Contents

[1 INTRODUCTION 2](#_Toc54209824)

[1.1 ABOUT THE ORGANIZATION 2](#_Toc54209825)

[2 SYSTEM CONFIGURATION 3](#_Toc54209826)

[2.1 HARDWARE CONFIGURATION 3](#_Toc54209827)

[2.2 SOFTWARE CONFIGURATION 3](#_Toc54209828)

[2.3 ABOUT THE SOFTWARE 4](#_Toc54209829)

[3 SYSTEM STUDY 10](#_Toc54209830)

[3.1 EXISTING SYSTEM 10](#_Toc54209831)

[3.2 DISADVANTAGES 10](#_Toc54209832)

[3.3 PROPOSED SYSTEM 11](#_Toc54209833)

[4 SYSTEM DESIGN 12](#_Toc54209834)

[4.1 INPUT DESIGN 12](#_Toc54209835)

[4.2 OUTPUT DESIGN 15](#_Toc54209836)

[5 CONCLUSION 37](#_Toc54209837)

[6 BIBLIOGRAPHY 38](#_Toc54209838)

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| INTRODUCTION | | | |
|  | 1.1 ABOUT THE ORGANIZATION | | |
| SYSTEM CONFIGURATION | | | |
|  | | | 2.1 HARDWARE CONFIGURATION Processor : Pentium  Hard Disk : 200 GB  RAM : 6 GB  Monitor : 15 inches  Mouse : Microsoft Mouse  Keyboard : 109 keys | | |
|  | 2.2 SOFTWARE CONFIGURATION Front End : HTML, CSS, JS, PHP  Software Tools : XAMP  Back End : MySQL server  Operating System : Windows | |
|  | 2.3 ABOUT THE SOFTWARE **FRONT END**    JavaScript is a [programming language](https://techterms.com/definition/programming_language)commonly used in [web development](https://techterms.com/definition/web_development). It was originally developed by Netscape as a means to add dynamic and interactive elements to websites. While JavaScript is influenced by [Java](https://techterms.com/definition/java), the [syntax](https://techterms.com/definition/syntax) is more similar to [C](https://techterms.com/definition/cplusplus) and is based on ECMAScript, a scripting language developed by Sun Microsystems.  A lot of the syntax of PHP is borrowed from other languages such as C, Java and Perl. However, PHP has a number of unique features and specific functions as well. The goal of the language is to allow Web developers to write dynamically generated pages quickly and easily. PHP is also great for creating database-driven Web sites. If you would like to learn more about PHP, the official site is [PHP.net](http://www.php.net/).  JavaScript is a client-side scripting language, which means the [source code](https://techterms.com/definition/sourcecode) is processed by the client's [web browser](https://techterms.com/definition/web_browser) rather than on the [web server](https://techterms.com/definition/web_server). This means JavaScript [functions](https://techterms.com/definition/function)can run after a webpage has loaded without communicating with the server. For example, a JavaScript function may check a web form before it is submitted to make sure all the required [fields](https://techterms.com/definition/field) have been filled out. The JavaScript code can produce an error message before any information is actually transmitted to the server.  HTML Stands for "Hypertext Markup Language." HTML is the language used to create [webpages](https://techterms.com/definition/webpage). "Hypertext" refers to the [hyperlinks](https://techterms.com/definition/hyperlink) that an HTML page may contain. "Markup language" refers to the way [tags](https://techterms.com/definition/tag) are used to define the page layout and elements within the page. The [web](https://techterms.com/definition/www) has gone through many changes over the past few decades, but HTML has always been the fundamental language used to develop webpages. Interestingly, while [websites](https://techterms.com/definition/website) have become more advanced and interactive, HTML has actually gotten simpler. If you compare the [source](https://techterms.com/definition/sourcecode) of an HTML5 page with a similar page written in HTML 4.01 or [XHTML](https://techterms.com/definition/xhtml) 1.0, the HTML5 page would probably contain less code. This is because modern HTML relies on [cascading style sheets](https://techterms.com/definition/css) or [JavaScript](https://techterms.com/definition/javascript) to format nearly all the elements within a page.  Like server-side scripting languages, such as [PHP](https://techterms.com/definition/php) and [ASP](https://techterms.com/definition/asp), JavaScript code can be inserted anywhere within the [HTML](https://techterms.com/definition/html) of a [webpage](https://techterms.com/definition/webpage). However, only the [output](https://techterms.com/definition/output) of server-side code is displayed in the HTML, while JavaScript code remains fully visible in the source of the webpage. It can also be referenced in a separate [.JS](https://fileinfo.com/extension/js) file, which may also be viewed in a browser.  JQuery  JQuery is a [JavaScript](https://techterms.com/definition/javascript) library that allows web developers to add extra functionality to their websites. It is [open source](https://techterms.com/definition/opensource) and provided for free under the MIT license. In recent years, jQuery has become the most popular JavaScript library used in [web development](https://techterms.com/definition/web_development). To implement jQuery, a web developer simply needs to reference the jQuery JavaScript file within the [HTML](https://techterms.com/definition/html) of a [webpage](https://techterms.com/definition/webpage). Some websites host their own local copy of jQuery, while others simply reference the library hosted by [Google](https://techterms.com/definition/google) or the jQuery server.  Features of php:   * It is very simple and easy to use, compare to other scripting language it is very simple and easy, this is widely used all over the world. * It is an interpreted language, i.e. there is no need for compilation. * It is faster than other scripting language e.g. asp and jsp. * Open source means you no need to pay for use php, you can free download and use. * PHP code will be run on every platform, Linux, Unix, Mac OS X, Windows. * PHP is case sensitive scripting language at time of variable declaration. In PHP, all keywords (e.g. if, else, while, echo, etc.), classes, functions, and user-defined functions are NOT case-sensitive. * PHP have some predefined error reporting constants to generate a warning or error notice. * PHP provides access logging by creating the summary of recent accesses for the user. * PHP supports variable usage without declaring its data type. It will be taken at the time of the execution based on the type of data it has on its value.   Features of javascript   * Using restrictions, you can better control the resources of the scheduler. This feature allows you to prevent drag and resize capabilities for a particular event. Also, you can set criteria that will allow or prevent the creation and modification of events. * You can configure the selection feature if you want to allow users to select multiple events at once. They can do it using Ctrl+Click. Several selected tasks can be moved through the calendar grid using drag-and-drop. * You can use the timeline to provide end-users with an overview for scheduling purposes. The Timeline View is a perfect solution for those who need to visualize and monitor the progress of ongoing tasks or projects. * JavaScript is an easy language intended to be interpreted at runtime. It has vaguely C-formed syntax (save for pointer arithmetic); a straightforward classless object model, said to be inspired by a little-known programming language named Self; automatic garbage collection; and weak, dynamic typing.   BACK END  MYSQL    Relational database  A [relational database](https://searchdatamanagement.techtarget.com/definition/relational-database), invented by [E.F. Codd](https://searchoracle.techtarget.com/definition/E-F-Codd) at IBM in 1970, is a tabular database in which data is defined so that it can be reorganized and accessed in a number of different ways. Relational databases are made up of a set of tables with data that fits into a predefined category. Each table has at least one data category in a column, and each row has a certain data instance for the categories which are defined in the columns. The [Structured Query Language](https://searchsqlserver.techtarget.com/definition/SQL) (SQL) is the standard user and application program interface for a relational database. Relational databases are easy to extend, and a new data category can be added after the original database creation without requiring that you modify all the existing applications.  RDBMS technologies:   * **Database:** A database is a collection of tables, with related data. * **Table:** A table is a matrix with data. A table in a database looks like a simple spreadsheet. * **Column:** One column (data element) contains data of one and the same kind, for example the column postcode. * **Row:** A row is a group of related data, for example the data of one subscription. * **Redundancy:** Storing data twice, redundantly to make the system faster. * **Primary Key:** A primary key is unique. A key value cannot occur twice in one table. With a key, you can find at most one row. * **Foreign Key:** A foreign key is the linking pin between two tables. * **Compound Key:** A compound key is a key that consists of multiple columns, because one column is not sufficiently unique. * **Index:** An index in a database resembles an index at the back of a book.   Database  A database is a collection of [information](https://searchsqlserver.techtarget.com/definition/information) that is organized so that it can be easily accessed, managed and updated. Data is organized into rows, columns and tables, and it is indexed to make it easier to find relevant information. Data gets updated, expanded and deleted as new information is added. Databases process workloads to create and update themselves, querying the data they contain and running applications against it.  MYSQL Database:   * MySQL is an Oracle-backed open source relational database management system ([RDBMS](https://searchdatamanagement.techtarget.com/definition/RDBMS-relational-database-management-system)) based on Structured Query Language ([SQL](https://searchsqlserver.techtarget.com/definition/SQL)). MySQL runs on virtually all platforms, including [Linux](https://searchdatacenter.techtarget.com/definition/Linux-operating-system), [UNIX](https://searchdatacenter.techtarget.com/definition/Unix) and [Windows](https://searchwindowsserver.techtarget.com/definition/Windows). Although it can be used in a wide range of applications, MySQL is most often associated with web applications and online publishing. * MySQL is an important component of an open source enterprise stack called [LAMP](https://whatis.techtarget.com/definition/LAMP-Linux-Apache-MySQL-PHP). LAMP is a web development platform that uses Linux as the operating system, [Apache](https://whatis.techtarget.com/definition/Apache) as the web server, MySQL as the relational database management system and  [PHP](https://whatis.techtarget.com/definition/PHP-Hypertext-Preprocessor) as the object-oriented scripting language. * MySQL is based on a [client-server](https://searchnetworking.techtarget.com/definition/client-server) model. The core of MySQL is MySQL server, which handles all of the database instructions (or commands). MySQL server is available as a separate program for use in a client-server networked environment and as a library that can be embedded (or linked) into seperate applications. | | |
|  |  | | |
| SYSTEM STUDY | | | |
|  | 3.1 EXISTING SYSTEM The existing system was a manual one. Whatever be the process involved in the system were done through register (files). There were lots of complexities involved in the system. . It was not easy to do several administrative works like managing the customer details, their account details, withdrawal details.etc.the process is more time consuming and it requires a great manual effort. | | |
|  | 3.2 DISADVANTAGES  * More time is consumed. * More Hard work to maintain all records. * Bulk of paper to be searched for a single paper. | | |
|  | 3.3 PROPOSED SYSTEM The system provides paperless maintenance. Initially a cashier or a clerk can be appointed to do all the transaction and update and maintained the records. In the new system the customer himself can do all the transaction and the computerized system automatically updates and maintains the records.  **Advantages**   * Less effort to complete transaction * Less time required. * No need to maintain the bulk of Papers. | | |
| SYSTEM DESIGN | | | |
|  | 4.1 INPUT DESIGN Input design is a part of overall system design, which requires careful attention. It is the process of converting user-originated inputs to a computer-based format. The major objective of the input design is to make data entry easy, logical and error free.  In PHP input to the system is entered through forms. A form is “any surface on which information is to be entered, the nature of which is determined by what is already on the surface”. If the data going into the system is incorrect, then processing and output will magnify these errors. So designer should ensure that form is acceptable and understandable by the user.  This application has been developed in a user-friendly manner. The layout of the form is made in such a way that the user will not find any difficulty in going from one field to other by just pressing the tab. During the processing the cursor is placed in the position where the data must be entered.  The user is also provided with an option of selecting an appropriate input from a list of values. Necessary dropdown list boxes and combo boxes are included for necessary fields so that the user need not remember all the data and can just select from it.  Validation is made for each and every data entered. Help messages are also provided whenever the users enter a wrong data into a particular field. This makes the user to understand what is to be entered, moreover whenever an erroneous data is entered the error message is displayed and the user can move to the next field only after entering the correct data.  The clear label for the menus and fields are also provided. Consultations are provided so that a user can view the details of any process at any time.  In an information system, input is the raw data that is processed to produce output. During the input design, the developers must consider the input devices such as PC, MICR, OMR, etc.  Therefore, the quality of system input determines the quality of system output. Well designed input forms and screens have following properties −   * It should serve specific purpose effectively such as storing, recording, and retrieving the information. * It ensures proper completion with accuracy. * It should be easy to fill and straightforward. * It should focus on user’s attention, consistency, and simplicity. * All these objectives are obtained using the knowledge of basic design principles regarding −   + What are the inputs needed for the system?   + How end users respond to different elements of forms and screens.   The Aim of input design are −   * To design data entry and input procedures * To reduce input volume * To design source documents for data capture or devise other data capture methods * To design input data records, data entry screens, user interface screens, etc. * To use validation checks and develop effective input controls.   Data Input Methods  It is important to design appropriate data input methods to prevent errors while entering data. These methods depend on whether the data is entered by customers in forms manually and later entered by data entry operators, or data is directly entered by users on the PCs.  A system should prevent user from making mistakes by −   * Clear form design by leaving enough space for writing legibly. * Clear instructions to fill form. * Clear form design. * Reducing key strokes. * Immediate error feedback.   Some of the popular data input methods are −   * Batch input method (Offline data input method) * Online data input method * Computer readable forms * Interactive data input   Input integrity controls include a number of methods to eliminate common input errors by end-users. They also include checks on the value of individual fields; both for format and the completeness of all inputs.  Audit trails for data entry and other system operations are created using transaction logs which gives a record of all changes introduced in the database to provide security and means of recovery in case of any failure. | | |
|  | 4.2 OUTPUT DESIGN One of the most important features of a system for users in the output it produces. Output design should improve the system’s relationship with the user and help in decision-making. Considering the future use of output required, and depending on the nature, it is displayed on the monitor for immediate need of obtaining the hard copy.  The objective of output design is to define the controls and format of all printed documents and reports and of screens that will be produced by the system. Computer output is the most important and direct source of information to the user.  **Objectives Of Output Design**   1. Design output to serve the intended purpose. 2. Deliver the appropriate quality of output. 3. Choose the right output method. 4. Provide output on time. 5. To develop output design that serves the intended purpose and eliminates the production of unwanted output. 6. To develop the output design that meets the end users requirements. 7. To deliver the appropriate quantity of output. 8. To form the output in appropriate format and direct it to the right person. 9. To make the output available on time for making good decisions.   various types of outputs  External Outputs  Manufacturers create and design external outputs for printers. External outputs enable the system to leave the trigger actions on the part of their recipients or confirm actions to their recipients.  Some of the external outputs are designed as turnaround outputs, which are implemented as a form and re-enter the system as an input.  Internal outputs  Internal outputs are present inside the system, and used by end-users and managers. They support the management in decision making and reporting.  There are three types of reports produced by management information −   * **Detailed Reports** − They contain present information which has almost no filtering or restriction generated to assist management planning and control. * **Summary Reports** − They contain trends and potential problems which are categorized and summarized that are generated for managers who do not want details. * **Exception Reports** − They contain exceptions, filtered data to some condition or standard before presenting it to the manager, as information.   Output Integrity Controls  Output integrity controls include routing codes to identify the receiving system, and verification messages to confirm successful receipt of messages that are handled by network protocol.  Printed or screen-format reports should include a date/time for report printing and the data. Multipage reports contain report title or description, and pagination. Pre-printed forms usually include a version number and effective date.  Both forms and reports are the product of input and output design and are business document consisting of specified data. The main difference is that forms provide fields for data input but reports are purely used for reading. For example, order forms, employment and credit application, etc.   * During form designing, the designers should know   + who will use them   + the purpose of the form or report   + where would they be delivered * During form design, automated design tools enhance the developer’s ability to prototype forms and reports and present them to end users for evaluation.   Aim of Form Design  A form design is necessary to ensure   * To keep the screen simple by giving proper sequence, information, and clear captions. * To meet the intended purpose by using appropriate forms. * To ensure the completion of form with accuracy. * To keep the forms attractive by using icons, inverse video, or blinking cursors etc. * To facilitate navigation.   **DATABASE DESIGN**  The activity deals with the design of the database. A key is to determine how the access paths are to be implemented. A physical path is derived from a logical path. The general theme behind database is to handle information as a whole. A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently.  The general objective is to make information access easy, quick, inexpensive and flexible for the user. Database design is the most critical part of the design phase. An elegantly designed, well-defined database is a strong foundation for the whole system. Files in a relational database are called as tables. Columns of tables represent data and rows represent the records in conventional technology.  **Normalization of Database**  Database Normalization is a technique of organizing the data in the database. Normalization is a systematic approach of decomposing tables to eliminate data redundancy and undesirable characteristics like Insertion, Update and Deletion Anomalies. It is a multi-step process that puts data into tabular form by removing duplicated data from the relation tables.  Normalization is used for mainly two purposes,   * Eliminating redundant (useless) data. * Ensuring data dependencies make sense i.e. data is logically stored.   **Normalization rule**  Normalization rule are divided into following normal form.   * First Normal Form * Second Normal Form * Third Normal Form   **First Normal Form (1NF)**   * As per First Normal Form, no two Rows of data must contain repeating group of information i.e. each set of column must have a unique value, such that multiple columns cannot be used to fetch the same row. Each table should be organized into rows, and each row should have a primary key that distinguishes it as unique. * The **Primary key** is usually a single column, but sometimes more than one column can be combined to create a single primary key. For example consider a table which is not in First normal form.   **Second Normal Form (2NF)**  As per the Second Normal Form there must not be any partial dependency of any column on primary key. It means that for a table that has concatenated primary key, each column in the table that is not part of the primary key must depend upon the entire concatenated key for its existence. If any column depends only on one part of the concatenated key, then the table fails **Second normal form**.  **Third Normal Form (3NF)**  **Third Normal form** applies that every non-prime attribute of table must be dependent on primary key, or we can say that, there should not be the case that a non-prime attribute is determined by another non-prime attribute. So this *transitive functional dependency* should be removed from the table and also the table must be in **Second Normal form**. For example, consider a table with following fields.  FEASIBILITY STUDY  The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company.  Three key considerations involved in the feasibility analysis are   * ECONOMICAL FEASIBILITY * TECHNICAL FEASIBILITY * OPERATIONAL FEASIBILITY * SOCIAL FEASIBILITY   ECONOMICAL FEASIBILITY  This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.  TECHNICAL FEASIBILITY  This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.  OPERATIONAL FEASIBILITY  The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.  SOCIAL FEASIBILITY  The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.  The first step in initial investigation is clarifying and strengthening the analyst's background about the organization. After studying the existing system one can gather information. One should know what information to be gathered, where and checked for completeness and accuracy. Analyzing the data involves identification of the system and interrelationships and identifying the strength and weakness of the system.  Major activities involved in the system analysis are fact finding and fact analysis. An analyst can collect data from two principal sources: written documents and personnel who are knowledgeable about or involved in the operation of the system under study.  Fact finding technique,which is the most appropriate one, includes four techniques   * Data collection * Questionnaires * Personnel interviews * Onsite-Observation   **Data Collection**  The data has been collected and organized from the documents such as records,reports from the project manager, then collected the details of the present system. The recent information is collected through personal interviews and by direct observation.  **Questionnaires**  Questionnaires are an important and effective type of correspondence. This method is effective to obtain information from a large number of peoples, particularly if they are widely scattered or in remote locations. It is also possible to ask to the manager about their requirements. Questionnaires should be brief in order to increase the promptness and probability of response. Questionnaires can also be used to solicit responses to specific questions from individuals.  **Personnel interview**  It is one of the most fruitful methods of obtaining information.Personal interviews has been conducted with the manager and also collected all the details related to the project.  **Onsite -observation**  The datas has been collected by observing the operations of the existing system and analyzed what are all the changes to be made to the existing system and formulated questions and drawn conclusions on the basis of observations.  **Implementation Planning:**  This section describes about the Implementation of the This application and the details of how to access this control from any application.  Implementation is the process of assuring that the information system is operational and then allowing users take over its operation for use and evaluation. Implementation includes the following activities.   * Obtaining and installing the system hardware. * Installing the system and making it run on its intended hardware. * Providing user access to the system. * Creating and updating the database. * Documenting the system for its users and for those who will be responsible for maintaining it in the future. * Making arrangements to support the users as the system is used. * Transferring ongoing responsibility for the system from its developers to the operations or maintenance part. * Evaluating the operation and use of the system.   **Implementation Phase in this project:**  This new system has been implemented. The database was put into the MySQL server.. The database is accessible through DriverManager. Documentation is provided well in such a way that it is useful for users and maintainers.  Your needs document defines the capabilities your new system needs so that you can focus on what’s most important during your selection stage.  And then again during the implementation and training stage, your needs document helps you avoid doing everything all at once.  The next critical step in your implementation journey is assembling the team(s) necessary for success. The makeup of an implementation team will vary for every business, depending on the unique needs of your business and the scale of implementation.  **To determine your needs**, identify how many business units will be using the new system and estimate of the total number of users. The more business units/users adopting the software, the larger you’ll want your implementation team to be.  o matter how great and intuitive the new system and tools you’ve selected are, implementation doesn’t equal adoption. You must put strategies in place to garner user acceptance and adoption of the new system.  Without positive engagement around the product, you risk adoption falling flat and ruining the software implementation plan.  To support the long-term implementation of your new system, start by prioritizing those capabilities that need to be mastered first. This will help influence training and provide benchmarks for regular check-ins.  Training is a central pillar in continuous improvement. But much like implementation, the right kind of training done when needed is better than poor training done often.  **Maintenance:**  Maintenance is any work done to change the system after it is in operational. The term maintenance is used to describe activities that occur following the delivery of the product to the customer. The maintenance phase of the software life cycle is the time period in which a software product performs useful work.  Maintenance activities involve making enhancements to products, adapting products to new environments, correcting problems.  In this be retrieve the data from the database design by searching the database. So, for maintaining data our project has a backup facility so that there is an additional copy of data, which needs to be maintained.  More over this project would update the annual data on to a CD, which could be used for later reference  **Methods of implementation:**  The four basic methods of implementation are:  1. Parallel system method.  2. Direct cut over method.  3. Pilot system method.  4. Phase in method.  Corrective Software Maintenance  Corrective software maintenance is what one would typically associate with the maintenance of any kind. Correct software maintenance addresses the errors and faults within software applications that could impact various parts of your software, including the design, logic, and code. These corrections usually come from bug reports that were created by users or customers – but corrective software maintenance can help to spot them before your customers do, which can help your brand’s reputation.  Adaptive Software Maintenance  Adaptive software maintenance becomes important when the environment of your software changes. This can be brought on by changes to the operating system, hardware, software dependencies, Cloud storage, or even changes within the operating system. Sometimes, adaptive software maintenance reflects organizational policies or rules as well. Updating services, making modifications to vendors, or changing payment processors can all necessitate adaptive software maintenance.  Perfective Software Maintenance  Perfective software maintenance focuses on the evolution of requirements and features that existing in your system. As users interact with your applications, they may notice things that you did not or suggest new features that they would like as part of the software, which could become future projects or enhancements. Perfective software maintenance takes over some of the work, both adding features that can enhance user experience and removing features that are not effective and functional. This can include features that are not used or those that do not help you to meet your end goals.  Preventive Software Maintenance  Preventative Software Maintenance helps to make changes and adaptations to your software so that it can work for a longer period of time. The focus of the type of maintenance is to prevent the deterioration of your software as it continues to adapt and change. These services can include optimizing code and updating documentation as needed.  Preventative software maintenance helps to reduce the risk associated with operating software for a long time, helping it to become more stable, understandable, and maintainable.  For all businesses and organizations, software maintenance is an essential part of the software development lifecycle. This isn’t something that one can skip or avoid. It is absolutely necessary for the success of your software and any evolution into the future. It is important to know that maintenance needs to go much further than fixing issues or bugs – that is only one steps of the software maintenance process.  TESTING METHEDOLOGIES  System Testing  System testing includes code testing which examines the logic of the program. Each and every part of the program is checked or executed individually to find out the errors. Once the errors in the program are found out, they are debugged. If wrong data is entered, an error message is displayed on the screen so that the user can correct the data at that time itself.  System testing is actually a series of different tests whose primary purpose is to fully exercise the computer based system. A series of testing are performed for the proposed system before the system is ready for the user acceptance test. A candidate system is subject to variety of tests – volume, stress, recovery, security and usability tests.  The steps in the system testing can be categorized as follows:   * Unit Testing * Requirement Testing * User Testing * Validation Testing * Integration Testing * User Acceptance Testing   Unit Testing  Unit testing focuses on verification efforts on the smallest unit of software design i.e., the module. The unit testing is always white box oriented and the step can be conducted in parallel for modules. The module interface is tested to ensure that information properly flows in and out of the program unit under test. The ‘local data structures ‘ are examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm execution.” Boundary Conditions” are tested to ensure that the module operates properly at boundaries established to limit or restrict processing. All ‘independent paths ‘ through the control structures are exercised to ensure that all statement in a module have been executed at least once. Finally all “ Error-Handling “ are tested.  Requirement Testing  The main aim of this test plan is to see whether the outputs created and inputs were given according to the user requirements and specifications that have been established. This was done in the security department by having the developer as a secondary person and another employee who conducted the actual test. Some Suggestion was made while requirement testing was done that has been incorporated.  User Testing  During the testing the tester places the role of the individual who desires to penetrate the system. The tester may attempt to acquire passwords through external clerical means and may attack the system with the custom software design to break down any defenses that have been constructed. The tester may also overwhelm the system thereby denying service to other s and may purposely cause system errors to penetrate during recovery and may browse through insecure data, hoping to find key to system entry.  Validation Testing  At the end of user testing, software is completely assembled as a package, interfacing errors have now being uncovered and correcting test begins. Software testing and validation are achieved through a serious black box tests that demonstrate conformity with the requirement.  A plan outlines the classes of tests to be conducted and test procedure defines specific cases that will be used to demonstrate conformity with requirements. Both the plan and the procedure are designed to ensure that all functional requirements are achieved, documentation is correct and other requirements are met. After the validation test, one of the conditions exists.  They are,  The function or performance characteristics confirm to specification and are accepted. A deviation from the specification is uncovered and a deficiency list is created. The deviation or error discovered at this stage in a project can rarely be corrected prior to scheduled completion. It is necessary to negotiate with the customer to establish methods.  Integration testing  Integration testing is a systematic technique for constructing the program structure while conducting tests to uncover errors associated with interfacing. The objective is to take unit testing modules and build a program structure that has been dictated by design.  There is often a tendency to attempt non incremental integration; that is to construct the program using “big-bang” approach. All modules are combined in advance. The entire program is tested as a whole. When a set of errors is encountered, correction is difficult because isolation of causes is complicated by the vast expanse of the entire program. Once these errors are corrected new ones appear and the process continues in a seemingly endless loop.  Incremental integration is the antithesis of big-bang approach. The program is constructed and tested in small sequence, where errors are easier to isolate and correct; interfaces are more likely to be tested completely; and a systematic test approach may be applied.  User acceptance testing  Acceptance testing involves planning and execution of functional tests, performance test, and stress tests to verify that the implemented system satisfies its requirements. Acceptance tests are typically performed by quality assurance and/or customer organizations. Functional and performance tests are performed to determine the limitations of the system. Typically , acceptance typically, acceptance test will incorporate test cases developed during unit testing and integration testing. Additional test cases are added to achieve desired level of functional , performance, and stress testing of the entire system. Tools of Special importance during acceptance testing include a test Coverage analyzer, a timing analyzer, and a coding standards checker. A test coverage analyzer records the control paths followed for each test case. Timing analyzer reports the time spent in various regions of the source code and different test cases.  White Box Testing  White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.  Black Box Testing  Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.  System maintenance is widely accepted part of SDLC now a days. It stands for all the modifications and updations done after the delivery of software product. There are number of reasons, why modifications are required, some of them are briefly mentioned below:   * **Market Conditions**- Policies, which changes over the time, such as taxation and newly introduced constraints like, how to maintain bookkeeping, may trigger need for modification. * **Client Requirements** - Over the time, customer may ask for new features or functions in the software. * **Host Modifications**- If any of the hardware and/or platform (such as operating system) of the target host changes, software changes are needed to keep adaptability. * **Organization Changes** - If there is any business level change at client end, such as reduction of organization strength, acquiring another company, organization venturing into new business, need to modify in the original software may arise.   Types of maintenance  In a software lifetime, type of maintenance may vary based on its nature. It may be just a routine maintenance tasks as some bug discovered by some user or it may be a large event in itself based on maintenance size or nature. Following are some types of maintenance based on their characteristics:   * **Corrective Maintenance** - This includes modifications and updations done in order to correct or fix problems, which are either discovered by user or concluded by user error reports. * **Adaptive Maintenance** - This includes modifications and updations applied to keep the software product up-to date and tuned to the ever changing world of technology and business environment. * **Perfective Maintenance** - This includes modifications and updates done in order to keep the software usable over long period of time. It includes new features, new user requirements for refining the software and improve its reliability and performance. * **Preventive Maintenance** - This includes modifications and updations to prevent future problems of the software. It aims to attend problems, which are not significant at this moment but may cause serious issues in future.   Corrective Software Maintenance  Corrective software maintenance is what one would typically associate with the maintenance of any kind. Correct software maintenance addresses the errors and faults within software applications that could impact various parts of your software, including the design, logic, and code. These corrections usually come from bug reports that were created by users or customers – but corrective software maintenance can help to spot them before your customers do, which can help your brand’s reputation.  Adaptive Software Maintenance  Adaptive software maintenance becomes important when the environment of your software changes. This can be brought on by changes to the operating system, hardware, software dependencies, Cloud storage, or even changes within the operating system. Sometimes, adaptive software maintenance reflects organizational policies or rules as well. Updating services, making modifications to vendors, or changing payment processors can all necessitate adaptive software maintenance.  Perfective Software Maintenance  Perfective software maintenance focuses on the evolution of requirements and features that existing in your system. As users interact with your applications, they may notice things that you did not or suggest new features that they would like as part of the software, which could become future projects or enhancements. Perfective software maintenance takes over some of the work, both adding features that can enhance user experience and removing features that are not effective and functional. This can include features that are not used or those that do not help you to meet your end goals.  Preventive Software Maintenance  Preventative Software Maintenance helps to make changes and adaptations to your software so that it can work for a longer period of time. The focus of the type of maintenance is to prevent the deterioration of your software as it continues to adapt and change. These services can include optimizing code and updating documentation as needed.  Preventative software maintenance helps to reduce the risk associated with operating software for a long time, helping it to become more stable, understandable, and maintainable.  For all businesses and organizations, software maintenance is an essential part of the software development lifecycle. This isn’t something that one can skip or avoid. It is absolutely necessary for the success of your software and any evolution into the future. It is important to know that maintenance needs to go much further than fixing issues or bugs – that is only one steps of the software maintenance process. | | |
| CONCLUSION I have tried to create a user friendly, useful, easy to handle application. The developed system is flexible, robust. The newly produced system provides facility which existing system were unable to provide such as remote access from anywhere, easy access, easier maintenance etc. The important benefits that have been found out through the implemented system are: reduced processing time, Increases accuracy, more reliability.  The system provides accuracy and updating facility to the users. Also the system is flexible and any amendments can be made with minor modifications. Testing has been done with actual data and system is much better than the existing one. This software is designed and developed using **PHP,** and the back-end tool for this application to store every event is **SQL Admin.** | | | |
| BIBLIOGRAPHY | | | |

1. PHP and MySQL for Beginners Mark Lassoff
2. Dynamic Web Design Made Easy David Powers
3. Build Gamified Websites with PHP and jQuery Detrick DeBurr
4. A Simple Start To PHP & MySQL Written By A Software Engineer Scott Sanderson
5. PHP for Absolute Beginners Jason Lengstorf, Thomas Blom Hansen
6. A Beginner's Guide to Programming Interactive Web Applications with PHP and mySQL Alan Forbes
7. PHP Objects, Patterns, and Practice Matt Zandstra
8. An Essential Beginners Guide to Learn the Realms of PHP From A-Z Logan Pratt