|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | Daffodil International University | | | | |  |  | |
|  |  | | |  | | |  | |  | |
|  | | | | | | | | | | |
| http://4.bp.blogspot.com/-SotLIkzmscA/UqQLE9BjdPI/AAAAAAAAARc/-SF46qFfVE0/s1600/daffodil-international-university-logo.png | | | | | | | | | | |
|  | | | | | | | | | | |
| Project Name: QUESTION AND ANSWER MANAGEMENT  Course Project  Software Engineering Project-II (Web Programming)  Course Code: SWE332 | | | | | | | | | | |
|  | | NAME | | | **:** | Md. Shahidul Islam | | | |  |
| ID | | | **:** | 143-35-825 | | | |
| 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)

[2.2.1 Admin 5](#_Toc497901985)

[2.2.2 User 5](#_Toc497901985)

[2.3 Use Case Model 5](#_Toc497901986)

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

[2.4.1 Register User 5](#_Toc497901988)

[2.4.2 Make question and give answer 5](#_Toc497901988)

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

[2.5.1 User registration 6](#_Toc497901990)

[2.5.1 Make question 6](#_Toc497901990)

[2.6 System Sequence Diagrams 6](#_Toc497901991)

[2.6.1 User registration 6](#_Toc497901992)

[2.6.1.1Success scenario 6](#_Toc497901992)

2.6.1.2 Failure scenario 6

[2.6.2 Make question 6](#_Toc497901992)

[2.6.2.1Success scenario 6](#_Toc497901992)

2.6.2 .2 Failure scenario 6

[2.7 Activity diagram 7](#_Toc497901994)

[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. Testing of the System 19](#_Toc497902006)

[5.1 Why Software Testing is Necessary? 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 19](#_Toc497902011)

[Chapter 6 20](#_Toc497902012)

[6. Conclusion 21](#_Toc497902013)

[6.1 System have nice useful features 21](#_Toc497902014)

[6.2 There have some procedure for getting the question answer 21](#_Toc497902015)

[Appendices 22](#_Toc497902017)

[7. User Manual 23](#_Toc497902018)

[8. References 24](#_Toc497902020)

Chapter 1

Introduction

# Introduction

## Backgrounds the System

In this system question and answer are managed. In our practical life we have many problems we can solve our problem in in platform. This platform is make for solving this kind of problem. There may be different kind of problem, especially in IT sector we faced many problems in every day if we want we can post our problem in this site and solve our problem. I think this site is very helpful for solving any problem as early as possible. Any registered user can make their question and get answer from other, also he/she can give answer in other question.

## Purpose

Solving our problem easily and instantly

Any registered user can make question as their demand

Any registered user can answer the question how many he/she want

We can justify our skill through answering the question

People can get benefit by getting their question answer instantly

## Scope

Any registered user can make the question and also can give the answer. Every user can make question how many they want. Without registration nobody can’t make any question and can’t make any answer. Without registration normal user only can view the site.

## Vision

People can easily solve their problem through internet.

## Why this system is necessary?

For solving our problem in our practical life. In everyday life we face many problems especially as a IT person we face many problems in everyday, so we can solve our problem early through this site

# 

Chapter 2

**System Analysis**

# System Analysis

## Use Case Model

QUESTION AND ANSWER MANAGEMENT

Internal User External User

## Actor Goal List

### Admin

Manage the user

Moderator Manage the question

Rating the user by default from system

Ranking the user from system

### User

Can make profile

Can edit the profile

Can make the question

Can give the answer

## Use Case Model

## Use Case Description (Briefly)

### Register User System

User

User need to make registration, because without no user can make the question also can’t answer any question.

### Make Question & Give Answer

Internal user

Every user can make the question and also can give the answer.

## Use Case Description (Detailed)

### User registration

|  |  |  |
| --- | --- | --- |
| Use Case ID | 1 | |
| Name | Register users | |
| Primary Actor | Admin | |
| Secondary Actor | users | |
| Goal | Registration to the system | |
| Precondition | Every user can get a username from database after registering.  E.g.Usrname , E-MAIL,Password,Confirm password | |
| Post Condition | User can get a username and password | |
| Main Success Scenario | Actor | System |
| 1. Admin handle the registration process.  2. only admin can handle the user data from database.  3. After registering every user can get a username and password for make question and give answer.  6. Register user data is saved in the database for handle the site smartly if there will face any trouble admin can handle it easily from database . | 2.1 There will be registration from for user in the system  2.2Admin of the system handle the database  3.1 There will be a specific roll for registering  3.2There will be a specific way of registering  6.1 submit data to server and return  Success message |
|  |  | |

### Make Question

|  |  |  |
| --- | --- | --- |
| Use Case ID | 2 | |
| Name | Make Question | |
| Primary Actor | User of the system | |
| Secondary Actor | Other user | |
| Goal | Make question with specific topic | |
| Precondition | Make question with Questiontitle, Category**,** Tags**,** QuestionDetails | |
| Post Condition | Make question specifically | |
| Main Success Scenario | Actor | System |
| 1 Registered user can make answer to question  2 There have a search option for question  3 without registration nobody can make any question  4 There have an option called post question  5 after posting the question another user can see it | 1.1admin will approve the question  1.1.1 without approval no question will not publish    4.1 every user can post more question how many they want . |
|  |  | |

## System Sequence Diagrams

### User registration

**2.6.1.1 Success scenario**

System

User

Fill the registartion form

Provide a usernname and password

user logn in

verify the log in

Fig: System sequence of user registration

**2.6.1.2 Failure Scenario**

System

User

Fill the registartion form

Provide a usernname and password

user logn in

verify the log in

Fig: System sequence of user registration

**2.6.2 Make Question**

**2.6.2.1 Success scenario**

System

User

**Make question**

**Show question in the system**

**Make answer**

**Provide rating**

Fig:System sequence of make question

**2.6.2.2 Failure scenario**

System

User

**Make question**

**Show question in the system**

**Make answer**

**Provide rating**

Fig:System sequence of make question

## Activity diagram

Activity diagrams are graphical representations of [workflows](https://en.wikipedia.org/wiki/Workflow) of stepwise activities and actionswith support for choice, iteration and concurrency. In the [Unified Modeling Language](https://en.wikipedia.org/wiki/Unified_Modeling_Language), activity diagrams are intended to model both computational and organizational processes .

Arrows run from the start towards the end and represent the order in which activities happen.

Activity diagrams may be regarded as a form of [flowchart](https://en.wikipedia.org/wiki/Flowchart)**.** Typical flowchart techniques lack constructs for expressing concurrency. However, the join and split symbols in activity diagrams only resolve this for simple cases; the meaning of the model is not clear when they are arbitrarily combined with decisions or loops.

The basic purposes of activity diagrams are similar to other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.

It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not. It shows different flows such as parallel, branched, concurrent, and single.

.

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.

Search Question

Make question

Answer question

User register

Rating answer

Manage profile

User (Registered)

Edit profile

View question and answer list

See Reputation

## 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.

Admin

-Id:int;

-UserName:string;

-Password:string;

+Address:string;

+Email:string;

+Mobile:int;

+AuthenticationUser();

+ControlSystem();

User

+Id:int;

+UserName:string;

-Password:string;

+Address:string;

+Email:string;

+Mobile:int;

+UserRegistration();

+UserInfo ();

1 1…\*

1 1…\*

\* 1…\*

QuestionAnswer

+QuestionId:int;

+QuestionType:int;

+QuestionTag:int;

+QuestionDescription:int;

+QuestionInfo();

+AnswerInfo();

Guest User

+Id:int;

+Address:string;

+Email:string;

+Mobile:int;

+UserType();

## 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

PHP

MySQL database

HTML5

CSS3

BOOSTRAP

JAVASCRIPT

Microsoft Visio

## Project Link

Chapter 5

**Testing of the System**

# 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 Necessary?

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: Project title

Author:

Date:

|  |  |
| --- | --- |
| Test case ID | TC 001 |
| Test case manager |  |
| 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

## System have nice useful features

## There has some procedure for getting the question answer

Appendices

# User Manual

# References