Software Requirements Specification

for

Pathology Referral System

Version 1.0

Client - **Rajesh Kumar Mishra** (9471192497)

rajesh@iitism.ac.in

By **WeCare Innovations** (Group 34)

Ankit Kachhi (18JE0127)

Basant Kumar (18JE0213)

Rashmikiran Pandit (18JE0667)

Ray Apoorva Nath (17JE003324)

Table Of Contents

1. Introduction

The software HealthISM is to be developed for making the pathology referring entirely online. Through HealthISM, different operations can be done efficiently and in a user-friendly way.

1.1 Purpose

This SRS defines the External Interface, Performance, and Software System Attribute requirements of HealthISM. This document is intended for the following group of people:-

- Developers for the purpose of maintenance and new releases of the software.
- Management of HealthISM.
- Documentation writers.
- Testers

1.2 Scope

The scope of the HealthISM is to create an online web -interface for the student/staff and health center staff where students/employee can view their medical records and can upload their tests report whereas health center staff will have full access they can add/delete/modify users tests and add/delete new labs contract. This will be an alternative to the traditional offline way of keeping records and reduces time to search particular. Apart from giving these features to the users and employees, it creates full analytics of how many tests are still going on and the total expenditure of the health center on these tests.

1.3 Definitions, Acronyms and Abbreviations

HC	Health Centre
MIS	Management Information System
SRS	Software Requirement Specification
UI	User Interface
DFD	Data Flow Diagram

1.4 References

- The following standards apply: IEEE STD 830 -1998, IEEE Standard for Software Requirement Specification

1.5 Overview

- Section 1.0 discusses the purpose and scope of the software.
- Section 2.0 describes the overall functionalities and constraints of the software and user characteristics.

- Details all the requirements needed to design the software.

2. Overall Description

2.1 Product Perspective

The motive of the project is to build an online platform for managing the activities of the health center of IIT(ISM) related to pathology and radiology. The software will be integrated with the MIS for easy access.

The vision is to digitally transform the manual management of patient records and provide accurate information related to pathology labs and tests in a lucid and convenient way to the users.

2.2 Product Functions

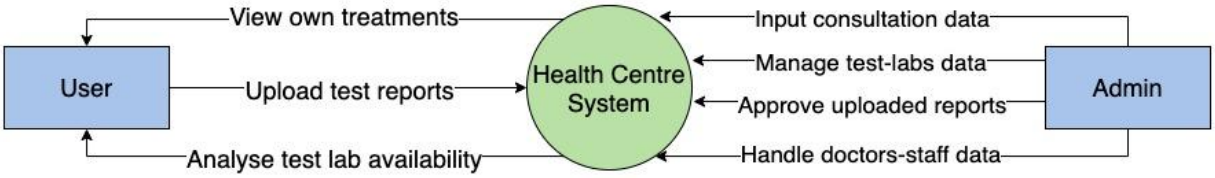
The primary functionalities of HealthISM are described as follows:-

➢ **Medical consultation record**: Whenever a patient visits the doctor at HC, the details of the consultation will be recorded by an official of the HC in the database through our platform. The details will include information regarding the symptoms, referred tests, date of visit etc. This allows HC to efficiently record and keep track of patients' ongoing and past treatments.

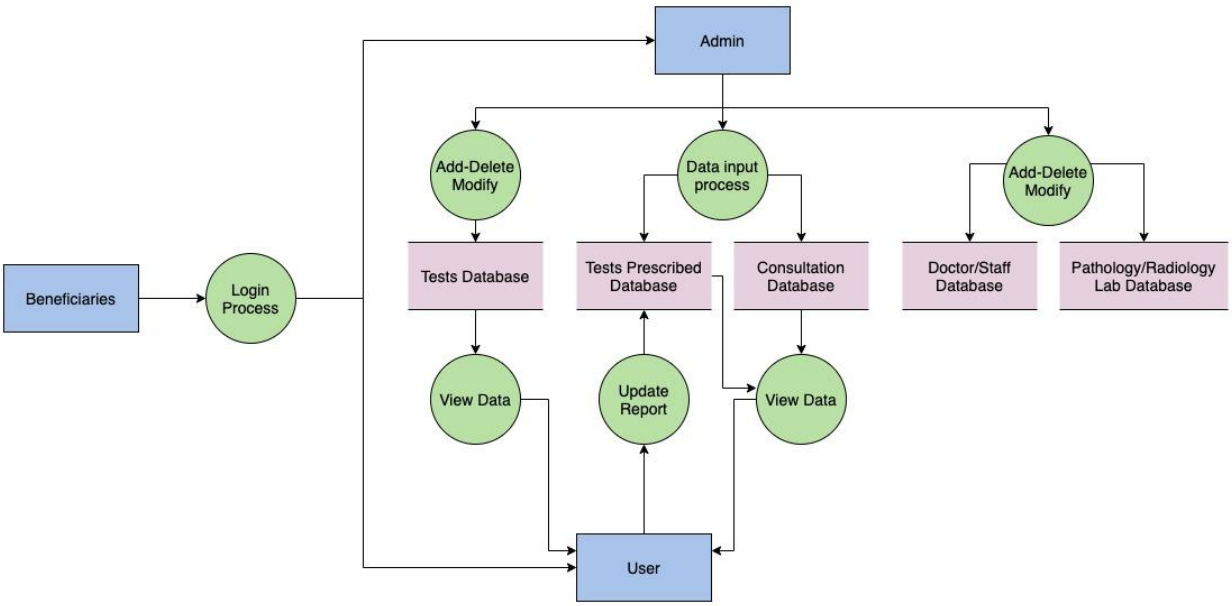
➢ **Patient's log book:** Upon successful login by the user, he/she can access all the past and ongoing treatment records along with the test prescribed and their respective statuses. The information is presented in a tabular format wherein users can click on the test prescribed and can view all the related lab centres that conduct that test against their prices. Upon taking the medical test in the lab, the user has to upload the outcome/result or report of the same. The report can be of any format from numbers, text to images and documents.

➢ **Manage lab centre's information:** In a scenario where there are multiple labs performing multiple tests such that a test can be conducted in more than one lab, the online tool will provide easy access to right information at a click of a button. Adding/deleting and modifying databases will become more efficient. As the platform is empowered with strong querying capabilities, information can be easily retrieved from the system. Analysis of annual statistics, finances, and patient records also becomes much more convenient.

Level 0 DFD



Level 1 DFD



2.3 User Characteristics

Different kinds of users will be interacting with the software. The intended users of the software are as follows:-

- **User:** The authenticated users of IIT(ISM) including employees, faculties, staff and students have the right to access the online portal for Health centre through MIS. They can view the complete history of medications issued, treatments obtained, assigned tests and their statuses. The users are required to upload the reports from their end after undergoing the tests in the lab.

- **Health Center staff:** This user has all the administrative permissions. The admin is solely responsible for managing the various entities and their information. The powers exercised by the admin are adding, modifying, deleting and updating records, querying information from the present data and performing timely analysis. They can even aid users to upload the reports on their behalf in case users are unable to do so by themselves.

2.4 Operating Environment

- This software is platform-independent and can run on any operating system. The only necessity to run this software is to have a web browser.

2.5 General Constraints

The general constraints of the project include but are not limited to:-

- This software should use the existing SQL database of the client.
- The software should be developed using LAMP tech-stack.
- This software should use CodeIgniter Version 2.2.0, PHP Version 5.4.16, and XAMPP compatible with PHP Version 5.
- The authentication and user ids will be processed based on the existing user database.

2.6 Assumption and Dependencies

- There already exists an MIS system that allows user identification and authentication.
- The beneficiaries of the HC system are registered users of the MIS system.
- The provision is only for the employees of the IIT(ISM) and not their dependents.
- All the prescribed tests are performed by the user and he has the report handy.
- The staff of the HC is familiar with the language 'English' and basic computer operations such as `point-and-click`.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The user interface should be really user-friendly and interactive to use. It should be self understandable and easy to operate by a layman with little or no knowledge of computing and coding.

As the platform will serve both the users and the administration, hence there should be separate user interfaces for the normal users and for the admin. The detailed design of UI is explained in the UI Design document

User-end Interface:

The functionalities needed to be delivered are:

➢ Viewing and updating self-treatments
➢ Plan test by comparing price/location across all the labs
➢ Enquiry facility

Admin-end User Interface:

The functionalities needed to be delivered are:

- ➢ Creating a new record for every medical consultation: This feature enables the admin to add new consultation entries in the database.
- ➢ Viewing and updating user's treatments records: Users exercise uploading of reports after they undergo the prescribed tests. Verification and approval of the uploaded reports shall be done by the admin. Admin beholds the power to upload reports on behalf of users, the same can also be facilitated through this interface.
- ➢ Managing labs and tests data store: Admin can manage the existing laboratories details and their tests using this interface.

  The abilities are:
  - Add a new lab to the list
  - Modify the data of an existing lab
  - Delete an existing lab entry
- ➢ Doctor and other staff's database management : The management of the doctor's database would be controlled through this interface.

3.1.2 Hardware Interfaces

Console input devices – either a keyboard and a mouse or a touch screen display. Data is fed into the system through these and the software processes it and performs the necessary actions. A functional printer if the need for a hard copy arises.

3.1.3 Software Interfaces

On the client-side the required software product is Internet Explorer / Google Chrome / Mozilla Firefox supporting at least HTML version 3.2, java enabled, and any operating system that can run the browsers.

3.1.4 Communication Interfaces

The default communication protocol for data transmission between the server and the client is Transmission Control Protocol/ Internet Protocol (TCP/IP). At the

upper-level Hypertext Transfer Protocol (HTTP) will be used for communication between the web server and client.

3.2 USE CASES

Actors: Students, Staff Members, Admin, Labs.

Main Use-Cases of this system:

1. Login
2. Patient views available tests.
3. Patient upload test report.
4. Patient views test report.
5. Admin views database.
6. Admin upload test report on the behalf of the patient.
7. Admin adds or remove lab.
8. Admin verifies the test rate of the lab.
9. Admin approves test report.

3.2.1 Login :

Primary Actor: Patient(Student,Staff Member),Admin,Lab.

Precondition: Patient has been prompted to sign in/up with valid credentials.

Success Scenarios:

1. Patient logs in using a validated user id and password.

Exception Scenarios:

1. Incorrect user id or password is used.

3.2.2 Patient views available tests :

Primary Actor: Student, Staff Member.

Precondition: Patient has logged in.

Success Scenarios:

1. Tests should be present in any of the labs.

2. The rate of the test should be verified by the admin.

3.2.3 Patient upload test report :

Primary Actor: Student, Staff Member.

Precondition: Patient has logged in and test details have been uploaded by labs.

Success Scenarios:

1. The patient sees their test details.
2. The patient uploads the report.

Exception Scenarios:

1. The lab has not updated test details.
2. The report has not been uploaded by the patient.

3.2.4 Patient view test report :

Primary Actor: Student, Staff Member.

Precondition: Patient has logged in and test details have been uploaded by labs.

Success Scenarios:

1. Patient sees their test details.

3.2.5 Admin views database :

Primary Actor: Admin.

Precondition: Admin has logged in.

Success Scenarios:

1. Admin sees all the tables of the database.

3.2.6 Admin upload test report on the behalf of the patient :

Primary Actor: Admin.

Precondition: Admin has logged in and the report has not been uploaded by the patient.

Success Scenarios:

1. Admin uploads the report.

Exception Scenarios:

1. Test details are not present in the database.

3.2.7 Admin add or remove lab :

Primary Actor: Admin.

Precondition: Admin has logged in.

Success Scenarios:

1. A new lab applied for services.
2. Test rates are in the given limit then add this lab.
3. Labs are not following guidelines then remove it.

3.2.8 Admin verifies test rate of the lab :

Primary Actor: Admin.

Precondition: Admin has logged in.

Success Scenarios:

1. The lab adds a test in their tests list.
2. If the test rate is within the given limit then add it.
3. If the test rate is out of the limit then remove it.

3.2.9 Admin approves test report :

Primary Actor: Admin.

Precondition:Admin has logged in and the patient has uploaded a report.

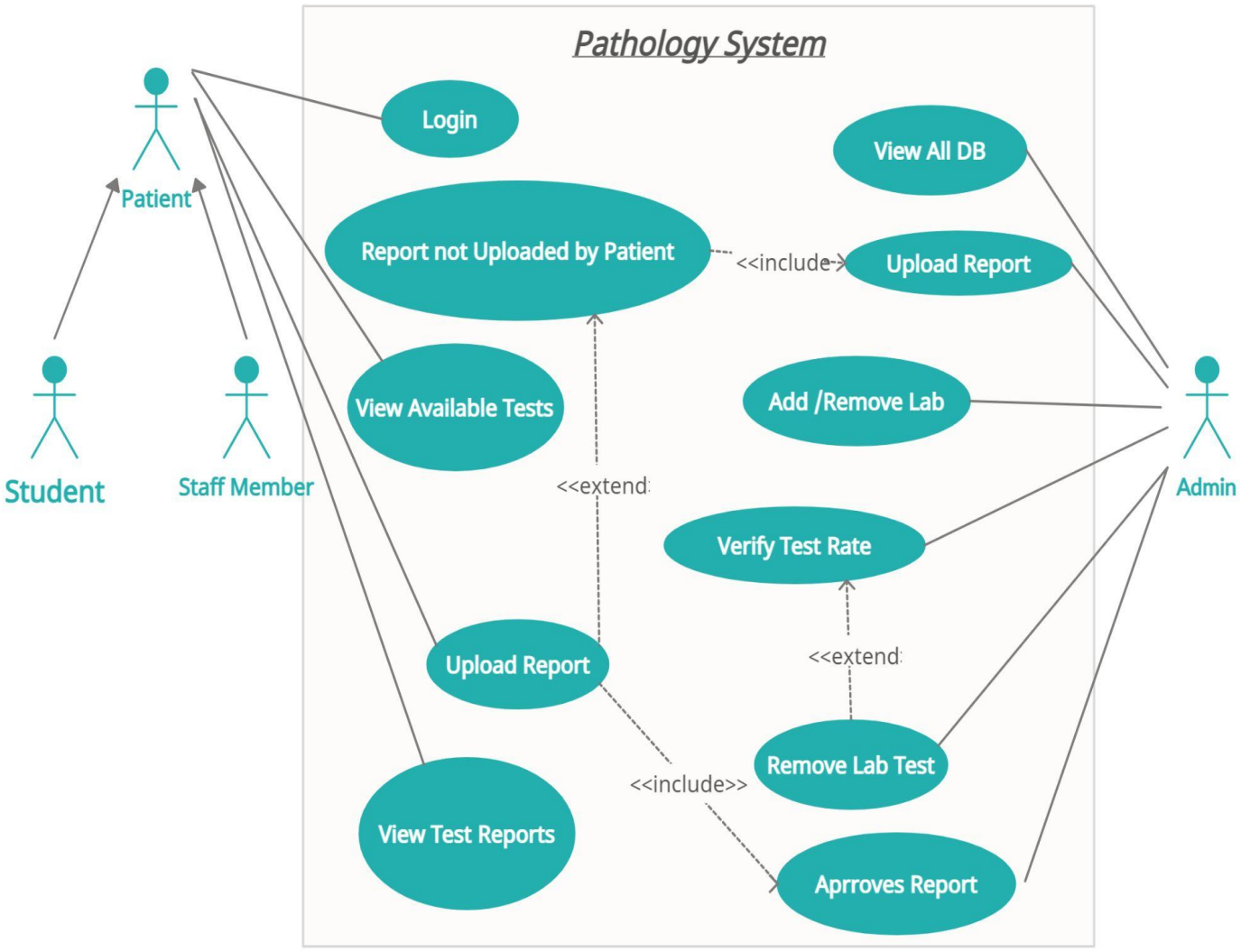
Success Scenarios:

1. Report is correct and authentic thus approved by the admin.

Exception Scenarios:

1. Uploaded report is falsy.
2. Report is not uploaded indefinitely.

3.2.10 Use case Diagram :



3.3 Database Requirements

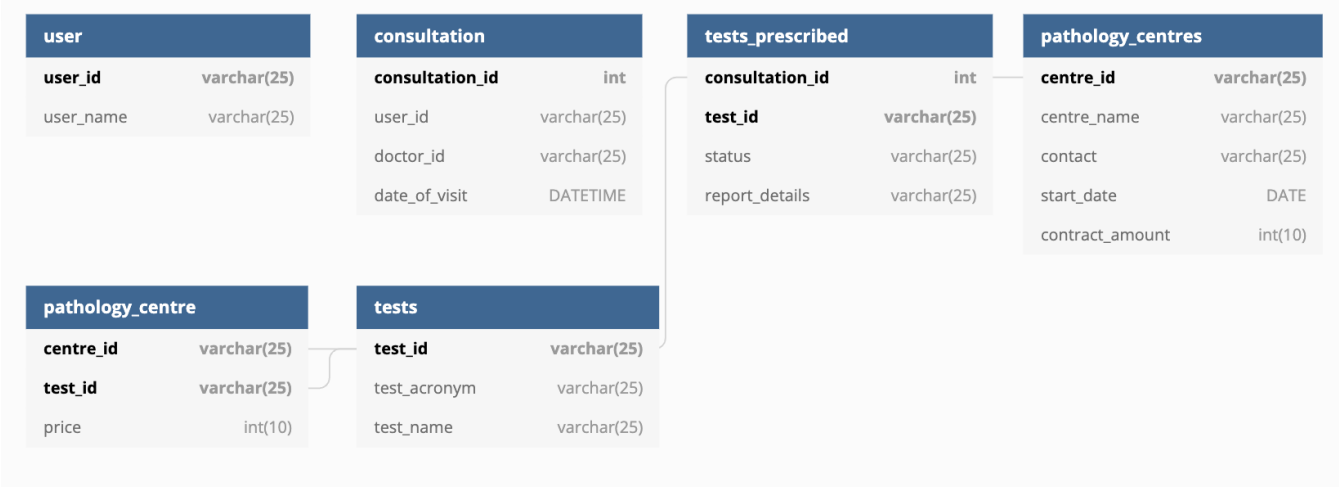
Keeping in view the existing database and the new requirements, below is the proposed system of required database tables.

A. Existing database
   a. User - user id, user name

B. New database requirements
   a. Doctors - doctor id, doctor name, specialty, salary, joining date, etc..
   b. Consultation - consultation id, user id, doctor id, date.
   c. Tests prescribed - consultation id , test id , status, report details.

d. Pathology centres - centre id, centre name, start date, contact details, contract amount.
e. Pathology centre - center id, test id, price.
f. Tests - test id, test acronym, test name.

The database tables will interoperate as described in the diagram below.



4. Other Nonfunctional Requirements

4.1 Performance Requirements

- Reliability and Accuracy: The records details generated by the software must be accurate and precise to prevent any impending chaos.
- Good response time: The system should have an acceptable response time depending on the user connection speed.
- User friendly: The system should interact and should display all the instructions clearly as it will result in a higher yield.
- Flexibility: The number of labs available, tests conducted and the prices of each test can be subject to frequent changes. The system should be flexible to be able to adapt to such changes.
- Capacity: The software should be available 24*7.

4.2 Design Constraints

- The primary tech stack will revolve around LAMP technology.
- This page can be accessed only through MIS and only authorized users can operate the system.
- The existing structure of the database of the users has to be incorporated in the new platform.
- The new features should be easy to integrate and interoperable with the existing login system.

4.3 Software System Attributes

- Maintainability: The software should be easy to maintain and should support the changes required by the admin.
- Security: The software should be secure and should display only relevant information to each user.
- Interoperability: The software should be able to coexist with other MIS features and should not hamper their work.
- Portability: The software should be able to run in a different environment.
- Understandability: The software should be easy to understand by the common users

4.4 Other Requirements

- None