**Coding Academy By Orange**

Text

Description automatically generated

**Osaid Ghnaimat**

**Coding academy by orange**

# CHAPTER ONE

# INTRODUCTION

|  |  |
| --- | --- |
| **1.1** | **Background** |
| **1.2** | **Objective** |
| **1.3** | **Overview of System** |
| **1.4** | **System Features** |
| **1.5** | **Textual Representation** |
| **1.6** | **Water Fall Model** |
| **1.7** | **System Requirements** |

* 1. **Background**

Nowadays, the internet has become one of the main requirements every person needs and that is why online platforms or communication sites have become so important to present a person, his accomplishments, works and even interests.

The objective of this project is to develop a website that links experts and specialists in various fields so that the expert can present his experiences and areas of specialization, and the user can easily book an appointment for a consultation in this field.

* 1. **Objective**

In the information age, people do not have the time and money to make decisions without prior experience. The system helps in arranging the meeting of the people and experiences they will direct The factors to consider in design are:

- The system allows new users to register.

- The system contains many experts.

- The system contains a follow-up for users.

- User-friendly and easy-to-use user interface system.

- The system allows the expert to add consultations easily.

- The system allows the student to view the comments and opinions of users

-The system allows students to book in many consultations

**1.3 System Overview**

**The solution is compatible with achieving the following use cases**

**Register now.**

**actore user representative.**

**The user enters and saves the main information.**

**The user sees the experiences of each expert.**

**User can view categories.**

**The user communicates with all experts.**

* 1. **System feature**

There are some things that should be checked through the system, such as registering for the purpose of adding follow-ups to Users.

And when the user exits the system, his information disappears from follow-ups, conversations and questions, so the student can protect his information or the list of likes that he must display at a later time.

We'll assume the system on which this program will run is based on Windows 10/11. The latest version is installed and ready to use. Xampp is installed and ready to use. This is required for the database; we assume that end users are able to use experienced and physical computer users trained to use applications and Internet servers if necessary. The interface is designed as a web page, so we will assume that the user has used a similar interface before and will not need basic help. We will also assume that the user will only need help on how to perform the main function of the system.

Allows multiple computers to access the database from different locations. Two approaches can be used.

Use php my admin and grant access and privileges to all the IP addresses you need. Use an SQL database file, and share it across the network system users are using. This should be password protected to prevent unwanted users.

Each of these solutions requires a system setup for all computers. The startup file that starts the server and opens the Internet Explorer window will be on the login page on each system desktop. Also, one computer system will be required to host Xampp, this must be done at all times.

* 1. **Textual Representation**

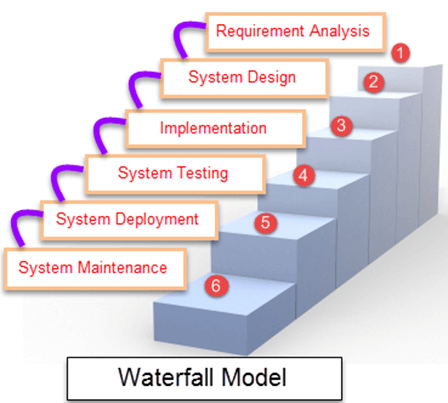
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Activity** | **Duration** | **Stat Date** | **End Date** |
| 1 | Analysis | (20)Days | 05/11/22 | 25/11/22 |
| 2 | Design | (30)Days | 26/11/22 | 26/12/22 |
| 3 | Implementation | (60)Days | 1/1/22 | 1/3/22 |
| 4 | Testing | (10)Days | 2/3/22 | 12/3 |

**Table (1.1)**

* 1. **Waterfall Model**

We followed Water-fall model to build our system.

A process model in which the software engineer works in a series of stages.



**1.6.1 Waterfall Principles**

Software engineers should work in a series of stages

Before completing each stage.

They should perform quality assurance so that the next stage can be built on a good foundation Software engineers sometimes have to step back to earlier stages when the problem in a subsequent stage.

**1.6.2 Waterfall limitations**

It has an overly rigid viewpoint that you should attempt to complete the entire specification before moving on to the design, and the entire design before moving on to implementation.

It makes no allowances for prototyping.

It implies that you can get the requirements right by simply writing them down and reviewing them. Something that only. Works for simple, Well-understood types of software development

It implies that once the product is finished, everything else is Maintenance.

* 1. **Technologies that used to bult this project are:**

PHP (Laravel: framework)

Java script

Html

Css

Php my admin (to handel the database tabels)

Mero (to make a wireframe)

# CHAPTER Two

# SYSTEM ANALYSIS AND DESIGN

|  |  |
| --- | --- |
| **2.1** | **Gathering Requirements** |
| **2.2** | **System Problem Statement** |
| **2.3** | **User Requirements** |
| **2.4** | **Functional, Non-Functional requirements** |
| **2.5** | **UML Diagram** |

**Requirements**

The project requirement phase is one of the most important parts of the project and especially. Working on a software project, you should use Software Engineering techniques in this phase. Requirements are prepared by communicating with your customers and suppliers which is in general your advisor

**Introduction**

Requirement analysis is critical to the success of a systems or software project, the requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design.

**2.1 Gathering Requirements:**

Techniques involving visualization of the requirements like storyboards, prototypes, scenarios are helpful when you have a business customer who may not be worried about the ins and outs of a technical solution or have long attention duration for legalizing the Requirements with users to let the analyst drive his discovery efficiently than just reading a document with a prospective user.

**2.2 System Problem Statement**

The area of the construction environment is involved with the handling of Customer sensitive data. The security and privacy of these data are very important. While doing online transfer of these secret data over the public network, it can be viewed and/or modified by the attackers. It can be also accessed by unauthorized persons who can break the privacy of the Customer's data.

**2.3 User Requirements:**

It is a document usually used in software engineering that specifies the requirements the user expects from software to be constructed in a software project. The important and difficult step in designing a software product is to determine what the customer wants. This is because the customer is often notable to reach the appropriate Tool/Service, and information about the Suppliers may be incomplete or unfavorable to him. Where the owner and customer have to register in the program to perform several operations, including a reservation.

**2.4 Functional, Non-Functional requirements:**

2.4.1 Functional requirements:

In Software engineering, a Functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behavior, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish.

|  |  |
| --- | --- |
| Stakeholder | Role |
| user | * Login * Logout * Search Consultation * Create Account * Book a courses |
| expert | * Login * Logout * Create Account * add a Consultation * delete a Consultation |

**Table (2.1): Functional requirements**

2.4.2 Non Functional requirements

In Software engineering, a non-Functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. This should be contrasted with functional requirements that define specific behavior or functions.

1. Security:

My web application is safe to use because we decided to link it to database SQL Server with three layers Models, DataAccessLayer, and DataBusinessLayers, which gives the web application the authority to authenticate each user's data individually.

2. Ease of use:

My web application is easy to use as we have facilitated building the user interface and data flow in it and entering information accurately and easily, and we worked on the web application in two versions for a student as a data entry for his information and the second for the student where they can check the procedures for communicating with each other.

3. Availability:

In the future, I aspire to have My web application to work on all different work environments, from smartphones and websites to facilitate the process of follow-up procedures between the parties. The web application is available to the user anytime, anywhere, making the process of collaborating between the owner and the customer easier.

4. Reliability

Describes the degree to which the system must work for users. Specifications for reliability typically refer to availability, mean time between failures, means time to repair, accuracy, and maximum acceptable bugs. For

\*The system meets the terms of a Service Level Agreement.

\*The mean time to failure will be at least four months.

5. Portability:

The system is based on the PHP language software adaptation and works on the vs code platform consists of different hardware and operating system.

6. Performance:

Specifications typically refer to response time, transaction throughput, and capacity.

\* All Web pages must download within three seconds during an average load, and five seconds during a peak load.

7. Supportability

This indicates the system’s ability to easily adjust or maintains to accommodate technical use or scenario change. For example, in our system. How easy was it to add a new appointment for the support framework.  
\*The system allows users to create new workflows, without the need for additional programming.

# CHAPTER four

**TESTING**

**4.1 Key Concepts**

**4.2 Strategic Approaches to Software Testing**

**4.3 Verification and Validation**

**4.4 Unit Testing Procedures**

**4.1 Key Concepts**

A strategy for software testing integrates test case design methods into a well-planned series of steps that result in the successful construction of software the strategy provides a road map that describes the steps to conduct as part of testing when these steps are planned and then undertaken and how much effort time and resources will be required.

Therefore testing strategy must incorporate test planning test case design test execution and result data collection and evaluation.

Software testing should be flexible enough to promote a customized testing approach. At the same time, it must be rigid enough to promote reasonable planning and management tracking as the project progresses.

**4.2 A Strategic Approach to Software Testing**

Testing is an asset of activities that can be planned in advance and conducted systematically. A number of software testing strategies 'Which provide the software developer with a template for testing and all have the following general characteristics:

1- To perform effective testing software team should conduct effective formal technical reviews.

2-Testing begins at the component level and works outward the integration of the entire computer-based system.

3- Different testing techniques are appropriate at different points in time.

4- Testing is conducted by the developer of the software and (for large projects) an independent test group.

5. Testing and debugging are different activities, but debugging in any testing strategy.

**4.3 Verification and Validation**

Software testing is one element of a broader topic that is often referred to as verification and validation (V&V). Verification refers to the set of activities that ensure that software correctly implements a specific function. Validation refers to a different set of activities that ensure that the software that has been built is traceable to customer requirements.

Test strategies for Conventional software

1 -Unit testing

Unit testing focuses verification effort on the smallest unit of software design —the software component or module. Using the component-level design description as a guide, important control paths are tested to uncover errors within the boundary of the module.

Complexity of the test and the errors those tests uncover is limited by the constrained scope established for unit testing -the unit test focuses on the internal processing logic and data structures within the boundaries of a component. This type of testing can be conducted in parallel for multiple components

**4.4 Unit Test Procedures**

Unit testing is normally considered as an adjunct to the coding step. The design of unit tests can be performed before coding begins or after source code has been generated. Each test case should be coupled with a set of expected results a component is not a standalone program driver and/or stub software must be developed for each unit test. In most applications, a driver is nothing more than a "main program" that accepts test case data passes such data to the component (to be tested), and prints relevant results.

Stubs serve to replace modules that are subordinate to (called by) the component to

tested la stubs to replace modules that are subordinate to (called by) the component to be tested. A stub or dummy subprogram" uses the subordinate module's interface, may do minimal data manipulation, provides verification of entry and returns control to the module undergoing testing. Drivers and stubs represent overhead. That is, both are software that must be written but are not delivered '.vith the final software product.

Unit testing is simplified when a component with high cohesion is designed. when only one function is addressed by a component, the number of test cases is reduced and errors can be more easily predicted and uncovered.

# CHAPTER FIVE

**CONCLUSION**

**AND**

**FUTURE WORK**

**5.1 Conclusion**

**5.2 Related and Future work**

**5.1 Conclusion**

The first project was designed to meet the requirements of users for communication between users. It was developed in ASP. It complies with the system specifications.

For system design we used a simple sequence diagram. )

This site will assist users in the communication process and benefit from the experiences.

**5.2 Related and future**

**Developing this system to become global**

**• Make this website with a variety of languages**

**• Developing the site to work in any new program**

**References:**

* https://laravel.com/docs/9.x
* https://www.w3schools.com/js/
* https://www.w3schools.com/css/
* https://developer.mozilla.org/en-US/docs/Web/HTML
* https://developer.mozilla.org/en-US/docs/Web/HTML
* https://www.php.net/