CHRIST
(DEEMED TO BE UNIVERSITY)
BANGALORE · INDIA

Title of Project

by

StudentName
(Roll Number)

Under the guidance of

Guide Name &

Designation

A Project report submitted in partial fulfillment of the requirements for the award of degree of Bachelor of Computer Applications of CHRIST (Deemed to be University)

April - 2024

CERTIFICATE

This is to certify that the report titled ***Project Title*** *is a bona fide record of work done by* ***Student Name (Roll Number)*** *of CHRIST (Deemed to be University), Bengaluru, in partial fulfillment of the requirements of 4th Semester BCA during the year 2023-24.*

Head of the Department **Project Guide**

Valued-by:

1. Name :
   Register Number :
   Examination Centre : CHRIST (Deemed to be University)
2. Date of Exam :

ACKNOWLEDGEMENT

ABSTRACT

TABLE OF CONTENTS

ABBREVIATIONS

DFD: - Data Flow Diagram

HMS: - Hospital Management System

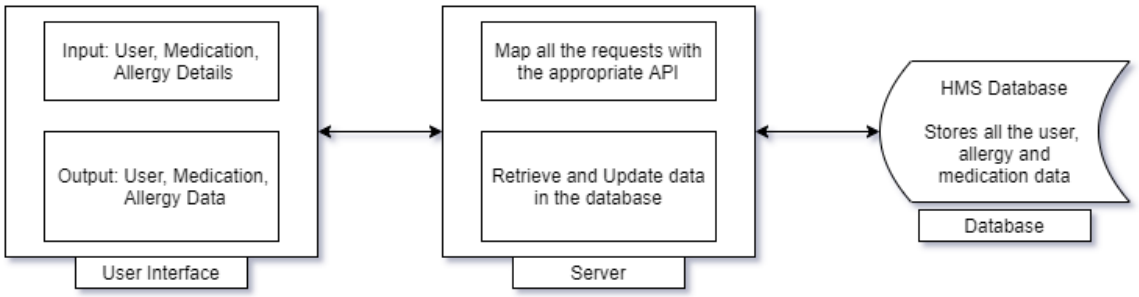

Fig. 2.1 Block diagram

3.2.1 Data Flow Diagram (DFD)

Data flow diagram helps in understanding the flow of data of a system or a process is represented by DFD. It also gives insight into the inputs and outputs of each entity and the process itself. DFD does not have control flow and no loops or decision rules are present. Specific operations depending on the type of data can be explained by a flowchart. Data Flow Diagrams can be represented in several ways. The DFD belongs to structured-analysis modelling tools. Data Flow diagrams are very popular because they help us to visualize the major steps and data involved in software-system processes.

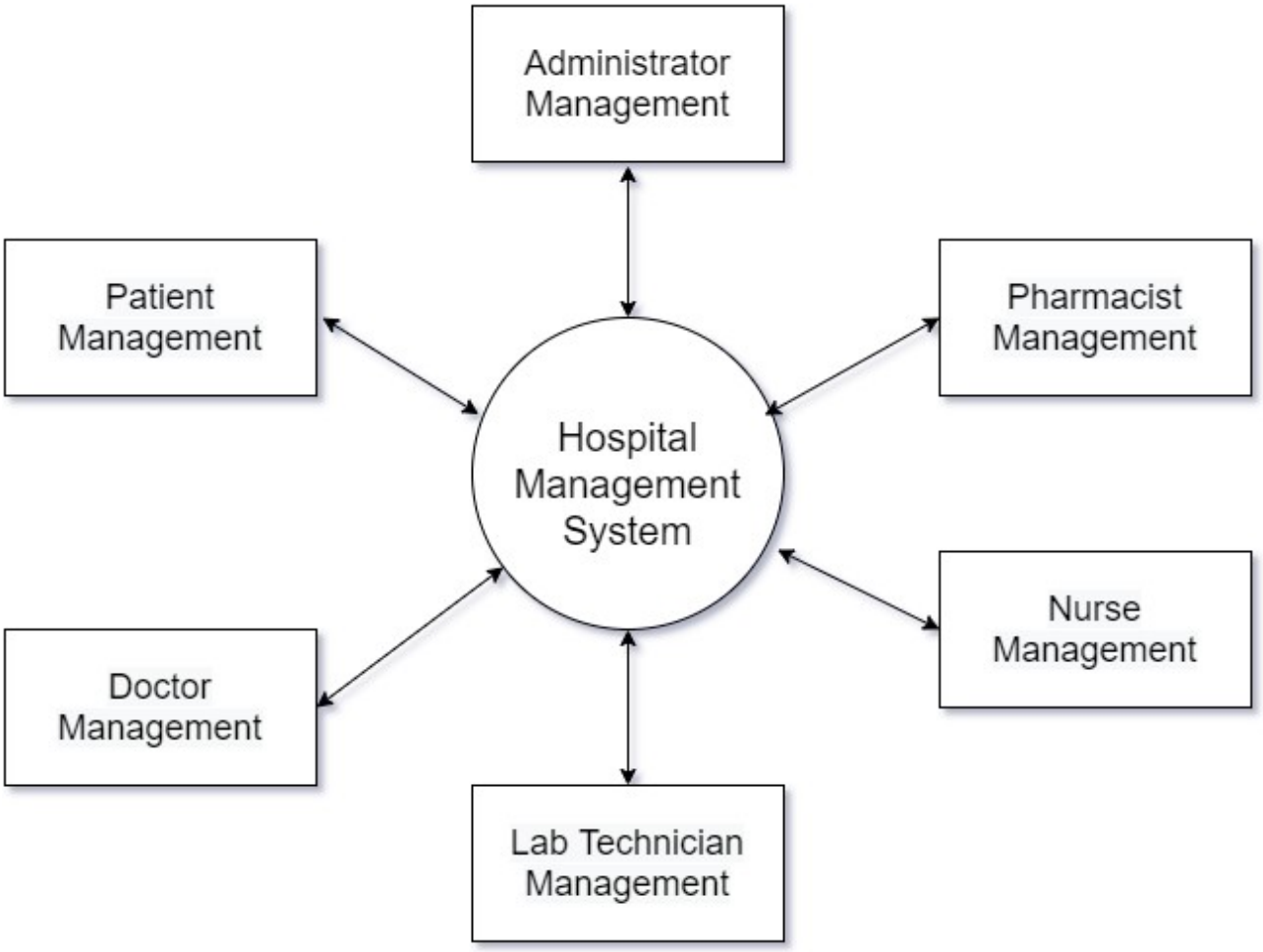

Fig 3.1 Level - 0

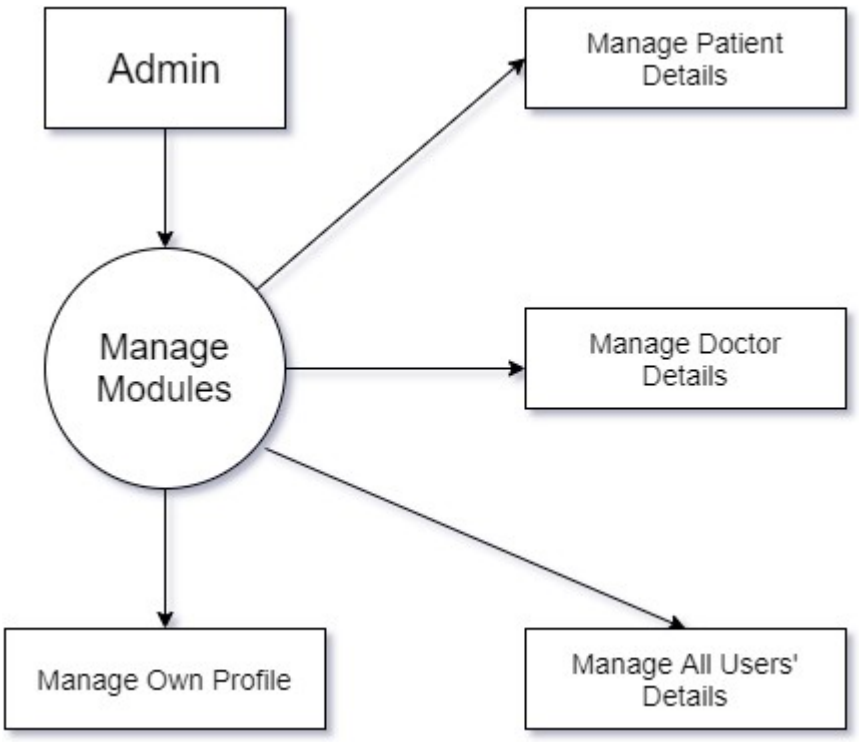

Fig 3.2 Level - 1, Admin

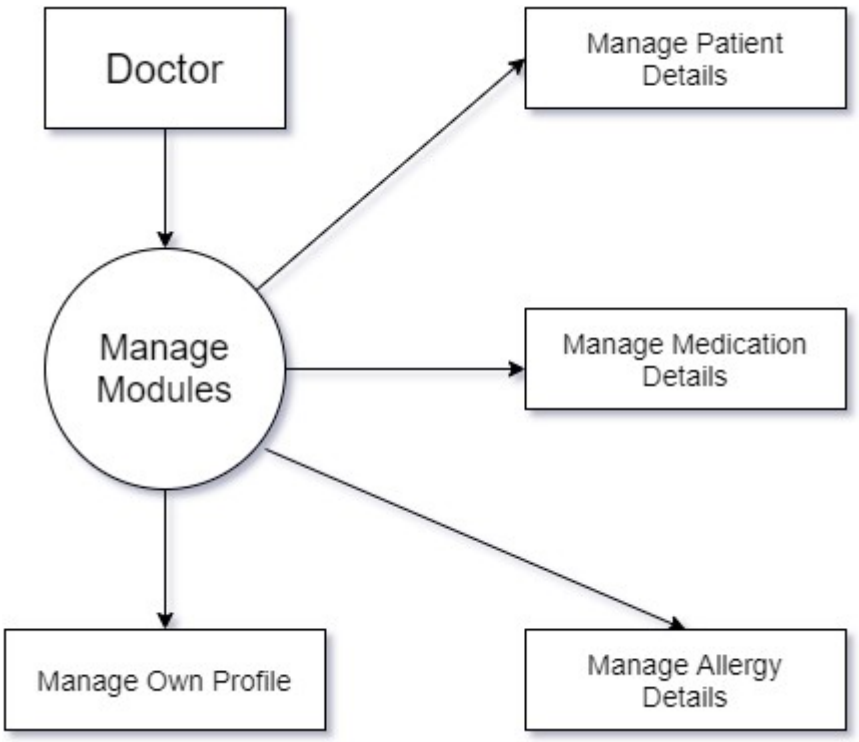

Fig 3.2 Level - 1, Doctor

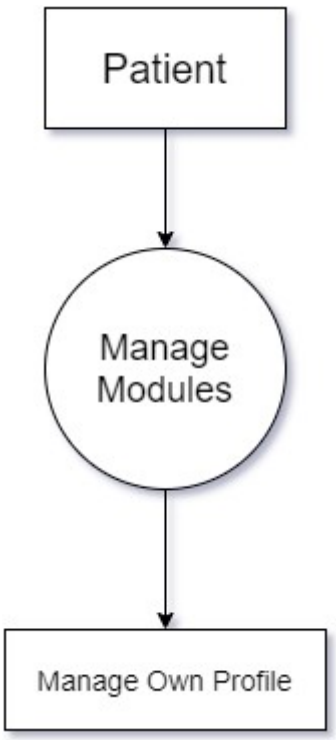

Fig 3.3 Level - 1, Patient

TESTING

This section discusses the various methods used for testing the modules and their results.

5.1. TEST CASES

All the modules have been unit tested with a Code Coverage of 100% and Path coverage of 95%.

Also, Integration tests for the APIs are also written and all the tests have passed Successfully.

5.2. TESTING APPROACHES

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation.

A test strategy is an outline that describes the testing approach of the software development cycle. It is created to inform project managers, testers, and developers about some key issues of the testing process. This includes the testing objective, methods of testing new functions, total time and resources required for the object, and the testing environment.

Test strategies describe how the product risks of the stakeholders are mitigated at the test- level, which types of test are to be performed, and which entry and exit criteria apply. System design documents are primarily used and occasionally conceptual design documents may be referred to. For every stage of development design, a corresponding test strategy should be created to test the new feature sets.

Software testing can be stated as the process of validating and verifying that a software product for the following:

i. Meeting the requirements that guided its design and development.
ii. Works as expected.
iii. Is it possible to be implemented with the same characteristics

Software testing, depending on the testing method employed, can be implemented at any time in the development process. However, most of the test effort occurs after the requirements have been defined and the coding process has been completed. As such, the methodology of the test is governed by the software development methodology adopted. Following are the types of testing that are utilized popularly in the software industry.

1. White Box Testing

It is conducted when the tester has access to the internal data structures and algorithms including the code that implements these. White-box testing methods can also be used to evaluate the completeness of a test suite that was created with black-box testing methods. This allows the software team to examine parts of a system that are rarely tested and ensures that the most important function points have been tested.

Two common forms of code coverage are:

i. Function coverage, which reports on functions executed.
ii. Statement coverage, which reports on the number of lines executed to complete the test.

They both return code coverage metric, measured as percentage.

2. Black-Box Testing

It treats the software as a "black box" without any knowledge of internal implementation. Black-box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, fuzz testing, model-based testing,

exploratory testing and specification- based testing.

3. System Testing

System testing is very vital. While integrating the system there might be certain errors in the system. The error needs to be detected and eliminated otherwise the system performance may go down drastically. The entire system when ready must be thoroughly checked for with the system requirement specification. Once this has been done the majority of the work in the phase is over.

4. Stress Testing

Stress testing implies testing the system under extreme conditions. I have made sure that the system works properly no matter how the user inputs data.

5. Condition Testing

Condition testing is a test case design method that exercises the logical condition contained in a program module.

This testing method focuses on testing each condition in the program. The aim of conditional testing is not only to locate error to locate in the condition but also to locate the error in the program.

6. Data Flow Testing

This method selects the test path of a program according to the location of the definition and uses of variables in the program. The flow of the data or the variable from one module to another has been checked.

6. CONCLUSION

REFERENCES

1. https://www.cerner.com/
2. https://getbootstrap.com/
3. https://spring.io/projects/spring-framework
4. https://spring.io/projects/spring-boot