University of Asia Pacific

Computer Science and Engineering



Emergency Medicine and Doctor Services

SYSTEM ANALYSIS & DESIGN LAB PROJECT

Course Teacher: Abdullah Al Omar, Lecturer, CSE UAP

Nabiha Tahsin (18201024) | Shawon Barman (18201043) | Jesmin Aktar (18201044)

system analysis & design lab report | CSE 306 | April 6, 2021

Table of Contents

Chapter One: PROJECT PROPOSAL	
1.1 Project proposal-1 Emergency Medicine and Doctor Services	5
1.2 Project proposal-2 Car Rental Management System	
1.3 Project proposal-3 Online Doctor Appointment System	
1.4 Choose one project and why we choose it?	
Chapter Two: INTRODUCTION	
2.1 ABSTRACTPg. 8	
2.2 Introduction	
2.3 Problem Statement	
2.4 Goals of our system	
2.5 Objective of our system	0
2.6 Modules	O
2.7 Admin Modules)
2.8 User Module (Patient))
2.9 Doctor Module	
2.10 Pharmacist Module	
Chapter Three: REQUIREMENT ANALYSIS AND FEASIBILITY ANALYSIS	
3.1 Requirement Analysis	
3.2 Feasibility Analysis	
3.2.1 Economic feasibility	
3.2.2 Technical feasibility	
3.2.3 Organizational feasibility	
Chapter Four : DESIGN PHASE	
4.1 Use Case Diagram	
4.2 Data Flow Diagram	
4.3 ER Diagram	
4.4 UML Class Diagram	
Chapter Five : METHODOLOGY	
5.1 Methodology	
5.2 Which methodology is best for develop our system? Pg. 27	
Chapter Six: COMPLEX ENGINEERING PROBLEM	
Chapter Seven : SAMPLE SCREENSHOT	
7.1 Home Page	
7.2 Medicine details and order page	
7.3 Doctor details and take appointment page	
7.4 Login page	

7.5 Profile page	Pg. 38
7.6 Register page	Pg. 39
Chapter Eight: SYSTEM IMPLEMENTATION	
8.1 Login page code	
8.2 Register page code	Pg. 42
8.3 Profile code	pg. 44

Chapter One: PROJECT PROPOSAL

1.1Project proposal-1 Emergency Medicine and Doctor Services (Presented by - 18201043):

People who are from remote areas are deprived of proper treatment facilities. For this reason, they need to go to Dhaka city for getting better treatment, which is so much painful and also expensive. On the other hand, during emergencies, it's become very pathetic for them to collect necessary medicines. Using technology and the internet this system will be helpful to get proper treatment and get to collect necessary medicines.

The goal of the project is to provide proper medicine and doctor services through online for remote area.

Features:

- 1. Easiest way to get Medicine and Doctor information
- 2. Emergency Medicine delivery service
- 3. Contact with doctor by live video conferencing or voice call conferencing
- 4. A website and Application for android and ios platform

1.2 Project proposal-2 Car Rental Management System (Presented by - 18201024):

Car Management System is a software application that meets General management requirements related to the details of the Car, staff, expenses and fleet and report to an organization, including booking options. It provides users with a graphical interface to manage daily transactions. Provides management reports such as car request details, car request status, car fuel concepts, car distance records, car repair and maintenance records, and user responses.

The goal of the system is to develop a web-based system where customers will register and reserve the car through online and for the company to manage their car rental business and to relieve customers task whenever they need to rent a car.

Features:

- 1. Easy reservation
- 2. Vehicle tracking system
- 3. Easy and secure payment system
- 4. Localization

1.3 Project proposal-3 Online Doctor Appointment Booking System (Presented by 18201024):

The Doctor Appointment System provides the power of direct interaction between doctors of customer choice as and when required for your small problems. Patients will able to fill online from in just few seconds before entering to the virtual office room by using this system. It can also enable customer to upload their lab results such as x-ray copies, health history etc, which can be viewed by customer referred doctors. It can also maintain list of doctors, available appointment time, book appointment, cancel appointment, view his appointment history.

The goal of the system provides ease and comfort to patients while taking appointments from doctors and it also resolves the problems that the patients have to face while making an appointment.

Features:

- 1. Admin can add new user and remove patient details
- 2. Admin can manage the appointment
- 3. Admin can add doctor or remove doctor
- 4. User can take appointment from doctor

1.4 Choose one project and why we choose it?:

We have choose Emergency Medicine and Doctor Services project.

Many remote areas are available in our country and many people live in these remote areas. They are facing many problems. One of the biggest problems is that they are deprived of proper medical care and it becomes very painful for them to procure the necessary medicines in case of emergency. This is why human life in remote areas is dying day by day. To solve all these problems and to give a better healthy life and reduce mortality and morbidity rates, we choose this emergency medicine and doctor services project. Our goal is to provide the right medicine and doctor services online.

Chapter Two: INTRODUCTION

2.1 ABSTRACT:

The most important health care service is the Emergency Medicine and Doctor Service (EMDS) because it plays an important role in saving lives and reducing mortality and morbidity. In rural areas, it is impossible to get emergency treatment, and when situations demand emergency care arise, they rush and deploy expensive agencies. However, do not necessarily increase the cost. Efforts to Improve Emergency Care. The need for emergency medical care preserves the way to develop applications for emergency medicine and medical services. Patients can easily get treatment by logging into our system. It also provides information on the availability of physicians and medications. We will play a key role in making our system readily available to patients and physicians. In this research paper, we apply the concept of virtual organization to EMDS as a framework for managing the collaboration of different participants in EMDS. To visualize how the concept of virtual organization can facilitate the management of potential collaborations, we use DFD, use case, ER, and UML class diagrams to model the structural and behavioral aspects of the virtual organization in this EMDS case study.

2.2 Introduction:

The project Emergency Medicine and Doctor Services includes registration of patients, storing their details into the system, a personal profile of patients, contact with doctors by live video conferencing or voice call conferencing, order any medicine in our system, and also computerized billing in the medicine shops. Users can search the availability of a doctor or medicine using the name and the details of a patient using the id. Users can choose an appropriate doctor for treatment and then take an appointment from a doctor. Users can read blogs about health-related in our system.

Emergency Medicine and Doctor Services can be accessed using a username and password. The admin is the main part of our system because only the admin can add each data to the database. Data can be easily recovered. The interface in our system is very user-friendly. Data is secure for personal use and makes data processing very fast.

The Emergency Medicine and Doctor Services system is powerful, flexible, and easy to use, and is designed and developed to deliver real imaginable benefits to medical and home delivery medicine service.

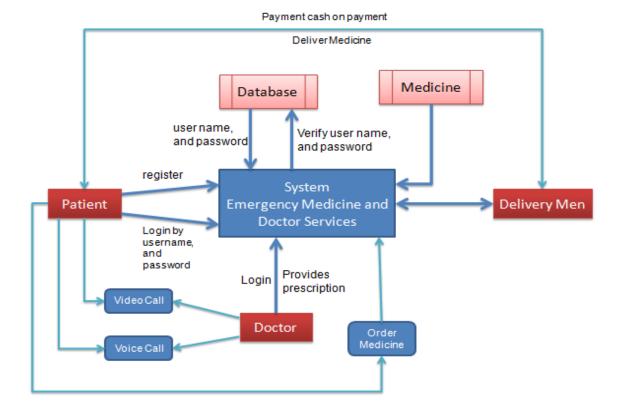


Figure-1: System Diagram

1.3 Problem Statement:

People who are from remote areas are deprived of proper treatment facilities. For this reason, they need to go to Dhaka city for getting better treatment, which is so much painful and also expensive. On the other hand, during emergencies, it's become very pathetic for them to collect necessary medicines.

1.4 Goals of our system:

- must be User Friendly
- ❖ low cost and very effective
- Provide proper medicine and best doctor services
- ❖ Easily contact with doctor through our system
- Home delivery medicine

1.5 Objective of our system :

- * Recording information about the patients that come
- * Recording the conferencing information between patient and doctor
- * Keeping information about medicines for healing
- * Recording information about the sell medicines

1.6 Modules:

The entire project mainly consists of 4 modules, which are:

- Admin module
- User module (Patient)
- Doctor module
- Pharmacist module

1.7 Admin Modules:

- Can add user, doctor, staff, medicine, medicine shop, medicine supplier, and delivery man.
- Can remove user, doctor, staff, medicine, medicine shop, medicine supplier, and delivery man.
- Can manage user and doctor account
- Watch and make appointment lists
- Watch transaction reports of patient payment
- Watch order details

1.8 User Modules (Patient):

- Can log in and register new account and manage own account
- View medicine and doctor details
- Take an appointment
- Order any medicine
- View prescription details
- Pay bill

1.9 Doctor Modules:

- Manage own account
- Create, and manage appointment list
- Give prescription details

1.10 Pharmacist Modules:

- Maintain shop
- Keep records of stock medicines and status
- View order details
- Add medicine
- Deliver medicine
- Get cash on payment

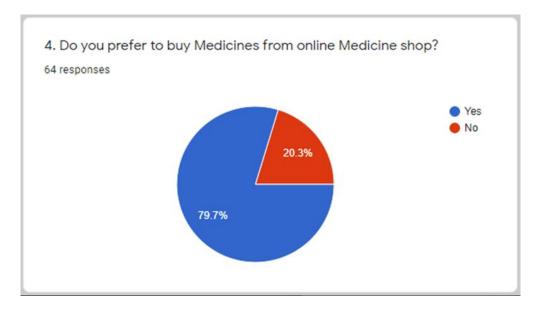
Chapter Three: REQUIREMENT ANALYSIS AND FEASIBILITY ANALYSIS

3.1 Requirement Analysis:

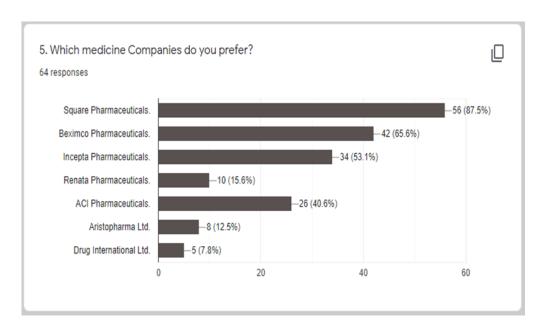
For finding the requirement, we have created a survey with some questions and 64 students responded to our survey.



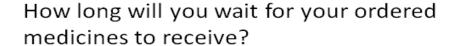
Here 67.2% of students said yes and 20.3% of students said maybe. Since many students said yes that's why we are very excited to implement this system.

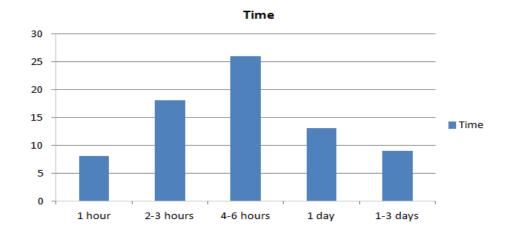


Here 79.7% of students preferred to buy medicines from online medicine shops. So we are sure that if we sell medicines from online medicine shops then many people buy medicine from our system.

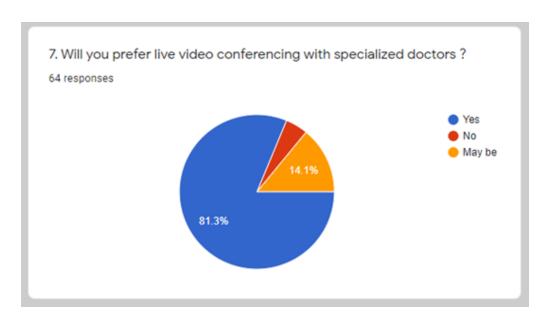


Here 87.5% of students preferred to sell Square Pharmaceuticals company, 65.6% of students preferred to sell Beximco Pharmaceuticals company, 53.1% of students preferred to sell Incepta Pharmaceuticals company, and 40.6% of students preferred to sell ACI Pharmaceuticals company. So these four medicine companies will be on the top of our system and we will store more and more these medicines for sale.

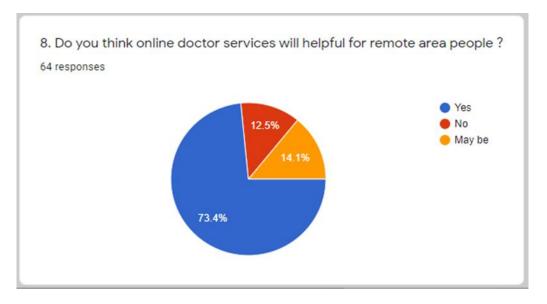




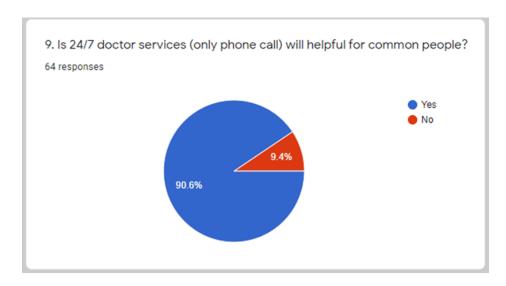
Here a maximum of students waits to received medicine within 6hours. But we can try our best to deliver the medicines within 1 hour.



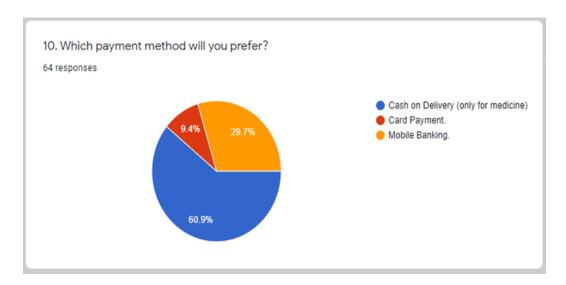
Here 81.3% of students preferred to live video conferencing with doctors. But we can also add voice call conferencing to our system.



Here 73.4% of students preferred that online doctor services will very helpful for remote areas people.



Here 90.6% of students preferred that only voice call doctor services will helpful for common people.



Here 60.9% of students preferred to cash on delivery method and 29.7% of students preferred to mobile banking payment method.

According to survey, we have decided to use:

- Include medicine in our system for sale.
- Delivery medicine within 1-2hours.
- Include live video conferencing or voice call conferencing with doctors.
- Include cash on delivery payment method, and mobile banking payment method.

3.2 Feasibility Analysis:

Feasibility analysis is one of the major Phase of SDLC in which an organization discusses about the cost and benefits of the software or system. Its also called the decision-making phase. Because profit from the system plays an important role if cost is very high then the company may face less. Feasibility analysis can be divided into three key components, which are Economic feasibility, Technical feasibility, and Organizational feasibility.

After the feasibility Studies –

- 1. The project may be accepted with some modification.
- 2. The project may be rejected.

3.2.1 Economic Feasibility:

Economic feasibility is determined by identifying costs and benefits associated with the system, assigning values to them, calculating future cash flows, ROI, BEP and measuring the financial worthiness of the project. There is two method to do that.

For our system, we will use simple cash flow method:

	Year 0	Year 1	Year 2	Year 3	Year 4	Total	
Benefits							
Increased sales		500,000	530,000	561,800	595,508	2,813,308	
Recluction in customer		70,000	70,000	70,000	70,000	350,000	
complaint calls							
Recluced invertory		68,000	68,000	68,000	68,000	340,000	
costs							
Total Benefits		638,000	668,000	699,800	733,508	3,503,308	
Development Costs							
2 servers	250,000	0	0	0	0	250,000	
Software licenses	34,825	0	0	0	0	34,825	
Server software	10,945	0	0	0	0	10,945	
Development labor	1,236,525	0	0	0	0	1,236,525	
Total Development	1,532,295	0	0	0 0		1,532,295	
Costs							
Operational Costs							
Hardware		50,000	50,000	50,000	50,000	250,000	
Software		20,000	20,000	20,000	20,000	100,000	
Operational labor		115,000	119,600	124,384	129,359	622,787	
Total Operational		185,000	189,600	194,384	199,359	972,787	
Costs							

Total Costs	1,532,295	185,000	189,600	194,384	199,359	2,300,638
Total Benefits – Total	[1,532,295]	453,000	478,400	505,416	534,149	1,202,670
Costs						
Commutative Net Cash	[1,532,295]	[1,079,295]	[600,895]	[95,479]	438,670	
Flow						

Return on Investment (ROI) =
$$\frac{3,503,308 - 2,300,638}{2,300,638} = 0.5227$$
 or 52.27%

Break Even Point (BEP) =
$$3 + \frac{534,149 - 438,670}{438,670} = 3.217$$

From this analysis -

- Acceptable ROI
- Benefit start from the first year

3.2.2 Technical Feasibility:

A technical feasibility study assesses the details of how we intend to deliver a product or service to customers.

The source of risks for our system are

- Users have a lack of familiarity with our system?
- Whether the treatment that doctors will give to patients will be right?
- Our system will give medicines to users will be original?
- Will user data safe with us?

3.2.3 Organizational Feasibility:

Organizational feasibility shows that how well the system ultimately will be accepted by its users and incorporated into the ongoing operations of the organization. It is important to think about strategic alignment. Either the project goals aligned with the business strategy.

If we will build it will they come –

- Our system will be one of the most efficient system available on the market.
- People from remote areas will be able to get the essential service at home without coming to the city at low costs.

Chapter Four: DESIGN PHASE

4.1 User Case Diagram:

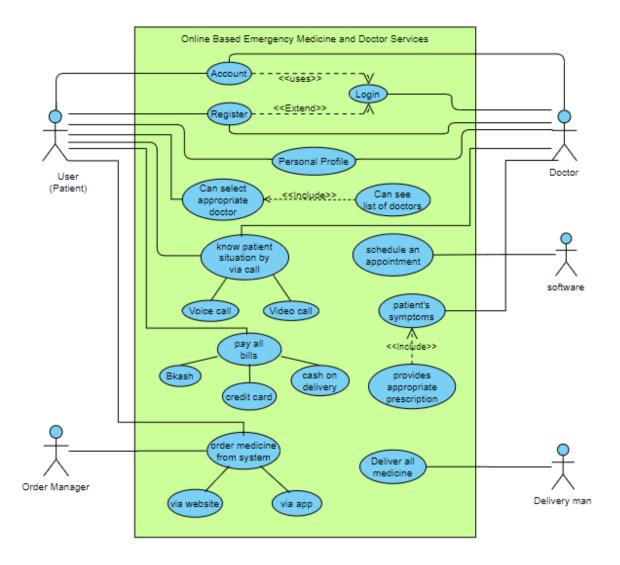


Figure-2: Use case Diagram

This is our use case diagram and there are 5 actors which are Users, Doctors, Software, Order manager, and Deliveryman. Also, there are many use cases and many connectors and one boundary. If users and doctors have an account then login into our system and then access our system. If users and doctors both have no account then register a new account and then login into our system and then easily access our system. For every user and doctor who has an account, our system provides him a personal account. Users can select the appropriate doctor by view the list of doctors. Users can take appointment from

a doctor. Then Software actor makes schedules an appointment. Then user and doctor will follow this schedule for conferencing. The user and doctor both will communicate by live video conferencing or voice call conferencing. In this conferencing doctor will ask many questions to know the situation of the patient. In this way, the doctor understands the patient's symptoms. According to this conferencing, the doctor provides an appropriate prescription and patients take this prescription and must follow this prescription. After conferencing, the user or patient pays the doctor charge with vat for our system by using the bkash payment or credit card payment. Users can also order any medicine from our system by seeing the list of medicine. Users can choose one or more medicine from our system and then order this. Order manager actor will prepare this order and send to delivery man for delivering. The Delivery Man actor will deliver this order who order it. User pays the medicine bill by using bkash payment or credit card payment or cash on delivery payment.

4.2 Data Flow Diagram (DFD):

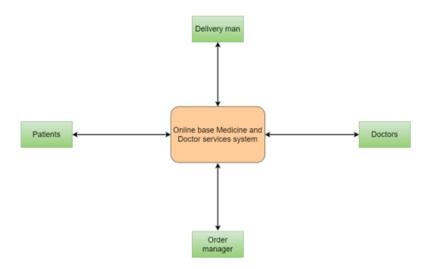


Figure-3: Level-0 DFD Diagram

This is a Level 0 Data Flow Diagram. DFD Level 0 is also called a Context Diagram. It's a basic overview of the whole system or process being analyzed or modeled. It's designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities. Here Patient, Doctor, Order manager, and Deliveryman these four entity can access our system by login into our system.

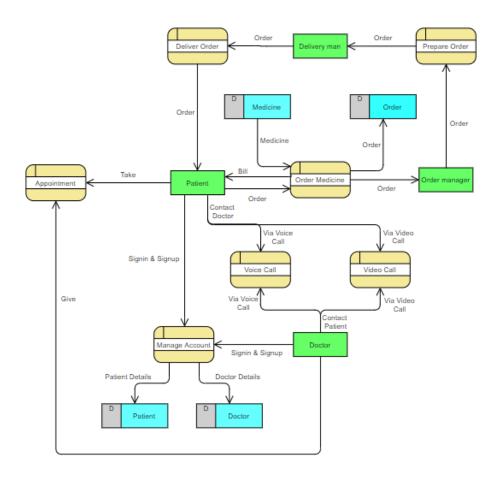


Figure-4: Level-1 DFD Diagram

That is a level-1 data flow diagram for our system. We can see that there are four entities which are Patient, Doctor, Order manager, and Deliveryman. Also, there are four data stores and seven processes, and many connectors. If patients have an account then they can sign in to our system by using the user name and password and then our system will send this user name and password to the patient data store. Then will check this user name and password have stores or not. If data are stored then the patient will perfectly sign in and then will access our system. If data are not stored that's mean the patient has no account in our system. In this case, they will signup for a new account and then sign in to access our system. Every patient's information is stored in the patient data store. The same cases are run for doctors. If the doctor has an account then will sign in for access to our system. If don't have an account then will signup for a new account and then will sign in to access. Every doctor's information is stored in the doctor data store. The patient can take an appointment and the Doctor will give this appointment. Then Software will make schedule an appointment. Then patient and doctor will follow this schedule for

conferencing. The patient and doctor both will contact by live video conferencing or voice call conferencing. According to this conferencing, the doctor will provide an appropriate prescription and patients take this prescription. The patient can see the list of medicine from our system and every medicine are stored in Medicine data store. The patient can order any medicine in our system. Our system will store this order in the Order data store and provide a bill receipt to the patient and this order will send to the order manager. The order manager checks this order and then will prepare this order for sent to the delivery man. When the delivery man will collect this order and then will deliver this order to who orders it. Then the patient will collect this order.

4.3 E-R Diagram:

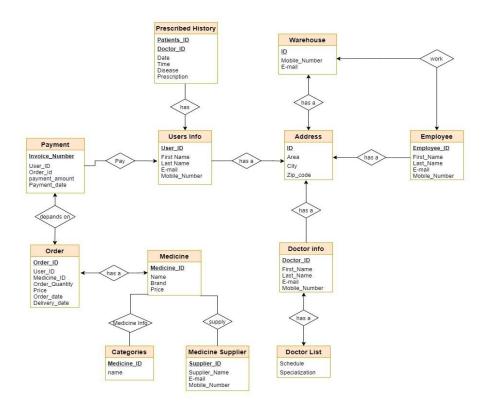


Figure-5: E-R Diagram

This is our Entity Relation Model. There are twelve tables. User's or patient's information is stored in the Users info table where User_id is the primary key for this table. Every doctor's information is stored in the Doctor info table where Doctor_id is the primary key. Doctor info tale has a Doctor list table. Every employee's information is stored in the

Employee table where Employee_id is the primary key. The employee works in a warehouse and every warehouse information is stored in the Warehouse table. User info, Doctor info, Employee, and Warehouse table has an Address table for stored the location. After conferencing doctor provides a prescription to the patient. That information is stored in the Prescribed history table where Patient_id and Doctor_id are foreign keys for this table. Users may or may not order any medicine in our system. All medicine information is stored in the Medicine table where Medicien_id is the primary key. Someone supplies this medicine in the warehouse. So every medicine supplier information is stored in medicine supplier table where supplier_id is the primary key. If a user orders any medicine in our system then this order information is stored in the Order table where Order_id is the primary key and User_id and Medicine_id are foreign keys for this table. The order table has depends on the Payment table. If the user has any order then the user must pay the bill for this order. This payment information is stored in payment table where invoice_number is the primary key and user_id and order_id are foreign key for this table.

4.4 UML Class Diagram:

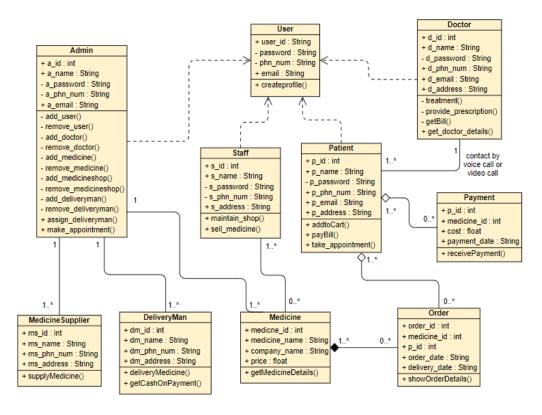


Figure-6: UML Class Diagram

This is our UML Class Diagram. There are ten classes. The User class is the root class. In the user class, there are four attitudes which are user_id, password, phn_num, and email. And also there has one operation which is createprofile(). Here password and phn_num are used for sign-in in our system, and user_id used for identifying users, and email used for contact with the user. The plus sign means public and minus sign means private. The user dependency classes are Admin, Staff, Patient, and Doctor. Admin class is the main class that plays a very important role. Admin class has five attributes. Admin can add the user or remove the user, add a doctor or remove a doctor, add medicine or remove medicine, add medicine shop or remove medicine shop, assign deliveryman and make or watch appointment list for doctor and user. The Staff class has five attributes and two operations which are maintain_shop() and sell_medicine(). The patient class has six attributes and three operations which are addtoCart() for order, payBill(), and take appointment(). The doctor class has six attributes and four operations which are checkup patient, provide a prescription, getPayment, and get_doctor_details(). One admin can deal with one or more medicine suppliers. The medicine supplier class has four attributes and one operation which is supplymedicine(). One admin can assign one or more delivery men for delivering medicine. The delivery man class has four attributes and two operations. One admin can add one or more medicine into our system. The medicine class has four attributes and one operation. One or more staff can sale zero or more medicine. One patient can order zero or more medicine and one medicine can order one or more patients. The order class has a composition relationship with the medicine class. Without medicine class, the order class has not worked. If patients order any medicine then they have to pay. The patient class has an aggregation relationship with the order class and payment class.

Chapter Five: METHODOLOGY

5.1 Methodology:

A formalized approch to implement the SDLC. Every project manager knows how crucial is to decide the right methodology for the project. There are many methodology but implementing depends on the type of project we have choosen.

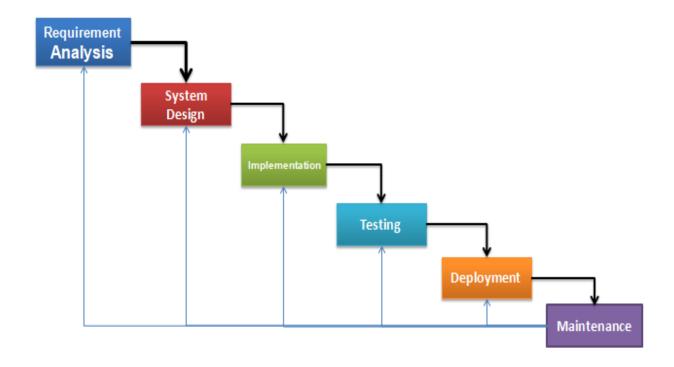
There are some methodology which are use in our project:

- Waterfall Methodology
- Parallel Methodology
- V-model Methodology
- Iterative Methodology
- System Prototyping Methodology
- Agile Development Methodology

5.2 Which Methodology is best for develop our system? :

We think that Waterfall Methodology is best for develop our system.

General Overview of "Waterfall Model"



We use every six of this phase to develop our project. Details given below

Requirement Analysis:

In our project which is known as "Emergency Medicine and Doctor Services". We gather requirement and analyzing every single detail which was related about our project. This phase is the main focus of project managers, doctors, and stakeholders. After requirement analysis we have decided to use:

- Include medicine in our system for sale.
- Delivery medicine within 1-2hours.
- Include live video conferencing or voice call conferencing with doctors.
- Include cash on delivery payment method, and mobile banking payment method.

Design:

In this phase, the system and software design is prepared from the requirement specifications which were studied in the first phase. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.

In our project, we also use this phase. We designed our project/ Now we elaborate on the design of our project.

We have used Use Case Diagram, Data Flow Diagram, E-R Diagram, and UML Class Diagram in our project.

Implementation/Codeing:

On receiving system design documents, the work is divided into modules/units and actual coding is started.

Our project is implemented by the web application is developed by HTML, CSS, Bootstrap, jQuery, JavaScript, Django, and Database is also used for manage the full data system. Django is used to develop the server, HTML and CSS used for developing the used user, admin, doctor and order manager side. JavaScript also does the same thing as HTML & CSS. The database manages the system data processing of the project.

Testing:

Testing is after the code is developed it is tested against the requirements to make sure that the product is actually solving the needs addressed and gathered during the requirements phase. During this phase unit testing, integration testing, system testing, acceptance testing are done.

After we complete our project we tested it. We test every single detail of it. Also, see that is our project really fulfills the demand of users requirements. Also, test the server and we system also.

Deployment:

Deployment is after successful testing the product is delivered/deployed to the customer for their use.

Maintenance:

Once when the customers start using the developed system then the actual problems come up and need to be solved from time to time. This process where the case is taken for the developed product is known as maintenance.

Chapter Six: COMPLEX ENGINEERING PROBLEM

A PROJECT ABOUT EMERGENCY MEDICINE AND DOCTOR SERVICES

Motivation: People who are from remote areas are deprived from proper treatment facilities. For this reason they need to go to Dhaka city for getting better treatment, which are so much painful and also expensive. One the other hand, during emergency situation its became very pathetic for them to collect necessary medicines. Using technology and internet this system will be helpful to get proper treatment and get to collect necessary medicines.

Objective: The goal of the project is to provide proper medicine and medical services through online for remote area.

Critical Challenges: Working with the medicine company and doctors will be a critical challenges.

Conflicting requirements: Without seeing the patient, prescribe a prescription can be risky for doctors. That is why they will hesitate to work. Thus, managing the doctors will be one of the most critical challenges.

No healthcare services for 5 years in Bagarchar UP



Figure: Bagarchar UP Health Center

-Various problems forced the Bagarchar UP Health Center to shut its services for more than five years at Bakashiganj Upazila in Jamalpur. As a result, the local patients have to go to the 12-kilometer-away government and private hospitals of the Upazila and district.

Source : https://www.daily-

<u>bangladesh.com/english/No-healthcare-services-for-</u> <u>5-years-in-Bagarchar-UP/53100</u> - 04:10 PM, 8 November 2020

Some components of Complex Engineering Problem:

Knowledge Profile (K)
K1 – natural sciences
K2 – mathematics
K3 – engineering fundamentals
K4 – specialist knowledge
K5 – engineering design
K6 – engineering practice
K7 – comprehension
K8 – research literature

Attribute	P1 and some or all of P2 to P7:
Depth of knowledge required	P1: one or more of K3, K4, K5, K6 or K8
Range of conflicting requirements	P2: wide-ranging or conflicting technical, engineering and other issues
Depth of analysis required	P3: no obvious solution
Familiarity of issues	P4: Involve infrequently encountered issues
Extent of applicable codes	P5: outside problems encompassed by standards and codes of practice
Extent of stake-holder involvement and conflicting requirements	P6: diverse groups of stakeholders with widely varying needs
Interdependence	P7: many component parts or sub-problems

Let's explore how a few P's could be addressed through this project:

P1 (*Depth of knowledge required- one or more of K3, K4, K5, K6 or K8*): This project generally requires a study of similar work with the same purpose as ours (K8 – Research Literature), Design the program for system(k5 - engineering design), We must need some medicine knowledge, and medical knowledge (K4 – Specialist Knowledge), Developing an App for this project (K6 – Engineering Practice).

P2 (*Range of conflicting requirements*): Data Analysis with proper regularization, while limited real data is available, will be create a Conflict for this project.

P6 (Extent of stake-holder involvement and conflicting requirements- diverse groups of stakeholders with widely varying needs): Various groups of stakeholders also medicine stores can be benefited from this project.

P7 (*Interdependence- many component parts or sub-problems*): This project involves three subsystems mainly:

1. Application model.

List of activities (As):

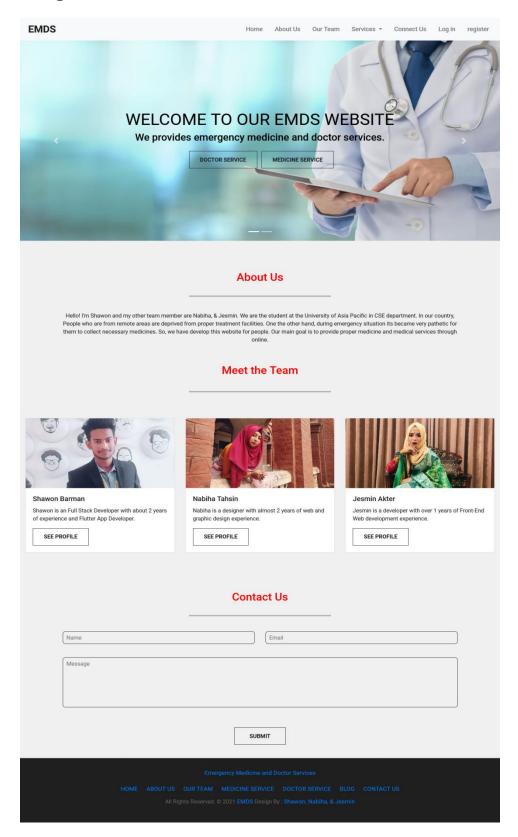
Attribute	Some or all of the following:
Range of resources	A1: use of diverse resources (include people, money, equipment, materials, information and technologies)
Level of interaction	A2: resolution of significant problems arising from interactions between wide-ranging or conflicting technical, engineering or other issues
Innovation	A3: creative use of engineering principles and research based knowledge in novel ways
Consequences for society and the environment	A4: consequences in a range of contexts, characterized by difficulty of prediction and mitigation
Familiarity	A5: Can extend beyond previous experiences by applying principles-based approaches

Let's explore how a few A's could be addressed through this project:

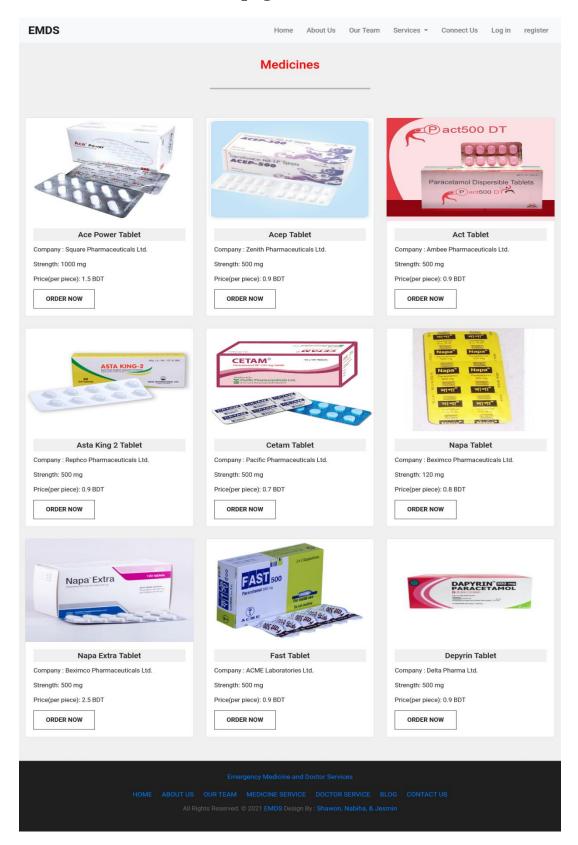
Attribute	Some or all of the following					
Range of resources	A1 (<i>Range of resources</i>): This plan will require the collaboration of various organizations including people (survey), finance (consideration of project creation), information and technology.					
Level of interaction	A2 (<i>Level of interaction</i>): A good level of interaction with the doctors, pharmacists and the faculty members of pharmacy departments is very important.					
Familiarity	A5 (<i>Familiarity</i>): Solving a medical-related problem will be a challenge for CSE students.					

Chapter Seven: SAMPLE SCREENSHOT

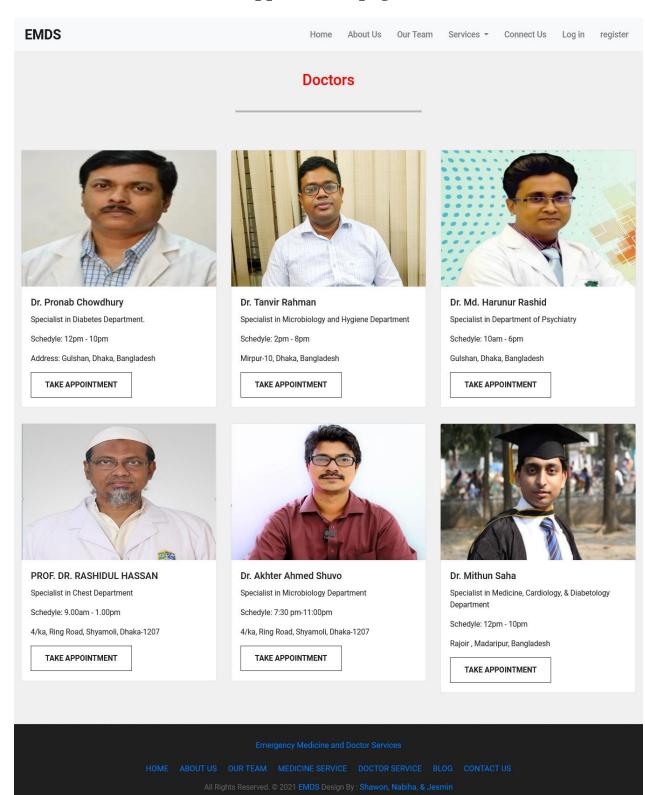
7.1 Home Page:



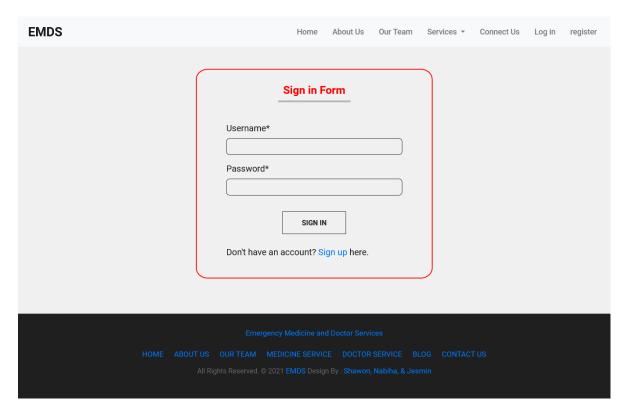
7.2 Medicine details and order page:



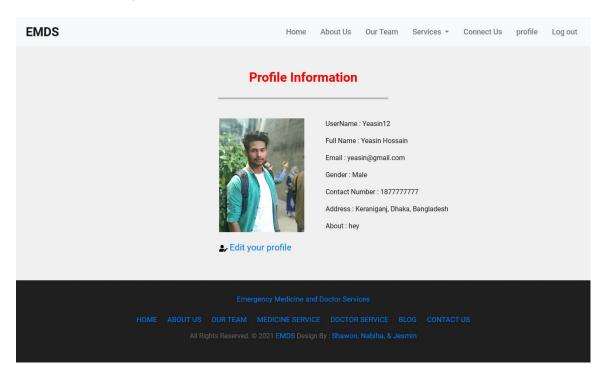
7.3 Doctor details and take appointment page:



7.4 Log in Page:



7.5 Profile Page:



7.6 Register Page:

EMDS	Home	About Us	Our Team	Services *	Connect Us	Log in	register
EMDS	Username* Required. 150 characters or fewer. Let First name* Last name* Email*	Form		Services *	Connect Us	Login	register
	Password* • Your password can't be too siminformation. • Your password must contain at • Your password can't be a common eventure of the entirely password confirmation* Enter the same password as before, SIGN Already have an account? Sig	least 8 characteronly used passinumeric.	ers.				
HOME AE	Emergency Medicine a OUT US OUR TEAM MEDICINE SERV All Rights Reserved. © 2021 EMDS Des				T US		

Chapter Eight: SYSTEM INPLEMENTATION

8.1 Log in page code:

• User_login.html:

```
{% extends 'base.html' %}
{% block title %}
    EMDS | log in
{% endblock %}
{% block content %}
    {% load crispy forms tags %}
    <div class="container signinform">
        <div class="heading">
            Sign in Form
        </div>
        <form action="" class="form-group" method="POST">
            {% csrf token %}
                {{ form | crispy }}
            <div class="submit">
                <button class="form-group btn btn-border">
                    sign in
                </button>
            </div>
            <div class="formfooter">
                Don't have an account? <a href="{% url
'register' %}">Sign up</a> here.
        </div>
        </form>
{% endblock %}
```

• urls.py:

```
from django.contrib import admin
from django.urls import path
from user import views as user_views
urlpatterns = [
    path('admin/', admin.site.urls),
    path('login/',
auth_views.LoginView.as_view(template_name='user_login.html'),
name='login'),
]
```

8.2 Register Page Code:

• user_registration.html:

```
{% extends 'base.html' %}
{% block title %}
    EMDS | registration
{% endblock %}
{% block content %}
    {% load crispy forms tags %}
    <div class="container registerform">
        <div class="heading">
            Sign up Form
        <form action="" class="form-group" method="POST">
            {% csrf token %}
                {{ form | crispy }}
            <div class="submit">
                <button class="form-group btn btn-border">
                    sign up
                </button>
        </form>
        <div class="formfooter">
            Already have an account? <a href="{% url
'login' %}">Sign in</a> here.
        </div>
{% endblock %}
```

• form.py:

• views.py:

```
def registration(request):
    if (request.method == 'POST'):
        form = UserRegistrationForm(request.POST)
        if(form.is valid()):
            form.save()
            username = form.cleaned data.get('username')
            password = form.cleaned data.get('password1')
            new user = authenticate(username=username,
password=password)
            login(request, new user)
            profile = ProfileDetails(user=request.user)
            profile.save()
        return redirect('home')
    else:
        form = UserRegistrationForm()
    context={
        'form': form
    return render(request, 'user registration.html',context)
```

• urls.py:

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path('', user_views.homepage, name = 'home'),
    path('login/',
auth_views.LoginView.as_view(template_name='user_login.html'),
name='login'),
    path('register/', user_views.registration,
name='register'),
]
```

8.3 Profile Code:

• User_profile.html :

```
{% extends 'base.html' %}
{% block title %}
    EMDS | profile
{% endblock %}
{% block content %}
    {% load static %}
    <div class="container-fluid" id="profileinfo">
        <div class="header color">Profile Information</div>
        <div class="about">
            <div class="profile pic">
                <img class="center" src="{{</pre>
user.profiledetails.profile pic.url }}" alt="{{
user.profiledetails.profile pic.url }}">
            </div>
            <div class="link">
                <a href="{% url 'profileUpdate' %}"><img
src="{% static 'user/img/user-edit-solid.svg' %}"
alt="edit logo"> Edit your profile</a>
            </div>
        </div>
        <div class="information">
            UserName : {{ name }}
            Full Name : {{ first name }} {{ last name}}
} } 
            Email : {{ email }}
            Gender: {{ user.profiledetails.gender}
```

• views.py:

```
@login_required
def profile(request):
    user = get_user(request)

context={
        'name': user.username,
        'first_name': user.first_name,
        'last_name': user.last_name,
        'email': user.email,
    }
    return render(request, 'user_profile.html',context)
```

• Urls.py:

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path('', user_views.homepage, name = 'home'),
    path('login/',
auth_views.LoginView.as_view(template_name='user_login.html
'),name='login'),
    path('logout/',
auth_views.LogoutView.as_view(template_name='user_logout.ht
ml'),name='logout'),
    path('register/', user_views.registration,
name='register'),
    path('profile/', user_views.profile, name='profile'),
    path('profileUpdate/', user_views.profile_update,
name='profileUpdate'),
]
```