SOFTWARE REQUIREMENTS
SPECIFICATION

Enclosed Application Agreement for Product Selling Website and Web Application System

Name: MD ABDUR RAHMAN ROKY
ID: 221-15-4972
SEC: 61_V
Department of CSE

Young IT

Enclosed Application Agreement for Product Selling Website and Web Application System

Contents

KUDROT-E-KHUDA
Managing Director GHURI
Shah Ali Plazza,10th Floor.
112-113, Metro-Rail pillar side, Mirpur 10
Dhaka -1100

Dear KUDROT-E-KHUDA,

Re: Enclosed Application Agreement for Product Selling Website and Web Application System.

Young IT knows that developing a client-focused application requires both technical prowess and effective communication, therefore we only recruit the best to make sure you get both. Knowing that every customer is different, we work hard to always present a distinctive, affordable, and original proposal. We then follow it up with a superb delivery that is both on time and within budget.

We create a product selling website and web application system for you in accordance with your business requirements. In this agreement paper, we detail all characteristics that apply to our program. Please take the time to read it carefully, and if it fulfills your needs, sign to affirm your agreement.

Yours Sincerely
Delayer Hosain
Head
Brach for Application Development
Young IT Enclosed

1.Project Overview

Take use of technology and digitalization efforts in the twenty-first century. The technology we use every day has advanced. We create a method to simplify our business or professional lives. A website and a web application make up this project; the website is entirely information-based, and the web application has two modules with iCloud functionality.

2.Obstacles

The website will provide options for registering as a user, looking through courses, interactive lessons, tests, user profiles, and a discussion forum.

2.1 User Classes and Characteristics

- Low Budget
- Platform
- Manage Storage File System

3.User Features in Details:

Product Selling Website and Web Application System Has two parts, they are:

3.1 Website

- Home
- About Us
- Properties
- On Click a Product View Project Details
- Include Picture Slider

3.2 User Registration and Authentication

- Users must be able to create an own username and password for registration.
- Users can sign in with their login information.
- Encryption should be used to store passwords safely.

3.3 Google Place

- Contact
- Login (Go for the Web Application)
- Web Application System
- User
- Login

3.4 User Id and Password are provided by admin

- Payment Status
- Payment History
- Download Receipt

- Create User
- Submit User Payment
- See User Payment Status
- Payment History
- Generated Receipt

4.Technology Require (Application and Hardware)

There are some of technology & Hardware required, these are:

- Framework: Slim.
- Database: MySQL.
- Design: Standard.
- Coding Architecture: HTML.
- Security: Standard.

5.Application

Product Selling Website and Web Application System

6.Milestones and Reporting

Milestone	Tasks	Reporting	Time
Analysis		Submit The Design	5 days
Requirements Collection	Submit To Us All Data		10 days
Development		Review The Work	40 days
Testing			15 days
Deployment	Must Ready the Server	Review Final Work	7 days
Delivery		Live On Server	7 days

7.Deployment

The Application will completely base on the following requirement which is given by your company and this Application covers those features which is right down. For development time developers only focus on the feature. If client want more feature, then He must pay based on new feature.

8.Testing

The testing process shall be as follows:
- Application will be tested by PHP Unit.
- Application also tested By Codetection.

9.Support

- 24/7 support based on payment.

10.Pricing

Our fee for seeing the project through from start to completion will be Forty Thousand Taka only (40000Tk).

11.Payment Terms

We propose the following payment terms:

10% (10%) Paid on acceptance of this proposal.

40% (50%)
Paid on signing of our application development agreement.
25% (75%)
Paid at 70% Application Demonstration.

25% (100%)
Paid at completion the Application.

12.Software Development Life Cycle (SDLC)

Based on the requirements we would like to choose "Iterative Model" in order to make the website. Because we know that "Iterative Model" has a lot of advantage. Such as:

1.   Some working functionality can be developed quickly and early in the life cycle.

2.   Results are obtained early and periodically.

3.   Parallel development can be planned.

4.   Progress can be measured.

5.   Less costly to change the requirements.

6.   Testing and debugging during smaller iteration is easy.

7.   With every increment, operational product is delivered.

8.   Issues, challenges and risks identified from each increment can be applied to the next increment.

9.   Risk analysis is better.

10.   It supports changing requirements.

11.   Initial Operating time is less.

12.   During the life cycle, software is produced early which facilitates customer evaluation and feedback.

After taking all these advantage into consideration we think it would be better to use "Iterative Model" rather than using any other model in order to make your website and web application system.

12.1 Models of SDLC

Software Development Life Cycle (SDLC) is a process used by the software industry to design, develop and test high quality software.

Some models of SDLC:

A. Waterfall Model: The Waterfall Model was the first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

Waterfall Model
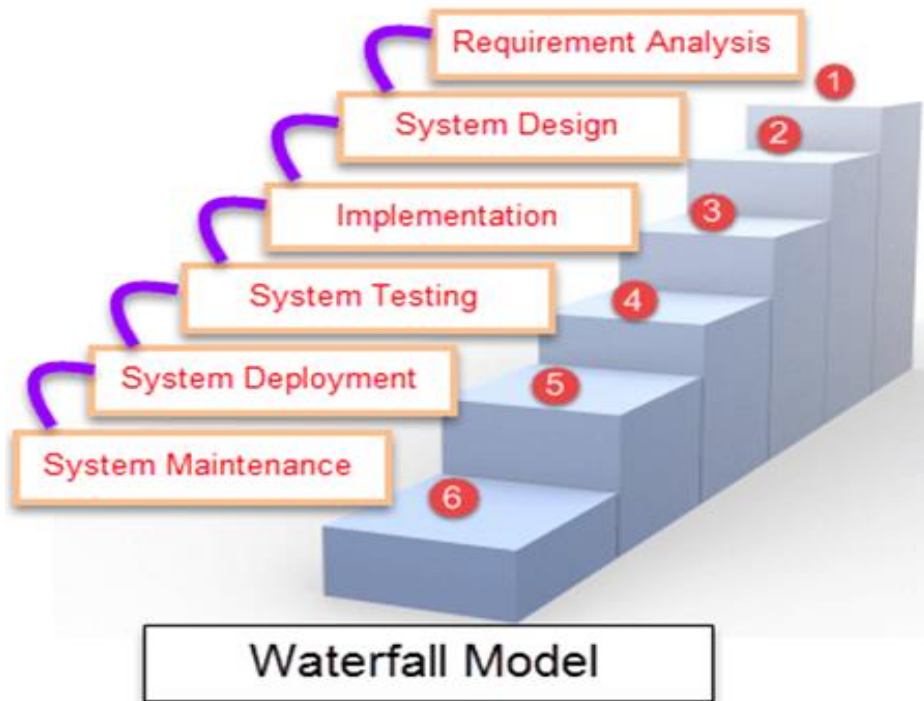
B. Iterative model: In the Iterative model, iterative process starts with a simple implementation of a small set of the software requirements and iteratively enhances the evolving versions until the complete system is implemented and ready to be deployed. An iterative life cycle model does not attempt to start with a full specification of requirements.
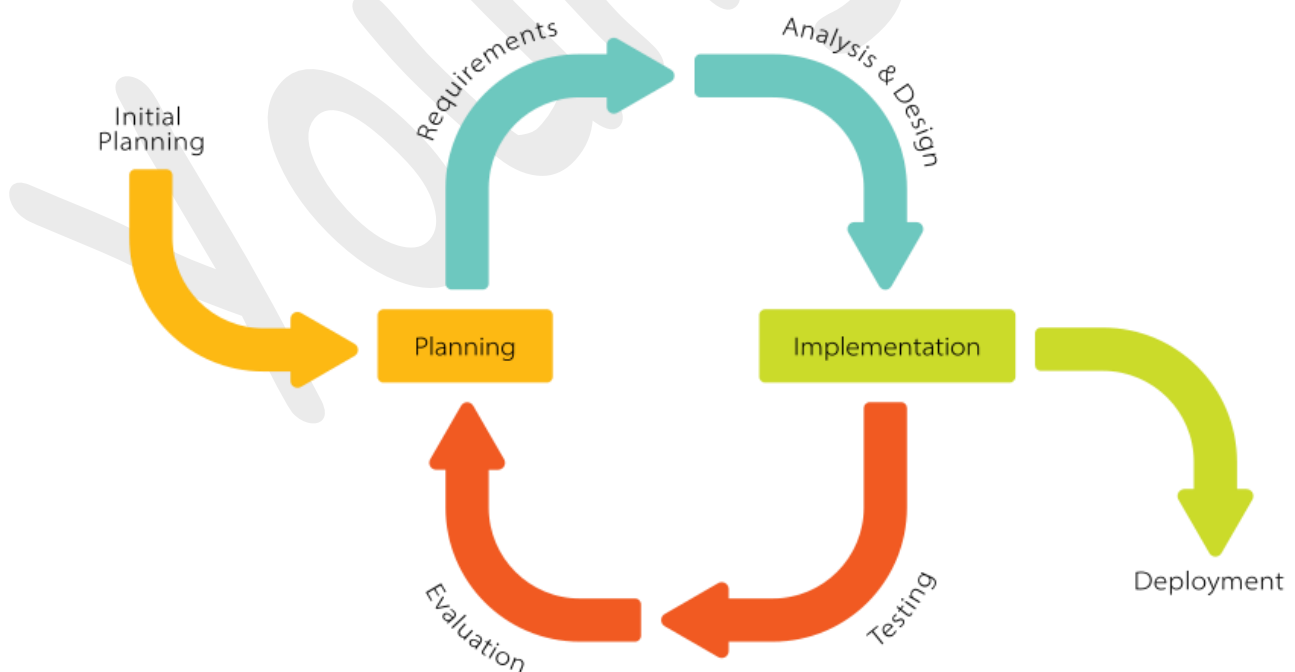
C. Spiral model: The spiral model combines the idea of iterative development with the systematic, controlled aspects of the waterfall model. This Spiral model is a combination of iterative development process model and sequential linear development model.
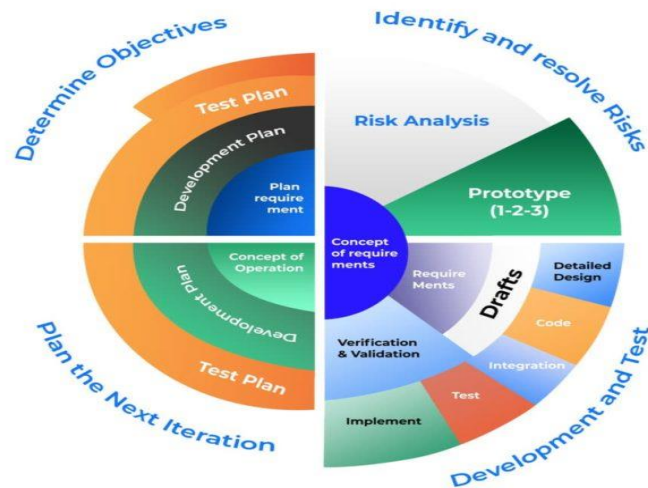

D. V-model: The V-model is an SDLC model where execution of processes happens in a sequential manner in a V-shape. It is also known as Verification and Validation model. This is a highly-disciplined model and the next phase starts only after completion of the previous phase.
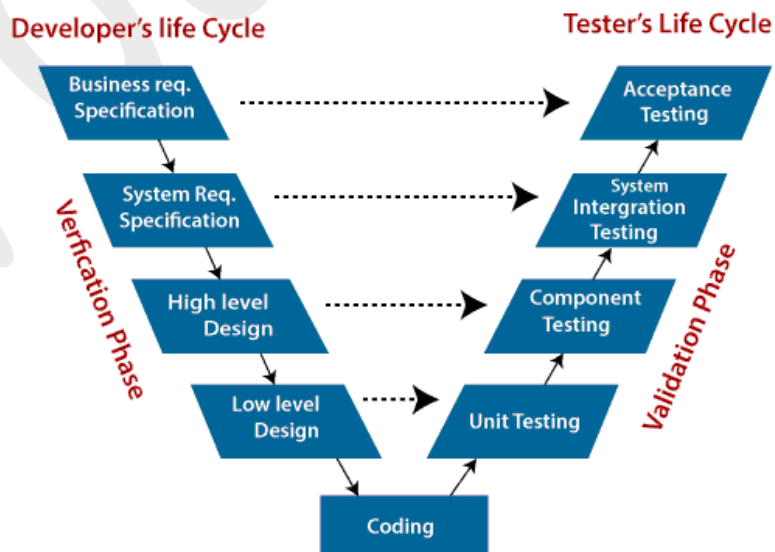
E. Big Bang Model: The Big Bang model is an SDLC model that does not adhere to any particular procedure. The necessary funds and labor are simply needed to begin the development process, and the finished product is software that may or may not meet client requirements.

Big Bang SDLC MODEL
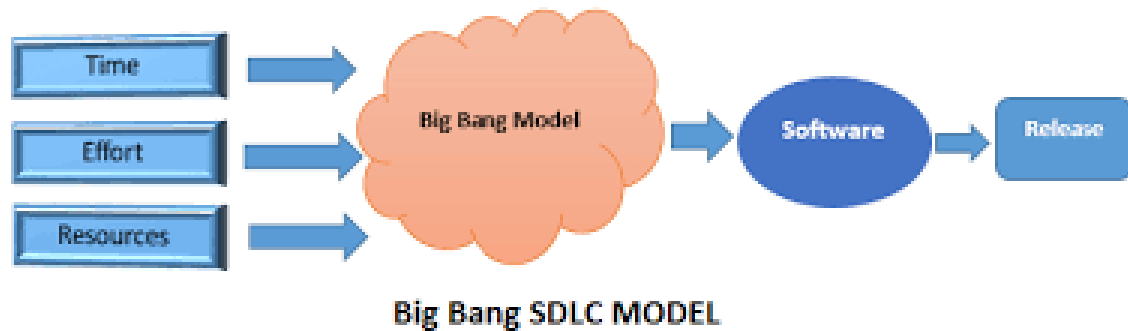
F. Agile Model: The agile software development life cycle (SDLC) paradigm combines incremental and iterative process models, emphasizing process flexibility and customer satisfaction through quick delivery of functional software. Using agile methods, the product is built in tiny, incremental increments. Iterations of these builds are supplied. Usually, an iteration lasts between one and three weeks.
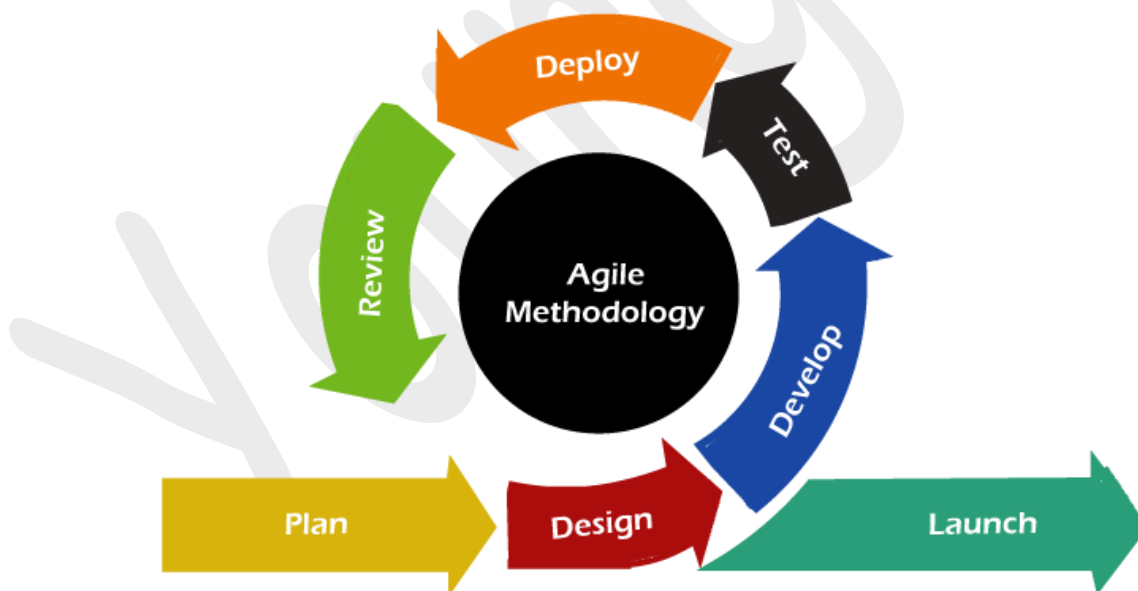
G. RAD Model: The RAD (Rapid Application Development) model is based on prototyping and iterative development with no specific planning involved. The process of writing the software itself involves the planning required for developing the product.

RAD Model Diagram

H. Software Phototype Model: This model describes the process of creating software application prototypes that show off the features of the product that is being developed, but they might not follow the exact logic of the original program.
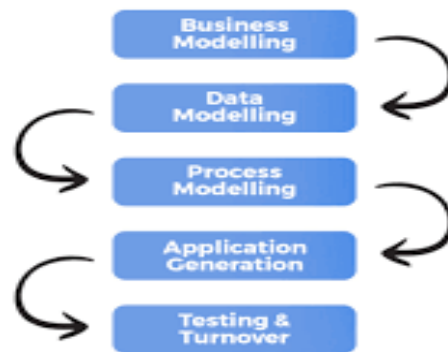
PROTO TYPE MODEL
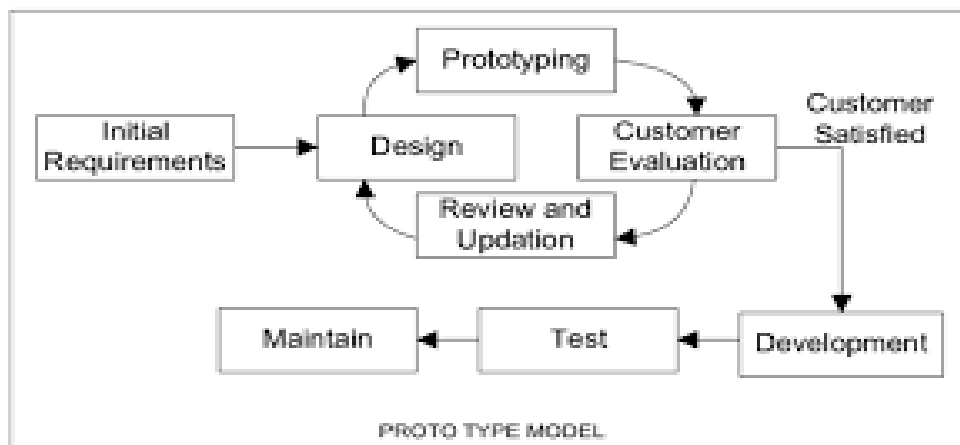
The needs of the project, the experience level of the team, and the specific environment all play a role in the choice of Software Development Life Cycle (SDLC) model. While each of the eight popular SDLC models has advantages and disadvantages of its own, some are generally seen as being easier to use and more suited for specific project kinds.

13.Use Case

«Subsystem»
Online Shopping

View Items

Registered Customer

Web Customer

Make Purchase

«include»

«include»

Checkout

New Customer

Client Register

«Service»
Authentication

Identity Provider

Credit Payment Service

PayPal
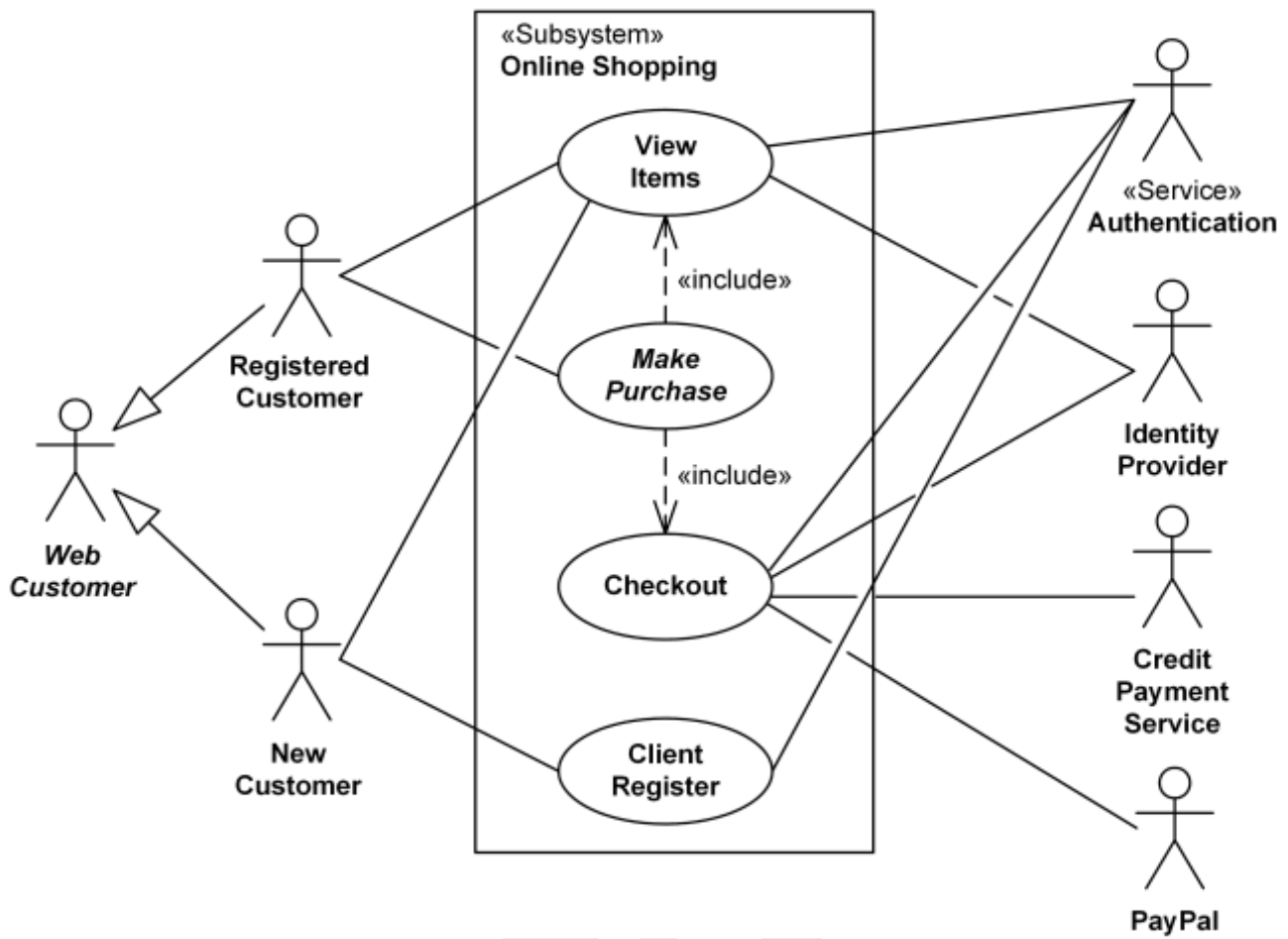
14.UML Diagram
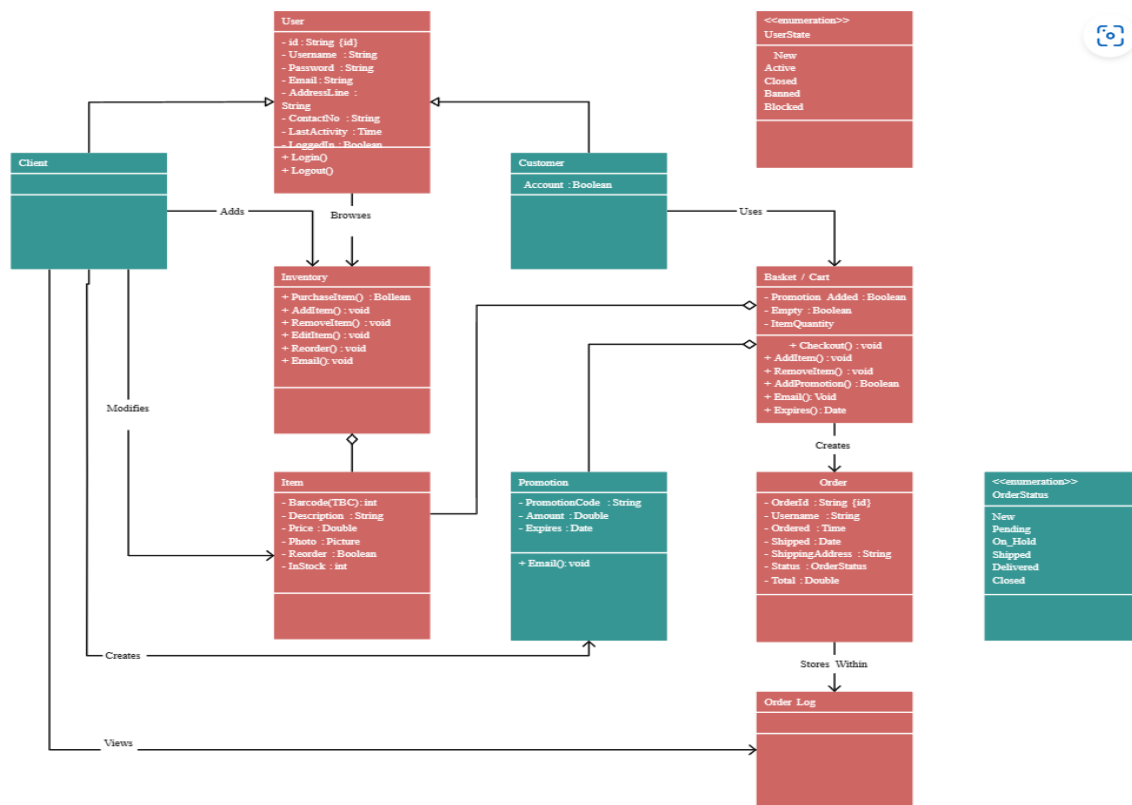
15.Testing Approach

The project's implementation of the test strategy is the test approach, which outlines the testing technique.

That testing is –

 • Static

• Dynamic

• Passive

1.Static testing: It entails inspections, walkthroughs, and reviews. Static testing, like proofreading, is often implicit and happens when programming tools, text editors, or compilers (pre-compilers) look over a program's grammar and data flow.

2.Dynamic testing: It involves running programmed code with a predetermined set of test cases.

3.Passive testing: Passive testing is the process of confirming system behavior without having to interact with the software product. Rather than supplying any test data, testers look over system logs and traces as opposed to actively testing.

15.1 Testing Levels

Unit testing: The first layer of testing that is integrated into the codebase is unit testing. It comprises features that are tested before the app is constructed, as well as many particular processes and modules.

Functional testing: Functional testing involves front-end manual user interface testing of features and components. Usually, at this stage, all of the information is false.

System/UI testing: Developers can now work closely with the QA team to conduct automated testing and analysis of a test version of the user interface that has successfully integrated with the back-end service.

15.2 Testing Tools

Test cases are executed using two different approaches:

- Automated
- Manual

15.3 The testing approach types on the basis of our strategies

When evaluating a website design firm, we want to focus on a few distinct aspects of our work to ensure that the user and client meet our quality and performance criteria. We consider the following testing and assessment techniques:

Installation testing: An excerpt from installation testing is included in this section. The majority of software systems require installation in order to be utilized for the intended purpose. Installation testing is the process of testing these steps to produce an installed and functioning software system. These procedures could include install and uninstall procedures in addition to full and partial updates.

Compatibility testing: Incompatibility with other application software, operating systems (or operating system versions, old or new), or target environments that differ significantly from the original software are frequently the root causes of software failure (real or perceived).

Acceptance testing: A build acceptance test, similar to a smoke test, comes before extra testing, such as regression or integration testing.

One of two terms can be used to describe acceptance testing:

- After a build acceptance test, such as a smoke test, preliminary testing, such as regression or integration testing, is conducted.
- User acceptability testing (UAT) is acceptance testing carried out by the client, typically on their own hardware in a lab setting.
- The hand-off approach might include acceptance testing at either of the two stages of the development process.

Alpha testing: Other test teams or potential users/customers conduct simulated or real-world operational testing at the developer's location. Before beta testing, commercial software is often subjected to alpha testing, a type of internal acceptability testing.

Beta testing: Beta testing is a type of external user acceptance testing that comes after alpha testing. Beta testers are members of the development team who are given access to software versions that are made available to a select group of people.

Functional testing: Verifying a specific code action or function is known as functional testing. Although use cases or user stories serve as a basis for the work of other development methodologies, these are usually found in the coding requirements definition.

Non- Functional testing: It's a term used to describe software characteristics like security, behavior under specific constraints, scalability, or other performance that might not be connected to a particular feature or user action.

Security testing: Security testing is necessary for software that manages sensitive data in order to keep hackers out of the system.

This kind of testing is "conducted to evaluate the degree to which a test item, and associated data and information, are protected so that authorized persons or systems are not denied access to them and unauthorized persons or systems cannot use, read, or modify them," according to the International Organization for Standardization (ISO).

Development testing: To lower the risks, expenses, and time related to software development, a range of defect prevention and detection techniques are employed simultaneously throughout the development testing process. The software development lifecycle's building phase is finished by the software developer or engineer. Before submitting the code to additional testing, development testing checks for construction errors. This tactic aims to improve both the overall efficacy of the development process and the quality of the software produced.

Development testing may include traceability, unit testing, code coverage analysis, data flow analysis, metrics analysis, peer code reviews, static code analysis, and other software testing depending on the organization's software development goals.

We think that in order to evaluate the overall quality and performance of a website design, the other components that have been stated are more extensive and call for a combination of testing approaches and evaluation techniques. This is predicated on giving careful thought to these crucial elements that are crucial to the testing methodology for web applications and website design firms. Development testing is compliant with our requirements. Before launching or revamping your website, a thorough test of the design should be conducted. This makes it feasible to locate the broken parts and replace them as soon as possible with the least amount of disruption.

This is particularly true for legal practices hoping to gain a large clientele through advertising campaigns, press attention, and other means. Performance hazards are mitigated by your web server's ability to handle high traffic volumes.

Even while single-user testing might not always be able to spot these risks, you can avoid any negative effects on your website's functionality by understanding how to put up the proper testing procedures.

Performance testing: Usually, the goal is to ascertain how responsive and stable a system or subsystem is to a specific workload. It can also be used to look into, quantify, confirm, or verify other aspects of the system quality, like resource utilization, scalability, and dependability.

15.4 Testing techniques & Tactics

For a website design company, you can approach website testing in the following ways:

- Usability Assessment: Test the usability of the website by having actual users interact with it to assess its overall navigability, user-friendliness, and friendliness. Get input on usability problems and potential enhancements.

- Tests for Multiple Browsers and Devices: Cross-Browser Testing: Make sure the website functions properly across a range of web browsers by using testing tools and services.

- Testing for functionality and content: Test the responsive design of the website to make sure it works properly across a range of platforms (desktop, tablets, smartphones).

- Functional testing: Verify that each feature and function operates as intended by testing them separately.

- Content testing: Ensure that all information is correct, current, and error-free.

- Security Scanning: Perform assessments and scans to find vulnerabilities, such as penetration

- Testing for Compatibility: Confirm that the website works with various platforms, operating systems, and third-party integrations.

- Load testing: Use programs like Apache Meter or equivalent to evaluate how well the website functions when there is a lot of user traffic.

- Speed testing: Check loading times and optimize for quicker loads by using performance monitoring tools.

- Testing for accessibility: To make sure a website is usable by individuals with disabilities, test for accessibility using the WCAG guidelines.

- SEO Testing: Check for on-page SEO components like headers, meta tags, and content optimization.

- Client Satisfaction and Portfolio Evaluation: Get input from past customers about how satisfied they were with the work done by the company.

Examine the company's portfolio to judge the caliber and range of previously accomplished work.

By combining these testing techniques and approaches, you can thoroughly assess a website design company's work, ensuring that the websites they create are of high quality, meet user expectations, and adhere to best practices in terms of usability, security, performance, and contractual obligations.

15.5 Testing process

The testing procedures include:

- An example testing cycle;
- An agile or XP development model;
- The conventional waterfall development model.

We use the Agile development model, or XP. Agile development is important because it ensures that development teams complete projects more quickly and more cheaply than their rivals. Another advantage is that it will improve communication between the development team and the product owner. Moreover, the Agile development process may help reduce the risks associated with complex projects.

15.6 Measurement in software testing

Examples of quality measures include accuracy, comprehensiveness, security, and ISO/IEC 9126 standards such as capability, dependability, efficiency, portability, maintainability, compatibility, and usability.

Several widely used software metrics, or measures, are available to aid in evaluating the software's current condition or the efficacy of the testing.

15.7 Testing difficulty hierarchy

A testing difficulty hierarchy has been proposed, based on the number of test cases needed to build an exhaustive test suite in each context; that is, a test suite such that, when applied to the implementation under test, we gather enough data to precisely determine whether the system is correct or incorrect in accordance with some specification. The following testability classes are included in it:

Class I: A limited number of comprehensive test suites are known to exist.

Class II: Any partial distinguishing rate or incomplete capacity to differentiate between correct and faulty systems can be achieved using a finite test suite.

Class III: There is a countable complete test suite at hand.

Class IV: There is a complete test suite at your disposal.

Class V: every situation.

15.8 Test Plan

A test plan is a written document that outlines the strategy to be used for the planned test operations. Among other things, the plan might address goals, scope, personnel needs, policies and procedures, and backup plans.

The test plan can be provided as a master test plan that offers an overview of several detailed test plans (a plan of a plan) or it can be provided as a single plan that encompasses all test types (such as an acceptance or system test plan) and planning considerations. Sometimes a more detailed "test strategy" that outlines broad testing techniques includes a test plan. This might be a master test plan or an alternate artifact.

15.9 Test Case

An identifier, references to requirements from a design specification, preconditions, events, a set of procedures (also referred to as actions) to be carried out, input, output, the anticipated result, and the actual outcome are often included in a test case.

A test case is defined clinically as an input plus an anticipated outcome. In addition to the test case ID, there are optional fields for test step or sequence of execution, associated requirement(s), depth, test category, author, and check boxes indicating whether or not the test is automated. Larger test cases may include contain explanations and essential states or procedures.

The actual outcome should also be included in a test case. Word processing documents, spreadsheets, databases, and other common repositories can be used to store these stages. A database system may allow you to see past test results as well as details about who made them, what system configuration they were created with, and other relevant information. Usually, these previous results would be stored in a separate table.

16.Language used for this Project

Front End & Back End Interface	HTML5, CSS, JAVASCRIPT
Database	PHP, MySQL
Framework	Bootstrap 5, Laravel 6.0
Development Platform	Sublime Text
Server	Xampp

17.Scheduling & Estimate

Analysis		Submit The Design	5 days
Requirements Collection	Submit To Us All Data		10 days
Development		Review The Work	40 days
Testing			15 days
Deployment	Must Ready the Server	Review Final Work	7 days
Delivery		Live On Server	7 days

18.Responsibility:

This Application Ordered by KUDROT-E-KHUDA, Managing Director of GHURI, all responsibility goes on him.

This Application will Developed by Sahinur Rahman and Dilruba Safa on a project of Young IT Development Group supervised by Delayer Hosain.

19.Contact Us

You can get in touch with us in any of the below ways:

By Phone:
+8801706999666
Md Abdur Rahman Roky

By Email
rokyvai01r@gmail.com

On our website www.roky's.us

By post
Room-113, 4 No, DSC,
Daffodil Smart City,
Birulia, Savar, Dhaka-1206

We look forward to hearing from you soon!

20.Agreement Signed By:

………………………….   …………………………..   ……………………………

Client Signature   Order Provider Signature   Authority Signature
KUDROT-E-KHUDA   Officer   MD ABDUR RAHMAN
Managing Director   Young IT   Managing Director
GHURI

Young IT