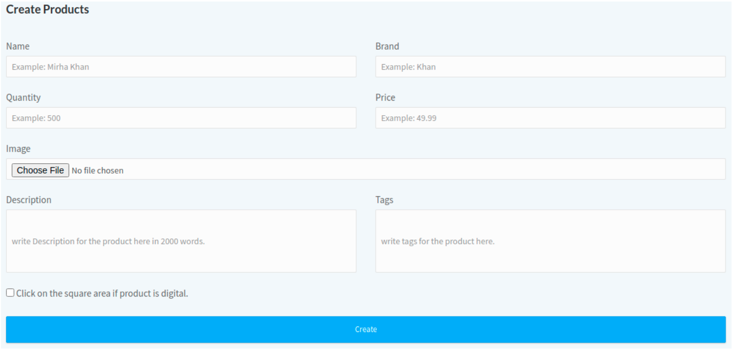
Program Specification 17.0 for crud product	
Name	crud product
Purpose	Admins to fill up create and update product form and delete product
Programmer	Abdul Hadi Mazbah
Languages	HTML, CSS JAVASCRIPT, PYTHON3
Events	Admins need to fill the form and click on submit and upload button respectively.
Input name	**type**
Name	String (200)
Brand	String
Quantity	Int
Price	Decimal(7,2)
Image	image
Description	String (1000)
Tags	String (100)
Digital	boolean
Click on delete button	click
Output name	**Type**
Name	String (15)
Brand	String (15)
Quantity	Int

Price	Decimal(3,2)
Image	image
Description	String (189)
Tags	String (10)
Digital	Boolean
Clicked on delete button	deleted

Pseudocode

```
(crud_products)

if user is authenticated == true

from = createForm()
if form is valid = true
form.save
return redirect(show products)

form1 = updateForm()
if form1 is valid == true
form.save()
return redirect(show products)

product = m_Product.id
onclick == ‘delete’
product.remove()
```

Screen	



Create Products

Name: Example: Mirha Khan

Brand: Example: Khan

Quantity: Example: 500

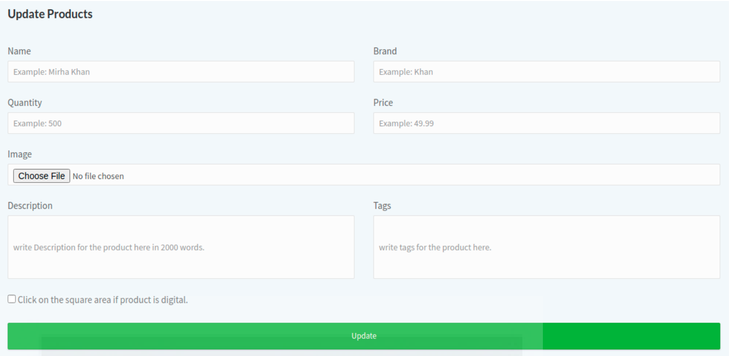
Price: Example: 49.99

Image: Choose File No file chosen

Description: write Description for the product here in 2000 words.

Tags: write tags for the product here.

□ Click on the square area if product is digital.



Update Products

Name: Example: Mirha Khan

Brand: Example: Khan

Quantity: Example: 500

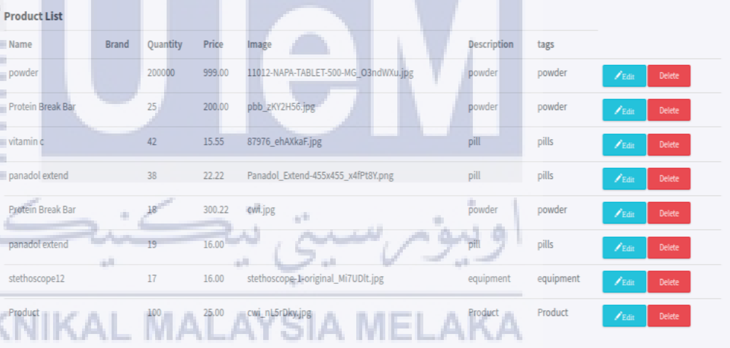
Price: Example: 49.99

Image: Choose File No file chosen

Description: write Description for the product here in 2000 words.

Tags: write tags for the product here.

□ Click on the square area if product is digital.



Product List

Name	Brand	Quantity	Price	Image	Description	tags	
powder		200000	999.00	11012-NAPA-TABLET-500-MG_O3ndWXu.jpg	powder	powder	Edit Delete
Protein Break Bar		25	200.00	pbb_zKY2H56.jpg	powder	powder	Edit Delete
vitamin c		42	15.55	87976_ehAXkaF.jpg	pill	pills	Edit Delete
panadol extend		38	22.22	Panadol_Extend-455x455_x4fPt8Y.png	pill	pills	Edit Delete
Protein Break Bar		18	300.22	cwi.jpg	powder	powder	Edit Delete
panadol extend		19	16.00		pill	pills	Edit Delete
stethoscope12		17	16.00	stethoscope-1-original_Mi7UDlt.jpg	equipment	equipment	Edit Delete
Product		100	25.00	cwi_nL5rDky.jpg	Product	Product	Edit Delete

4.3.2 Physical Database Design

Table 31: P.D.D Admin

Admin			
Column Name	Constraint	Example	Validation
AdminID	Primary key	0001	
Fname	Not Null	mark	Varhcar <= 200
Lname	Not Null	Hamil	Varhcar <= 200
Username	Not null	mark	Varhcar <= 200
email	Unique	mark@gmail.com	Varhcar <= 200
password1	Not null	*********	Password2 must be same.
password2	Not null	*********	Password1 must be same.
IC_no	Not null	BN612561	Varhcar <= 15

Table 32: P.D.D Doctor

Doctor			
Column Name	Constraint	Example	Validation
DoctorID	Primary key	0001	
Fname	Not Null	mark	Varhcar <= 200
Lname	Not Null	Hamil	Varhcar <= 200
Username	Not null	mark	Varhcar <= 200
email	Unique	mark@gmail.com	Varhcar <= 200
password1	Not null	*********	Password2 must be same.
password2	Not null	*********	Password1 must be same.
IC_no	Not null	BN612561	Varhcar <= 15
Doctor_reg_no	Not null	15669	Varhcar <= 25
institute_name	Not null	UPM	Varhcar <= 100

Table 33: P.D.D m_Product

m_Product			
Column Name	Constraint	Example	Validation
m_product ID	Primary key	0001	
Name	Not Null	napa	Varchar < =200
Brand	Null = true.	Square	Varchar < =200
quantity	Default = 0. null =true, blank =true.	200	Integer
Price	Not null	150.49	Decimal

Digital	Default = false,
	null = true, blank = true.
		False	Boolean

Image	Null = true. Blank = true.	Image file	image
Description	Null = true	This is napa tablet.	Varchar < =1000
Tags	Null = true	Pills, medicine.	Varchar < =100

Table 34: P.D.D Patient

Patient			
Column Name	Constraint	Example	Validation
user	Primary key	0001	
Name	Not Null	Bill	Varchar (200)
Email	Not null, unique = true	patient@gmail.com	Varchar (200)
Password	Not null	********	Password must be included with at least 1 character and alphabet and number.
IC_no	Not null	AF115665	Varchar (25)
Contact_no	Not null	+601548946679	Varchar (13)

Table 35: P.D.D Prescription

Prescription			
Column Name	Constraint	Example	Validation
prescriptionID	Primary key	0001	
patient	Null = true, on _delete = set null.	Bill	
Complete	Not null	False	Boolean
TransactionID	Not Null	1502546.154612	auto
Authorize	Not Null	Pending	Pending / Accepted / Rejected
date_ordered	Auto add.	15/05/2021	auto
PrescripImage	null	Image file	image

Table 36: P.D.D OrderItem

OrderItem			
Column Name	Constraint	Example	Validation
orderitemID	Primary key	0001	
product	Null = true, on _delete = set null.	0001	
Order	Null = true, on _delete = set null.	0001	
Quantity	Not Null	5	int
date_added	Auto add	15/05/2021	auto

Table 37: P.D.D Shipping Address

ShippingAddress			
Column Name	Constraint	Example	Validation
shippingAddressID	Primary key	0001	
patient	Null = true, on _delete = set null.	0001	
Order	Null = true, on _delete = set null.	0001	
address	Not Null	BBU Apartment	Varchar <= 200
date_added	Auto add.	15/05/2021	Auto
city	Not Null	Melaka	Varchar <= 200
state	Not Null	Melaka	Varchar <= 200
zipcode	Not Null	76450	Varchar <= 200
Contact no	-	+60149845569	Varchar <= 200

4.4 Conclusion

In conclusion, above in this chapter we covered the designs that illustrates E-DocPh system. We covered and discussed about the database and its component Database Design section. All the necessary components and relations has been described there. On the other hand, System architecture shows the flow of the system with flowchart diagrams. It illustrates how the system is flowing from one section to another. In user interface design, navigation design is helping to map the system. Input design is showing the forms and the data input requirements has been briefly described. Software design part holds information and data for the functions of the system according to Data-flow-diagram (DFD) which is described in chapter 3 analysis. Now, we can understand the designs that illustrate and defines the E-DocPh web application.

CHAPTER 5 : IMPLEMENTATION

5.1 Introduction

This chapter contains information about implementation of E-DocPh. Moving on this phase will illustrate network management of the system, development environment, implementation phase and its brief discussion about how all the modules are working and their purposes as well. Moreover, there will be setup configuration management system discussed in order to understand how the system configured in terms of implementation. Implementation Status will contain all the modules name and their development time period and how and why they were developed.

After Completing this phase, the system should be ready for testing. The system should have three main modules named store, doctor and admins each module for different type of users. The system backend and frontend should be ready for testing and should have space to reprogram the modules if any bug pop-up on during or post-testing. If no bug found program should be ready to go online anytime.

5.2 Software Development Environment setup

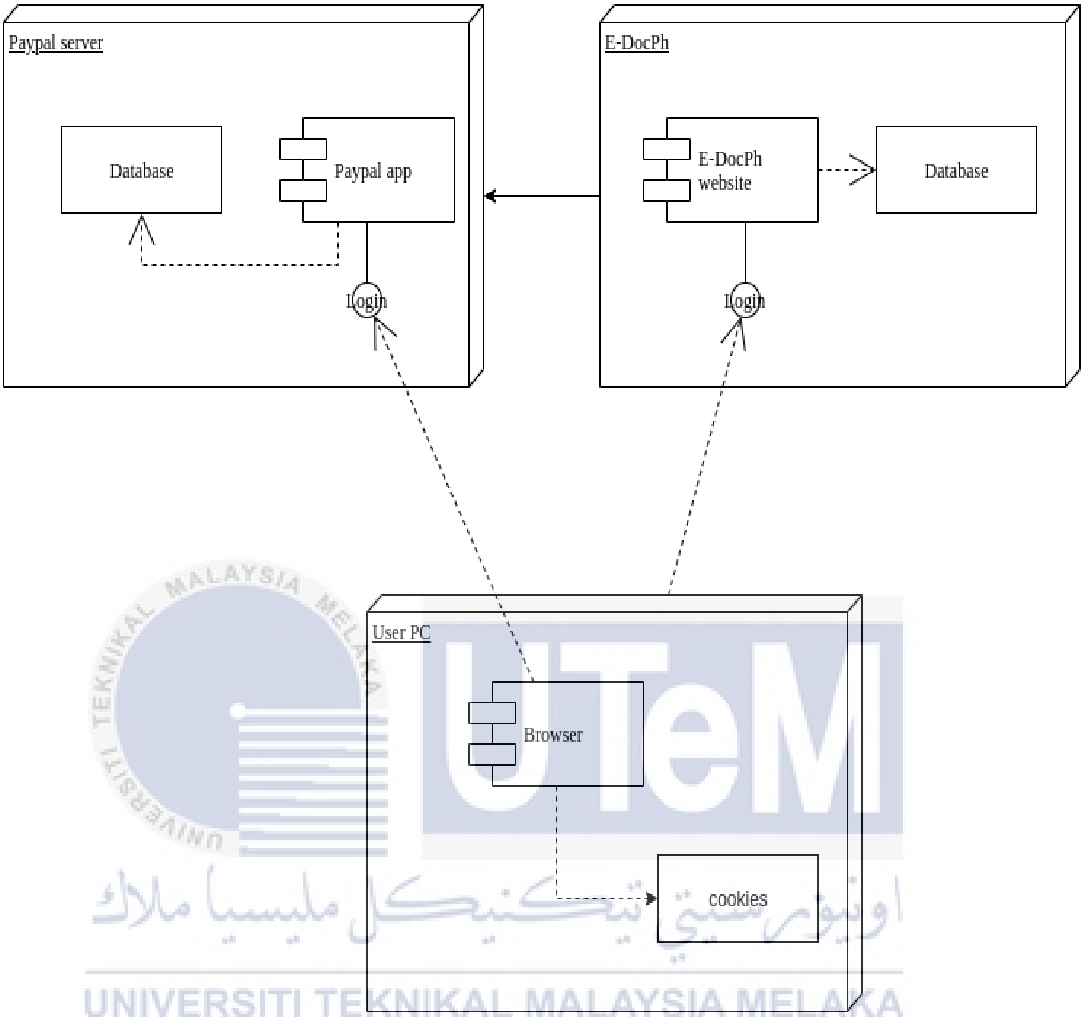
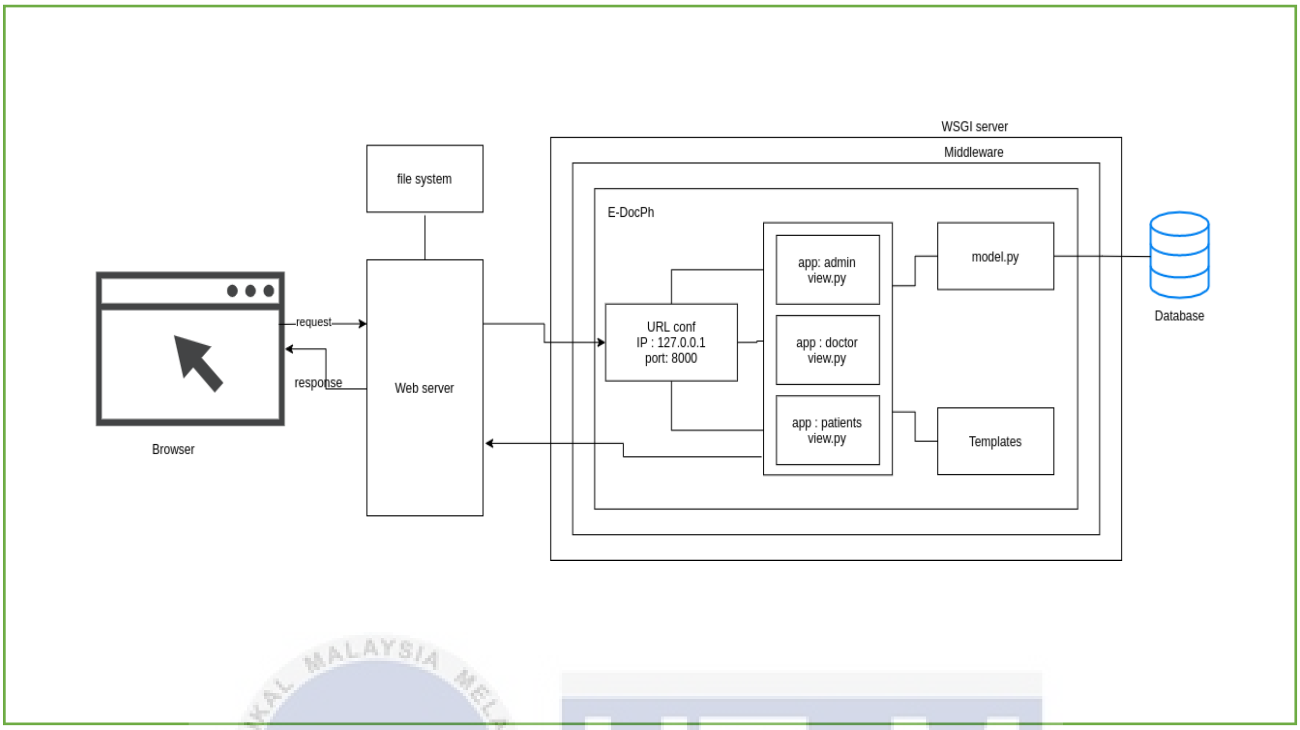


Figure 20: deployment diagram

As shown in figure 20 user can access to E-DocPh web application through login to their account, do so user can access the database of the system. If the user is patient and wants to buy any item, they can login to their PayPal account; note that PayPal integration is already connected with the system through PayPal API. Once User will arrive in checkout page, user can see the PayPal gateway button. On click PayPal button user will be redirected to the PayPal application and will be asked for their PayPal information. Once authentication is completed, the system will automatically send the amount data to PayPal and PayPal will charge amount basis of data PayPal received from the system. Upon completing all the procedures user’s payment should be completed.

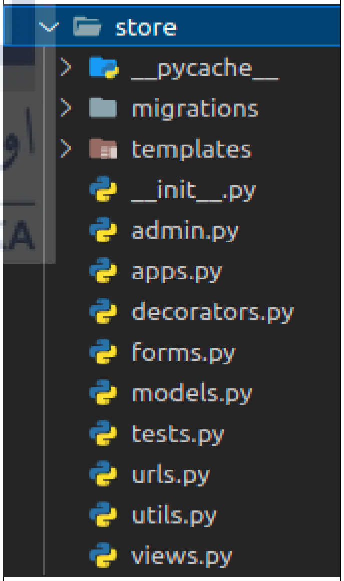


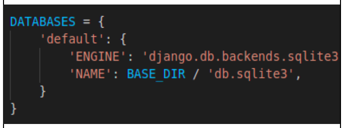
On above diagram, network setup for E-DocPh has been illustrated. As of mentioned diagram, user is accessing to the system through a browser where user will request data and in response will get data requested from web server. The system is connected to the web server through localhost or 127.0.0.1 and port for the connection is 8000. Before web server can access to system's URL configuration, first it needs to mate with WSGI server, middleware. Once web server connects with system's URL, view file starts to process the functions and gate data from database through model file and send data through response to the template enabling user to see the data itself.

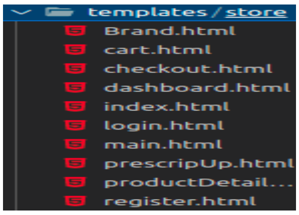
5.3 Software Configuration

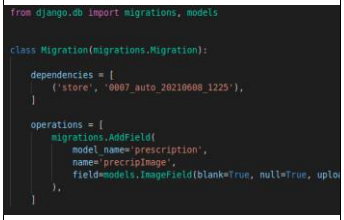
5.3.1 Configuration environment setup

Table 38: Configuration environment setup

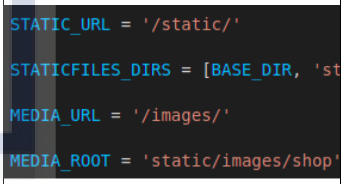
No	name	Code	Description	Screenshots(optional)
01	Creating Django project	django-admin startproject E-DocPh	This will create a project named E-DocPh and create some files like manage.py, wsgi.py, settings.py etc. automatically.	
02	Execute server	Python3 manage.py runserver	This will run server and check if the connection is good or not	
03	Creating app	Python3 manage.py startapp store	This will create an app named store and a store folder will be created. This folder will contain all the necessary data for this particular app.	 store __pycache__ migrations templates __init__.py admin.py apps.py decorators.py forms.py models.py tests.py urls.py utils.py views.py

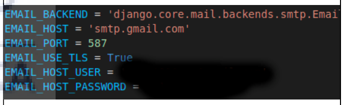
04	Create superuser	Django-admin create superuser	This will create a superuser for the database.	
				DATABASES = {
				    'default': {
				        'ENGINE': 'django.db.backends.sqlite3
				        'NAME': BASE_DIR / 'db.sqlite3',
				    }
				}

05	Templates	-	Developer need to create a folder name templates in the app(store) and then create another folder named same as app(store).	 templates / store Brand.html cart.html checkout.html dashboard.html index.html login.html main.html prescripUp.html productDetail... register.html

06	Migrating model into database	Python3 manage.py makemigrations	This will take all the data from models.py and write it in migrateable way automatically	
				`from django.db import migrations, models`
				
				`class Migration(migrations.Migration):`
				
				`    dependencies = [`
				`        ('store', '0007_auto_20210608_1225'),`
				`    ]`
				
				`    operations = [`
				`        migrations.AddField(`
				`            model_name='prescription',`
				`            name='precripImage',`
				`            field=models.ImageField(blank=True, null=True, uplo`
				`        ),`
				`    ]`

07	Migrate	Python3 manage.py migrate	This will migrate data of migrations and write it in the database.	

08	Setting media	-	This will balance all the media files routing.	
				`STATIC_URL = '/static/'`
				`STATICFILES_DIRS = [BASE_DIR, 'st`
				`MEDIA_URL = '/images/'`
				`MEDIA_ROOT = 'static/images/shop'`

09	Setting email automation	-	This will configure email automation.	
				`EMAIL_BACKEND = 'django.core.mail.backends.smtp.Emai`
				`EMAIL_HOST = 'smtp.gmail.com'`
				`EMAIL_PORT = 587`
				`EMAIL_USE_TLS = True`
				`EMAIL_HOST_USER =` [redacted]
				`EMAIL_HOST_PASSWORD =`

5.4 Implementation Status

Table 39: Implementation Status

Module	**Description**	**Duration**	**Dates Involved**	**Size of software**	**Application**
Registration Patient	This module will be used for registering patients	7 days	1-7 April 2021	175.8 KB	Store
Login patient	This module will be used for authenticating patients	7 days	1-7 April 2021	175.8 KB	Store
Product Detail	This module will get all the data of a specific product and show in the template	1 day	7 April 2021	175.8 KB	Store
Shop	This module will show all the products from the database	3 days	7-10 April 2021	175.8 KB	Store
Cart	This is to manage cart for a user	7 days	10-17 April 2021	175.8 KB	Store
Checkout	This is to pay for the order of a user	7 days	17-24 April 2021	175.8 KB	Store
Brand	This is to show all products under a brand only	4 days	11-14 August 2021	175.8 KB	Store
Doctors	This is the home page for doctors. Doctors can see all the pending orders of patients.	3 days	10-13 May 2021	62.8 KB	Doctor
View orders	Doctors can see all the items a patient ordered and can see the prescription uploaded by patient and can either reject	7 days	13-20 May 2021	62.8 KB	Doctor

	or approve the order from here.				
Registration Doctor	This module will be used for registering Doctor	3 days	20-23 May 2021	62.8 KB	Doctor
Login Doctor	This module will be used for authenticating Doctor	3 days	23-26 May 2021	62.8 KB	Doctor
Registration Admins	This module will be used for registering Admins	2 days	26-28 May 2021	92.4 KB	Admins
Login Admins	This module will be used for authenticating Admins	2 days	28-30 May 2021	92.4 KB	Admins
Dashboard	This module will show how many orders are in progress, how many are delivered, how many orders has been completed. It also shows last 10 orders.	7 days	7-14 June 2021	92.4 KB	Admins
Customers	This module shows all the patients information registered in the system	1 day	15 June 2021	92.4 KB	Admins
Create products	This module includes form which is used to get all the data of a product and store them into database	2 days	15-16 June 2021	92.4 KB	Admins
Update products	This module includes form which is used to get all the data of a product and update them into database	1 day	17 June 2021	92.4 KB	Admins
Delete products	This module will delete information of a product from the database	1 day	18 June 2021	92.4 KB	Admins

Products	This module shows all the product information in the system	1 day	19 June 2021	92.4 KB	Admins
Doctor	This module shows all the doctor information registered in the system	1 day	20 June 2021	92.4 KB	Admins
Orders	This module shows all the Orders information in the system	1 day	20 June 2021	92.4 KB	Admins
Charts	This module shows information in a line chart as how many patients registered in the system on date basis. This module also contains a bar chart where all the products sold of a brand shows as sum.	24 days	19 July – 10 August 2021 & 15-18 August 2021	92.4 KB	Admins

5.5 Conclusion

In this chapter implementation has been illustrated. All the necessary diagrams and tables had been shown briefly. The chapter started showing the network management, how the system is connected and how its data is passing through the server. Next, the system building management has been shown in a table. Which explains how to start the project and configure it with all the necessary components. At last, Implementation status shows the time duration of development and also describe each module’s primary objection, and all had been sorted as app name.

CHAPTER 6: TESTING

6.1 Introduction

Software testing is a widely used technique for ensuring and validating software quality. Software testing is the process of running a program or system in order to find failings. Testing software is an important part of the software development life cycle. It aids in the development of a developer's confidence that a program does what it is supposed to do. Two of the most popular testing methods, black box testing and top-down testing, will be used to test the system.

In this chapter testing format and all the other necessary components will be described in order complete the testing phase. In software development cycle, software testing is extremely important. This process refines the entire process and ensure that the product is of superior quality and working as it expected. This also reduces risk and less effects on maintenance and provides better usability and functionality.

6.2 Test Plan

6.2.1 Test Organization

Table 40: Test Organization

Tester ID	First name	Last name	Status	Roles
T_S_01	Abdul Hadi	Mazbah	Student	Author
T_S_02	Abdullah Al	Mamun	Student	User
T_S_03	Asif Jamil	Sahad	Student	User

6.2.2 Test Environment

Table 41: Test Environment

No	Name	Description	Type
01	Processor	Intel Core i3 4th gen	Hardware
02	Installed RAM	4 GB	Hardware
03	Graphics Card	Intel HD 4400	Hardware
04	Hard Drive	128 GB HDD	Hardware
05	Python	3.6 and above	Language
06	HTML	5	Language
07	CSS	3	Language
08	JavaScript	ES6	Language
09	Django	3.1	Language
10	Operating System	Any	Software
11	Editor	Any	Software
12	Browser	Any	Software

6.2.3 Test Schedule

Table 42: Test Schedule

No	Module	Start date	End date	Duration	Application
01	Registration Patient	19 Aug 2021	20 Aug 2021	1 day	Store
02	Login patient	19 Aug 2021	20 Aug 2021	1 day	Store
03	Product Detail	20 Aug 2021	22 Aug 2021	2 days	Store
04	Shop	20 Aug 2021	22 Aug 2021	2 days	Store
05	Cart	20 Aug 2021	22 Aug 2021	2 days	Store
06	Checkout	20 Aug 2021	22 Aug 2021	2 days	Store
07	Brand	20 Aug 2021	22 Aug 2021	2 days	Store
08	Doctors	21 Aug 2021	22 Aug 2021	1 day	Doctor
09	View orders	21 Aug 2021	22 Aug 2021	1 day	Doctor
10	Registration Doctor	19 Aug 2021	20 Aug 2021	1 day	Doctor
11	Login Doctor	19 Aug 2021	20 Aug 2021	1 day	Doctor
12	Registration Admin	19 Aug 2021	20 Aug 2021	1 day	Admins
13	Login Admin	19 Aug 2021	20 Aug 2021	1 day	Admins
14	Dashboard	23 Aug 2021	25 Aug 2021	3 days	Admins
15	Customers	23 Aug 2021	25 Aug 2021	3 days	Admins
16	Create products	23 Aug 2021	25 Aug 2021	3 days	Admins
17	Update products	23 Aug 2021	25 Aug 2021	3 days	Admins
18	Delete products	23 Aug 2021	25 Aug 2021	3 days	Admins
19	Products	23 Aug 2021	25 Aug 2021	3 days	Admins
20	Doctor	23 Aug 2021	25 Aug 2021	3 days	Admins
21	Orders	23 Aug 2021	25 Aug 2021	3 days	Admins
22	Charts	24 Aug 2021	26 Aug 2021	3 days	Admins

6.3 Test Strategy

Top-Down Testing:

Top-down testing is an incremental integration categorized testing way where two or more modules need to be joined and these should be move down from top to bottom throughout architecture structure control flow. High level modules are tested first and then comes low level modules to be tested in these. At the end integration should be done ensuring the system is working properly and perfectly.

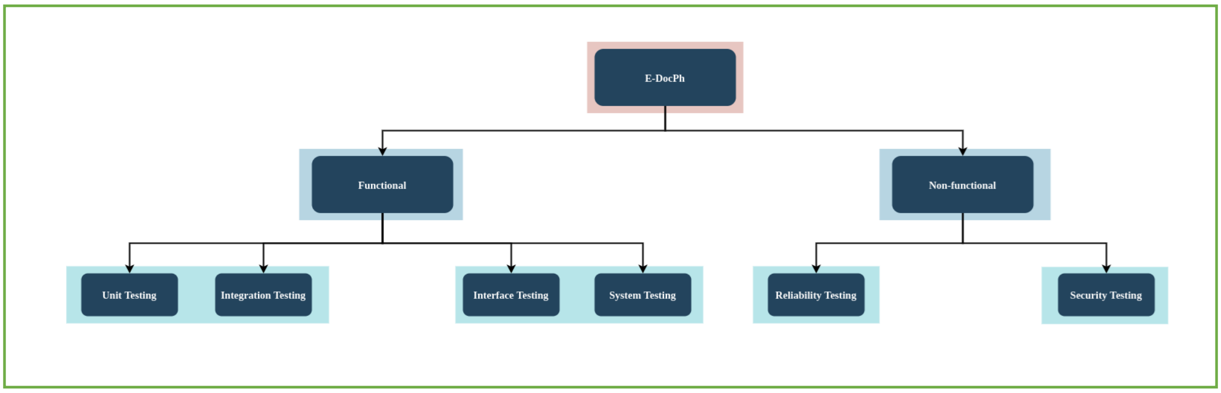
In this project top-down testing should be appropriate to test the system. This is because the system is written in both simple and complex way to build respective modules. Since top-down testing will start from complex modules and since these complex modules are some combination of lots of simple modules, functions and queries, it will test the complex modules and at the same time those simple components which used to build these complex modules. As far complex modules are okay after testing simple component were tested with them as benefit. So, after testing simple or low-level modules it is already tested two times at the time.

Black Box Testing:

Black box Testing is a testing where code structure, paths and implementation details knowledge is not required. It originally focuses on input and output of the applications. Another name for black box testing is behavioral testing.

In this project Top-Down testing already took part. Therefore, black box testing has been chosen over white box test. This is because black box testing usually done by people who does not have knowledge of the applications implementation processes or in simple words users.

6.3.1 Classes of tests



6.4 Test Design

6.4.1 Test Description

Table 43: Test Description

Test Case Identification	Test Case	Expected Result	Module	Application	Test Case Type
T_C_01	Unit Testing	Patients should be able to fill up the form given and if input data correctly should be redirect to login page.	Registration Patient	Store	Functional
T_C_02	Unit Testing	Patients should be able to login if input correct information in the given form and submit.	Login patient	Store	Functional
T_C_03	Unit Testing	Patient should be able see chosen products detail.	Product Detail	Store	Functional
T_C_04	Unit Testing	Patient should see shop page and its components.	Shop	Store	Functional
T_C_05	Unit Testing	Patients should be able to choose their wanting products and on click cart button should be added into their respective cart. Patient	Cart	Store	Functional

		should also be able to see their chosen products to buy and be able to update them.			
T_C_06	Unit Testing	Patients should be able to get confirmation emails from the system if doctor approves their order and be able to redirect to the checkout page where patient should be able to find a form and if input correctly should be able to redirect to PayPal gateway for payment.	Checkout	Store	Functional
T_C_07	Unit Testing	Patients should be able to see products based on their own brand.	Brand	Store	Functional
T_C_08	Unit Testing	Doctors should be able to see all the orders currently pending.	Doctors	Doctor	Functional
T_C_09	Unit Testing	Doctors should be able to see each pending orders details.	View orders	Doctor	Functional
T_C_10	Unit Testing	Doctors should be able to fill up the form given and if input data correctly should be redirect to login page.	Registration Doctor	Doctor	Functional
T_C_11	Unit Testing	Doctors should be able to login if input correct information in the given form and submit.	Login Doctor	Doctor	Functional
T_C_12	Unit Testing	Admins should be able to fill up the form given and if	Registration Admin	Admins	Functional

		input data correctly should be redirect to login page.			
T_C_13	Unit Testing	Admins should be able to login if input correct information in the given form and submit.	Login Admin	Admins	Functional
T_C_14	Unit Testing	Admins should see last 10 orders and pending orders, completed orders and total orders.	Dashboard	Admins	Functional
T_C_15	Unit Testing	Admins should be able see all the data of patients.	Customers	Admins	Functional
T_C_16	Unit Testing	Admins should find a form and if input properly should be storing the data in the database.	Create products	Admins	Functional
T_C_17	Unit Testing	Admins should find a form and if input properly should be updating the data in the database.	Update products	Admins	Functional
T_C_18	Unit Testing	Admins should be able to delete any product at a time.	Delete products	Admins	Functional
T_C_19	Unit Testing	Admins should be able see all the data of Products.	Products	Admins	Functional
T_C_20	Unit Testing	Admins should be able see all the data of doctors.	Doctor	Admins	Functional
T_C_21	Unit Testing	Admins should be able see all the data of orders.	Orders	Admins	Functional
T_C_22	Unit Testing	Admins should be able to see how many patients has registered in the system on basis as line chart. Admins	Charts	Admins	Functional

		should also be able to see how many products has been sold on basis of Products name.			
T_C_23	System Testing	Doctors should be able to see pending orders and on click view should see respected order details.	Doctor/View orders	Doctor	Functional
T_C_24	System Testing	Admins should be able to see all the products and be able to redirect to update page on clicking update button and be able to delete any specific product by clicking on delete button at a time.	Show orders/update product/delete product	Admins	Functional
T_C_25	Security Testing	The user’s password will be accepted if it adheres the rules.	Login Admins	Admins	Non-Functional
T_C_26	Security Testing	The user’s password will be accepted if it adheres the rules.	Login Patient	Patients	Non-Functional
T_C_27	Security Testing	The user’s password will be accepted if it adheres the rules.	Login Doctor	Doctor	Non-Functional
T_C_28	Interface Testing	Should be generate a form and two cards	Registration Patient	Store	Functional
T_C_29	Interface Testing	Should be generate a form and two cards	Login patient	Store	Functional
T_C_30	Interface Testing	Show specific product details and photo	Product Detail	Store	Functional
T_C_31	Interface Testing	Show six products in one page and show cart and menu	Shop	Store	Functional

		button and drop down for brands.			
T_C_32	Interface Testing	Show selected products and up down button- and drop-down menu	Cart	Store	Functional
T_C_33	Interface Testing	Show a form and a button	Checkout	Store	Functional
T_C_34	Interface Testing	Show all the products of a brand and each page should have maximum of six products.	Brand	Store	Functional
T_C_35	Interface Testing	Show a table and a button.	Doctors	Doctor	Functional
T_C_36	Interface Testing	Should show a table and two buttons.	View orders	Doctor	Functional
T_C_37	Interface Testing	Should be generate a form and two cards	Registration Doctor	Doctor	Functional
T_C_38	Interface Testing	Should be generate a form and two cards	Login Doctor	Doctor	Functional
T_C_39	Interface Testing	Should be generate a form and two cards	Registration Admin	Admins	Functional
T_C_40	Interface Testing	Should be generate a form and two cards	Login Admin	Admins	Functional
T_C_41	Interface Testing	Should generate three cards and a table	Dashboard	Admins	Functional
T_C_42	Interface Testing	Should generate a table	Customers	Admins	Functional
T_C_43	Interface Testing	Should be generate a form and a button	Create products	Admins	Functional
T_C_44	Interface Testing	Should be generate a form and a button	Update products	Admins	Functional

T_C_45	Interface Testing	A button	Delete products	Admins	Functional
T_C_46	Interface Testing	A table and two buttons.	Products	Admins	Functional
T_C_47	Interface Testing	A table	Doctor	Admins	Functional
T_C_48	Interface Testing	Table with a Boolean button.	Orders	Admins	Functional
T_C_49	Interface Testing	One card with line chart another card with bar chart.	Charts	Admins	Functional

6.4.2 Test Data

Table 44: T.D Patient

Register Patients					
First name	**Last name**	**Last name**	**Username**	**Email**	**Password**
Sheikh	Fazlur	Fazlur	Fazlur	fazlur@gmail.com	0147qwer
Rezaul	karim	Karim	Rezaul	karim@gmail.com	
John	Kennedy	Kennedy	Kennedy		45625poi
Isaac	Newton	Newton		isaac@gmail.com	0147qwer
Login Patients					
Username		**Password**			
Fazlur		0147qwer			
Rezaul		0147qwer			
Kennedy					

	0147qwer
Fazlur	45625poi

Table 45: T.D Doctor

Register Doctor					
First name	**Last name**	**Last name**	**Username**	**Email**	**Password**
Richard	Feynman	Fazlur	Feynman	richard@gmail.com	0147qwer
Albert	Einstein	Karim	Einstein	albert@gmail.com	
Peter	Higgs	Kennedy	Peter		45625poi
Sheldon	Cooper	Newton		cooper@gmail.com	0147qwer

Login Doctor	
Username	**Password**
Feynman	0147qwer
Einstein	0147qwer
Peter	
	0147qwer
Einstein	45625poi

Table 46: T.D Admins

Register Admins					
First name	**Last name**	**Last name**	**Username**	**Email**	**Password**
Bill	Gates	Fazlur	Bill	Gates@gmail.com	0147qwer
Steve	Jobs	Karim	Steve	Jobs@gmail.com	
Elon	Musk	Kennedy	Elon		45625poi
Jeff	Bezos	Newton		Jeff@gmail.com	0147qwer

Login Admins	
Username	**Password**
Bill	0147qwer
Steve	0147qwer
Elon	
	0147qwer
Steve	45625poi

Table 47: T.D Create Products

Create Product							
Name	**Brand**	**Quantity**	**Price**	**Digital**	**Image**	**Description**	**Tags**
ramipril	Altace	500	50.50	False		This is a Capsule	Pill Capsule
glimepiride		100	49.49	False			Pill Capsule
zolpidem	Ambien		30.30	True		This is a Capsule	Pill Capsule
lorazepam	Ativan	260	50.00	False		This is a Capsule	Pill Capsule
warfarin	Coumadin	-5	6.50	True		This is a Capsule	Pill Capsule
glyburide	Diabeta	0	68.15	False		This is a Capsule	Pill Capsule
phenytoin	Dilantin	200	-5	True		This is a Capsule	Pill Capsule
	Flonase	100	30.23	False		This is a nassal Spray	Pill Capsule

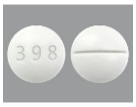
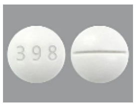
glipizide	Glucotrol	3000	90.58	True		This is a Capsule	

Table 48: T.D Update Product

Update Product							
Name	**Brand**	**Quantity**	**Price**	**Digital**	**Image**	**Description**	**Tags**
ramipril	Altace	300	50.50	False		This is a Capsule	Pill Capsule
glimepiride	Amaryl	100	49.49	False			Pill Capsule
zolpidem	Ambien		-5	True		This is a Capsule	Pill Capsule
lorazepam	Ativan	260	50.00	False		This is a Capsule	Pill Capsule
warfarin	Coumadin	300	6.50	True		This is a Capsule	Pill Capsule
glyburide	Diabeta	0	68.15	False			Pill Capsule

phenytoin	Dilantin	200	50.23	True		This is a Capsule	
	Flonase	100	30.23	False		This is a nassal Spray	Pill Capsule
glipizide	Glucotrol	3000	90.58	True		This is a Capsule	Pill Capsule

6.5 Test Results and Analysis

Table 49: Test Results and analysis

Test Case Identification	**Tester Identification**	**Result**	**Details (if result is Failed)**
T_C_01	T_S_01	Passed	
T_C_02	T_S_01	Passed	
T_C_03	T_S_01	Passed	
T_C_04	T_S_01	Passed	
T_C_05	T_S_01	Passed	
T_C_06	T_S_01	Failed	After Payment is completed user should redirect to home page but it failed. User kept in the same page after checkout complete.
T_C_07	T_S_01	Passed	
T_C_08	T_S_01	Passed	
T_C_09	T_S_01	Passed	
T_C_10	T_S_01	Passed	

T_C_11	T_S_01	Passed	
T_C_12	T_S_01	Passed	
T_C_13	T_S_01	Passed	
T_C_14	T_S_01	Passed	
T_C_15	T_S_01	Passed	
T_C_16	T_S_01	Passed	
T_C_17	T_S_01	Passed	
T_C_18	T_S_01	Passed	
T_C_19	T_S_01	Passed	
T_C_20	T_S_01	Passed	
T_C_21	T_S_01	Passed	
T_C_22	T_S_01	Passed	
T_C_23	T_S_01	Passed	
T_C_24	T_S_01	Passed	
T_C_25	T_S_02	Passed	
T_C_26	T_S_02	Passed	
T_C_27	T_S_02	Passed	
T_C_28	T_S_02	Passed	
T_C_29	T_S_02	Passed	
T_C_30	T_S_02	Passed	
T_C_31	T_S_02	Passed	
T_C_32	T_S_02	Passed	
T_C_33	T_S_02	Passed	
T_C_34	T_S_02	Passed	
T_C_35	T_S_02	Passed	
T_C_36	T_S_02	Passed	
T_C_37	T_S_02	Passed	
T_C_38	T_S_02	Passed	
T_C_39	T_S_03	Passed	
T_C_40	T_S_03	Passed	

T_C_41	T_S_03	Passed	
T_C_42	T_S_03	Passed	
T_C_43	T_S_03	Passed	
T_C_44	T_S_03	Passed	
T_C_45	T_S_03	Passed	
T_C_46	T_S_03	Passed	
T_C_47	T_S_03	Passed	
T_C_48	T_S_03	Passed	
T_C_49	0	Passed	

6.6 Conclusion

This chapter contains the test strategy that has been selected for this project and described briefly why those strategies has been chosen. Afterwards, all the necessary documents have been attached in order to complete the testing phase smoothly. Before that testers information was declared where these testers were linked in test results to identify who tested which module and which phase. The requirements had been mentioned earlier to conduct the testing. All the necessary equipment were described along with their versions to avoid version complexation.

CHAPTER 7. CONCLUSION

7.1 Observation on Weaknesses and Strengths

Each project has its own set of strengths and weaknesses, in order to keep the quality of the project it must be updated over time. In general, a product's quality is determined by comparing with its objectives. If a product meets all of the requirements, it is considered a good product; however, this does not rule out the possibility of faults. When the objectives aren't clear, it's difficult to assess the product's quality. Below are some of the strengths and weaknesses that will be discussed.

This web application has met all its objectives. It is impossible for a patient and doctor to change anything of the system. They cannot access to each other module with their own identification. Same thing applies for the admin as well. However, admins can see information of doctors and patients, but they cannot see their password because password is encoded. On the other hand, unless doctors approve patient order patients cannot get to the checkout page because patient will only get the link to access to their checkout through their email upon doctors' approval. With this feature no can buy anything from this web application without any prescription. Which was a major objective of this project. No one can misuse the system to buy drugs and use them. Moreover, this project has also enabled easier way to buy medicines and secure buying. As concern of payment, the project has PayPal gateway which is one of the easiest ways to pay for anything. Afterwards, once payment is complete admins can see transaction appear on the order module where they ensure that payment has been completed. After that, admins can start the process of delivering and once delivery is complete, they can press a button and an email will be forwarded to the patient that their order has been delivered.

With all these strengths the system has some weaknesses too. Such as, when order has been submitted for approval doctors should get a notification that a new order is waiting. The system does not have that feature. Next, when the user is done with their payment, they should redirect to the home page but in the system it doesn’t. Lastly the Doctors registration form should have more field for registering doctor information. This problem occurred because of Django model limited customization opportunities. However, this project was focused on the doctor approval ordering system so, this type of features has very small role to play.

7.2 Propositions for Improvements

Any application made for a lifelong service. Every time in a while the system needs to update itself and add new features. These updates can contain error fixing or to add new feature even to entirely remove a module. So, there is always room for change. However, the main improvement for this system would be notification system. Which should have been used more broadly. On update this problem can be solved. Since, this system is based on Django framework, there are some slight problems to registration form. This problem can be solved in next update with little bit of time.

7.3 Project Contribution

This project has been developed for any individual or group who is willing to do business on pharmacy sector. Usually in pharmacy more than any person needed to maintain the business. Normally business need to hire pharmacist who can only give some opinions about medications. Here in this project owner can get help of doctors rather than pharmacists. On the other hand, owner doesn’t have to expend money on decorating and rent of the pharmacy store because this is going online. Instead of hire people for store owner can hire admins to maintain the system. This way owner is also helping people to get their medicines from home and also stopping drugs crime.

7.4 Conclusion

In conclusion, the project's objectives were met successfully. The projects have met the basic requirements, but they are still operational. Web application development is something I've learned. I have learned a new framework and understood the difference between raw codes and framework. Django framework is something I enjoyed very much to learn. This project passed all of its initial tests. This system still has a lot of room for improvement to make it more user-friendly.

References

Accessed 3rd June 2021 at 9.53 pm,<https://stackoverflow.com/>

Accessed 18th June 2021 at 10.19 am,<https://docs.djangoproject.com/en/3.2/topics/http/shortcuts/>

Accessed 21st July 2021 at 3.59 pm,<https://docs.djangoproject.com/en/3.2/topics/auth/default/>

Accessed 2nd August 2021 at 1.27 am,<https://django-analytical.readthedocs.io/en/latest/>

Accessed 21st August 2021 at 7.21 pm,<https://docs.djangoproject.com/en/3.2/topics/auth/default/>

Roberta M. Roth, Alan Dennis, Barbara Haley Wixom (2013). System Analysis and Design. Asia: John Wiley & Sons, Inc.

BIBLIOGRAPHY

Bate, R., Ginger, A., Jin, Z., Mathur, A., & Attaran, A. (2016). *Poor Quality Drugs and Global Trade: A Pilot Study*. Retrieved from http://fortune.com/2013/05/15/dirty-medicine/

Flodgren, G., Rachas, A., Farmer, A. J., Inzitari, M., & Shepperd, S. (2015). Interactive telemedicine: Effects on professional practice and health care outcomes. *Cochrane Database of Systematic Reviews*, Vol. 2015. John Wiley and Sons Ltd. https://doi.org/10.1002/14651858.CD002098.pub2

Ignjatović, D., & Stanić, M. (2019). *CONTEMPORARY MARKETING IN PHARMACY WITH THE FOCUS ON THE E-PHARMACY CONCEPT*. https://doi.org/10.15308/finiz-2019-78-84

Karimy, M., Rezaee-Momtaz, M., Tavousi, M., Montazeri, A., & Araban, M. (2019). Risk factors associated with self-medication among women in Iran. *BMC Public Health*, *19*(1), 1–7. https://doi.org/10.1186/s12889-019-7302-3

Lee, D. H. (2018). Strategies for technology-driven service encounters for patient experience satisfaction in hospitals. *Technological Forecasting and Social Change*, *137*, 118–127. https://doi.org/10.1016/j.techfore.2018.06.050

Lee, S. M., & Lee, D. H. (2020, March 1). “Untact”: a new customer service strategy in the digital age. *Service Business*, Vol. 14, pp. 1–22. Springer. https://doi.org/10.1007/s11628-019-00408-2

Nandal, R., & Research Scholar, Mt. (2017). A Research Paper OnWebsite Development OptimizationUsing Xampp/PHP. *International Journal of Advanced Research in Computer Science*, *8*(5), 1231–1235. Retrieved from www.ijarcs.info

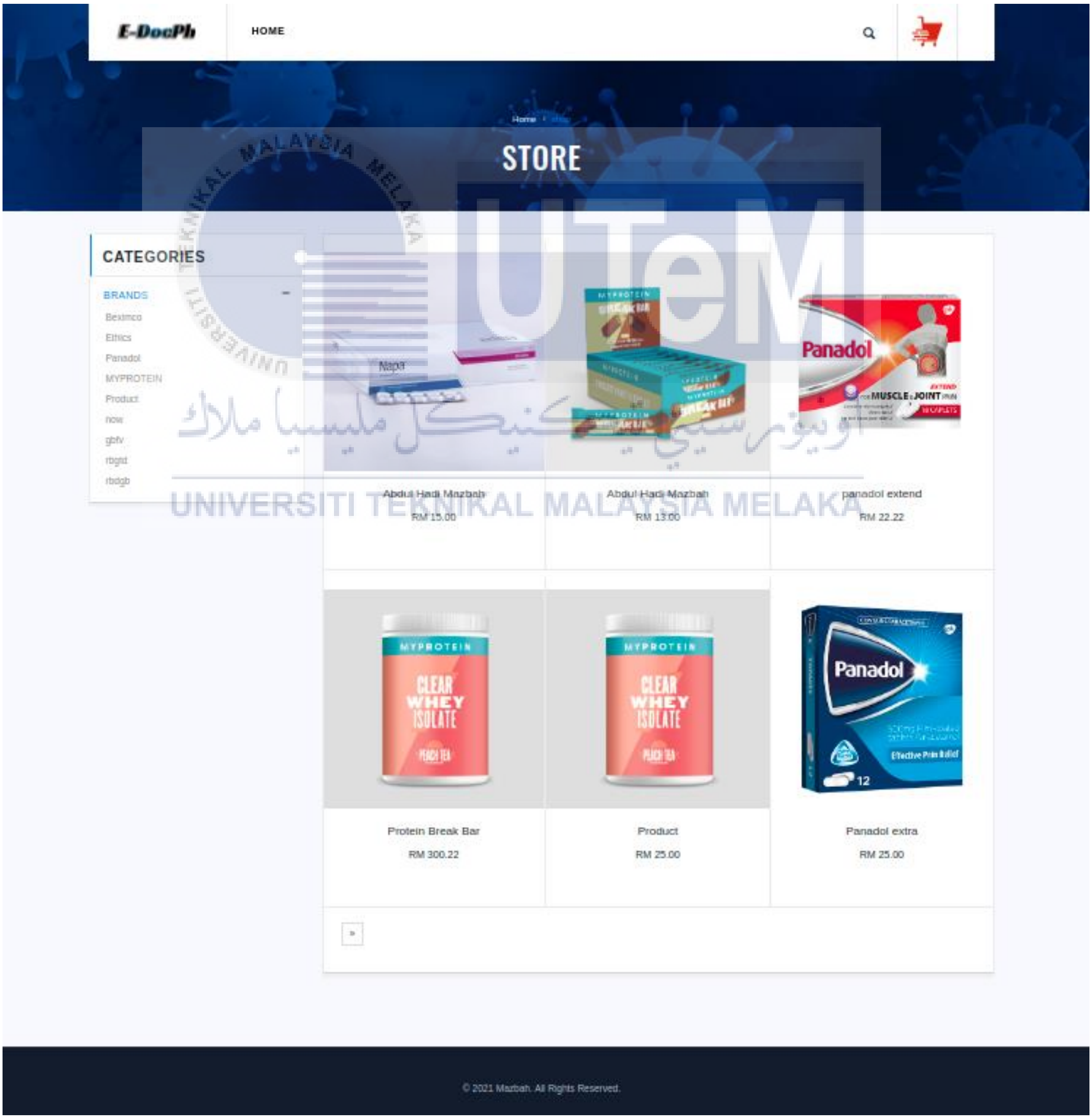
Patil, A. S., Patil, R. S., More, S. P., Sankpal, S. S., & Student, B. E. (2019). Web Application for Online Pharmacy. *International Research Journal of Engineering and Technology*, 1499. Retrieved from www.irjet.net

Tates, K., Antheunis, M. L., Kanters, S., Nieboer, T. E., & Gerritse, M. B. E. (2017). The Effect of Screen-to-Screen Versus Face-to-Face consultation on doctor-patient communication: An experimental study with simulated patients. *Journal of Medical Internet Research*, *19*(12), e421. https://doi.org/10.2196/jmir.8033

APPENDIX A

USER INTERFACE

1. Patient Home page



E-DocPh

HOME

Home / [illegible]

STORE

CATEGORIES

BRANDS

- Bextmco
- Ethics
- Panadol
- MYPROTEIN
- Product
- now
- gbfv
- rbgfd
- rbdgb

Abdul Hadi Mazbah
RM 15.00

Abdul Hadi Mazbah
RM 13.00

panadol extend
RM 22.22

Protein Break Bar
RM 300.22

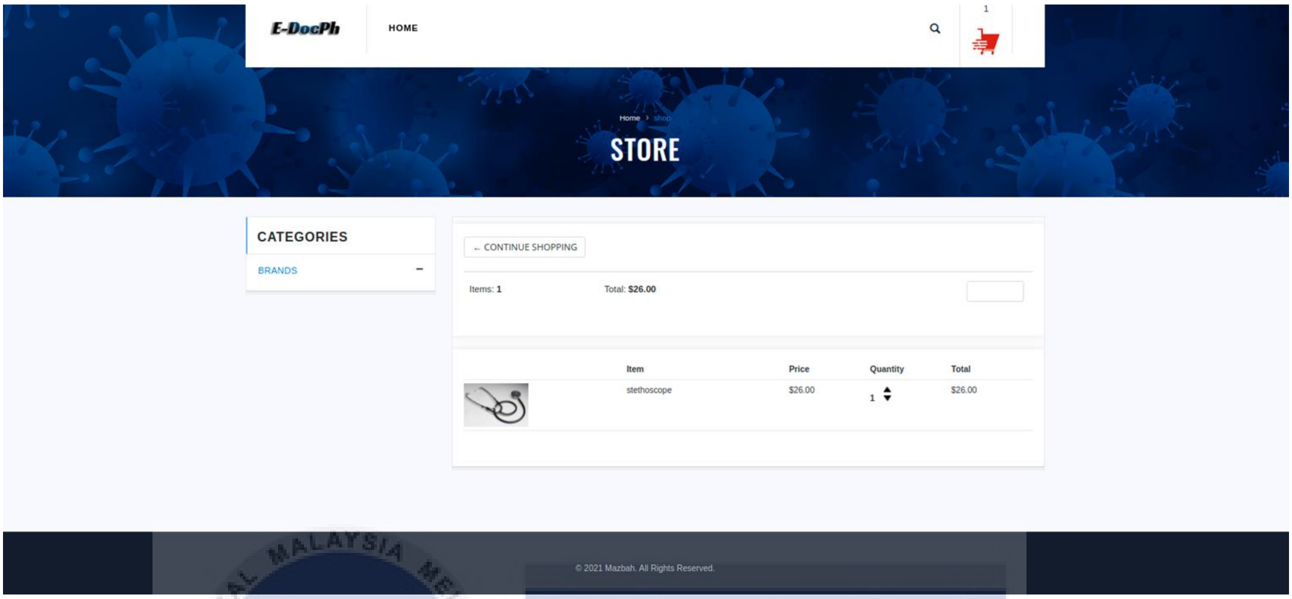
Product
RM 25.00

Panadol extra
RM 25.00

»

© 2021 Mazbah. All Rights Reserved.

2. Cart

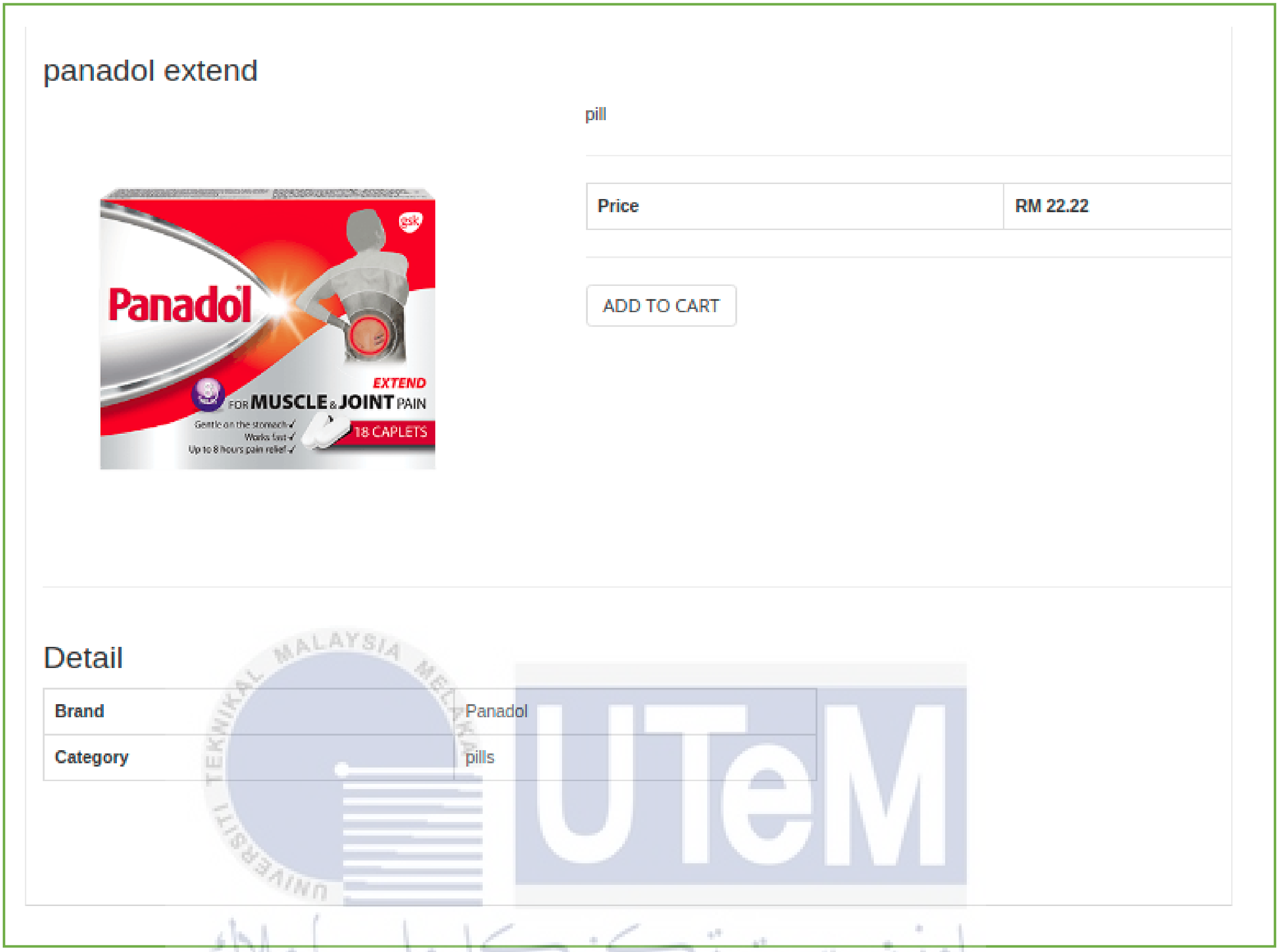


Items: **1** Total: **$26.00**

	Item	Price	Quantity	Total
	stethoscope	$26.00	1	$26.00

© 2021 Mazbah. All Rights Reserved.

3. Product Details



panadol extend

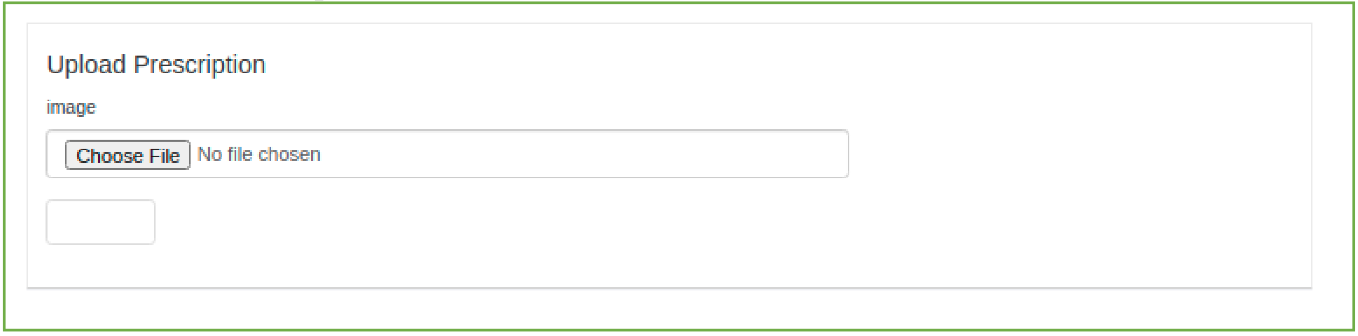
pill

Price	RM 22.22

Detail

Brand	Panadol
Category	pills

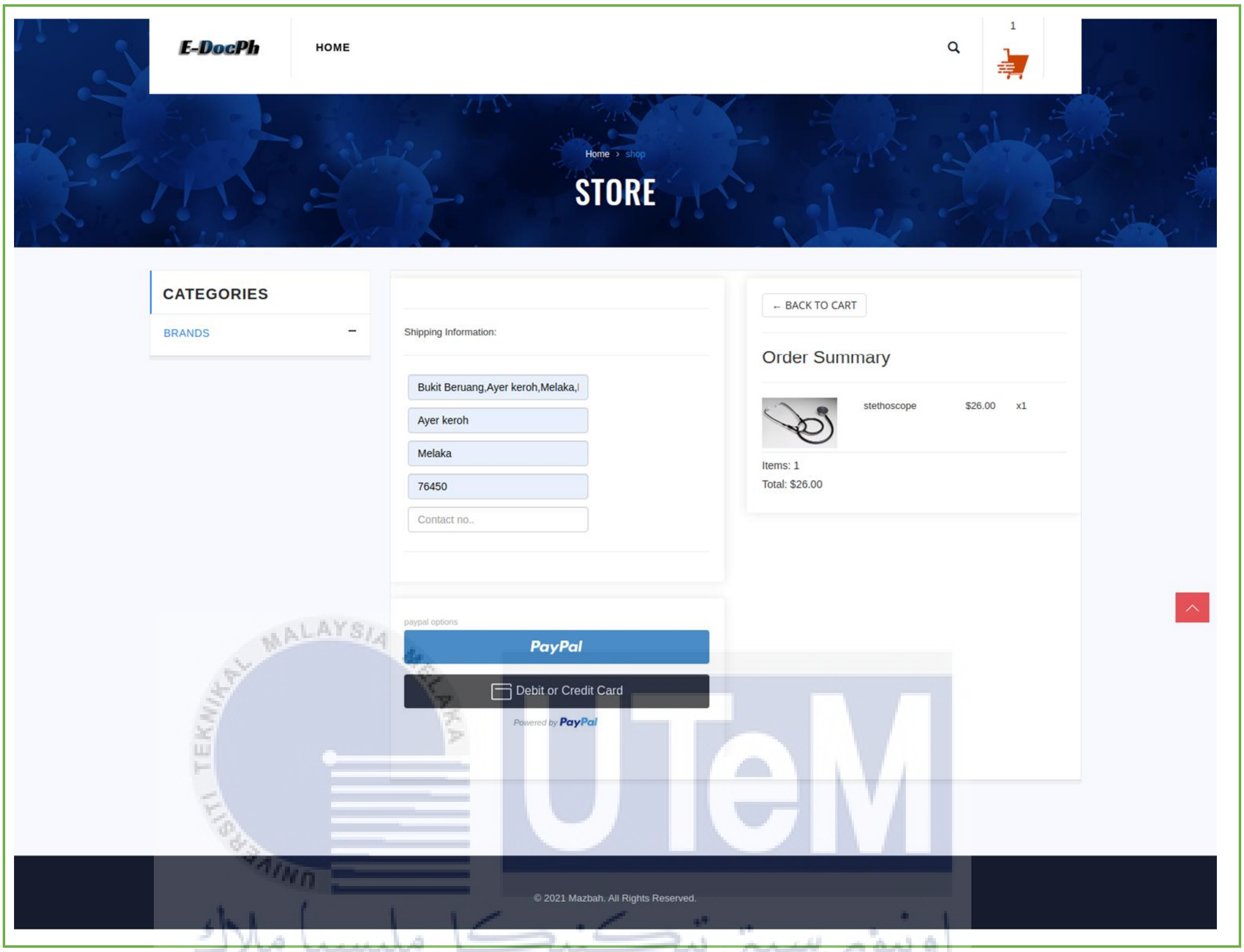
4. Prescription



Upload Prescription

image

5. Checkout



Shipping Information:

Bukit Beruang,Ayer keroh,Melaka,|

Ayer keroh

Melaka

76450

Contact no..

Order Summary

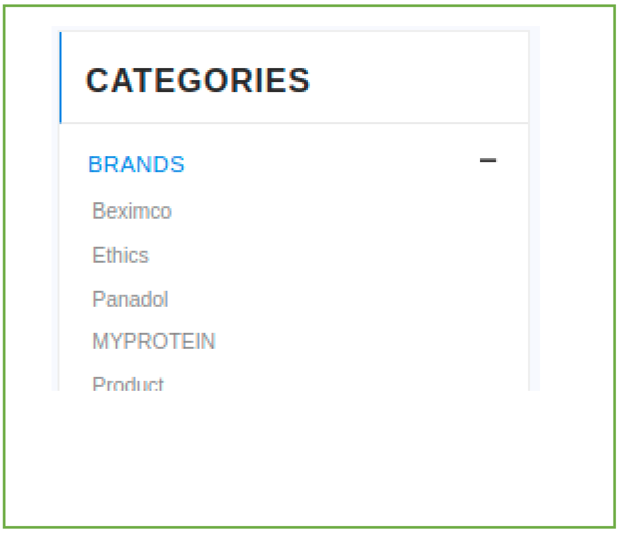
stethoscope	$26.00	x1

Items: 1

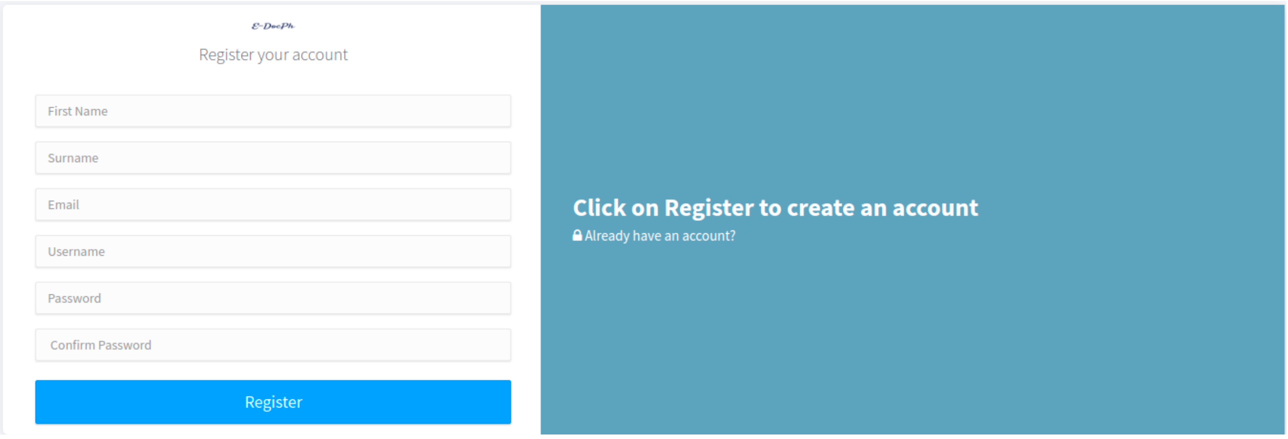
Total: $26.00

© 2021 Mazbah. All Rights Reserved.

6. Brands



7. Register Patients

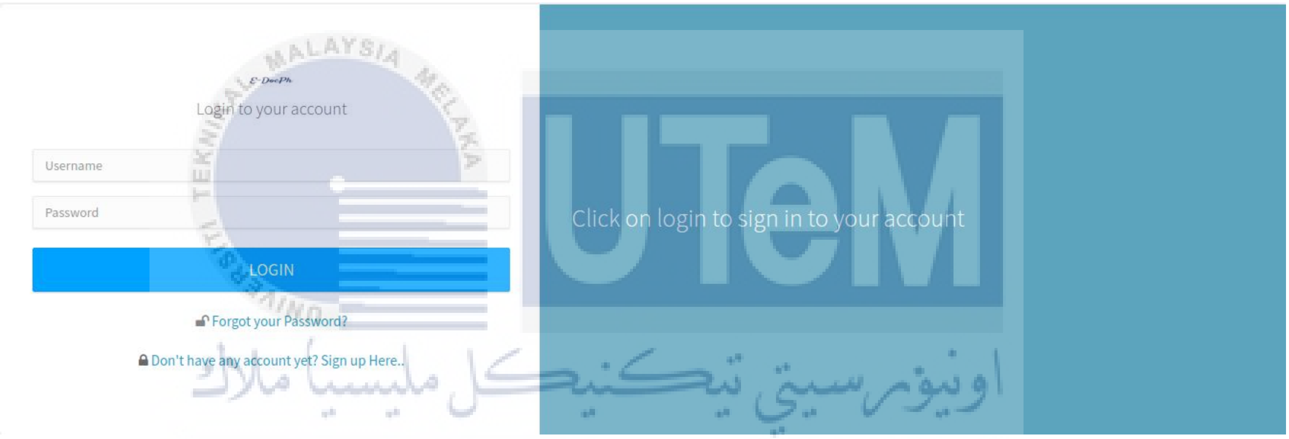


Register your account

Click on Register to create an account

Already have an account?

8. Login Patients



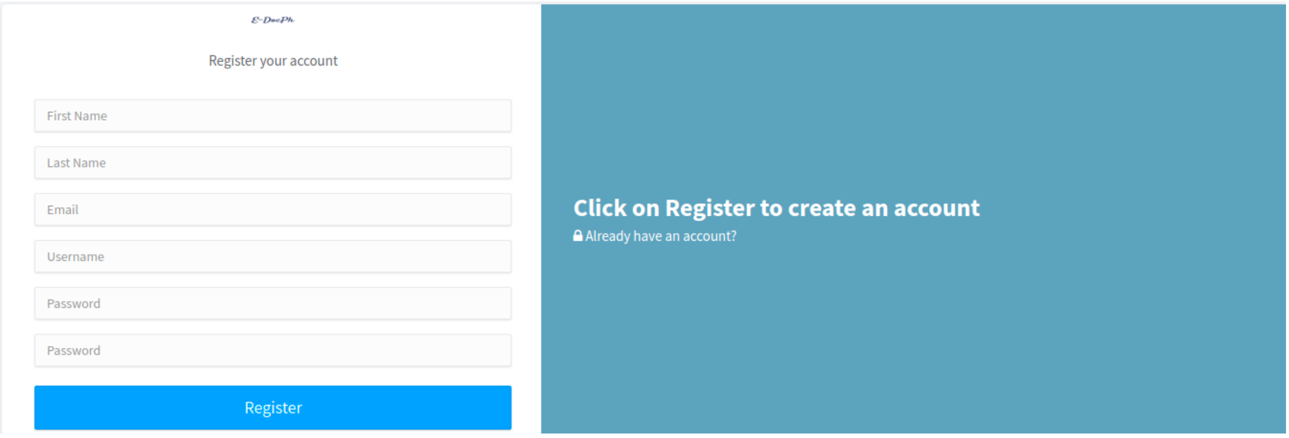
Login to your account

Forgot your Password?

Don't have any account yet? Sign up Here..

Click on login to sign in to your account

9. DoctorRegister

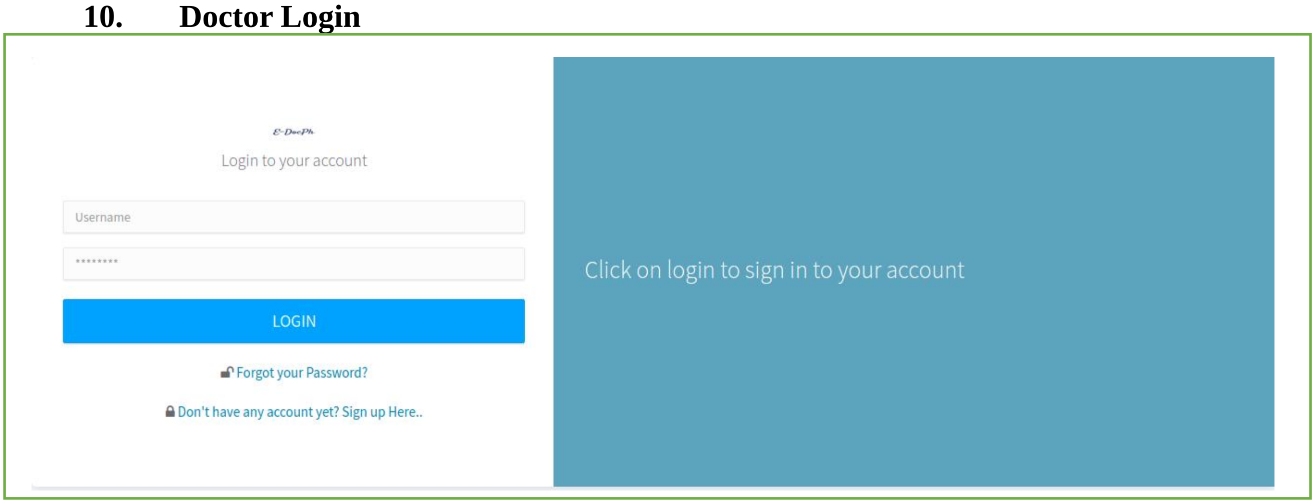


Register your account

Click on Register to create an account

Already have an account?

10. Doctor Login



Login to your account

Username

****

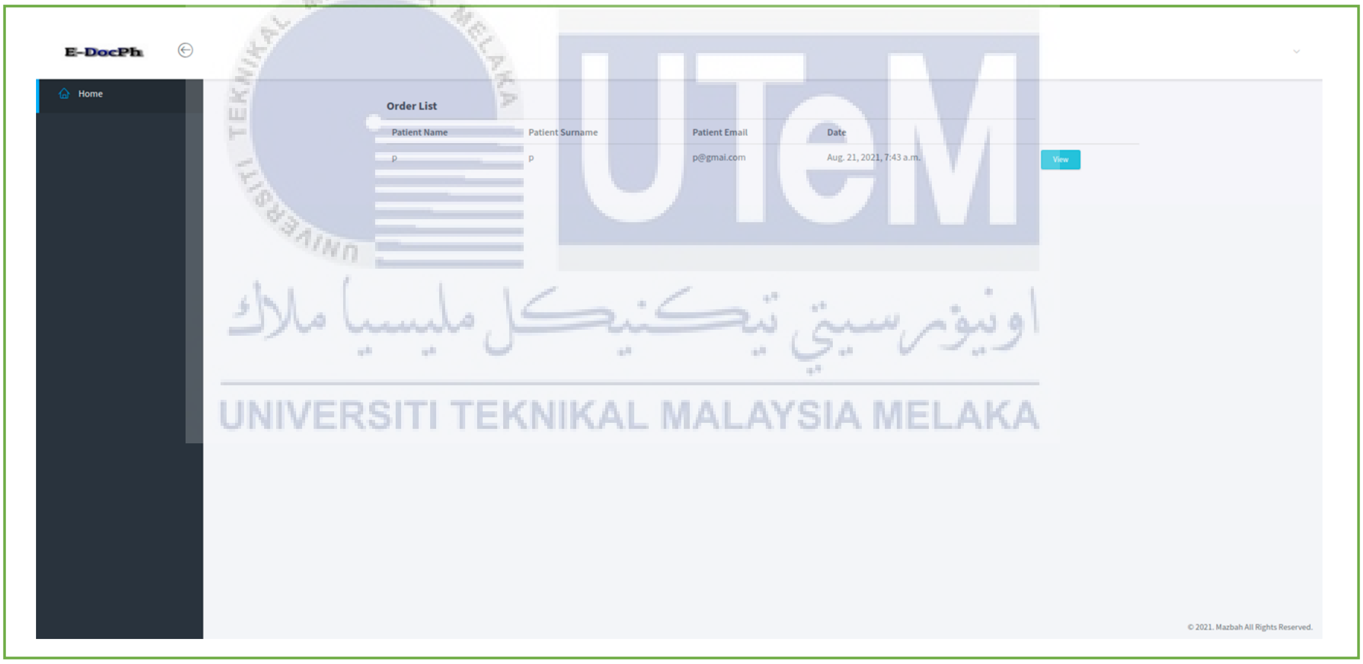
LOGIN

Forgot your Password?

Don't have any account yet? Sign up Here..

Click on login to sign in to your account

11. Doctor

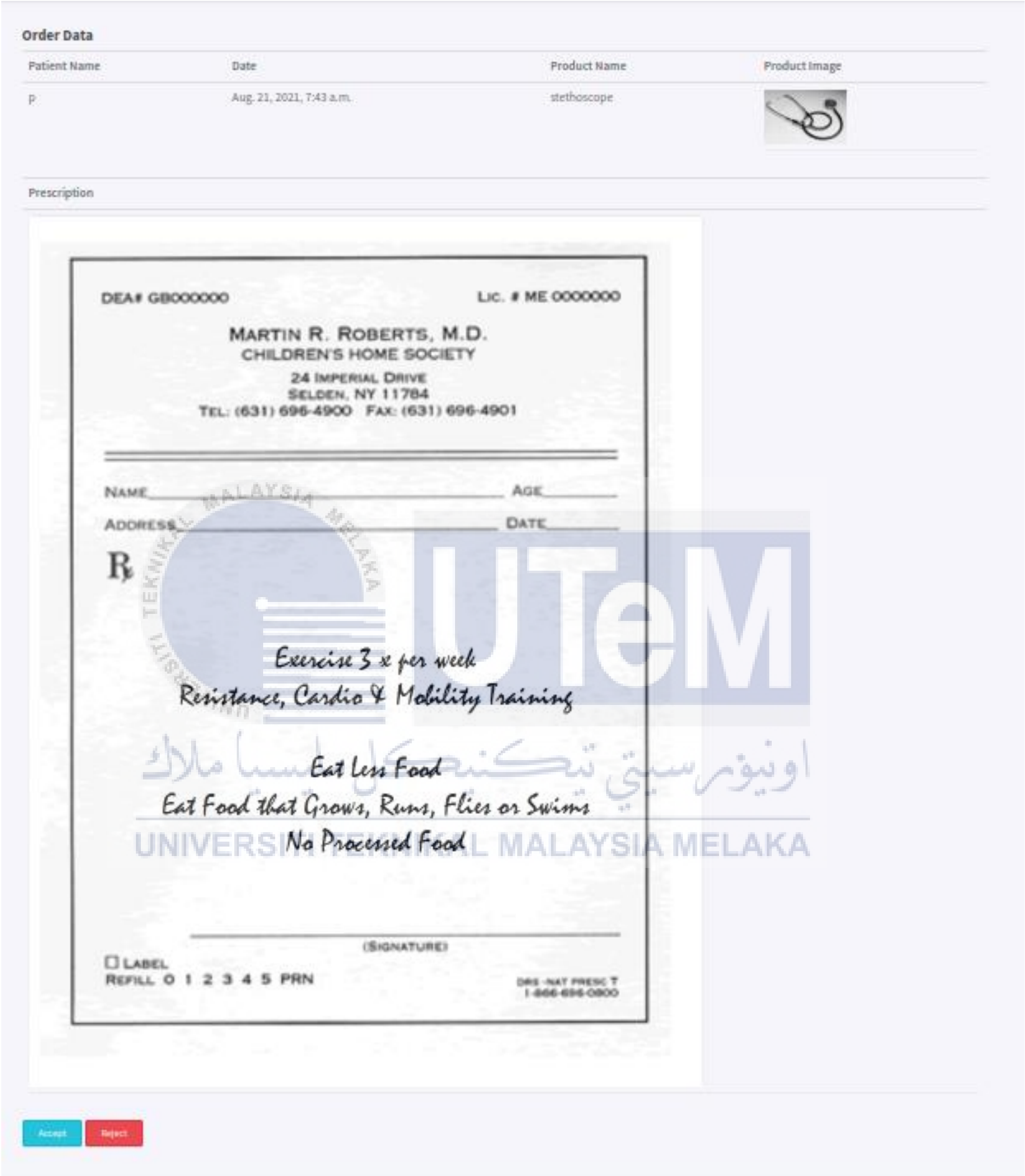


Order List

Patient Name	Patient Surname	Patient Email	Date	
p	p	p@gmai.com	Aug. 21, 2021, 7:43 a.m.	View

© 2021. Mazbah All Rights Reserved.

12. View Orders



Order Data

Patient Name	Date	Product Name	Product Image
p	Aug. 21, 2021, 7:43 a.m.	stethoscope	

Prescription

DEA# GB0000000 LIC. # ME 0000000

MARTIN R. ROBERTS, M.D.
CHILDREN'S HOME SOCIETY
24 IMPERIAL DRIVE
SELDEN, NY 11784
TEL: (631) 696-4900 FAX: (631) 696-4901

NAME ____________ AGE ______

ADDRESS ____________ DATE ______

℞

Exercise 3 x per week
Resistance, Cardio & Mobility Training

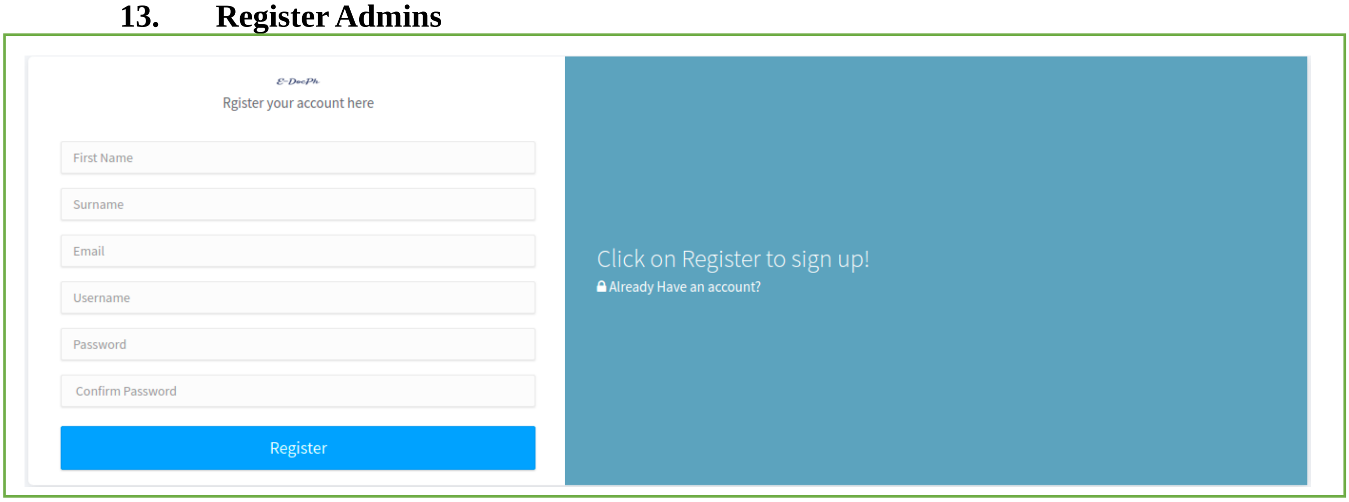
Eat Less Food
Eat Food that Grows, Runs, Flies or Swims
No Processed Food

(SIGNATURE)

□ LABEL
REFILL 0 1 2 3 4 5 PRN

DRS -NAT PRESC T
1-866-696-0800

13. Register Admins



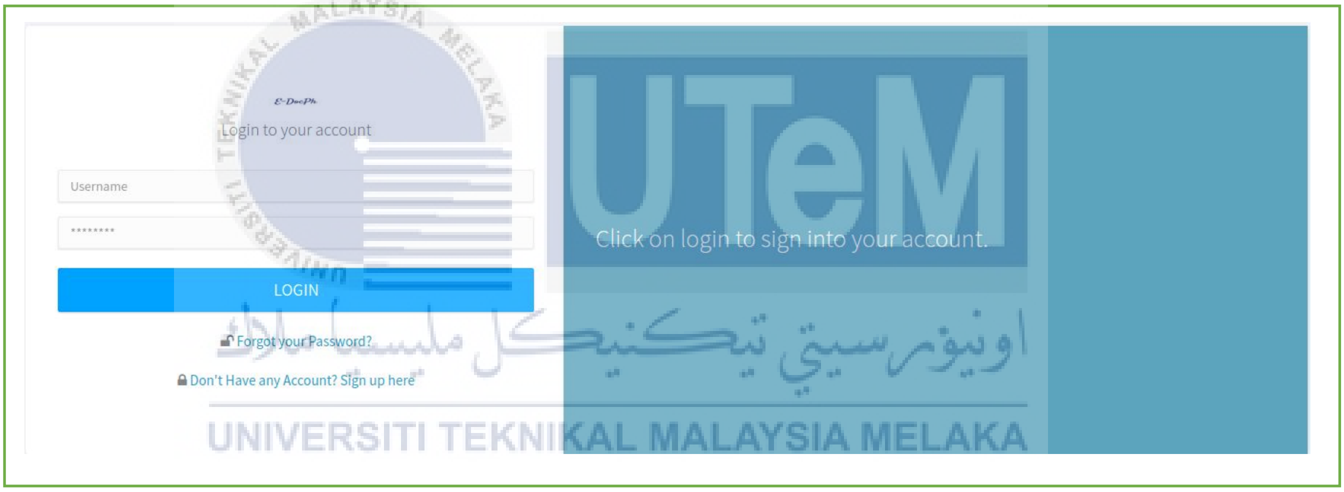
E-DocPh

Rgister your account here

Click on Register to sign up!

Already Have an account?

14. Login Admins

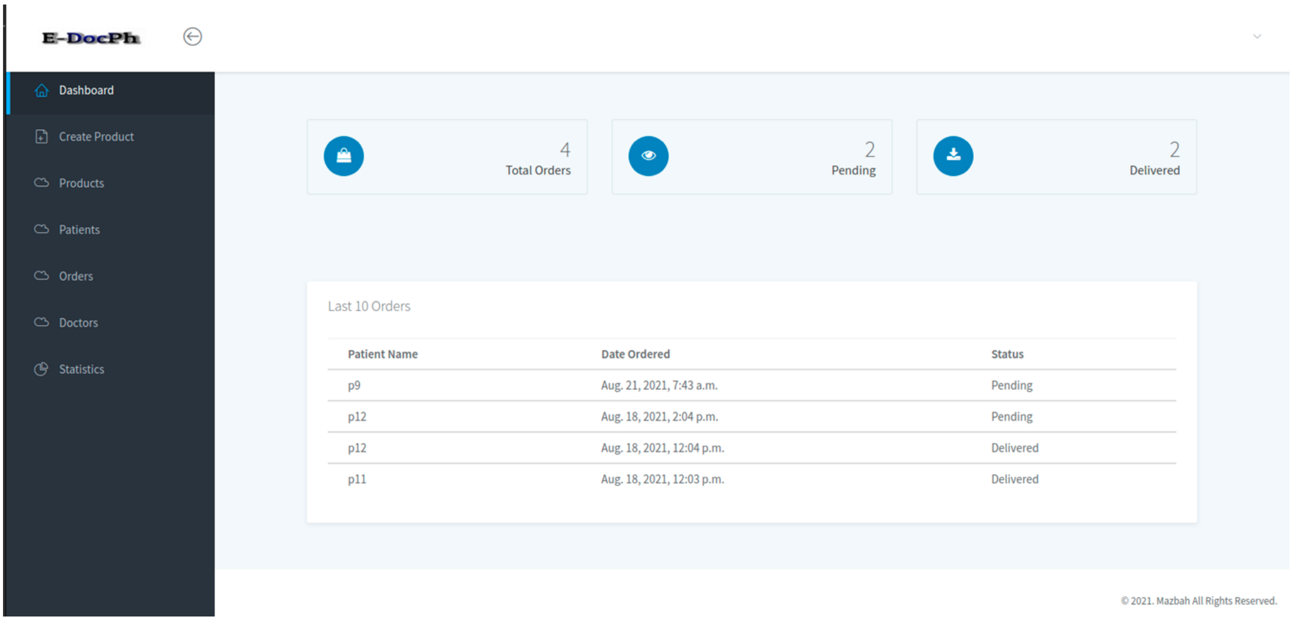


E-DocPh

Login to your account

Click on login to sign into your account.

15. Dashboard



4 Total Orders

2 Pending

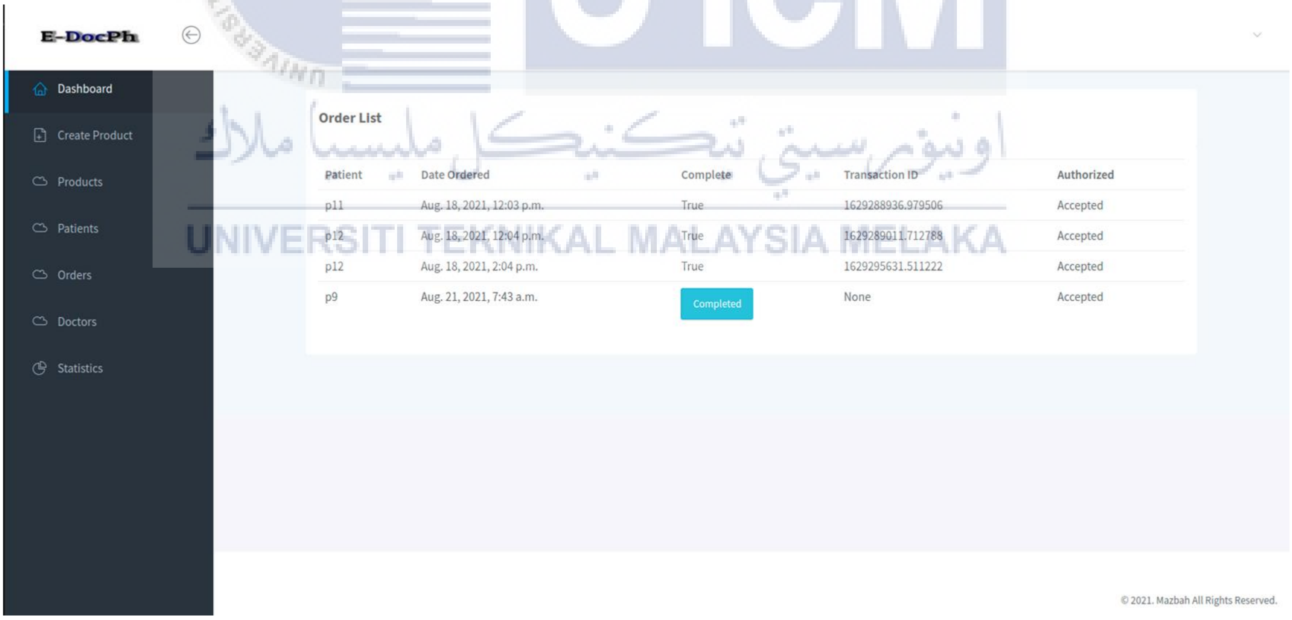
2 Delivered

Last 10 Orders

Patient Name	Date Ordered	Status
p9	Aug. 21, 2021, 7:43 a.m.	Pending
p12	Aug. 18, 2021, 2:04 p.m.	Pending
p12	Aug. 18, 2021, 12:04 p.m.	Delivered
p11	Aug. 18, 2021, 12:03 p.m.	Delivered

© 2021. Mazbah All Rights Reserved.

16. Orders

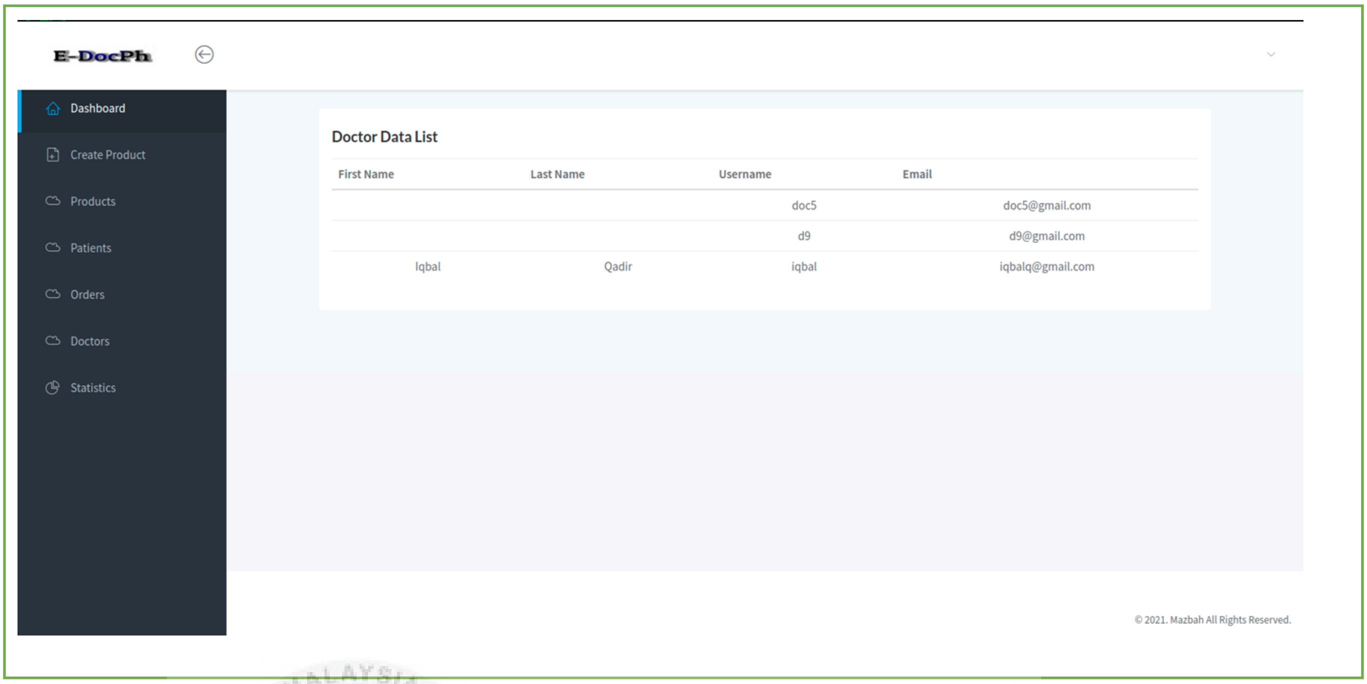


Order List

Patient	Date Ordered	Complete	Transaction ID	Authorized
p11	Aug. 18, 2021, 12:03 p.m.	True	1629288936.979506	Accepted
p12	Aug. 18, 2021, 12:04 p.m.	True	1629289011.712788	Accepted
p12	Aug. 18, 2021, 2:04 p.m.	True	1629295631.511222	Accepted
p9	Aug. 21, 2021, 7:43 a.m.	Completed	None	Accepted

© 2021. Mazbah All Rights Reserved.

17. Doctors

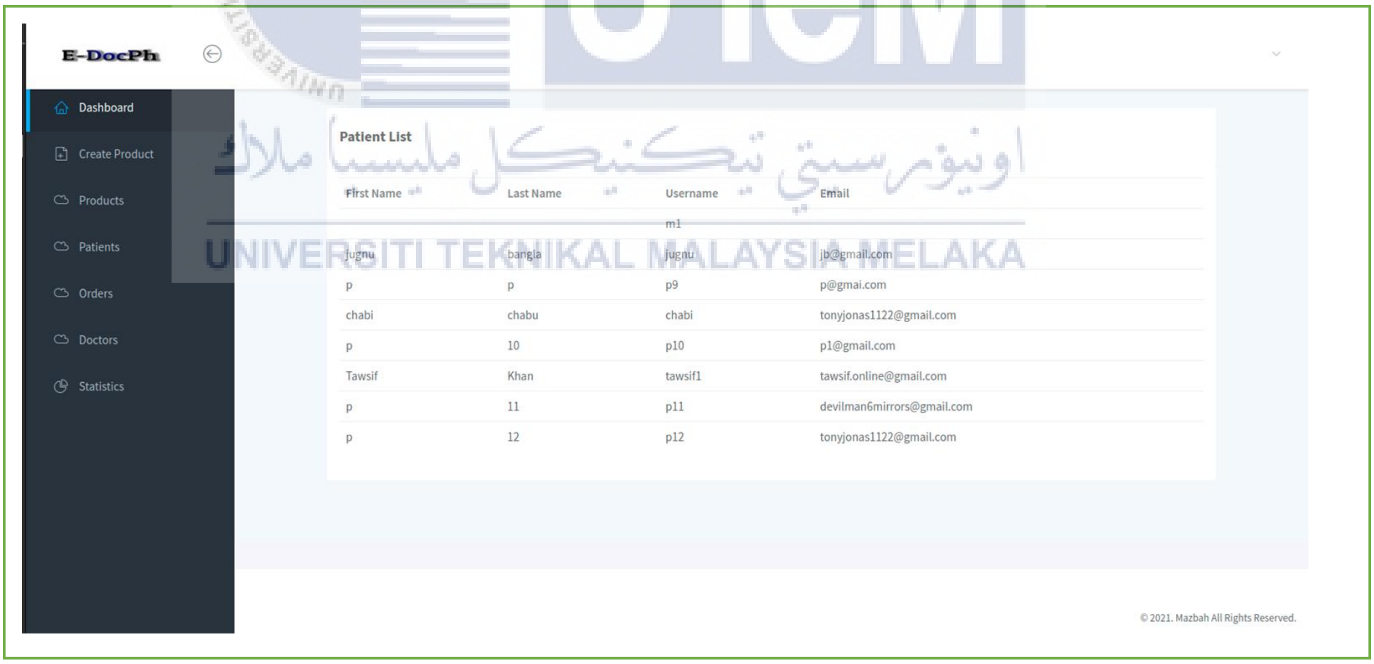


Doctor Data List

First Name	Last Name	Username	Email
		doc5	doc5@gmail.com
		d9	d9@gmail.com
Iqbal	Qadir	iqbal	iqbalq@gmail.com

© 2021. Mazbah All Rights Reserved.

18. Patients

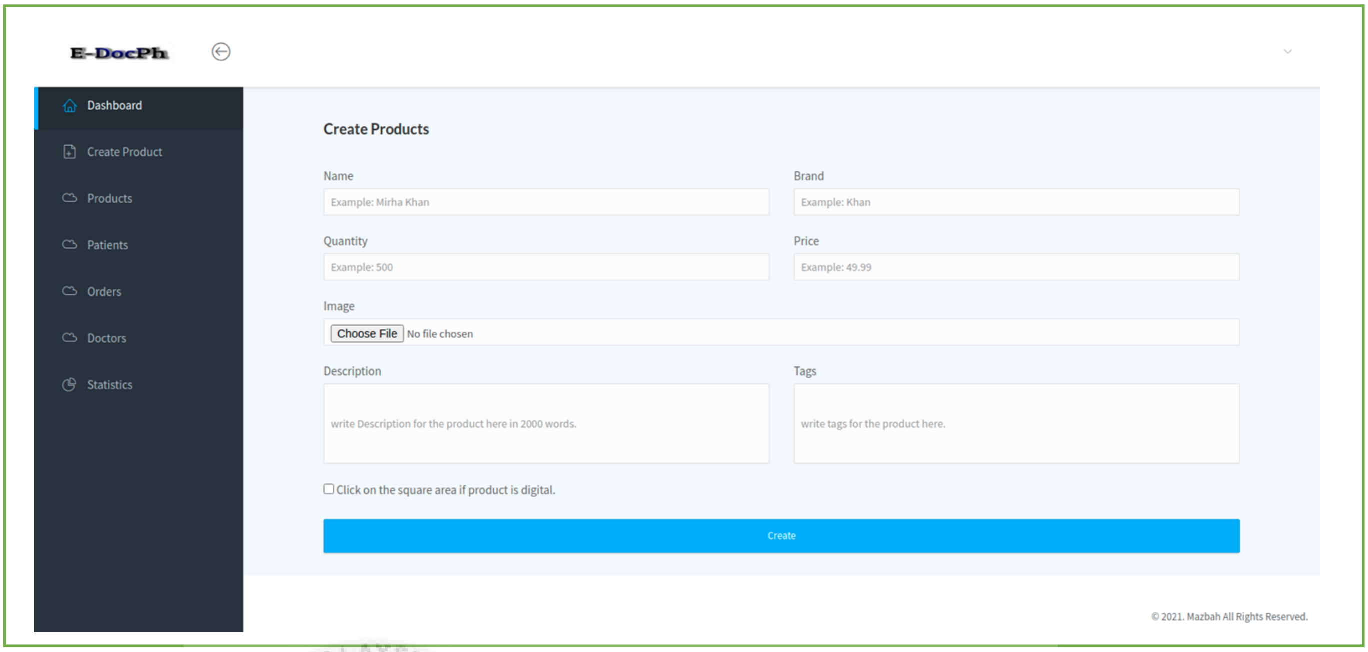


Patient List

First Name	Last Name	Username	Email
		m1	
jugnu	bangla	jugnu	jb@gmail.com
p	p	p9	p@gmai.com
chabi	chabu	chabi	tonyjonas1122@gmail.com
p	10	p10	p1@gmail.com
Tawsif	Khan	tawsif1	tawsif.online@gmail.com
p	11	p11	devilman6mirrors@gmail.com
p	12	p12	tonyjonas1122@gmail.com

© 2021. Mazbah All Rights Reserved.

19. Create Products



Create Products

Name: Example: Mirha Khan

Brand: Example: Khan

Quantity: Example: 500

Price: Example: 49.99

Image: Choose File No file chosen

Description: write Description for the product here in 2000 words.

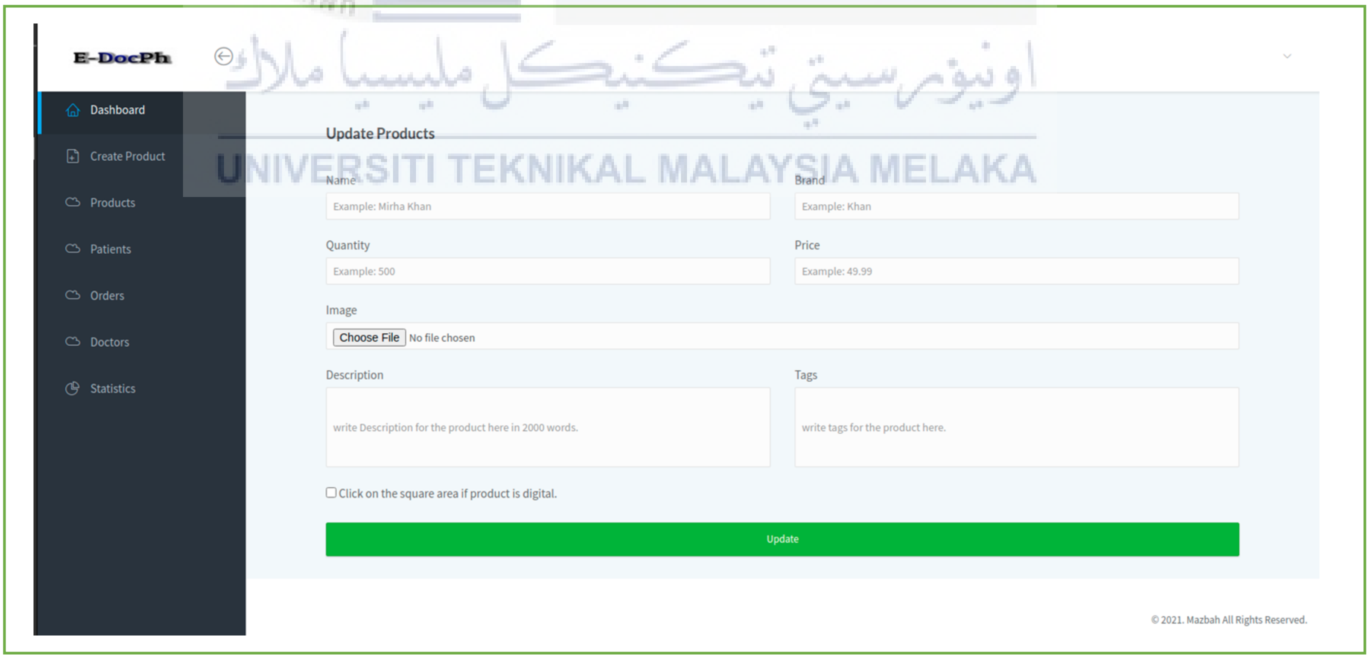
Tags: write tags for the product here.

□ Click on the square area if product is digital.

Create

© 2021. Mazbah All Rights Reserved.

20. Update Products



Update Products

Name: Example: Mirha Khan

Brand: Example: Khan

Quantity: Example: 500

Price: Example: 49.99

Image: Choose File No file chosen

Description: write Description for the product here in 2000 words.

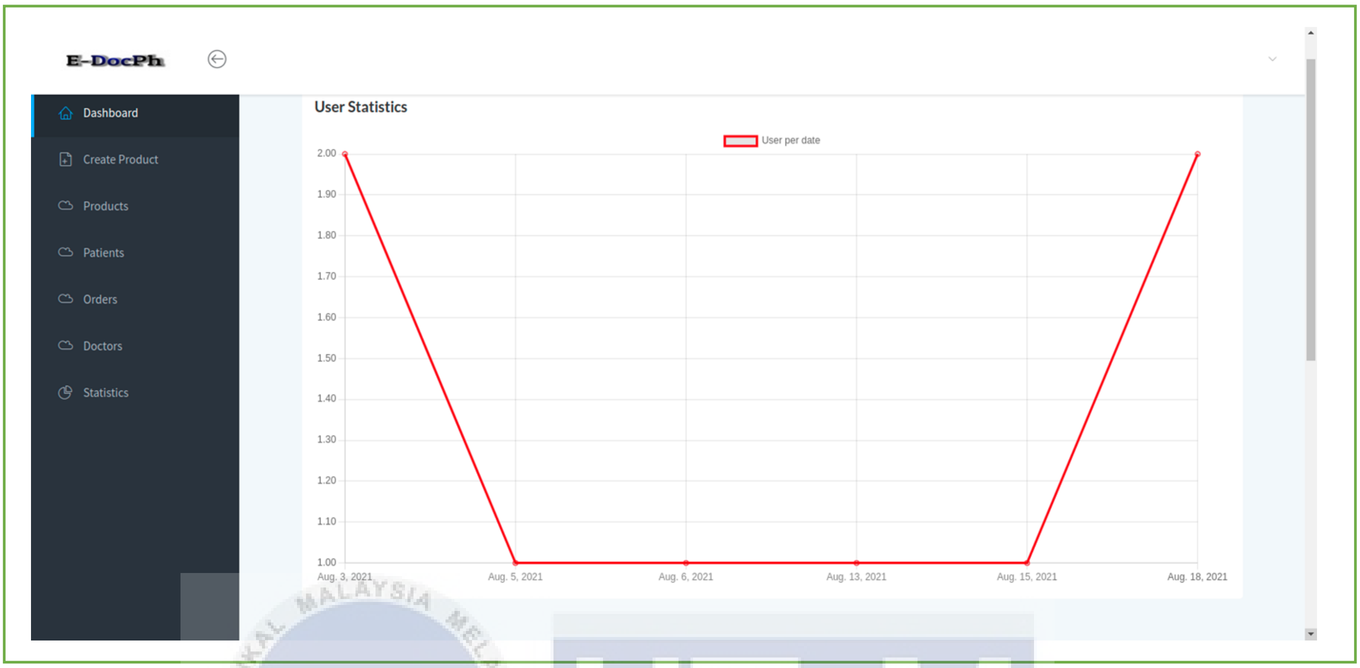
Tags: write tags for the product here.

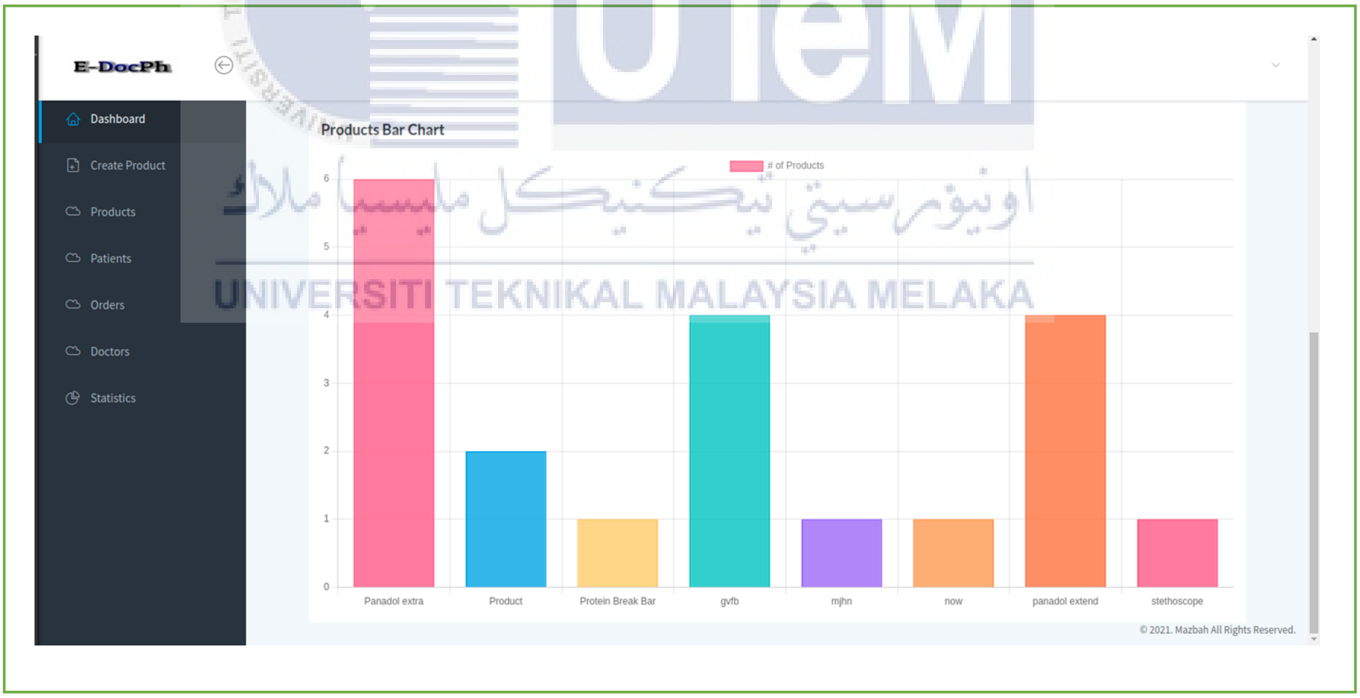
□ Click on the square area if product is digital.

Update

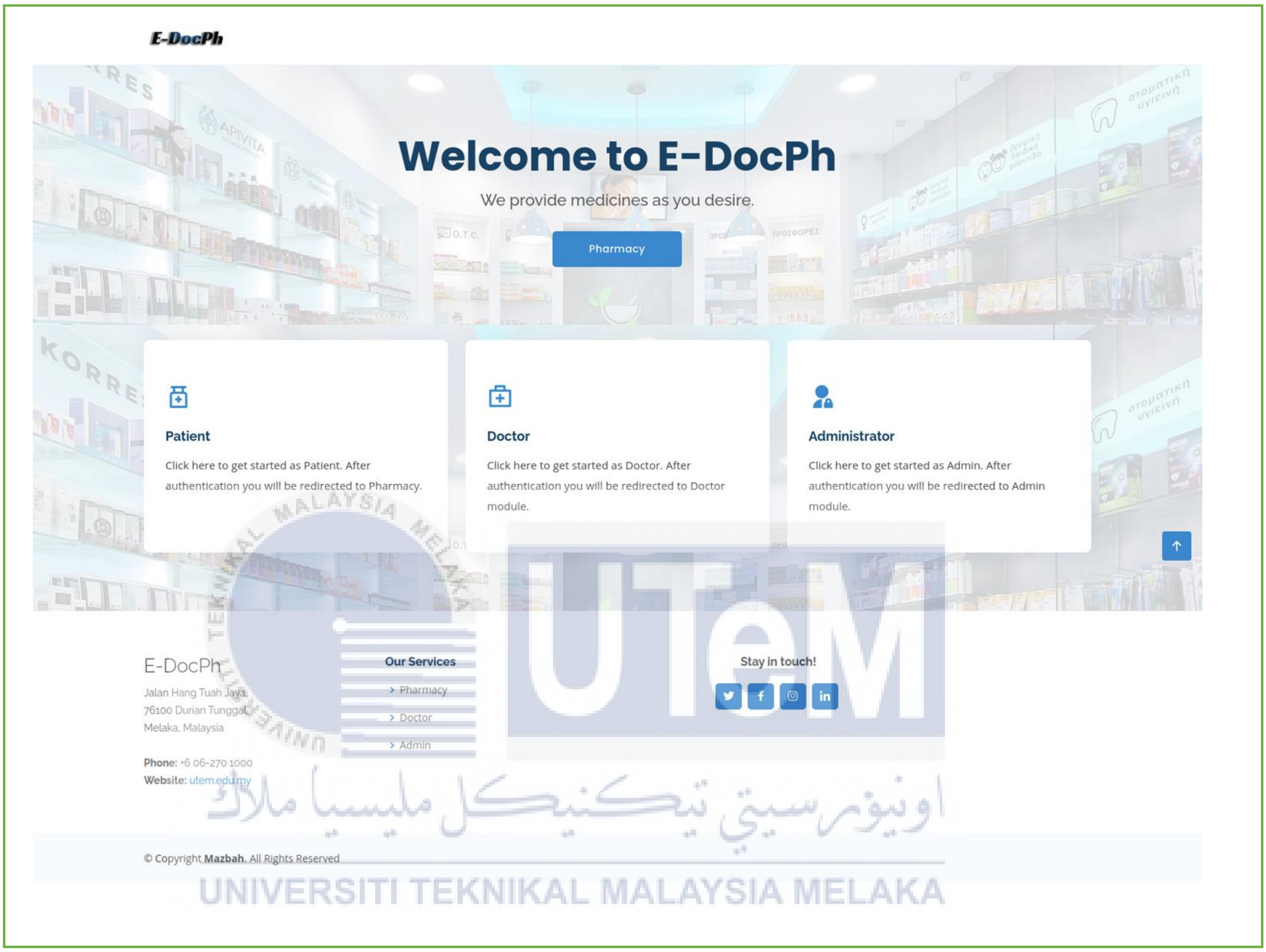
© 2021. Mazbah All Rights Reserved.

21. Statistics





22. Welcome Page



E-DocPh

Welcome to E-DocPh

We provide medicines as you desire.

Patient

Click here to get started as Patient. After authentication you will be redirected to Pharmacy.

Doctor

Click here to get started as Doctor. After authentication you will be redirected to Doctor module.

Administrator

Click here to get started as Admin. After authentication you will be redirected to Admin module.

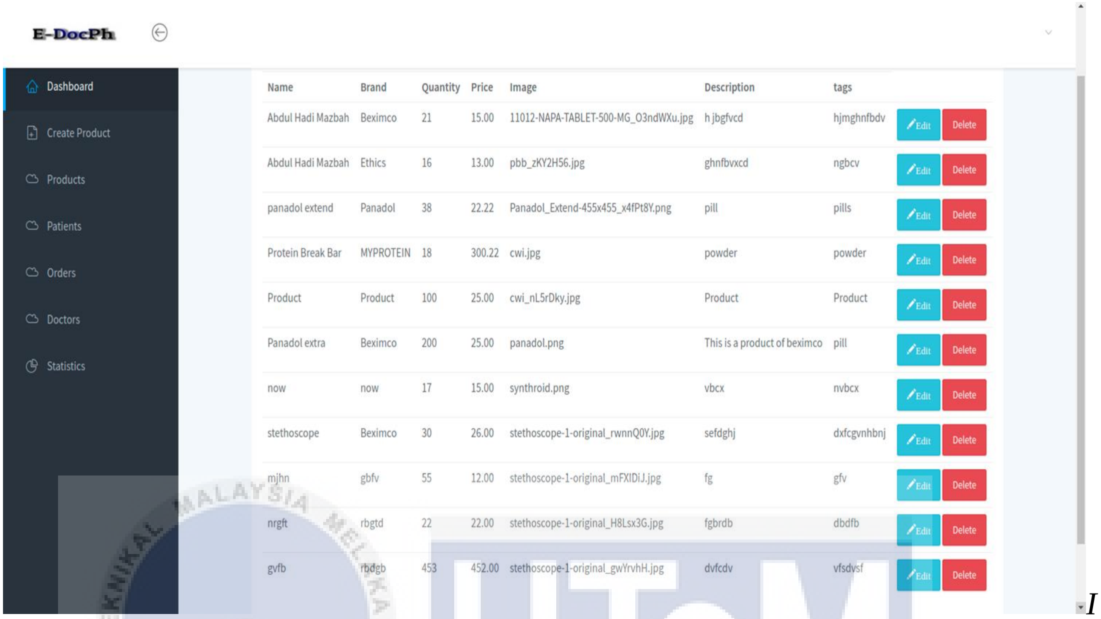
E-DocPh

Jalan Hang Tuah Jaya,
76100 Durian Tunggal,
Melaka, Malaysia

Phone: +6 06-270 1000
Website: utem.edu.my

© Copyright **Mazbah.** All Rights Reserved

23. Products



Name	Brand	Quantity	Price	Image	Description	tags	
Abdul Hadi Mazbah	Beximco	21	15.00	11012-NAPA-TABLET-500-MG_O3ndWXu.jpg	h jbgfvcd	hjmghnfbdv	Edit Delete
Abdul Hadi Mazbah	Ethics	16	13.00	pbb_zKY2H56.jpg	ghnfbvxcd	ngbcv	Edit Delete
panadol extend	Panadol	38	22.22	Panadol_Extend-455x455_x4fPt8Y.png	pill	pills	Edit Delete
Protein Break Bar	MYPROTEIN	18	300.22	cwi.jpg	powder	powder	Edit Delete
Product	Product	100	25.00	cwi_nL5rDky.jpg	Product	Product	Edit Delete
Panadol extra	Beximco	200	25.00	panadol.png	This is a product of beximco	pill	Edit Delete
now	now	17	15.00	synthroid.png	vbcx	nvbcx	Edit Delete
stethoscope	Beximco	30	26.00	stethoscope-1-original_rwnnQ0Y.jpg	sefdghj	dxfcgvnhbnj	Edit Delete
mjhn	gbfv	55	12.00	stethoscope-1-original_mFXIDiJ.jpg	fg	gfv	Edit Delete
nrgft	rbgtd	22	22.00	stethoscope-1-original_H8Lsx3G.jpg	fgbrdb	dbdfb	Edit Delete
gvfb	rbdgb	453	452.00	stethoscope-1-original_gwYrvhH.jpg	dvfcdv	vfsdvsf	Edit Delete

I