

Table of Contents

Project Topic.....	2
ABSTRACT.....	2
CHAPTER 1 – INTRODUCTION	3
1.1 System Specifications	3
CHAPTER 2 – LITERATURE REVIEW	4
CHAPTER 3 OVERALL DESCRIPTION OF THE PROPOSED SYSTEM.....	5
3.1 Module Description	5
CHAPTER 4 – DESIGN.....	7
4.1UML Diagrams:	7
4.1.1Usecase Diagrams:	8
4.1.2 Sequence Diagram:	8
4.1.3 Collaboration Diagram:	8
4.1.4 Architecture Design	9
4.1.5. Data Flow Diagram	9
4.1.6 Class Diagram	9
4.1.7 Table Design	9
4.1.8 ER Design	9
4.1.9 Activity Design	9
CHAPTER 5 - OUTPUT SCREENSHOTS	9
CHAPTER 6 – IMPLEMENTATION DETAILS	10
6.1 Introduction to Html Framework	10
6.2 Cascading Style Sheets (CSS).....	10
6.3 MYSQL Server	10
6.4PHP	10
6.5 ANGULAR JAVA SCRIPT	10
CHAPTER 8-TECHNICAL FEASIBILITY	12
CHAPTER 9-SYSTEM TESTING.....	14
CHAPTER 10– CONCLUSIONS	18
CHAPTER 11- REFERENCES	18

Project Topic

ABSTRACT

CHAPTER 1 – INTRODUCTION

Objectives

1.1 System Specifications

Hardware Requirements:-

Processor : Intel(R) 2.10GHz

Installed memory (RAM) : 4 GB

Hard Disk : 160 GB

Operating System : Windows (7)

Software Requirements: -

Front-End : HTML, CSS, and Bootstarp

Back-End : Angular JS, PHP, MYSQL

Tool :

Android Emulator

xampp-win32-5.5.19-0-VC11

Android SDK - adt-bundle-windows-x86

CHAPTER 2 – LITERATURE REVIEW

2.1 Existing Solution:

2.2 Proposed Solution:

CHAPTER 3 OVERALL DESCRIPTION OF THE PROPOSED SYSTEM

3.1 Module Description

The system after careful analysis has been identified to be presented with the following modules **User, Administrator and Mechanic.**

User

- **Register** – User has to register their basic details to get access with this application service.
- **Login** – Once they have registered they need to login to avail the service at the needy time.
- **View Details** – Logging in with the application will provide you the lists of mechanics that have the approval of the application.
- **Search records & call** – So that the users can search the mechanic among the list according to their place and time.
- **Post feedback** – After all the process every user has to give their feedback with this application to about their adopted service through this medium.

Administrator

- **Provide Approval** – Administration has to give approval to the registering mechanic after verifying their licensing details for the effective service.

Mechanic

- **Register** – At first every mechanic has to register their details with the admin for getting approval.
- **Login** – Registered mechanics can login their accounts if they got their approval from the admin.
- **Post details** – Here mechanics have to post their details like name, location, services available, etc.
- **View feedback** – Using the feedbacks provided by the users/customers of the service mechanics have to maintain or improve their service.

3.2 System Features

In the life of the software development, problem analysis provides a base for design and development phase. The problem is analyzed so that sufficient matter is provided to design a new system. Large problems are sub-divided into smaller ones to make them understandable and easy for finding solutions. Same in this project all the tasks are sub-divided and categorized.

System Modules

User modules:

- Register
- Login
- View Details
- Search records & call
- Post feedback

Admin modules:

- Provide Approval

Mechanic modules:

- Register
- Login
- Post details
- View feedback

CHAPTER 4 – DESIGN

Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization.

Once the software requirements have been analyzed and specified the software design involves three technical activities - design, coding, implementation and testing that are required to build and verify the software.

The design activities are of main importance in this phase, because in this activity, decisions ultimately affecting the success of the software implementation and its ease of maintenance are made. These decisions have the final bearing upon reliability and maintainability of the system. Design is the only way to accurately translate the customer's requirements into finished software or a system.

Design is the place where quality is fostered in development. Software design is a process through which requirements are translated into a representation of software. Software design is conducted in two steps. Preliminary design is concerned with the transformation of requirements into data.

4.1UML Diagrams:

UML stands for Unified Modeling Language. UML is a language for specifying, visualizing and documenting the system. This is the step while developing any product after analysis. The goal from this is to produce a model of the entities involved in the project which later need to be built. The representation of the entities that are to be used in the product being developed need to be designed.

There are various kinds of methods in software design:

- Use case Diagram
- Sequence Diagram
- Collaboration Diagram

4.1.1 Usecase Diagrams:

Use case diagrams model behavior within a system and helps the developers understand of what the user require. The stick man represents what's called an actor. Use case diagram can be useful for getting an overall view of the system and clarifying that can do and more importantly what they can't do.

Use case diagram consists of use cases and actors and shows the interaction between the use case and actors.

- The purpose is to show the interactions between the use case and actor.
- To represent the system requirements from user's perspective.
- An actor could be the end-user of the system or an external system.

4.1.2 Sequence Diagram:

Sequence diagram and collaboration diagram are called INTERACTION DIAGRAMS. An interaction diagram shows an interaction, consisting of set of objects and their relationship including the messages that may be dispatched among them.

A sequence diagram is an introduction that empathizes the time ordering of messages. Graphically a sequence diagram is a table that shows objects arranged along the X-axis and messages ordered in increasing time along the Y-axis.

4.1.3 Collaboration Diagram:

A **collaboration diagram** is a type of visual presentation that shows how various software objects interact with each other within an overall IT architecture and how users can benefit from this **collaboration**. A **collaboration diagram** often comes in the form of a visual chart that resembles a flow chart.

4.1.4 Architecture Design

4.1.5. Data Flow Diagram

4.1.6 Class Diagram

4.1.7 Table Design

4.1.8 ER Design

4.1.9 Activity Design

CHAPTER 5 - OUTPUT SCREENSHOTS

CHAPTER 6 – IMPLEMENTATION DETAILS

6.1 Introduction to Html Framework

6.2 Cascading Style Sheets (CSS)

6.3 MYSQL Server

6.4PHP

6.5 ANGULAR JAVA SCRIPT

CHAPTER 7- SYSTEM STUDY

7.1 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. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ◆ **ECONOMICAL FEASIBILITY**
- ◆ **TECHNICAL 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.

CHAPTER 8-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. This will lead to high demands 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.

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.

8.1 Non Functional Requirements

Non-functional requirements are the quality requirements that stipulate how well software does what it has to do. These are Quality attributes of any system; these can be seen at the execution of the system and they can also be the part of the system architecture.

8.2 Accuracy:

The system will be accurate and reliable based on the design architecture. If there is any problem in the accuracy then the system will provide alternative ways to solve the problem.

8.3 Usability:

The proposed system will be simple and easy to use by the users. The users will comfort in order to communicate with the system. The user will be provided with an easy interface of the system.

8.4 Accessibility:

The system will be accessible through internet and there should be no any known problem.

8.5 Performance:

The system performance will be at its best when performing the functionality of the system.

8.6 Reliability:

The proposed system will be reliable in all circumstances and if there is any problem that will be affectively handle in the design.

8.7 Security:

The proposed system will be highly secured; every user will be required registration and username/password to use the system. The system will do the proper authorization and authentication of the users based on their types and their requirements. The proposed system will be designed persistently to avoid any misuse of the application.

CHAPTER 9-SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as shown by successfully unit testing, the combination of components is correct and

consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

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 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.

9.1 Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page

9.2 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

9.3 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

CHAPTER 10– CONCLUSIONS

CHAPTER 11- REFERENCES

- Adler M. J. (2009). Harvard Business Review. Health Care Requires Big Changes to Complement New IT, 87, 20.
- Appolinário, F. (2004). Dicionário de Metodologia Científica: um Guia para a Produção do Conhecimento Científico.
- Arnhold, I.; Schmidt, S. & Bohnenberger, M.C. (2008). Seleção e Implantação de ERP em uma Cooperativa Médica no RS: Um Estudo de Caso. In: Encontro Anual da ANPAD. 32, Rio de Janeiro, 2008. Anais... Rio de Janeiro: ANPAD.
- Bahensky, J. A.; Jaana, M. & Ward, M. M. (2008). Health Care Information Technology in Rural America: Electronic Medical Record Adoption Status in Meeting the National Agenda. The Journal of Rural Health.
- Balfour III, D.C. (2009). et al. Health Information Technology — Results From a Roundtable Discussion. JMCP Supplement to Journal of Managed Care Pharmacy. (January/February).

