|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | Daffodil International University | | | | |  |  | |
|  |  | | |  | | |  | |  | |
|  | | | | | | | | | | |
| http://4.bp.blogspot.com/-SotLIkzmscA/UqQLE9BjdPI/AAAAAAAAARc/-SF46qFfVE0/s1600/daffodil-international-university-logo.png | | | | | | | | | | |
|  | | | | | | | | | | |
| Project Name: Medicine Search Engine  Course Project  Software Engineering Project-II (Web Programming)  Course Code: SWE332 | | | | | | | | | | |
|  | | NAME | | | **:** | Kazi Kaysar | | | |  |
| ID | | | **:** | 143-35-785 | | | |
| DEPARTMENT | | | **:** | Software Engineering | | | |
| Faculty of Science and Information Technology | | | | | | | |
|  | | |  |  | | | |
| December 2017 | | | | | | | | | | |

Table of Contents

[Chapter 1 1](#_Toc497901973)

[1. Introduction 2](#_Toc497901974)

[1.1 About the System 2](#_Toc497901975)

[1.2 Purpose 2](#_Toc497901976)

[1.3 Scope 2](#_Toc497901977)

[1.4 Vision 2](#_Toc497901978)

[1.5 Why this system is necessary? 2](#_Toc497901979)

[1.6 Proposed Solution 2](#_Toc497901980)

[Chapter 2 3](#_Toc497901981)

[2. System Analysis 4](#_Toc497901982)

[2.1 Use Case Model 5](#_Toc497901983)

[2.2 Actor Goal List 5](#_Toc497901984)

[2.3 Use Case Model 5](#_Toc497901986)

[2.4 Use Case Description (Brief) 5](#_Toc497901987)

[2.5 Use Case Description (Detailed) 6](#_Toc497901989)

[2.5.1 Input Syndrome 6](#_Toc497901990)

[2.5.1 Show Disease 6](#_Toc497901990)

[2.5.1 Give Medicine 6](#_Toc497901990)

[2.6 System Sequence Diagrams 6](#_Toc497901991)

[2.6.1 User Registration 6](#_Toc497901992)

[2.7 Domain/Conceptual Model 7](#_Toc497901993)

[Chapter 3 9](#_Toc497901995)

[3. System Design 10](#_Toc497901996)

[3.1 Sequence Diagrams 10](#_Toc497901997)

[3.2 Class Diagram 11](#_Toc497901999)

[3.3 Entity Relationship Diagram 13](#_Toc497902000)

[Chapter 4 15](#_Toc497902001)

[4. Implementation 15](#_Toc497902002)

[4.1 Tools &Technologies 16](#_Toc497902003)

[4.2 Project Link 16](#_Toc497902004)

[Chapter 5 17](#_Toc497902005)

[5. System Testing 19](#_Toc497902006)

[5.1 Why Software Testing is Essential? 19](#_Toc497902007)

[5.1.1 Black box testing 19](#_Toc497902008)

[5.1.2 White box testing 19](#_Toc497902009)

[5.2 Test Cases 19](#_Toc497902010)

[5.2.1 Test case: Sign up(Registration process) 19](#_Toc497902011)

[Chapter 6 20](#_Toc497902012)

[6. Conclusion 21](#_Toc497902013)

[6.1 Good Features of the System 21](#_Toc497902014)

[6.2 Limitations of the System 21](#_Toc497902015)

[Appendices 22](#_Toc497902017)

[7. User Manual 23](#_Toc497902018)

[8. References 24](#_Toc497902020)

Chapter 1

Introduction

# Introduction

## Backgrounds the System

This project is developing for the people who have been suffering for disease or health problem. If they want to know what is his/her disease they can login this site give their syndrome and they can get their disease name and medicine.

## Purpose

Anyone can visit and know about their problem.

Anyone can take medical treatment from this system.

Anyone can know about his/her disease.

Anyone can take information about blood donate.

## Scope

* people can easily know their disease and it’s medicine. As a overpopulated country we have face many problems to visit a doctor. So it will be very important service for the people to get a good health.

## Vision

People can easily know their health problem through this site.

## Why this system is necessary?

By the use of this site people can easily know their disease and it’s medicine. . If they want they can give their syndrome and they can get their disease name and medicine. So it will be very important service for the people to get a good health.

# 

Chapter 2

**System Analysis**

# System Analysis

## Use Case Model



## Actor Goal List

### Admin

User, Admin

### User

Can Input Syndrome

Can get health tips

Can get blood donate information

## Use Case Model

## Use Case Description (Brief)

## Use Case Description (Detailed)

### Input Syndrome

|  |  |  |
| --- | --- | --- |
| Use Case ID | 1 | |
| Name | Input Syndrome | |
| Primary Actor | User | |
| Secondary Actor | Admin | |
| Goal | Give Information to the system | |
| Precondition | 1. If the user give proper information than it will be match with syndrome. | |
| Post Condition | If the information is wrong than system will be fail. | |
| Main Success Scenario | Actor | System |
| If the user give proper information than it will be match with syndrome. | When the user give proper information than it will be match with syndrome. |
|  |  | |

### Show Disease

|  |  |  |
| --- | --- | --- |
| Use Case ID | 2 | |
| Name | Show Disease | |
| Primary Actor | Admin | |
| Secondary Actor | User | |
| Goal | Give the problem information. | |
| Precondition | If the information match with syndrome than system will show the disease name. | |
| Post Condition | If the information is wrong than system will be fail. | |
| Main Success Scenario | Actor | System |
| If the information match with syndrome than system will show the disease name. | Syndrome match than system will show the disease name. |
|  |  | |

### Give Medicine

|  |  |  |
| --- | --- | --- |
| Use Case ID | 3 | |
| Name | Give Medicine | |
| Primary Actor | Admin | |
| Secondary Actor | User | |
| Goal | Give the solution of that problem. | |
| Precondition | When system will show the disease name than it will give the medicine of that disease. | |
| Post Condition | If the other information is correct than system will not fail. | |
| Main Success Scenario | Actor | System |
| When system will show the disease name than it will give the medicine of that disease.  . | When how disease name than it will give the medicine of that disease. |
|  |  | |

## System Sequence Diagrams

### User registration

**2.6.1.1 Success scenario**

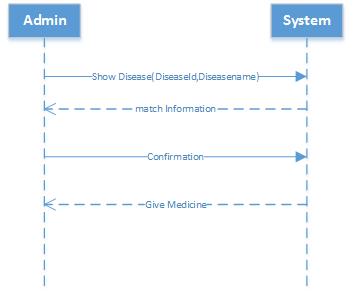


Fig: System sequence of use case 1

**2.6.1.2 Success scenario**

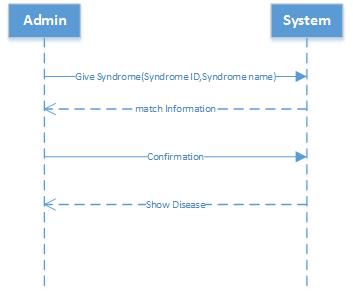
****

Fig:System sequence of use case 2

**2.6.1.3 Success scenario**

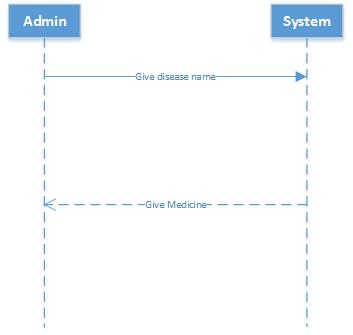
****

Fig:System sequence of use case 3

Chapter 3

**System Design**

# System Design

System design is the process of defining the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system. It is meant to satisfy specific needs and requirements of a business or organization through the engineering of a coherent and well-running system.

Systems design implies a systematic approach to the design of a system. It may take a bottom-up or top-down approach, but either way the process is systematic wherein it takes into account all related variables of the system that needs to be created—from the architecture, to the required hardware and software, right down to the data and how it travels and transforms throughout its travel through the system. Systems design then overlaps with systems analysis, systems engineering and systems architecture.

The systems design approach first appeared right before World War II, when engineers were trying to solve complex control and communications problems. They needed to be able to standardize their work into a formal discipline with proper methods, especially for new fields like information theory, operations research and computer science in general.

## Sequence Diagrams

A sequence diagram is an[interaction diagram](https://en.wikipedia.org/wiki/Interaction_diagram)that shows how objects operate with one another and in what order. It is a construct of a [message sequence chart](https://en.wikipedia.org/wiki/Message_sequence_chart)**.**

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

A sequence diagram shows, as parallel vertical lines, different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

## Sequence Diagram

**3.2.1.** Sequence diagram for use case 1:

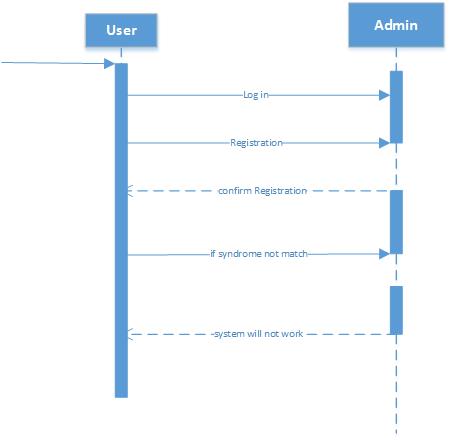
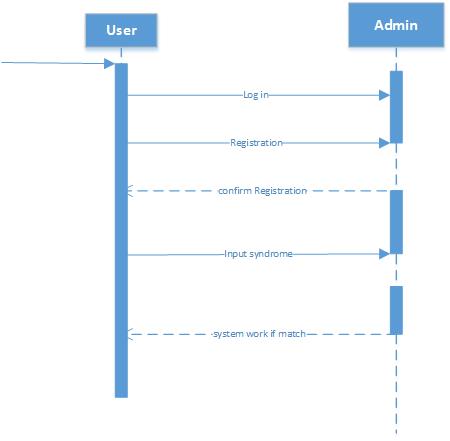
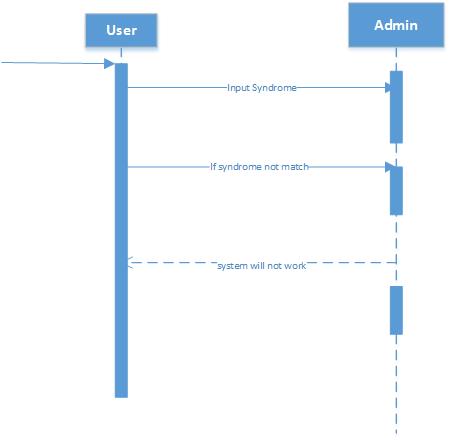
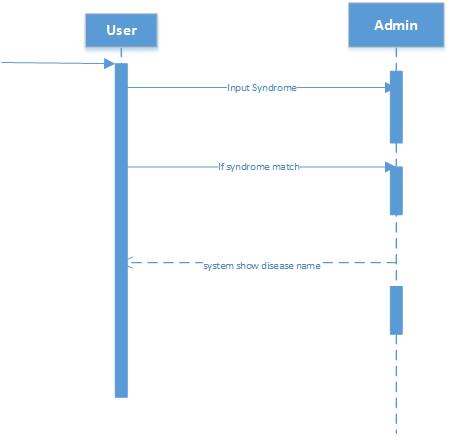
****

Fig: Success Scenario Fig: Failure Scenario

**3.2.2.** Sequence diagram for use case 2:



. Fig: Success Scenario Fig: Failure Scenario

**3.3.3.** Sequence diagram for use case 3:

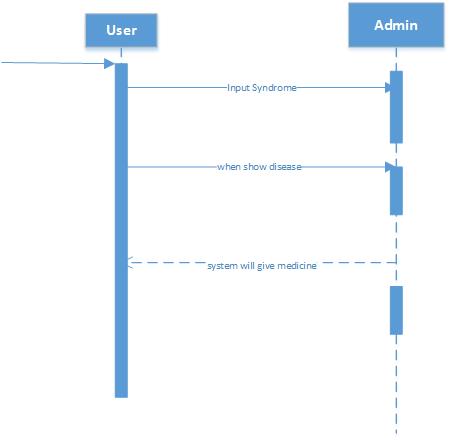


Fig: Success Scenario

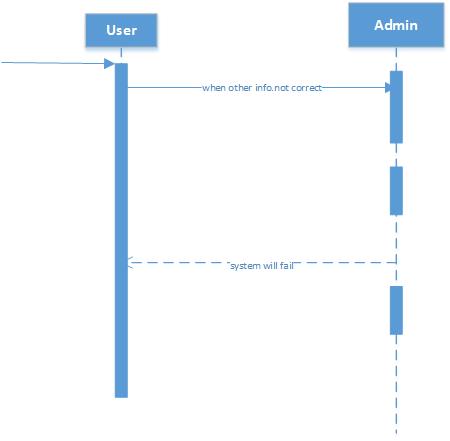


Fig: Failure Scenario

## Class Diagram

A class diagram is an illustration of the relationships and [source code](http://searchsoa.techtarget.com/definition/source-code) dependencies among classes in the Unified Modeling Language (UML). In this context, a [class](http://whatis.techtarget.com/definition/class) defines the [method](http://searchcio-midmarket.techtarget.com/definition/method)**s** and [variable](http://whatis.techtarget.com/definition/variable)**s** in an [object](http://searchsoa.techtarget.com/definition/object), which is a specific entity in a program or the unit of code representing that entity. Class diagrams are useful in all forms of object-oriented programming (OOP). The concept is several years old but has been refined as OOP modeling paradigms have evolved.

In a class diagram, the classes are arranged in groups that share common characteristics. A class diagram resembles a [flowchart](http://whatis.techtarget.com/definition/flowchart) in which classes are portrayed as boxes, each box having three rectangles inside. The top rectangle contains the name of the class; the middle rectangle contains the [attribute](http://searchsoa.techtarget.com/definition/attribute)**s** of the class; the lower rectangle contains the methods, also called operations, of the class. Lines, which may have arrows at one or both ends, connect the boxes. These lines define the relationships, also called associations, between the classes.

Disease Name

+Id: int;

+User Name: string;

+Info: string;

+Disease();

User Informatin

-User Name: string;

-Password: string;

+Address: string;

+Email: string;

+Mobile: int;

+Registration();

+Login();

1…\*

1

1

1

\* \* 1…\*

Syndrome

+Id: int;

+Name: string;

+Info: string;

+Syndrome();

Medicine

+Id: int;

+Name: string;

+Info: string;

+Medicine();

+QuestionInfo();

+AnswerInfo();

## Entity Relationship Diagram

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

At first glance an entity relationship diagram looks very much like a [flowchart](https://www.smartdraw.com/flowchart/). It is the specialized symbols, and the meanings of those symbols, that make it unique.

Chapter 4

**Implementation**

# Implementation

A product softwareimplementation method is a systematically structured approach to effectively integrate a software based service or component into the workflow of an organizational structure or an individual end-user.

A product software implementation method is a blueprint to get users and/or organizations running with a specific software product.

The method is a set of rules and views to cope with the most common issues that occur when implementing a software product: business alignment from the organizational view and acceptance from human view.

The implementation of product software, as the final link in the deployment chain of software production, is in a financial perspective of a major issue.

It is stated that the implementation of (product) software consumes up to 1/3 of the budget of a software purchase (more than hardware and software requirements together).

when the main point of the computer study program is to implement counter measures to bots and bugs.

## Tools &Technologies

Following are the tools and technologies used in development of this project:

Microsoft office word

Laravel

MySQL database

HTML5

CSS3

BOOSTRAP

Microsoft Visio

## Project Link

Chapter 5

**System Testing**

# System Testing

SystemTesting is a level of the softwaretesting where a complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements. Definition by ISTQB. systemtesting: The process of testing an integrated system to verify that it meets specified.

## Why Software Testing is Essential?

Systemtesting of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. Systemtesting falls within the scope of black-box testing, and as such, should require no knowledge of the inner design of the code or logic.

### Black box testing

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of **test** can be applied virtually to every level of software testing: unit, integration, system and acceptance.

### White box testing

White-box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) is a method of testing software thattests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing).

## Test Cases

### Test case: Sign up

Project: Medicine Search Engine

Author: User, Admin

Date:

|  |  |
| --- | --- |
| Test case ID | TC 0012 |
| Test case Admin |  |
| Functional area | Sign up |
| Test name | Sign up with validated information |
| Objective | For opening a username , every user need to filled up the sign up form .there have no chance to keep empty any field in registration. |
| Pre-requisite | Need to registration |
| Steps to perform | 1. fill the form with proper information and validate info  2. then create account |
| Expected result | If have any empty field error message will display in the field |
| Test result | 1.registration may be success  2. registration may be unsuccessful |

Chapter 6

**Conclusion**

# Conclusion

## The system is very essential for the people.

## This will provide health services for the all kind of people.

Appendices

# User Manual

# References:

<https://www.draw.io/>

<https://www.lucidchart.com/pages/system-sequence-diagram-overview>

<https://en.wikipedia.org/wiki/Sequence_diagram>

<https://en.wikipedia.org/wiki/Class_diagram>

<https://project-proposal.casual.pm/>