**4. Testing Report**

**4.1Methodologies used for Testing**

Software testing is an integral part of the software development life cycle (SDLC).

Testing a piece of code effectively and efficiently is equally important, if not more, to writing it. So what is software testing? Well, for those who are new to software testing and quality assurance, here are few useful facts. Software testing is nothing but subjecting a piece of code to both, controlled and uncontrolled operating conditions, in an attempt to observe the output and examining whether it is in accordance with certain pre-specified conditions.

Different sets of test cases and testing strategies are prepared, all of which are aimed at achieving one common goal i.e. removing bugs and errors from the code and making the software error-free and capable of providing accurate and optimum outputs. There are different types of software testing techniques and methodologies. A software testing methodology is different from a software testing technique. We'll take a look at few of the software testing methodologies in the latter part of this article.

**Software Testing Methodology**

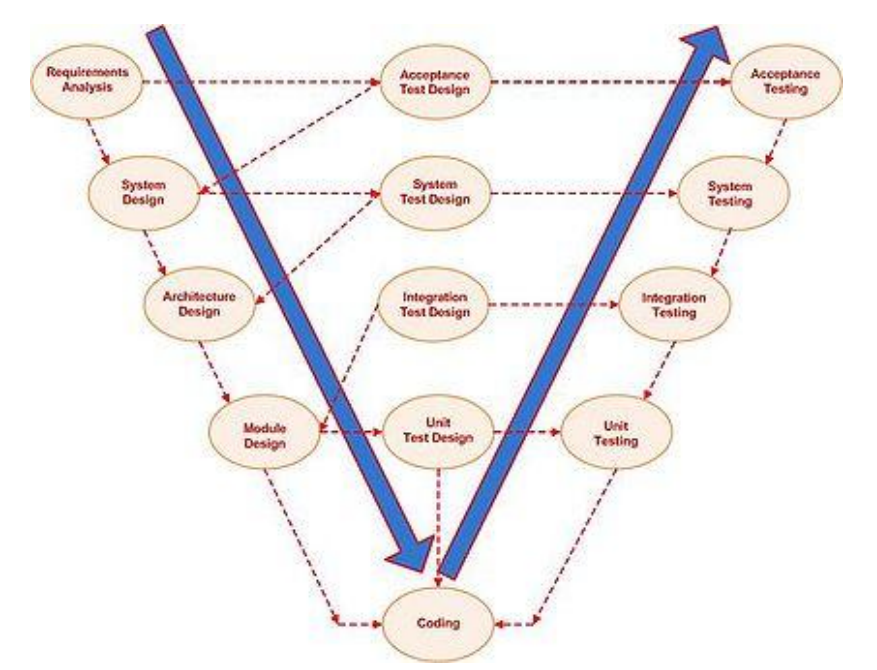
**V Model**

The issues seen in the traditional waterfall model gave birth to the V-Model;

It was developed with an intention to address some of the problems found in waterfall model. As you can see that in waterfall model defects were found very late in the development life cycle because testing was not involved until the end of the project.

In V-Model testing begins as early as possible in the project life cycle, it is always a good practice to involve testers at earlier phases of product life cycle. There are varieties of test activities that need to be carried out before end of the coding phase. These activities should be carried out in parallel to the development activities so that testers can produce a set of test deliverables.

The V-Model illustrates that testing activities (Verification and Validation) can be integrated into each phase of the product life cycle. Validation part of testing is integrated in the earlier phases of the life cycle which includes reviewing end user requirements, design documents etc. There are variants of V-Model however we will take a common type of V-model example.

****

The V-model generally has four test levels

.

* Component Testing
* Integration Testing
* System Testing
* Acceptance Testing

**Advantages**

* Each phase has specific deliverables.
* Higher chance of success over the waterfall model due to the development of test plans early on during the life cycle.
* Time concern in comparison with the waterfall model is low or even we can say 50% less.
* Works well for small projects where requirements are easily understood.
* Utility of the resources is high.

**Disadvantages**

* Very rigid, like the waterfall model.
* Little flexibility and adjusting scope is difficult and expensive.
* Software is developed during the implementation phase, so no early prototypes of the software are produced.
* Model doesn’t provide a clear path for problems found during testing phases

**4.2 Types of Testing**

**4.2.1 White Box Testing**

White box testing strategy deals with the internal logic and structure of the code. It is also called glass, structural, open or clear box testing. The tests that are written based on the white box testing strategy incorporate coverage of the code written, branches, paths, statements and internal logic of the code, etc. In order to implement white box testing, the tester has to deal with the code, and hence is required to possess knowledge of coding and logic i.e., internal working of the code. White box test also needs the tester to look into the code and find out which unit/statement/chunk of the code is malfunctioning. In other words, it is imperative that the tester has 'structural' knowledge about how the system has been implemented. Not only the code, but even the data flow and control flow have to be assessed. The areas of the code that are tested using white box testing are:

1. Code Coverage

2. Segment Coverage

3. Branch Coverage

4. Condition Coverage

5. Loop Coverage

6. Path Testing

7. Data Flow Coverage

There are three aspects of the code, which are validated in white box testing, namely

1. If the software has been designed as per the original design of the software.

2. If security measures have been implemented into the software and it is robust.

3. Find out vulnerabilities in the said software.

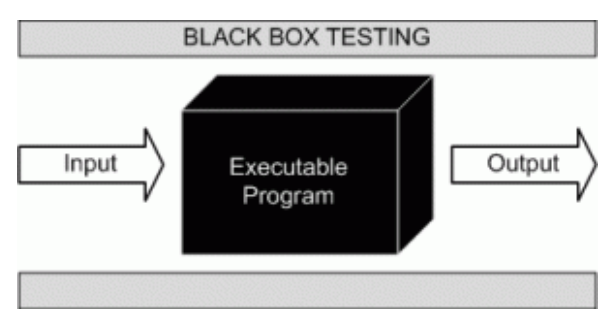
**Advantages of White Box Testing**

* + As the knowledge of internal coding structure is prerequisite, it becomes very easy to find out which type of input/data can help in testing the application effectively.
  + Yet another advantage of white box testing is that it helps in optimizing the code.
  + It helps in removing the extra lines of code, which can introduce defects in the code.

**Disadvantages of White Box Testing**

* + As knowledge of code and internal structure is a prerequisite, a skilled tester is needed to carry out this type of testing, and this, in turn, increases the cost of the software.
  + It is nearly impossible to look into every bit of code to find out hidden errors, which may create problems, resulting in failure of the application.

**4.2.2 Black Box Testing**

****

Black box testing is not a type of testing; it instead is a testing strategy, which does not need any knowledge of internal design or code, etc. As the name "black box" suggests, not knowledge of internal logic or code structure is required. The types of testing under this strategy are totally based/focused on the testing for requirements and functionality of the work product/software application. Black box testing is sometimes also called "Opaque Testing",

"Functional/Behavioral Testing" and "Closed Box Testing".

**These testing types are again divided in two groups:**

1. Testing in which the user plays a role of tester

2. User is not required

**4.3 Test Cases**

While testing a Website development application we need to consider following cases:

**4.3.1 Functionality Testing**

In functionality testing we have conducted following tests

* + **Web Forms**

i. Field validation

ii. Error message for wrong input

iii. Optional and Mandatory fields

**4.3.2 Usability Testing**

Usability testing is the process by which the human-computer interaction characteristics of a system are measured, and weaknesses are identified for correction.

* + - Ease of learning
    - Navigation
    - Subjective user satisfaction
    - General appearance

**Database**

Testing will be done on the database integrity

**4.3.3 Performance Testing**

Performance testing can be done to check the performance of the system in other operating environment. Here we have check how it is operated in other versions of the Internet Explorer, Google Chrome and Firefox and other configurations of the system

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case No** | **Purpose** | **Step to be Executed** | **Expected Result** | **Actual Result** | **Pass/ Fail** |
| 1. | Validation for | 1) Enter the valid data in email id Textbox.  2) Enter the valid data in Password textbox.  3)Click on Login Button | User Should be entering in Application | User is entering in Application | Pass |
| 2. | Validation for ‘Password’ field | 1) Enter the Invalid data in email id textbox.  2) Enter the valid data in Password textbox.  3)Click on Login Button | User Should be entering in Application | User is not entering  Application | Pass |
| **3.** | Validation for ‘Invalid Email Id’ field | 1)Enter the Invalid data in email id text box  2) Enter the valid data in Password textbox.  3) Click on Login Button | User Should be entering in Application | User is entering in Application | Pass |
| **4.** | Validation for ‘Invalid Password’ field | 1) Enter the valid data in email id text box.  2) Enter the Invalid data in Password textbox.  3)Click on Login Button | User Should be entering in Application | User is not entering in Application | Pass |
| **5.** | Creating new user | 1)Enter valid information in all textboxes  2)Click on Save button | User should be created | User should not be created | Fail |
| **6.** | Showing last point of device | 1)Click on Last Point checkbox | It should display last point of device on map | Last point should be displayed | Pass |
| **7.** | Geofence alert | 1)Click on Show Geofence checkbox | It should give alert if device goes outside of geofence area | Gives alert if device goes outside of area | Pass |
| **8.** | Show infowindow for each marker | 1)mousehover on each marker | It should display infowindow on marker | Infowindow should be displayed | Pass |

**5. System Implementation**

**5.1 Hardware requirement at Client Side**

* + Hard Disk :Free Space 1GB and more
  + Processor speed 250 MHz and more

The processor does all the processing for the contents of the program and when the clock speed increases so does the processing speed.

* + RAM 256 and more.

The memory selection is done and preferred for higher memory because the program before running is flushed into memory buffer of computer, then it is executed. More the memory, more will be the speed and hence less time for execution.

* + Sim Card: Standard Gsm sim
  + Tk102 tracing device

**5.2 Software requirement at Client Side**

Software plays an important role in any project development. One should understand that which software he/she should use to develop the project. Window XP was used as the operating system.

The application has been developed using:

* Internet Connection on any Computer.

**5.3 Testing done as End User**

An alpha test is a preliminary software field test carried out by a team of users in order to find bugs that were not found previously through other tests. The main purpose of alpha testing is to refine the software product by finding (and fixing) the bugs that were not discovered through previous tests.

The team that conducts the alpha test is often an independent test team, perhaps made up of potential users/customers. Alpha testing involves simulating a real user environment by carrying out tasks and operations that the actual users might perform. Once software passes the alpha test, it is considered for the next phase of testing called beta testing. The meaning of alpha also can differ based on whether the project is custom software one for a client. In this case, alpha testing implies an initial meeting between software vendor and client to ensure that the client's requirements are properly met by the developer in terms of the performance, functionality and durability of the software program. Compare this versus the context of a web application, where alpha testing can be interpreted as an online application that isn't completely ready for usage, but that has been opened up to get some initial feedback. Commonly an alpha might allow power users to get their first look at the system via a private invitation. The meaning of alpha also can differ based on whether the project is custom software one for a client. In this case, alpha testing implies an initial meeting between software vendor and client to ensure that the client's requirements are properly met by the developer in terms of the performance, functionality and durability of the software program

**6. System Maintenance and Evaluation**

**6.1 Maintenance**

The term maintenance, when accompanied to software, assumes a meaning profoundly different from the meaning it assumes in any other engineeringdiscipline. In fact, many engineering disciplines intend maintenance as the process of keeping something in working order, in repair. The key concept is the deterioration of an engineering artifact due to the use and the passing of time; the aim of maintenance is therefore to keep the artifact’s functionality in line with that defined and registered at the time of release. Of course, this view of maintenance does not apply to software, as software does not deteriorate with the use and the passing of time.

A predominant proportion of changes are to meet ever changing user needs. This is captured by the first law of Lehman: “A program that is used in a real world environment necessarily must change or become progressively less useful in that environment”. Significant changes also derive from the need to adapt software to interact with external entities, including people, organizations, and artificial systems.

Software maintenance is a very broad activity often defined as including all work made on a software system after it becomes operational. This covers the correction of errors, the enhancement, deletion and addition of capabilities, the adaptation to changes in data requirements and operation environments, the improvement ofperformance, usability, or any other quality attributes. The IEEE definition is as follows:

“Software maintenance is the process of modifying a software system orComponent after delivery to correct faults, improves performances or other attributes, or adapt to a changed environment.”This definition reflects the common view that software maintenance is a post-deliveryactivity: it starts when a system is released to the customer or user and encompasses all activities that keep the system operational and meet the user’s needs Software maintenance activities are classified into following four classes:

1. **Adaptive -** modifying the system to cope with changes in the software environment
2. **Perfective** - implementing new or changed user requirements which concernfunctional enhancements to the software

**3. Corrective -** diagnosing and fixing errors, possibly ones found by users

**4. Preventative -** increasing software maintainability or reliability to prevent problems in the future

Application performance has top most priority these days, as organization face tremendous customer demands and competitions in the entire business sector. Application maintenance is a challenging job for any software solution. Over a period of usage, the application may run out of performance due to unexpected problems .Based on user experience there could be suggestions for changes on the functionalities or mode of operation. Also for enterprise applications there may require emergency updates due to unexpected business requirements. All these come under software application maintenance namely preventive maintenance and scheduled maintenance. Preventive maintenance is performed based on the analysis of modules or product before it fails for effective operations.

There are many underlying factors that need to be taken care of while providing an application. The constant support and maintenance services, of course, are top priority.

Defect free system:

We provide 100% error free application to our clients all over the world.

We understand the requirements at all user levels of our clients.

Technical and functional expertise:

We have the rich technology blended technical and functional expertise to maintain the scalability, stability and strategy of business process to keep them up to date.

**Cost effective:**

The software models that we have delivered ensures, cost effectiveness in all ways.

**Flexibility to change:**

The application maintenance services, sometimes requires the system to undergo sufficient changes.

**6.2 Evaluation**

In the present day market situation, there are several alternatives available when a customer wants to purchase a product or adopt a software system that meets the customer’s requirements. Software technologies have been evolving rapidly and for a given set of functional and non-functionalrequirements there usually exist several competing software products. The present day users are faced with a challenging situation that requires evaluation and selection of a suitable software product that satisfies the users’ operational and business needs. Unfortunately this evaluation is usually carried on in anad-hoc and informal way and with various degree of success.

Software evaluation methodologies can be divided into two categories.

The First category is used to evaluate software development methodology or processes such as those used to evaluate variousagent-based development methodologies.

The Second category is used to evaluate software products such as Cost evaluation and selection methodologies.

System Evaluation recognizes and integrates the important features of other frameworks, overcomes any obvious deficiencies, and adopts new features that generalize and extend its usability.

The evaluation stage of the software development process requires the client and developer to review the software.

Broadly speaking, they evaluate against the following questions:

1. Does this software meet the user requirements?

2. Is it fit for purpose?

**Evaluation - Criteria:**

To answer these questions, the original aims of the software must be evaluated against the

Following criteria:

1. Robustness

2. Reliability

3. Portability

4. Efficiency

5. Maintainability

.

**7. User / Operational Manual**

1) This is a guide for the user of this system.

2) This contains all the relevant screen display that will help the user to understand the system.

3) There are proper uses of comments that will help the user to understand in case of error.

4) Proper validation is done and message box are prompted to the user before entering data into the database.

5) The comments describe what exactly the code block does.

**7.1 Security Aspects, Access Rights**

Does the protect itself and its data against unauthorized access and use? Does it allow its operator to enforce security policies? Are appropriate security mechanisms in place? Are those security mechanisms implemented correctly? Can the software withstand attacks that must be expected in its intended environment? Is the software free of errors that would make it possible to circumvent its security mechanisms? Does the architecture limit the impact of yet unknown errors? Security testing is any develop system is about finding loops and weaknesses of the system

There are two types of the user who can access the GPS Tracking System, they are:

* Administrator
* Normal User

Administrator has a right to access all the web forms of the project.

Normal user also has the permission to access the web forms of the Details Menu and the information about the hotels, restaurant, and recipes.

**User's perspective:**

In addition to the technical qualities of software, the end user's experience also determines the quality of software. This aspect of software quality is called usability. It is hard to quantify the usability of a given software product. Some important questions to be asked are:

* Is the user interface intuitive?
* Is it easy to perform easy operations?
* Is it feasible to perform difficult operations?
* Does the software give sensible error messages?
* Do widgets behave as expected?
* Is the software well documented?
* Is the user interface self-explanatory/ self-documenting?
* Is the user interface responsive or too slow?

Also, the availability of (free or paid) support may determine the usability of the software.

**7.2 Back Up’s**

In information technology, backup refers to making copies of data so that these additional copies may be used to restore the original after a data loss event. These additional copies are typically called "backups." Backups are useful primarily for two purposes. The first is to restore a state following a disaster (called disaster recovery). The second is to restore small numbers of files after they have been accidentally deleted or corrupted. Backups are typically that last line of defense against data loss, and consequently the least granular and the least convenient to use.

Since a backup system contains at least one copy of all data worth saving, the data storage requirements are considerable. Organizing this storage space and managing the backup process is a complicated undertaking. There are also many different ways in which these devices can be arranged to provide geographic redundancy, data security, and portability. Before data are sent to their storage location, they are selected, extracted, and manipulated.

Many organizations and individuals try to have confidence that the process is working as expected and work to define measurements and validation techniques. It is also important to recognize the limitations and human factors involved in any backup scheme.

Due to a considerable overlap in technology, backups and backup systems are frequently confused with archives and fault-tolerant systems. Backups differ from archives in the sense that archives are the primary copy of data and backups are a secondary copy of data. Backup systems differ from fault-tolerant systems in the sense that backup systems assume that a fault will cause a data loss event and fault-tolerant systems assume a fault will not.

* + Database Backups is taken by Admin on weekly basis.
  + Backups help in Data Integrity.

**7.3 Screen Shots**

**8. Future Enhancement**

**9. Limitations**

The limitations of the website are as follows:

* The user has to be computer literate.
* Website will not work in case of internet unavailability
* The website will not work in case of power failure or disconnect the internet

.

**10.CONCLUSION**

To display things on the website i.e all information about the hotels, restorant, and any recepies with address, phone number in any time any place.

The objective of this project was to build the website for all information for hotels, restaurant, recepies with address in one place, any time. The website developed is able to meet all the basic information hotels, restaurant, recepies with address in one place, any time.

. The important thing is that the website should be flexible enough for future modifications. The website has been factored into different modules to make system adapt to the further changes. Every effort has been made to cover all user requirements and make it user friendly.

Goal achieved: The website is able provide the interface to the owner so that he can replicate his desired data.

User friendliness: Though the most part of the website is supposed to act in the background, efforts have been made to make the foreground interaction with user (owner) as smooth as possible. Also the integration of the existing system with the project has been kept in mind throughout the development phase.

**11. Bibliography**

**Books**

* [**PHP and MYSQL for Dummies**](http://www.amazon.com/exec/obidos/ASIN/0764516507/ref=nosim/forlagetadland00) by Valade.John Wiley & Sons Inc. Paperback- 30 April, 2002
* [**PHP Bible**](http://www.amazon.com/exec/obidos/ASIN/0764549553/ref=nosim/forlagetadland00) by Converse.John Wiley & Sons Inc. Paperback- 4 October, 2002.
* [**Core PHP Programming**](http://www.amazon.com/exec/obidos/ASIN/0130893986/ref=nosim/forlagetadland00) by Leon Atkinson.Prentice Hall PTR. Paperback- 1 August, 2000.

# Websites

* + http://www.dish\_is\_wish.com