

ONLINE DIAGNOSTIC LAB MANAGEMENT SYSTEM

Done by

ADWIN GEORGE SAJAN -- Reg no: 210021086886

BHARATH S -- Reg no: 210021086910

Under the guidance of

ELIZABETH POULOSE

In partial fulfillment of requirements for the award of the

Degree of Bachelor of Computer Applications

Of

Mahatma Gandhi University ,Kottayam, Kerala



NIRMALA COLLEGE

(Affiliated to M.G.University,Kottayam)

MUVATTUPUZHA



CERTIFICATE

Certified that this is a bonafide report on the project work entitled

.....

done by.....

*Reg.No.....during the year..... in the partial
fulfillment of the requirements for the award of the degree of Bachelor of
Computer Applications of Mahatma Gandhi University, Kottayam
,Kerala.*

Internal Guide

Head of the Department

Submitted for the Viva-voce Examination held on.....

External Examiner

ACKNOWLEDGEMENT

We give all honor and praise to the Lord who gave wisdom and enabled us to complete our project successfully.

We express our sincere and heartfelt thanks to respected **Prof Dr.Thomas K.V**, our beloved principal for providing necessary facilities for the completion of our project successfully.

We thank profusely **Ms. Preethy George**, Course Coordinator of the Computer Science Department for her guidance and inspiration throughout our course of study.

We express our sincere gratitude to our project guide (Name of Guide) Assistant Professor in the Computer Science Department for the valuable advice and guidance throughout the completion of our project.

We also express our gratitude and thanks to all my teachers and friends for their sincere and friendly cooperation in the successful completion of our project.

CONTENTS	Page No
1. Introduction	
1.1 Objective of the project	
2. System Analysis	
2.1 Existing System	
2.2 Proposed System	
2.3 System Requirement Specification (SRS)	
2.3.1 Hardware specification	
2.3.2 Software specification	
2.3.3 Front End	
2.3.4 Back End	
2.4 Feasibility Analysis	
2.5 Data Flow Diagram	
2.5.1 Introduction to Data Flow Diagram	
3. System Design	
3.1 Input Design	
3.2 Output Design	
3.3 Table Design	
4. System Testing & Implementation	
4.1 System Testing	
4.2 System Implementation	
5. Security Technologies & Policies	
6. Maintenance	

7. Scope for Future Enhancements	
8. Conclusion	
9. Bibliography	
10. Appendix	
10.1 Screen Shots	

1. INTRODUCTION

Online Diagnostic Lab Management System is web-based technology that brings up various diagnosis work online. The Online Diagnostic Centre Management System brings numerous advantages that empower diagnostic centres to deliver efficient, patient-centric care while enhancing operational efficiency and productivity. It enables diagnostic centres to embrace the digital transformation of healthcare management and stay ahead in today's rapidly evolving healthcare landscape.

1.1 OBJECTIVE OF THE PROJECT

1. Patient Registration:

- To allow patients to create accounts and provide their personal information for identification and communication purposes.

2. Test Selection:

- To enable patients to view a comprehensive list of available medical tests and their associated costs for informed decision-making.

3. Booking an Appointment:

- To facilitate the scheduling of appointments for sample collection, providing patients with the flexibility to choose a convenient date and time.

4. Sample Collection:

- To arrange for a lab technician to collect samples from the patient's registered address, ensuring a smooth and efficient sample collection process.

5. Payment:

- To allow patients to make secure online payments for the selected tests, simplifying the financial transaction process.

6. Testing Process:

- To efficiently process and conduct the medical tests in the laboratory to ensure accurate and timely results.

7. Notification of Test Results:

- To notify patients promptly about the completion of their test results, improving communication and reducing wait times.

8. Report Generation and Access:

- To automate the generation of test reports and make them readily accessible to patients through their accounts, enhancing convenience and transparency.

2. SYSTEM ANALYSIS

System analysis is a step-by-step process used to identify and develop or acquire the software need to control the processing of specific application. System analysis is a continuing activity the stages of the systems development. System analysis is the process of gathering and interpreting facts, diagnosing problems and using the facts to improve the system. The outputs from the organization are traced through the various processing that the input phases through in the organization. This involves gathering information and using structured tools for analysis. A detailed study of this process must be made by various techniques like interviews; questionnaires etc.

2.1 EXISTING SYSTEM

In the existing system of diagnostic labs, the process typically involves two separate visits:

1. **Taking the Test:** Initially, patients are required to physically visit the diagnostic lab to undergo the necessary diagnostic test. During this visit, patients are typically guided by lab staff or healthcare professionals who perform the test.
2. **Collecting the Report:** After the test is conducted, patients need to return to the diagnostic lab for a second visit to collect the test report or results. The report contains the findings, interpretations, and any necessary recommendations based on the analysis of the test. During this visit, patients typically interact with lab staff or healthcare professionals to receive their reports.

DISADVANTAGES

1. **Time and Convenience:** Patients have to allocate separate time slots for both visits, which can be time-consuming and inconvenient. It requires making multiple trips to the lab, potentially disrupting daily routines or work schedules.
2. **Travel and Transportation:** Patients need to arrange for transportation to visit the diagnostic lab twice. This can involve additional costs, commuting time, and logistical challenges, especially for individuals with limited access to transportation or living in remote areas.

3. **Waiting Times:** Both visits may involve waiting times at the diagnostic lab. During the first visit, patients may need to wait for their turn to undergo the test. Similarly, during the second visit, patients may experience wait times before their report is ready for collection.
 4. **Coordination and Communication:** Patients need to ensure effective communication and coordination between the diagnostic lab and themselves for both visits.
 5. **Delayed Access to Results:** The need for a separate visit to collect the report introduces a potential delay in accessing the diagnostic results. Patients may have to wait for the report to be processed and made available before they can obtain critical information about their health status.
- .

2.2 PROPOSED SYSTEM

The patients are first allowed to register on the website and log in using registered details. Once registered with their address and contact details, the patients may now see a variety of tests conducted by the lab along with their costs. The patient will select the required test and book an appointment after that the lab centre send the lab boy a registered address to collect a sample. The cost of the test will be paid to the lab at the time of booking through online. After successful testing, the user now gets a notification of the test result. The system allows the admin to attach a copy of the report to the system and automatically show it on the user side so that, the user can download the report.

ADVANTAGES

1. **Streamlined Operations:** The system automates various processes, such as appointment scheduling, patient registration, test ordering, result reporting, reducing manual paperwork and streamlining operations. It eliminates time-consuming tasks, allowing staff to focus on providing quality healthcare services.
2. **Enhanced Efficiency:** With intelligent scheduling algorithms, the system optimizes appointment slots, reducing patient wait times and maximizing the utilization of resources. It also automates test result reporting, minimizing delays and ensuring timely access to crucial information.
3. **Improved Patient Experience:** Patients can easily book appointments online, eliminating the need for physical visits or long phone queues. They can access their test results securely through the system, saving time and improving convenience. The system also provides

automated reminders and notifications, keeping patients informed about their appointments and results.

4. **Real-Time Analytics and Reporting:** The system offers comprehensive analytics and reporting capabilities, providing healthcare providers with real-time insights into various aspects of their diagnostic centre operations. This data-driven approach enables informed decision-making, resource allocation, and strategic planning to improve efficiency and quality of care.

2.3 SYSTEM REQUIREMENT SPECIFICATION

A software requirements specification (SRS) is a comprehensive description of the intended purpose and environment for software under development. The SRS fully describes what the software will do and how it will be expected to perform. An SRS minimizes the time and effort required by developers to achieve desired goals and also minimizes the development cost. A good SRS defines how an application will interact with system hardware, other programs and human users in a wide variety of real-worked situations.

Problem to be Solved

By providing the students with relevant information at hand, we have made this portal more accessible and informative. This system helps user from usual server error problems and make result checking, revaluation and other purposes faster and easier.

Customer requirements

- The system should be fast.
- User friendly
- Maintaining security of data
- Efficiency in data retrieval and management

What the developer needs to know?

- Must know the existing system and its drawbacks.
- Must know what will be needed in the proposed system.

Business Requirements

The system should be feasible both to the developer and client. It should be effective and should be able to complete in time. Developer should be responsible for developing the system, install

the software and update the software whenever necessary, conducting any user training that may be needed for using the system.

User Requirements

The user requirement(s) specification is a document usually that specifies the requirements the user expects from software to be constructed in a software project.

Administrator has overall control in the system.

- Admin has overall control in the system.
- Provide test result.
- Faster processing.

Functional Requirements

Functional requirements define what a system is supposed to do. The system should perform the following functionalities.

- Login – Login of users
- Registration – User can register with the system.
- Update profile – User and Staff can update their profile.
- Test details=Admin can add new Test details.
- View status – Admin can view status of the new bookings.
- Logout – System users can logout from panel.

ADMIN

- Login- Admin can be login by using his login id and password.
- Staff Registration-Admin can register and remove Staff from the System
- Test-Admin can add or remove test from the system.
- Test Request- Admin can view test requests and assign an employee to collect the sample.
- Test Result- Admin can set their test result.
- Logout- Admin can logout this page

USER

- login- User can login to the page by entering their corresponding details and register.
- Test Booking-User can book their test from the system.
- Assignment-User can view which Staff is assigned to them and contact them when need arises.

- Test result- The user can view the result from the view booking page.
- Logout-User can logout from this page.

STAFF

- Login-Staff can login to the page by entering their corresponding details.
- Assignment-Staff can view which user is assigned to them and act accordingly.
- Delay-Staff report to the User if there is a delay in sample collection.
- Logout-Staff can logout from this page.

System Requirements

2.3.1 Hardware Specifications

Processor	: Intel core i3 or higher Processor
Speed	: 3.0GHz or higher
System bus	: 64bits
Memory	: 8GB RAM or Higher
Hard disk	: 40GB or Higher
Monitor	: 14" LCD Monitor
Keyboard	: 104 keys
Pointing Device	: Two or Three Button Mouse

2.3.2 Software Specifications

Operating System	: Windows
Front End	: PHP, HTML, CSS
Scripting Language	: Java script
Back End	: SQL Server Web Server WAMP 2.4
Browser	: Google Chrome

2.3.3 Front end

PHP

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. Originally created by Rasmus Lerdorf in 1994, the PHP reference implementation is now produced by The PHP Group. PHP originally stood for Personal Home Page, but it now stands for the recursive backronym PHP: Hypertext Preprocessors.

PHP code may be embedded into HTML code, or it can be used in combination with various web template systems, Web content management system and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications.

HTML

Hypertext Markup Language (HTML), the languages of the World Wide Web (WWW), allows users to produce Web pages that include text, graphics and pointer to other Web pages (Hyperlinks). HTML is not a programming language, but it is an application of ISO Standard 8879, SGML (Standard Generalized Markup Language), but specialized to hypertext and adapted to the Web. The idea behind Hypertext is that instead of reading text in rigid linear structure, we can easily jump from one point to another point. We can navigate through the information based on our interest and preference. A markup language is simply a series of elements, each delimited with special characters that define how text or other items enclosed within the elements should be displayed. Hyperlinks are underlined or emphasized words that load to other documents or some portions of the same document.

CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a mark-up language. Although most often used to set the visual style of

web pages and user interfaces written in HTML and XHTML. The language can be applied to any XML document, including plain XML. SVG and XUL and is applicable to rendering in speech, or on other media.

Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

CSS is designed primarily to enable the separation of document content from document presentation, including aspects such as the layout, colours, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification or presentation characteristics, enable multiple HTML pages to share formatting by specifying on all platforms except Windows. MySQL ships with no GUI tools to administer MySQL databases or manage data contained within the databases. Users may use the included command line tools or install MySQL Workbench via a separate download. Many third party GUI tools are also available.

JAVASCRIPT

JavaScript is a script-based programming language that was developed by Netscape Communication Corporation. JavaScript was originally called Live Script and renamed as JavaScript to indicate its relationship with Java. JavaScript supports the development of both client and server components of Web-based applications. On the client side, it can be used to write programs that are executed by a Web browser within the context of a Web page. On the server side, it can be used to write Web server programs that can process information submitted by a Web browser and then update the browser's display accordingly. Even though JavaScript supports both client and server Web programming, we prefer JavaScript at Client-side programming since most of the browsers supports it. JavaScript is almost as easy to learn as HTML, and JavaScript statements can be included in HTML documents by enclosing the statements between a pair of scripting tags,

`<SCRIPTS>...</SCRIPT>.`

`<SCRIPT LANGUAGE = "JavaScript">`

JavaScript statements

</SCRIPT>

2.3.4 Back end

MYSQL

MySQL server is powerful database, and it requires limited programs and used has back end. It supports GUI and more application is developed by help this server. Collection of tables which holds the data is called database. A beginner can create their own database by click home page. ships with no GUI tools to administer MySQL databases or manage data contained within the databases. Users may use the included command line tools or install MySQL Workbench via a separate download. Many third-party GUI tools are also available.

2.4 FEASIBILITY ANALYSIS

A feasibility study is an evaluation and analysis of the potential of the proposed project which is based on extensive investigation and research to give full comfort to the decision makers. Feasibility studies aim to objectively and rationally uncover the strength and weakness of existing business of proposed venture, opportunities and threads as presented by the environment, the resources required to carry through, and ultimately the process for success. In its simplest terms, the two criteria to judge feasibility are cost required and value to attain. As such, a well-designed feasibility study should provide a historical background of the business or project, description of the product or service, accounting statements, details of the operations and management, marketing research and policies, financial data, legal requirements and tax obligations.

The four aspects in the feasibility study are:

- Technical feasibility
- Economic feasibility
- Operational feasibility
- Behavioural feasibility

Technical Feasibility

The technical feasibility centres on the existing system and what extend it can support the proposed addition. The technical feasibility assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. The minimum requirements of the system are met by average user. The developer system has á modest technical requirement as only minimal or null changes are required for implementing system.

Normally associated with the technical feasibility includes:

- Development risk
- Resource availability
- Technology

The proposed system can work without any additional hardware or software support other than the computer system and networks. So, I analysed that the proposed system is much more technically feasible than other systems when comparing with the benefits of the new system.

Economic Feasibility

Economic feasibility analysis is also known as cost/benefit analysis. The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. The proposed system reduces the operating cost in terms of time by automating the process. This system is economically feasible.

Operational Feasibility

Operational feasibility is a measure of how well a proposed system solves the problems and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

Behavioural Feasibility

People are inherently resistant to changes and computer is known for facilitating the chances.

An estimate should be made to how strongly the users react towards the development of the system. The proposed system consumes less time. Thus, the people are made to engage in some other important work.

2.5 DATA FLOW DIAGRAM (DFD)

2.5.1 Introduction to data flow diagram

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. It differs from the flowchart as it shows the data flow instead of the control flow of the program. A data flow diagram can also be used for the visualization of data processing (structured design).

Data flow diagrams were invented by Larry Constantine, the original developer of structured design based on Martin and Estrin's "data flow graph" model of computation.

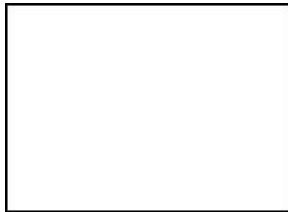
Data flow diagrams (DFDs) are one of the three essential perspectives of Structured System Analysis and Design Method SSADM. The sponsor of a project and the end users will need to be briefed and consulted throughout all stages of a system's evolution. With a data flow diagram, users can visualize how the system will operate, what the system will accomplish, and how the system will be implemented. The old system's data flow diagrams can be drawn up and compared with the new system's data flow diagrams to draw comparisons to implement a more efficient system. Data flow diagrams can be used to provide the end user with physical idea of where the data they input ultimately has an effect upon the structure of the whole system from order to dispatch to report. How any system is developed can be determined through a data flow diagram.

Developing a data flow diagram helps in identifying the transaction data in the data model. There are different notations to draw data flow diagrams, defining different visual representation for process, data stores, data flow, and external entities. The first step is to draw a data flow diagram (DFD). A DFD also known as "bubble chart" has the purpose of clarifying system requirements and identifying major transformation that will become program in system design. So, it is starting point of the design phase that functionally decompose the requirements

specification down to the lowest level of details DFD consists of series of bubbles joined by lines. The bubbles represent data transformation and the lines represent data flow in the system.

DFD Symbols: -

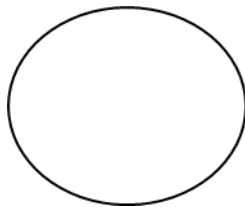
- Square- Defines source or destination of system.



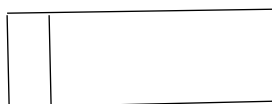
- Data flow - Identifies data flow Circle.

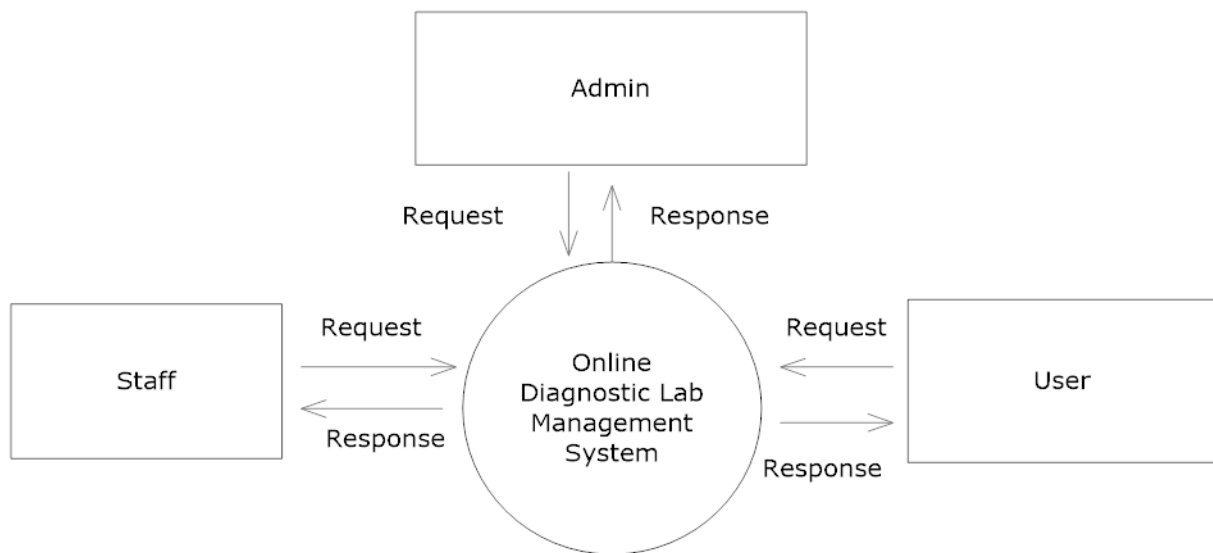


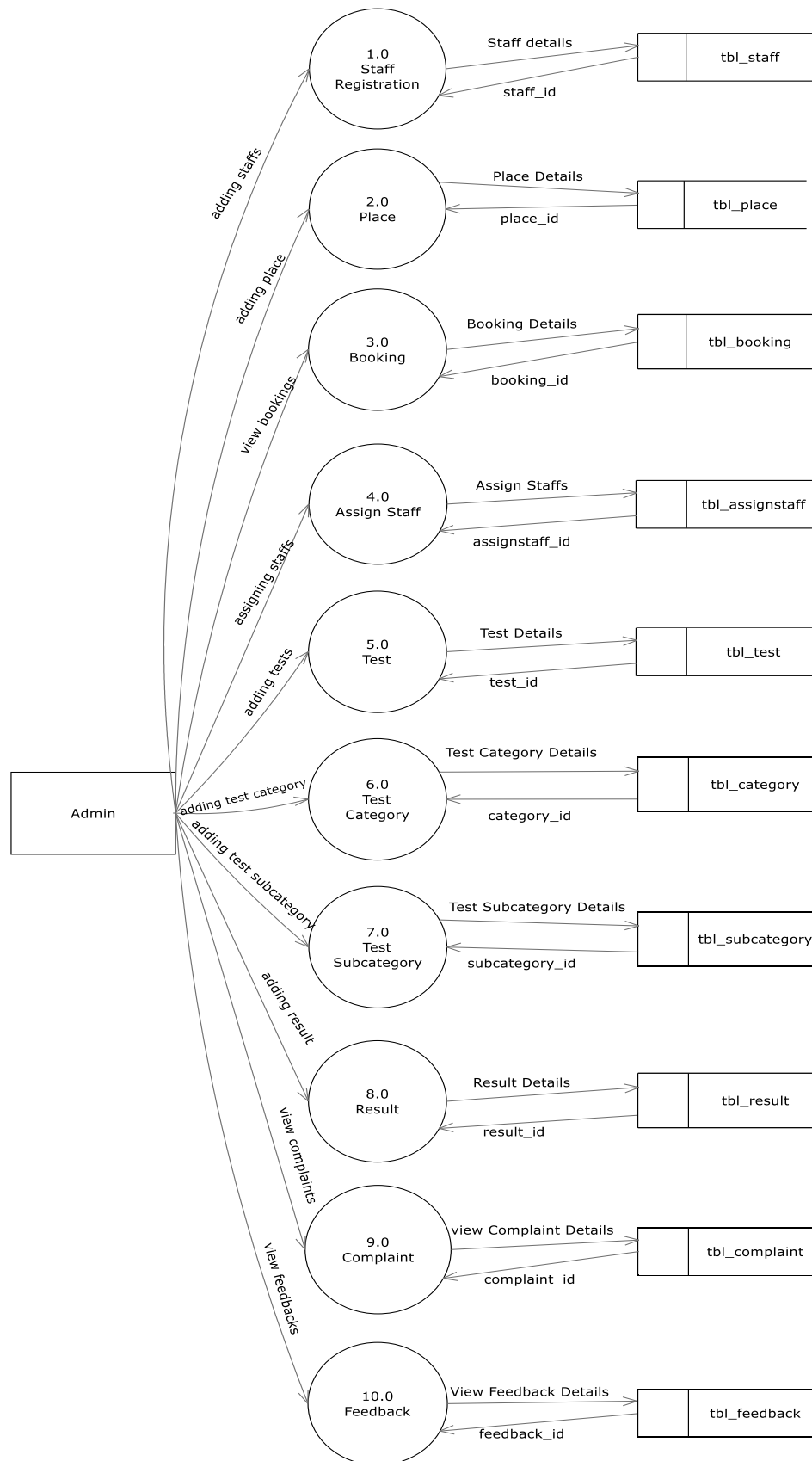
- Bubble - Represents a process that transforms incoming data to outgoing data.

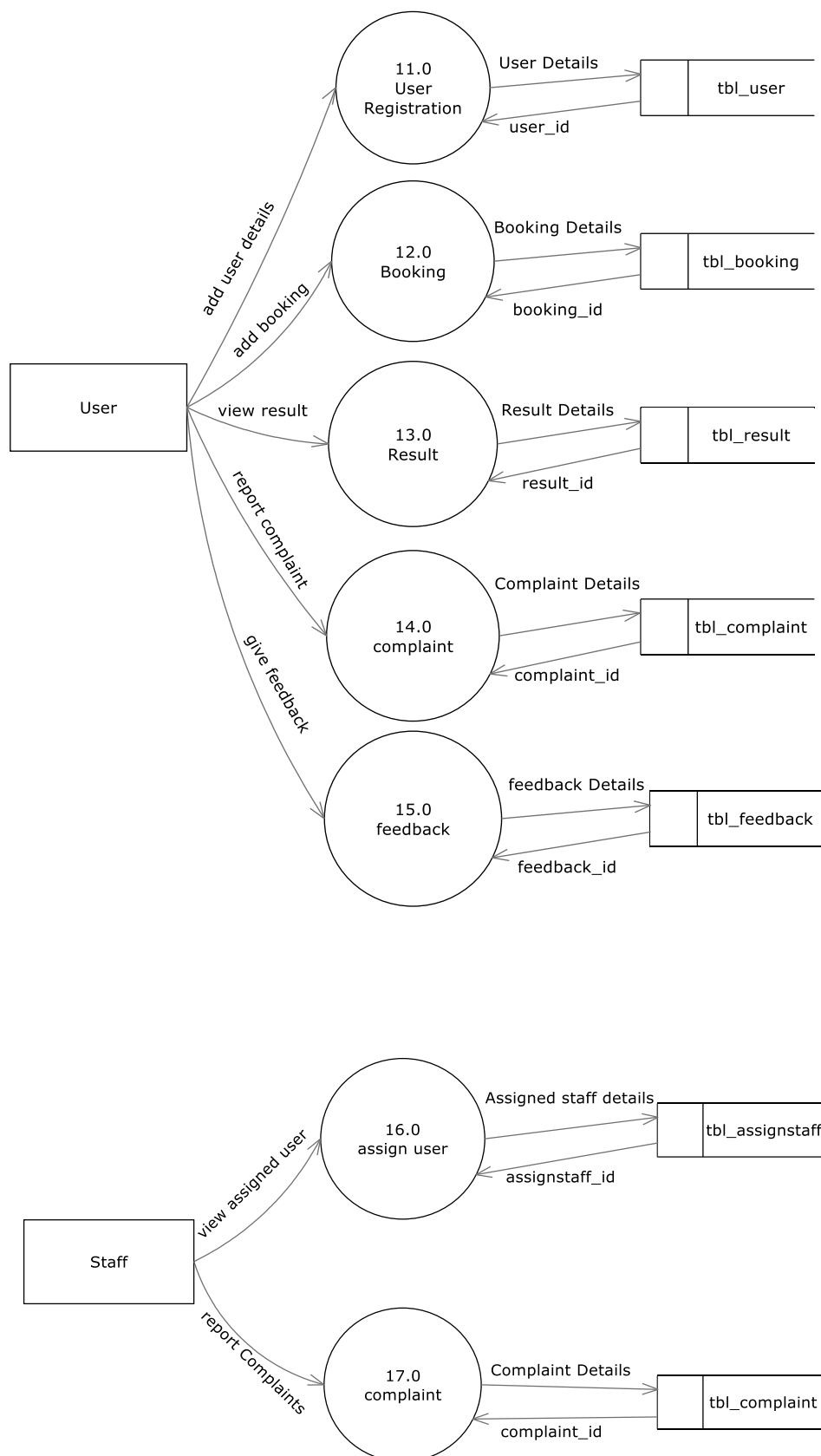


- Open rectangle- Data store



LEVEL0

LEVEL1



3. SYSTEM DESIGN

3.1 INPUT DESIGN

The quality of the system input determines the quality of the system output. Input specification describes the way data enter the system for processing. Input design features can ensure the reliability of the system and produce result from accurate data, or they can result in the production of erroneous information. The input design also determines whether the user can interact efficiently with the system.

In our system almost all inputs are being taken from the databases. To provide adequate inputs we have to select necessary values from the databases and arrange it to the appropriate controls.

Admin

Admin is the one who controls the whole system. The administrator is the super user of this application. Only admin have access into this admin page. Admin can access the page using their login id and password. Admin can add and remove test Details. Only admin can publish test result.

User

User can create their profile and upload their details. The user can request for test, also they can view that status of those bookings. And the main thing is user will also get the test result.

Staff

The staff can request for Time Delay in Sample collection if the schedule is busy. Staff can also update the status of booking.

3.2 OUTPUT DESIGN

One of the important features of an information system for users is the output produces. Output is the information delivered to users through the information system. Without quality of the output, the entire system appears to be unnecessary that users will avoid using it. Users

generally merit the system solely by its output. In order to create the most useful output possible. One works closely with the user through an interactive process. Until the result is considered to be satisfactory.

Admin

Admin can view the details of all the peoples who applied for test. Admin can add or remove test details. Admin can upload their test result. User can view the result.

User

User can view details of the staff that is assigned to them. Also, the user can view the test result. User can register complaints and give feedback.

Staff

Staff can view details of the user that they are assigned to. Also, the staff can register complaints.

3.3 TABLE DESIGN

The data design transforms the information domain model created during analysis into the data structures that will be required to implement the software. The data objects and relationships defined in the entity relationship diagram and the detailed data content depicted in the data dictionary provide the basis for the data design activity.

The overall objective in the development of database technology has been to treat data as an organizational resource and as an integrated whole. Database Management System allows data to be protected and organized separately from other resources. Database is a integrated collection of data. This is the difference between logical and physical data.

The organization of data in the database aims to achieve three major objectives:

- Data integration
- Data integrity
- Data independence

The databases are implemented using a DBMS package. Each DBMS has unique characteristics and general techniques for database design. There are 6 major steps in design process. The first 5 steps are usually done on paper and finally the design is implemented.

- Identify the table and relationships.
- Identify the data that is needed for each table and relationship.
- Resolve the relationship.
- Verify the design.
- Implement the design.

The database uses tables for storage. A table also contains records, which is a set of fields. All records, in a table have the same set of fields with different information. Uses 3 tables.

Each table contains key fields that establish relationships in the database and how the records are stored. There are primary key fields that uniquely identify a record in a table. There are also fields that contain the primary key from another table called foreign keys.

The various database tables that are used in this project are the following:

1. Table name: tbl_admin

Description: Admin Login Details

Primary Key: admin_id

Field	Type	Size	Description
admin_id	int		Admin id
admin_name	varchar	50	Admin name
admin_email	varchar	50	Admin email
admin_password	varchar	50	Admin password

Admin_contact	varchar	50	Admin contact number
---------------	---------	----	----------------------

2. Table name: tbl_category**Description: Test Category Details****Primary Key: category_id**

Field	Type	Size	Description
category_id	int		Category id of Test
category_name	varchar	50	Category name of test

3. Table name: tbl_subcategory**Description: Test Subcategory Details****Primary Key: subcategory_id****Foreign Key: category_id**

Field	Type	Size	Description
subcategory_id	int		Subcategory id of Test
subcategory_name	varchar	50	Subcategory name of Test
category_id	int		Category of test

4. Table name: tbl_test**Description: Test Details****Primary Key: test_id****Foreign Key: subcategory_id**

Field	Type	Size	Description
test_id	int		Test id
test_name	varchar	50	Test name
test_price	int		Test price

subcategory_id	int		Subcategory of test
----------------	-----	--	---------------------

5. **Table name:** tbl_place
Description: Place Details
Primary Key: place_id

Field	Type	Size	Description
place_id	int		Place id
Place_name	varchar	50	Place name

6. **Table name:** tbl_booking
Description: Booking Details
Primary Key: booking_id
Foreign Key: place_id, test_id

Field	Type	Size	Description
booking_id	int		Booking id
booking_time	varchar	50	Booking Time
Booking_date	varchar	50	Booking Date
user_id	int		User id
place_id	int		Place id
test_id	int		Test id
Booking_status	varchar	50	Status of booking
Booked_datetime	varchar	100	Current Date and Time

7. **Table name:** tbl_user
Description: user details
Primary Key: user_id
Foreign Key: place_id

Field	Type	Size	Description
user_id	int		User id

user_name	varchar	50	User name
user_address	varchar	100	Address of user
user_contact	varchar	50	User contact number
user_age	varchar	50	User age
user_gender	varchar	50	User gender
user_proof	varchar	100	Proof of user (Aadhar card or anything else)
user_photo	varchar	100	User photo
user_email	varchar	50	User email
user_password	varchar	50	User password
place_id	int		place id

- 8. Table name: tbl_result**
Description: Result Details
Primary Key: result_id
Foreign Key: booking_id

Field	Type	Size	Description
result_id	int		Result id
result_content	varchar	50	Result content
booking_id	int		Booking id

- 9. Table name: tbl_complaint**
Description: Complaint Details
Primary Key: compliant_id
Foreign Key: user_id, staff_id

Field	Type	Size	Description
-------	------	------	-------------

complaint_id	int		Complaint id
complaint_title	varchar	50	Complaint Title
complaint_content	varchar	200	Complaint Content
complaint_date	varchar	50	Complaint Date
complaint_reply	varchar	50	Complaint Reply
complaint_status	varchar	50	Status of Complaint
user_id	int		User id
staff_id	int		Staff id

10. Table name: tbl_staff**Description: Staff Details****Primary Key: staff_id****Foreign Key: place_id**

Field	Type	Size	Description
staff_id	int		Staff Id
staff_name	varchar	50	Staff Name
staff_address	varchar	50	Staff Address
staff_proof	varchar	100	Staff Proof
staff_contact	varchar	50	Staff Contact Number
Staff_age	varchar	50	Staff Age
Staff_gender	varchar	50	Staff Gender
staff_email	varchar	50	Staff Email
staff_password	varchar	50	Staff Password
staff_photo	varchar	100	Staff Photo
place_id	int		Place of Staff Operation

11. Table name: tbl_assignstaff**Description: Assigning Staffs****Primary Key: assignstaff_id****Foreign Key: booking_id,staff_id**

Field	Type	Size	Description
assignstaff_id	int		Assign staff id
booking_id	int		Id of Booking
staff_id	int		Staff id

12. Table name: tbl_feedback
Primary Key: feedback_id
Foreign Key: user_id

Field	Type	Size	Description
feedback_id	int		Feedback id
feedback_content	varchar	50	Feedback content
user_id	int		Id of user

4. SYSTEM IMPLEMENTATION AND TESTING

4.1 SYSTEM TESTING

Testing is the process of examining the software to compare the actual behaviour with that of the expected behaviour. The major goal of software testing is to demonstrate that faults are not present. In order to achieve this goal, the tester executes the program with the intent of finding errors. Though testing cannot show absence of errors but by not showing their presence it is considered that these are not present.

System testing is defined as the process by which one detects the defects in the software. Any software development organization or team has to perform several processes. Software testing is one among them. It is the final opportunity of any programmer to detect and rectify any defects that may have appeared during the software development stage. Testing is a process of testing a program with the explicit intention of finding errors that makes the program fail. In short system testing and quality assurance is a review in software products and related documentation for completion, correctness, reliability and maintainability.

System testing is the first stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct and the goal will be successfully achieved. A series of testing are performed for the proposed system before the proposed system is ready for user acceptance testing.

The testing steps are,

- Unit testing
- Integration testing
- Acceptance Testing
- Validation
- Output testing

System Testing provides the file assurance that software once validated must combined with all other system elements. System testing verifies whether all elements have been combined properly and that overall system function and performance is achieved. For the integration of modules, the validation test was carried out over the system. It was that all the modules work well together and meet the overall system function and performance.

1. Unit Testing

Unit testing is carried out screen-wise, each screen being identified as an object. Attention is diverted to individual modules, independently to one another to locate errors. This has enabled the detection of errors in coding and logic.

Various test cases are prepared. For each module these test cases are implemented, and it is checked whether the module is executed as per the requirements and outputs the desired result. In this test each service input and output parameters are checked.

In unit testing

- Module interface was tested to ensure that information properly flows into and out of the program under test.
- Boundary condition was tested to ensure that module operates properly at boundaries established to limit or restrict processing.
- All independent paths through the control structures were executed to ensure that all statements in the modules have been executed at least once.
- Error handling paths were also tested.

2. Integration Testing

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing.

Unit tested module were taken and a single program structure was built that has been dictated by the design. Incremental integration has been adopted here.

The modules are tested separately for accuracy and modules are integrated too. Then, using bottom-up integration i.e., by integrating from moving from bottom to the top of the system is checked and errors found during integration are rectified. In this testing individual modules were combined and the module wise Shifting was verified to be alright.

The entire software was developed and tested in small segments, where errors were easy to locate and rectify. Program builds (group of modules) were constructed corresponding to the successful testing of user interaction, data manipulation analysis, and display processing and database management.

3. Validation Testing

Validation testing is done to ensure complete assembly of the error-free software. Validation can be termed successful only if it functions in manner. Reasonably expected by the student under validation is alpha and beta testing. The student-side validation is done in this testing phase. It is checked whether the data passed to each student is valid or not. Entering incorrect values does the validation testing and it is checked whether the errors are being considered. Incorrect values are to be discarded. The errors are rectified.

In "Health Lab" verifications are done correctly. So, there is no chance for users to enter incorrect values. It will give error messages by using different validations. The validation testing is done very clearly and found it is error free.

4. Output Testing

After performing the validation testing the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in a specific format.

The output format on the screen was found to be correct as the format was designed in the system design phase according to the user needs. For the hard copy also, the output comes out as specified requirement by the user. Hence output testing does not result in any Correction in the system.

Output This project is developed based on the user choice. It is user friendly. The output format is very clear to user. Output testing is done on Smart builders correctly.

5. Acceptance testing

Acceptance involves running a suite of tests on the completed system. Each individual test, known as a Case, exercise particular operating condition of the operating condition of the user's environment or feature of the system, and will result in a pass fail, or Boolean outcome.

4.2 SYSTEM IMPLEMENTATION

The implementation is the final state, and it is an important phase. It involves the invalid programming system testing. user training and the operational running of developed proposed system that constitutes the application subsystems. A major task of preparing for implementation is education of users. which should really have been taken place much carrier in the project when they were belong involved in the investigation and design work. During the

implementation phase system take physical shape. In order to develop a system implemented planning is very essential.

The implementation phase of the software development is concerned with translating design specification into source code. The user tests the developed system and changes are made according to their needs. Our system has been successfully implemented.

Before implementation several tests have been conducted to ensure that no errors are encountered during the operation. The implementation phase ends with an evaluation of the system after placing into the operation for a period of time.

The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from old system to new system. The system can be implemented only after testing is done and is found to be working to specifications. The implementation stage is a systems project in its own right.

The implementation stage involves following tasks:

- Careful planning.
- Investigation of system and constraints.
- Design of method to achieve change over.
- Evaluation of the changeover method.

In the case of this project all the screens are designed first. For making it to be executable, codes are written on each screen and performs the implementation by creating the database and connecting to the server. After that the system, is Checked, whether it performs all the transactions correctly. Then databases are cleared and made it to be usable to the technicians.

Implementation Plans

The following are the step involved in the implementation plan of "Smart Builders":

- Test system with sample data
- Detection and correction of errors
- Make the necessary changes in the system.
- Check the existing system.
- Installation of hardware and software utilities
- Training and involvement of user personals

5.SECURITY TECHNOLOGIES& POLICIES

The protection of computer-based resources that includes hardware, software, data procedures and people against unauthorized use or natural disaster is known as System Security.

System Security can be divided into four related issues:

- Security
- Integrity
- Privacy
- Confidentiality

SYSTEM SECURITY refers to the technical innovations and procedures applied to the hardware and operation systems to protect against deliberate or accidental damage from a defined threat.

DATA SECURITY is the protection of data from loss, disclosure, modification and destruction.

SYSTEM INTEGRITY refers to the proper functioning of hardware and programs, appropriate physical security and safety against external threats such as eavesdropping and wiretapping.

PRIVACY defines the rights of the user or organizations to determine what information they are willing to share with or accept from others and how the organization can be protected against unwelcome, unfair or excessive dissemination of information about it.

CONFIDENTIALITY is a special status given to sensitive information in a database to minimize the possible invasion of privacy. It is an attribute of information that characterizes its need for protection.

SECURITY IN SOFTWARE System security refers to various validations on data in form of checks and controls to avoid the system from failing. It is always important to ensure that only valid data is entered, and only valid operations are performed on the system.

The system employs two types of checks and controls:

CLIENT-SIDE VALIDATION Various client-side validations are used to ensure on the client side that only valid data is entered. Client-side validation saves server time and load to handle invalid data. Some checks imposed are:

- Forms cannot be submitted without filling up the mandatory data so that manual mistakes of submitting empty fields that are mandatory can be sorted out at the client side to save the server time and load.
- Tab-indexes are set according to the need and taking into account the ease of user while working with the system.

SERVER-SIDE VALIDATION Some checks cannot be applied at client side. Server-side checks are necessary to save the system from failing and intimating the user that some invalid operation has been performed or the performed operation is restricted. Some of the server-side checks imposed is:

- Server-side constraint has been imposed to check for the validity of primary key and foreign key. A primary key value cannot be duplicated. Any attempt to duplicate the primary value results into a message intimating the user about those values through the forms using foreign key can be updated only of the existing foreign key values.
- User is intimating through appropriate messages about the successful operations or exceptions occurring at server side.
- Various Access Control Mechanisms have been built so that one user may not agitate upon another. Access permissions to various types of users are controlled according to the organizational structure. Only permitted users can log on to the system and can have access according to their category. User- name, passwords and permissions are controlled o the server side.
- Using server-side validation, constraints on several restricted operations are imposed.

6. MAINTENANCE

Software maintenance is the modification of a software product and delivery to correct faults, to improve performance or other attributes. Maintenance is the ease with which a program can be corrected if any error is encountered, adapted if its environment changes or enhanced if the customer desires a change in requirement. Maintenance follows conversation to extend that

changes are necessary to maintain satisfactory operations relative to changes in the user's environment.

Maintenance often includes minor enhancements or corrections to problems that surface in the system's operation. Maintenance is also done based on fixing the problems reported, changing the interface with other software or hardware enhancing the software.

CATEGORIES OF MAINTENANCE

Corrective Maintenance

Corrective maintenance is the most used maintenance approach, but it is easy to see its limitations. When equipment fails, it often leads to downtime in production, and sometimes damages other parts. In most cases, this is expensive. Also, if the equipment needs to be replaced, the cost of replacing it alone can be substantial. Reliability of systems maintained by this type of maintenance is unknown and cannot be measured. Corrective maintenance is possible since the consequences of failure or wearing out are not significant and the cost of this maintenance is not great.

Perfective Maintenance

Modification of a software product after delivery to improve performance or maintainability. This term is used to describe changes undertaken to expand the existing requirements of the system. A successful piece of software tends to be subjected to the Succession of changes resulting in an increase in its requirements. This is based on the premise that as the software becomes useful, the user experiments with new cases beyond the scope for which it was initially developed. Expansion requirements can take the form of enhancement of existing system functionality and improvement in computational efficiency.

Adaptive Maintenance

Modification of a software product performed after delivery to keep a product usable, changed or changing environment. Adaptive maintenance includes any work initiated because of moving the software to a different hardware or software platform. It is a change driven by the need to accommodate modifications in the environment of the software system. The environment

in this context refers to the totality of all conditions and influences which act from outside upon the system. A change to the whole or part of this environment will Warrant a corresponding modification of the software.

Preventive Maintenance

Preventive maintenance is a schedule of planned maintenance actions aimed at the prevention of breakdowns and failures. The primary goal of preventive maintenance is to prevent the failure of equipment before it occurs. It is designed to preserve and enhance equipment reliability by replacing worn components before they fail. Preventive maintenance activities include equipment checks, partial or complete overhauls at specified periods.

Long-term benefits of preventive maintenance include:

- Improved system reliability.
- Decreased cost of replacement.
- Decreased system downtime.

7. SCOPE FOR FUTURE ENHANCEMENT

The system has been designed in such a way that it can be modified with very little effort when such needs arise in the future. New features can be added with slight modifications of software which make it easy to expand the scope of this project. Though the system is working on various assumptions, it can be modified easily to any kind of requirements. The system is also expected to be improvised by adding various features. Now the system provided the information about the test booking for elderly people which includes only blood testing, but in the future, we hope to expand this to various other tests including vehicle services if the test needs to be done in lab and implementing mobile app for seamless and smooth working.

8. CONCLUSION

In conclusion, the online medical testing and reporting system presented in this project offers a streamlined and user-friendly approach to healthcare service delivery. By enabling patients to register, book tests, and access their results conveniently, it addresses the evolving needs of both healthcare providers and patients.

This system represents a significant step towards the digitization of healthcare services, enhancing accessibility and efficiency. However, there is ample room for improvement and expansion. Future enhancements may include the integration of emerging technologies, such as

telemedicine, wearable health devices, and advanced data analytics, to provide more comprehensive and proactive healthcare solutions. Additionally, ensuring data security, regulatory compliance, and quality assurance will be crucial in maintaining patient trust and system reliability.

In the ever-evolving landscape of healthcare technology, this system has the potential to grow and adapt, ultimately offering an even more comprehensive and patient-centric healthcare experience. As the project continues to develop and incorporate advancements in the field, it can play a pivotal role in transforming how healthcare services are accessed and delivered.

9. BIBLIOGRAPHY

1. Jalotte, Pankaj. " **SOFTWARE ENGINEERING**", Third Edition, Spring Publishers, Narosa Publishing House.
2. Ramakrishnan, Raghu. and_ Gehrke, Johannes. **DATABASE MANAGEMENT SYSTEM**", Third Edition, McGraw-Hill Professional Publication.
3. Awad, Elias M. "**SYSTEM ANALYSIS AND DESIGN**", Second Edition, McGraw-Hill Professional publication.
4. Rob, Peter. and Coronel, Carlos. "**DATABASE SYSTEMS DESIGN AND MANAGEMENT**", Fifth Edition, Thomson Learning Publication.
5. Pressman, Roger S. "**SOFTWARE ENGINEERING**", Fifth Edition, McGraw-Hill Publication.
6. Bayross, Ivan. "**SQL/MySQL**" Second Revised Edition, BPB Publications

10. APPENDIX

10.1 SCREEN SHOTS

