A

PROJECT REPORT

ON

**“Healthify - A Complete Medicare Solution”**

Towards partial fulfilment of the requirement in

**4th Semester I-MCA (2022-2023)**

**Submitted By: -**

Garvit Bucha (210510314001-IMCA-CSF)

Kartik Gupta (210510312003-IMCA-AI)

**Submitted To: -**



**Parul Institute of Computer Application,**

**Parul University**

Under the guidance of

**Prof. Manish Joshi,** Assistant Professor

**Acknowledgement**

*The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of our project. All that we have done is only due to such supervision and assistance and we would not forget to thank them.*

*We respect and thank* ***Dr Priya Swaminarayan****, Dean, FITCS for providing us an opportunity to do the project work in BCA and giving us all support and guidance, which made us complete the project duly. We are extremely thankful to Ma’am for providing her support and guidance, although she had busy schedule managing the academic affairs.*

*We would not forget to remember* ***Prof. Hina Chokshi****, HOD, BCA department for her encouragement and more over for her timely support and guidance till the completion of our project work.*

*We owe our deep gratitude to our project guide* ***Prof. Manish Joshi****, who took keen interest on our project work and guided us all along, till the completion of our project work by providing all the necessary information for developing a good system.*

*We are thankful to and fortunate enough to get constant encouragement, support and guidance from our Parents, all Teaching staffs of BCA Department which helped us in successfully completing our project work. Also, we would like to extend our sincere esteems to all staff in laboratory for their timely support.*

***Garvit Bucha (210510314001-IMCA-CSF)***

***Kartik Gupta (210510312003-IMCA-AI)***



PARUL INSTITUTE OF COMPUTER APPLICATION

**CERTIFICATE**

This is to certify that ***Garvit Bucha and Kartik Gupta*** the student(s) of Parul Institute of Computer Application, has/have satisfactorily completed the project entitled “***Healthify - A complete Medicare Solution****”* as a part of course curriculum in IMCA semester-IV for the academic year 2022-2023 under guidance of ***Prof. Manish Joshi.***

Enrolment Number: 210510314001

Enrolment Number: 210510312003

|  |  |  |
| --- | --- | --- |
| **Quality of work** | **Grade** | **Sign of Internal Guide** |
| **Poor / Average / Good /**  **Excellent** | **B / B+ / A / A+** |  |

Date of submission:

HOD, Principal,

**Prof. Hina Chokshi** **Dr Priya Swaminarayan**

**INDEX**

|  |  |
| --- | --- |
| Content | Page No. |
| 1. Research | 1 |
| 1. Feasibility Studies | 2 |
| * 1. Technical Feasibility | 2 |
| * 1. Economic Feasibility | 2 |
| * 1. Operational Feasibility | 2 |
| 1. System Requirement Specification | 4 |
| * 1. Introduction to SRS | 4 |
| * 1. Abstract | 4 |
| * 1. System Users | 4 |
| * 1. Modules | 5 |
| * 1. Modules Description | 5 |
| * 1. Hardware / Software Requirement | 5 |
| * 1. Flow Chart | 6 |
| * 1. TimeLine Chart | 7 |
| 1. Technology Description | 8 |
| * 1. Features and Limitations of New System | 8 |
| 1. Data Flow Diagram | 9 |
| * 1. Context Level DFD’s | 9 |
| * 1. Level 1 DFD’s | 9 |
| * 1. Level 2 DFD’s | 10 |
|  |  |
| 1. Use Case Diagram | 12 |
| 1. Class Diagram | 14 |
| 1. Activity Diagram | 15 |
| * 1. Description of Activity Diagram | 17 |
| 1. E-R Diagram | 18 |
| * 1. E-R Diagram Description | 18 |
| 1. Data Dictionary | 19 |
| * 1. Description of Data Dictionary | 21 |
| 1. Form Design (Screenshots Phase 1, 2, 3, 4 and Validation’s Screenshots) | 22 |
| 1. What is Testing? | 28 |
| * 1. Importance and Types of Testing | 29 |
| 1. Future Enhancement | 30 |
| 1. References and Bibliography | 31 |

1. **Research**
   1. **What is research?**

Research is a process of systematic inquiry that entails collection of data; documentation of critical information; and analysis and interpretation of that data/information, in accordance with suitable methodologies set by specific professional fields and academic disciplines.

Research is the careful consideration of study regarding a particular concern or problem using scientific methods. According to the American sociologist Earl Robert Babbie, “research is a systematic inquiry to describe, explain, predict, and control the observed phenomenon. It involves inductive and deductive methods.”

Inductive methods analyse an observed event, while deductive methods verify the observed event. Inductive approaches are associated with [qualitative research](https://www.questionpro.com/blog/qualitative-research-methods/), and deductive methods are more commonly associated with quantitative analysis.

* 1. **Types of Research Methodology**

1. **Qualitative methods**

[Qualitative research](https://www.questionpro.com/blog/qualitative-research-methods/) is a method that collects data using conversational methods, usually [open-ended questions](https://www.questionpro.com/blog/what-are-open-ended-questions/). The responses collected are essentially non-numerical. This method helps a researcher understand what participants think and why they think in a particular way.

1. **Quantitative methods**

[Quantitative](https://www.questionpro.com/blog/quantitative-research/) methods deal with numbers and measurable [forms](https://www.questionpro.com/forms.html). It uses a systematic way of investigating events or data. It answers questions to justify relationships with measurable variables to either explain, predict, or control a phenomenon.

1. **Feasibility Studies**

**What is Feasibility?**

A feasibility study evaluates a project's or system's practicality. As part of a feasibility study, the objective and rational analysis of a potential business or venture is conducted to determine its strengths and weaknesses, potential opportunities and threats, resources required to carry out, and ultimate success prospects. Two criteria should be considered when judging feasibility: the required cost and expected value.

* 1. **Technical Feasibility**

This assessment focuses on the technical resources available to the organization. It helps organizations determine whether the technical resources meet capacity and whether the technical team can convert the ideas into working systems. Technical feasibility also involves the evaluation of the hardware, software, and other technical requirements of the proposed system.

* 1. **Economic Feasibility**

This assessment typically involves a cost/ benefits analysis of the project, helping organizations determine the viability, cost, and benefits associated with a project before financial resources are allocated.

* 1. **Operational Feasibility**

This assessment involves undertaking a study to analyse and determine whether—and how well—the organization’s needs can be met by completing the project. Operational feasibility studies also examine how a project plan satisfies the requirements identified in the requirements analysis phase of system development.

* 1. **Importance of Feasibility Studies**

There are several benefits to feasibility studies, including helping project managers discern the pros and cons of undertaking a project before investing a significant amount of time and capital into it.

• Feasibility studies can also provide a company's management team with crucial information that could prevent them from entering into a risky business venture.

• Such studies help companies determine how they will grow. They will know more about how they will operate, what the potential obstacles are, who the competition is, and what the market is.

• Feasibility studies also help convince investors and bankers that investing in a particular project or business is a wise choice.

* 1. **Feasibility Study of our Proposed System**
     1. **Technical Feasibility**:

In our system, this involves evaluating the technical requirements of the application, such as the hardware and software resources required, and determining if they are available and suitable for the project.

* + 1. **Economic Feasibility:**

This involves evaluating the costs associated with the development and deployment of the application, such as the cost of hiring developers, purchasing hardware, and marketing the app.

* + 1. **Operational Feasibility:**

This involves evaluating the operational aspects of the system, including the ability of the system to meet the needs of the users, the availability of personnel to operate and maintain the system, and the ability of the system to integrate with existing processes and procedures.

1. **System Requirement Specification**
   1. **Introduction To SRS**
      1. **What is SRS?**

A software requirements specification (SRS) is a description of a software system to be developed. It lays out functional and non-functional requirements and may include a set of use cases that describe user interactions that the software must provide.

* + 1. **Need of SRS**

In order to fully understand one’s project, it is very important that they come up with a SRS listing out their requirements, how are they going to meet it and how will they complete the project. It helps the team to save upon their time as they can comprehend how are going to go about the project. Doing this also enables the team to find out about the limitations and risks early on.

* 1. **Abstract**

The purpose of the project entitled as “HOSPITAL MANAGEMENT SYSTEM” is to computerize the Front Office Management of Hospital to develop software which is user friendly simple, fast, and cost – effective. It deals with the collection of patient’s information, diagnosis details, etc. Traditionally, it was done manually. The main function of the system is register and store patient details and doctor details and retrieve these details as and when required, and to manipulate these details meaningfully System input contains patient details, diagnosis details, while system output is to get these details on to the screen. The Hospital Management System can be entered using a username and password. It is accessible either by an administrator or receptionist. Only they can add data into the database. The data can be retrieved easily. The data are well protected for personal use and makes the data processing very fast.

* 1. **System Users**

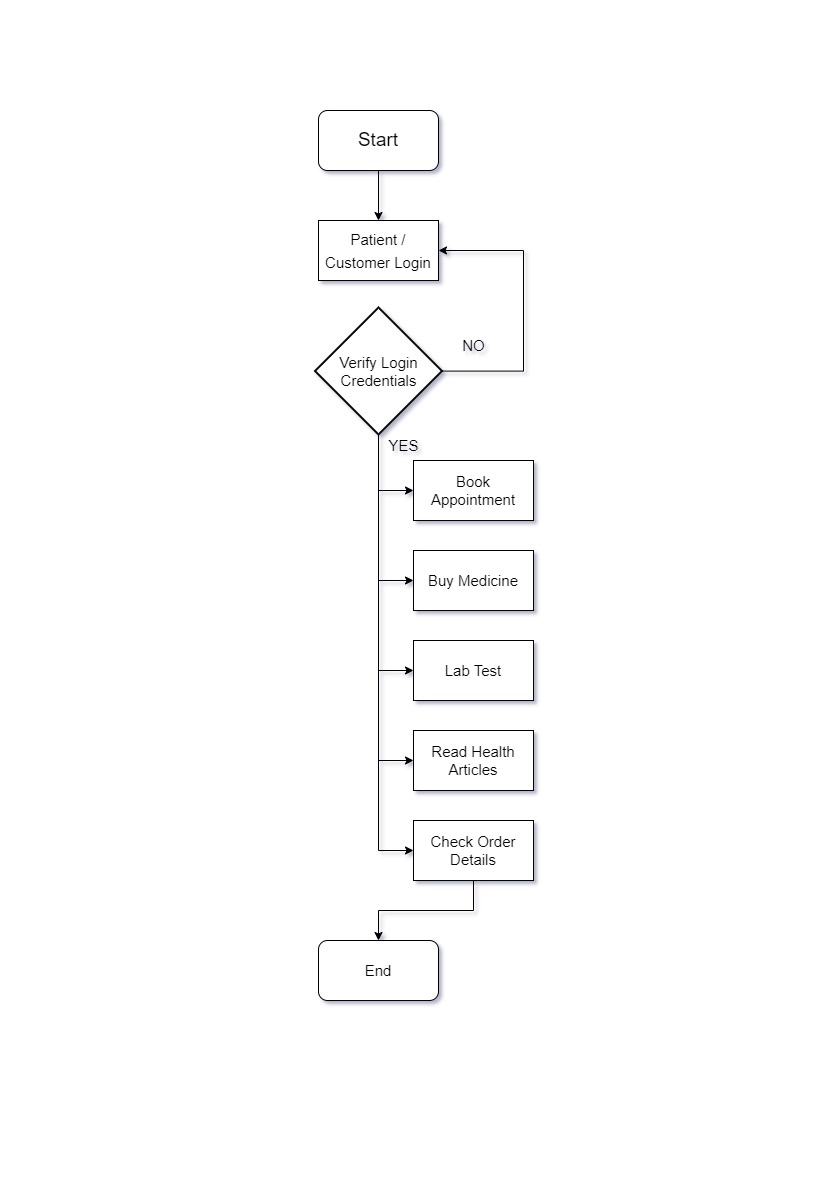
1. **Admin**
2. **Patient**
   * 1. **Description of User Role**
3. **Admin:** manage department of hospitals, user, doctor, nurse, pharmacist, laboratories accounts, watch appointment of doctors, watch transaction reports of patient payment, Bed, ward, cabin status, watch blood bank report, watch operation report, watch birth report, watch diagnosis report, watch death report.
4. **Patient:** View appointment list and status with doctors, View prescription details, View medication from doctor, View doctor list, View blood bank status, View operation history, Manage own profile
   1. **Modules**
5. Provide access to registered users only.
6. Registration of new patients.
7. Enable patient to view their record.
8. Enable patient to update their record.
9. Generate appointment date and timing.
10. Confirmation by doctor.
11. Patients can do Payment.
12. Modification in schedule by patient.
13. Admin access to patient’s record.
14. Admin Verify Payment and Generate Bill/Receipt.
15. Admin can view monthly/yearly records.
    1. **Modules Description**
16. Information about Patients is done by just writing the Patients name, age, and gender. Whenever the Patient comes up his information is stored freshly.
17. Bills are generated by recording price for each facility provided to patient on a separate sheet and at last, they all are summed up.
18. Keeping information about various diseases and medicines available to cure them.
19. Keeping record of the immunization provided to children/patients.
20. Recording information related to diagnosis given to patients.
21. Recording information about the Patients that come.
    1. **Hardware Requirements**

|  |  |
| --- | --- |
| **Name of Components** | **Specification** |
| Processor | Intel core i3 / Ryzen 3 or above |
| RAM | 2GB and above |
| Hard Disk | 32GB or above |

* 1. **Software Requirements**

|  |  |
| --- | --- |
| **Name of Components** | **Specification** |
| Operating System | Android 6+ |
| Software development Kit | Android Studio |
| Tools & languages | Java, XML, Json |

* 1. **Flow Chart**



yes

Figure 3.8.1. System Flow Chart

* 1. **Time Line Chart**

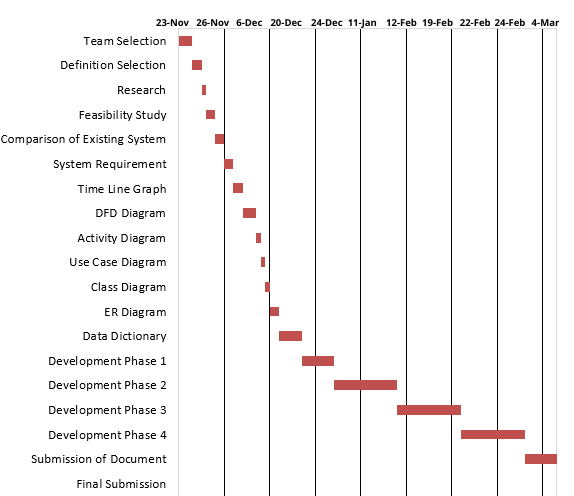


Figure 3.9.1. Time Line Chart

1. **Technology Description**
   1. **Features and Limitations of New System**

|  |  |
| --- | --- |
| **Existing System** | **New System** |
| Less Functionality | More Functionality |
| Not so Secured | Highly Secured |
| Bad User Experience | Modern and new user experience |
| Not economical Feasible | It is feasible |

**Features**

* + - 1. Patient Management.
      2. Appointment Scheduling
      3. Medical record management
      4. Secure Data Management
      5. Reporting and Analysis

**Limitations**

Complexity

Cost

Data Management

Integration with other system

User Adoption

1. **Data Flow Diagram**
   1. **Context Level DFD’s**

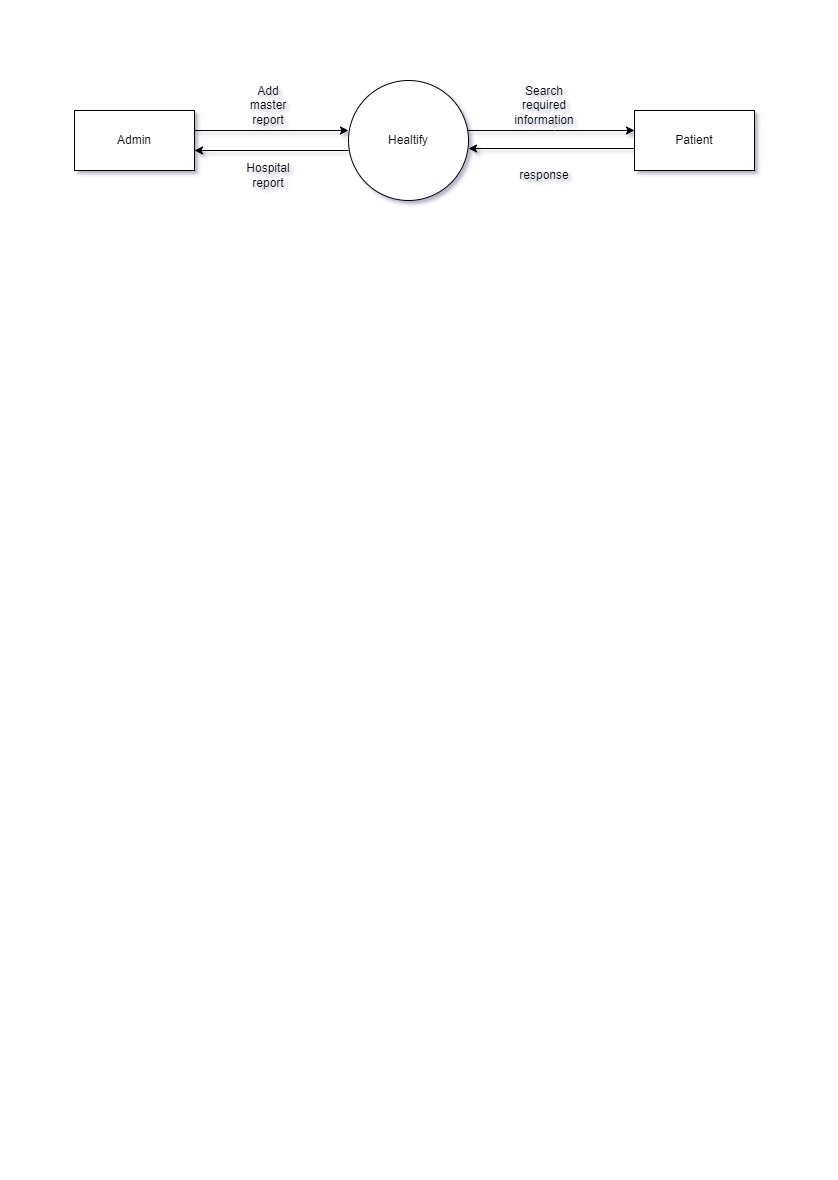


Figure 5.1.1. Context Level DFD: 0 Level

* 1. **Level 1 DFD’s:**

**ADMIN**

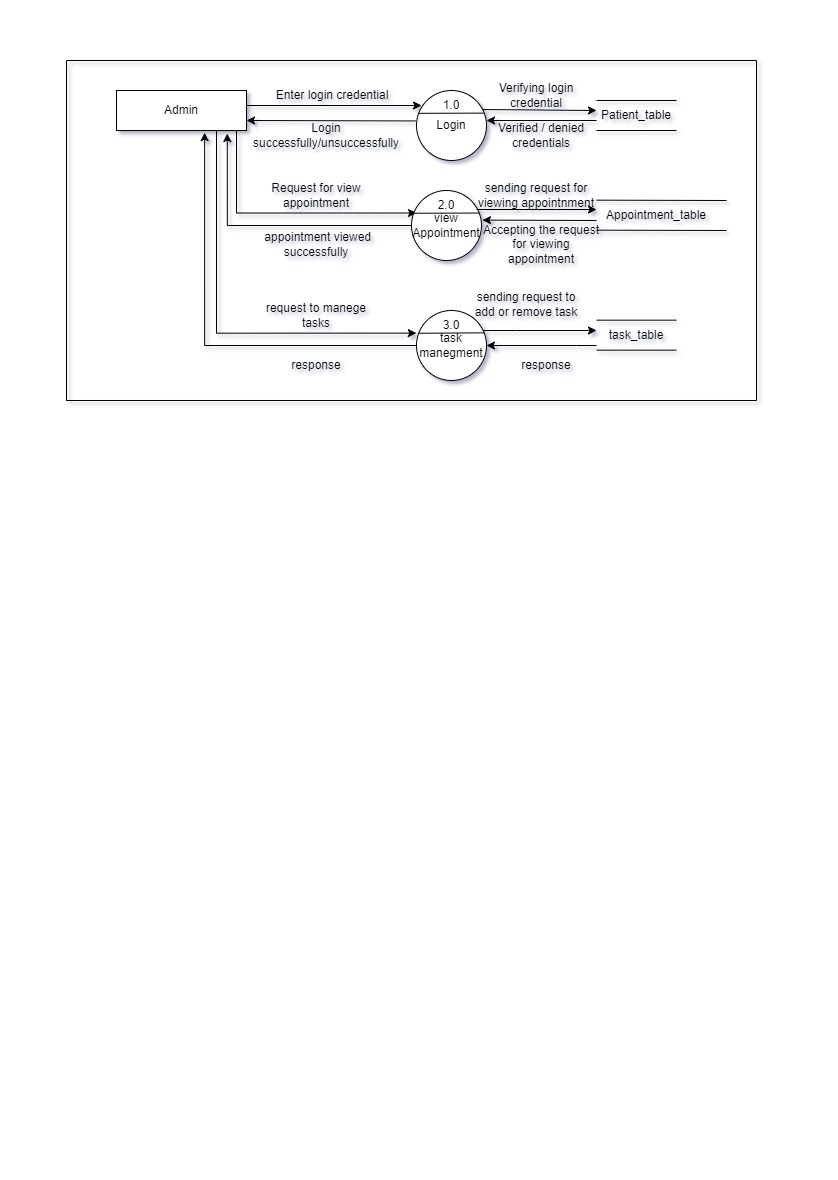


Figure 5.2.1. 1 Level DFD: Admin

**PATIENT**

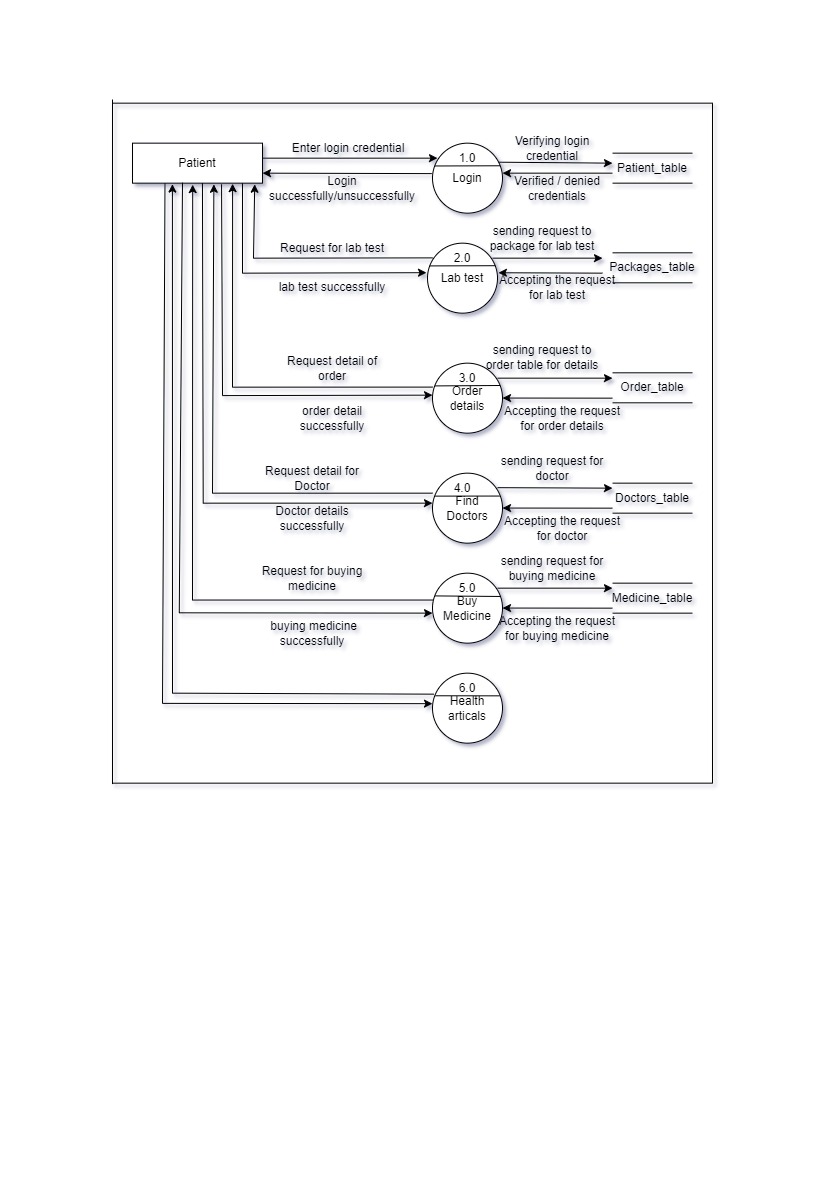


Figure 5.2.2 1 Level DFD: Patient

* 1. **Level 2 DFD’s**

**Admin**

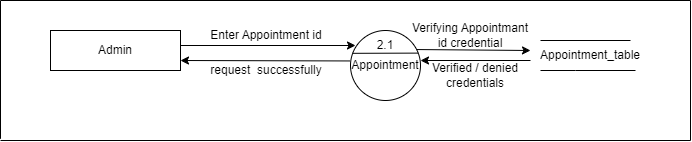
****

Figure 5.3.1 2 Level DFD: Admin

**Patient**

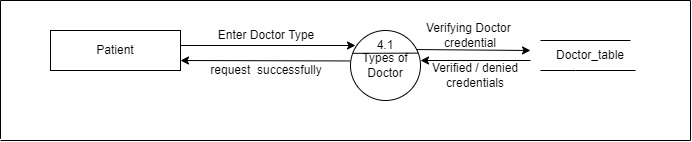
****

Figure 5.3.2 2 Level DFD: Patient

**Patient**

Figure 5.3.3 2 Level DFD: Patient

**Patient**

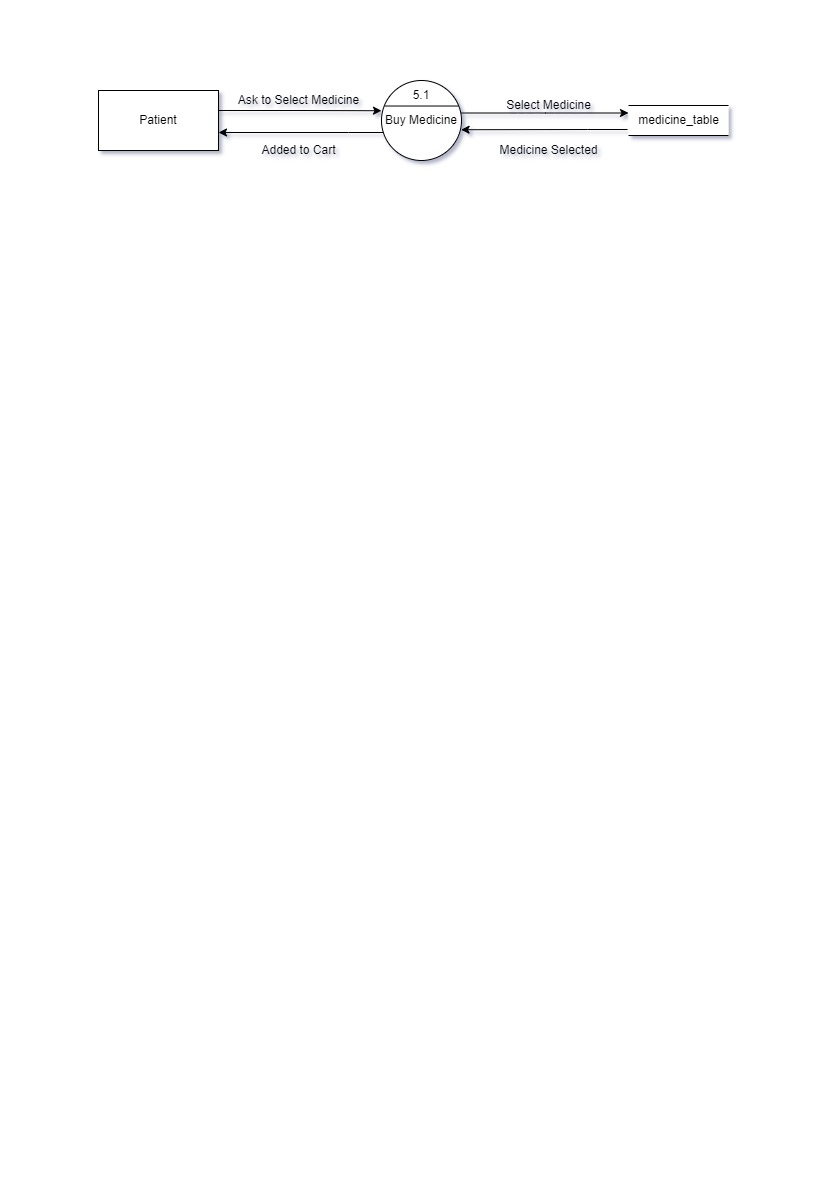


Figure 5.3.4 2 Level DFD: Patient

1. **Use Case Diagram**

**Use Case: ADMIN**

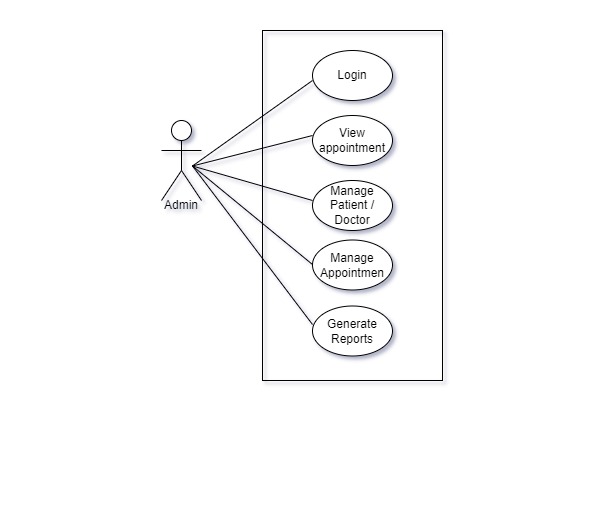


Figure 6.1.1. Use Case Diagram: Admin

**Use Case: PATIENT**

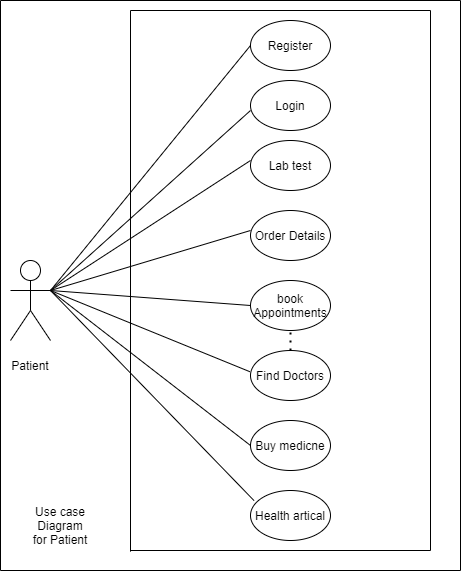


Figure 6.1.2. Use Case Diagram: Patient

1. **Class Diagram**

**Class Diagram: Healthify**

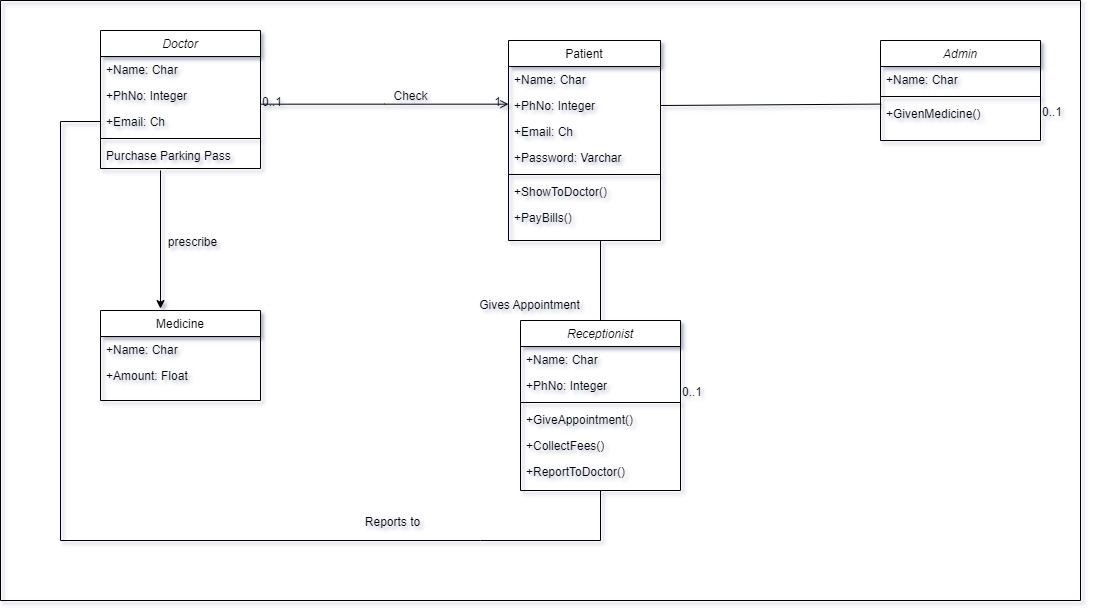


Figure 7.1. Class Diagram of Healthify

1. **Activity Diagram**

**Activity Diagram: Admin**

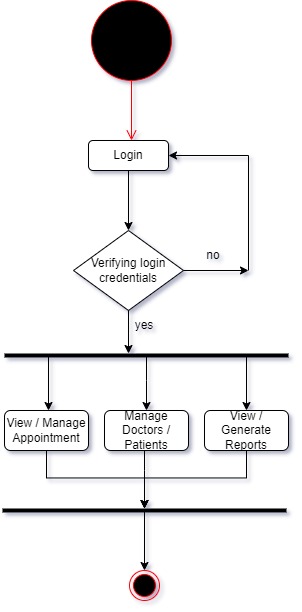


Figure 7.1. Activity Diagram: Admin

**Activity Diagram : Patient**

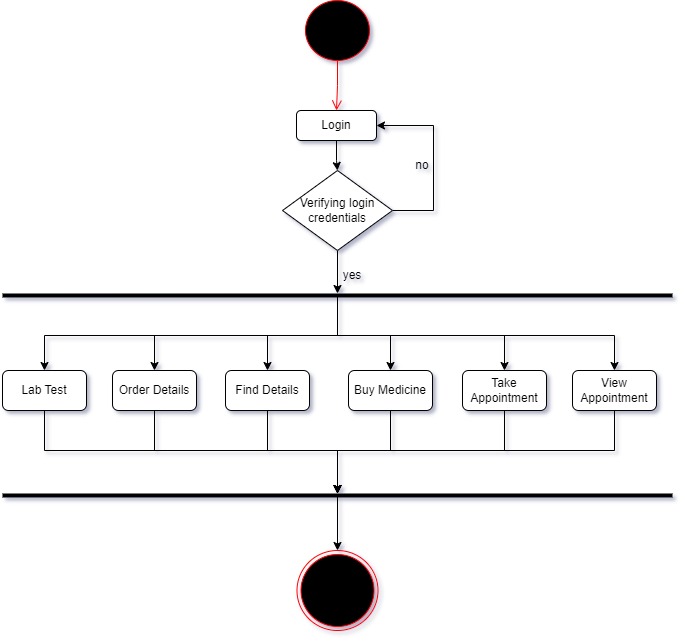


Figure 7.2. Activity Diagram: Patient

* 1. **Description of Activity Diagram**

The activity diagram for all users of a hospital management system summarizes the different activities and processes that occur for each user type, including patients, doctors, and administrators. The activity diagrams are designed to provide a visual representation of the flow of activities, and to help stakeholders understand the different processes and interactions that occur within the system.

For patients, the activity diagram includes steps such as patient registration, appointment scheduling, medical record access, and billing and financial management. For doctors, the activity diagram includes steps such as patient record management, prescription management, and appointment management. For administrators, the activity diagram includes steps such as user account management, appointment scheduling, and financial reporting and management.

In general, the activity diagrams are used to identify the steps and processes involved in each user scenario, to help identify potential bottlenecks or areas for improvement, and to guide the development of a robust and efficient hospital management system.

1. **E-R Diagram**

**Entity-Relationship Model**

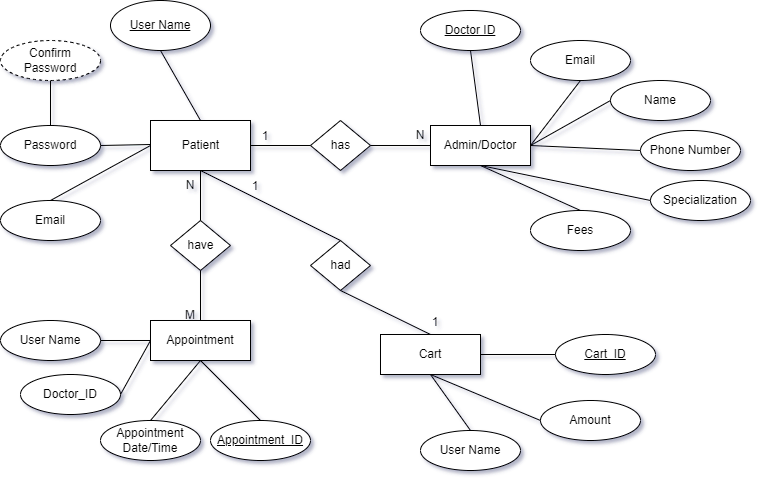


Figure 9.1. Entity Relationship Diagram: Admin

* 1. **Description of E-R Diagram:**

The ER diagram for the hospital management system is a graphical representation of the entities and relationships in the system. It provides a high-level view of the structure of the system and helps in identifying the relationships between different components.

The ER diagram includes entities such as patients, doctors, appointments, medical records, and billing information. Each entity is represented by a rectangle, and the relationships between entities are represented by lines connecting the rectangles.

For example, a patient can have many appointments, and a doctor can have many appointments. A medical record is associated with a specific patient and doctor, and billing information is associated with a specific patient.

The ER diagram helps in understanding the requirements of the system and in designing a database that meets those requirements. It can also help in identifying potential problems and limitations in the system, and in finding ways to resolve them.

1. **Data Dictionary**
   1. **Patients :**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sr. No** | **Field Name** | **Datatype** | **Size** | **Description** | **Constraint** | **Example** |
| 1 | Patient Id | int | 15 | Unique id of the Patient | PRIMAERY KEY | 1234567890 |
| 2 | Name | varchar | 15 | Name of the Patient | NOT NULL | Garvit Bucha |
| 3 | Address | varchar | 30 | Address of the Patient | NOT NULL | Rajkot Gujarat |
| 4 | Phone number | Int | 10 | 10 digits phone number of patient | NOT NULL and UNIQUE KEY | 1234567890 |
| 5 | Email | varchar | 15 | Email of Patient | NOT NULL and UNIQUE KEY | abc@gmail.com |

**10.2 Doctors :**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sr. No** | **Field Name** | **Datatype** | **Size** | **Description** | **Constraint** | **Example** |
| 1 | Doctor Id | int | 15 | Unique id of the Doctor | PRIMAERY KEY | 1234567890 |
| 2 | Name | varchar | 15 | Name of the Doctor | NOT NULL | Kartik Gupta |
| 3 | Specialization | char | 30 |  | NOT NULL |  |
| 4 | Phone number | Int | 10 | 10 digits phone number of patient | NOT NULL and UNIQUE KEY | 1234567890 |
| 5 | Email | varchar | 15 | Email of Doctor | NOT NULL and UNIQUE KEY | abc@gmail.com |
| 6 | Address | varchar | 15 | Address of doctor | NOT NULL | Rajkot Gujarat |

**10.4 Appointment:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sr. No** | **Field Name** | **Datatype** | **Size** | **Description** | **Constraint** | **Example** |
| 1 | Appointment Id | int | 15 | Unique id of the Appointment no. | PRIMAERY KEY | 1234567890 |
| 2 | Patient Id | int | 15 | Unique id of the Patient | PRIMAERY KEY | 1234567890 |
| 3 | Doctor Id | int | 15 | Unique id of the Doctor | PRIMAERY KEY | 1234567890 |
| 4 | Appointment date and time | int | 12 | Unique date and time | NOT NULL | 14-02-2023 / 11:00 |
| 5 | Appointment status | char | 10 | Conformation Status of appointment | UNIQUE KEY | Scheduled, Cancelled, Completed. |

* 1. **Medical Records:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sr. No** | **Field Name** | **Datatype** | **Size** | **Description** | **Constraint** | **Example** |
| 1 | Medical Records Id | int | 15 | Unique id of the Appointment no. | PRIMAERY KEY | 1234567890 |
| 2 | Patient Id | int | 15 | Unique id of the Patient | PRIMAERY KEY | 1234567890 |
| 3 | Doctor Id | int | 15 | Unique id of the Doctor | PRIMAERY KEY | 1234567890 |
| 4 | Date and time of record creation | int | 12 | Unique date and time of record creation | NOT NULL | 14-02-2023 / 11:00 |
| 5 | Description of medical condition and treatment | char | 30 | Description of medical condition and treatment | NOT NULL | Take paracetamol, etc |
| 6 | Test result and imaging data | varchar | 20 | Test result | NOT NULL | Fever, cold, etc |

* 1. **Billing and Financial Information :**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sr. No** | **Field Name** | **Datatype** | **Size** | **Description** | **Constraint** | **Example** |
| 1 | Billing Id | int | 15 | Unique id of the Doctor | PRIMAERY KEY | 1234567890 |
| 2 | Patient Id | int | 15 | Unique id of the Patient | PRIMAERY KEY | 1234567890 |
| 3 | Total amount charged | int | 5 | Amount paid by patient | NOT NULL | 1500/- |
| 4 | Payment Status | char | 10 | Payment paid status by patient | NOT NULL and UNIQUE KEY | Paid, outstanding, processing ,etc. |

* 1. **Description of Data Dictionary**

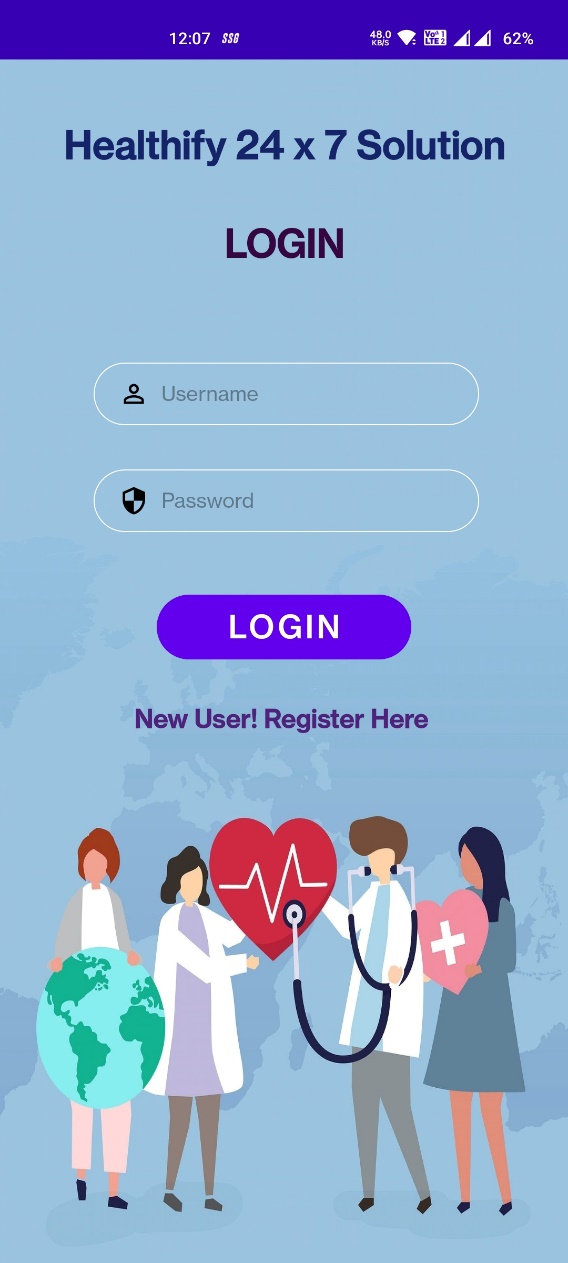
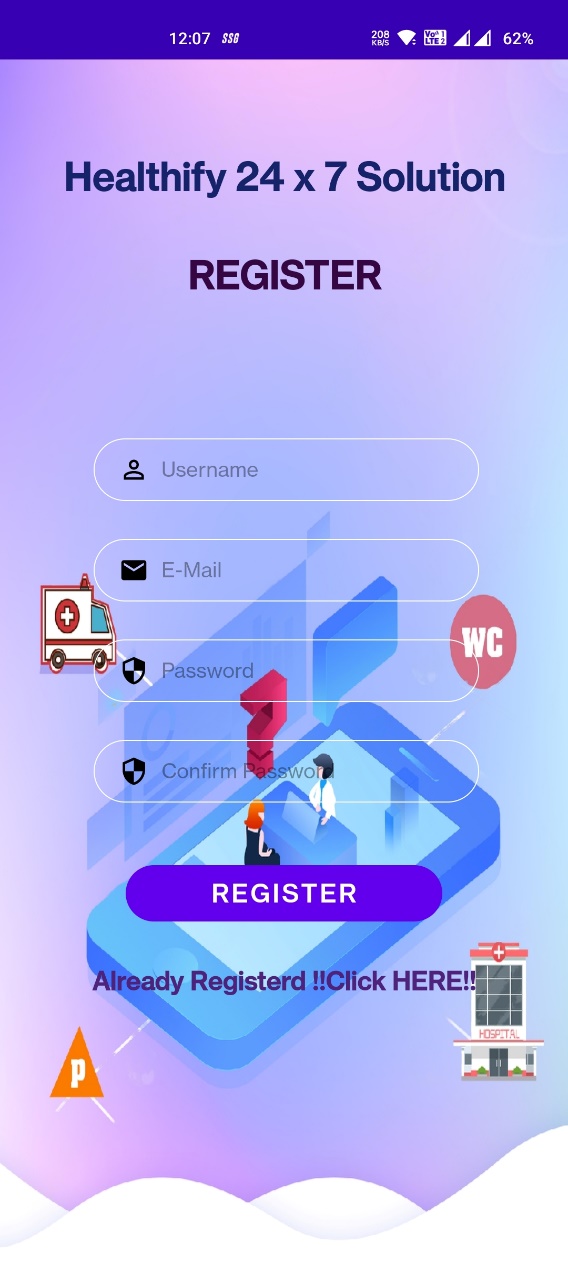
A data dictionary is a comprehensive list of all the data elements in a system and their associated metadata. In the context of a hospital management system, the data dictionary would contain information about the different types of data that the system needs to store and manage, such as patient information, doctor information, appointment information, and billing information.

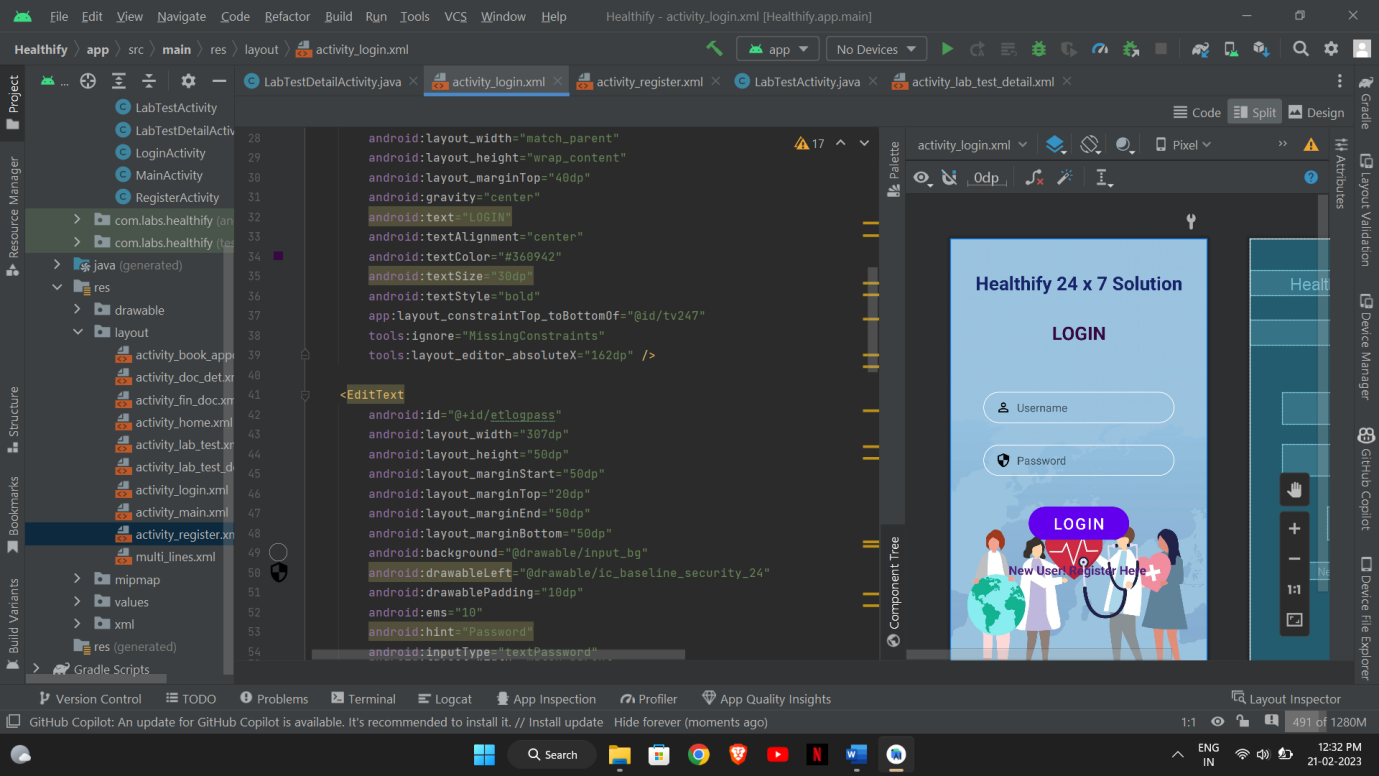
Each data element in the data dictionary would have a unique identifier and would be described in detail, including its data type, size, format, and any constraints or rules associated with it. The data dictionary would also include information about the relationships between data elements, such as the relationships between patients and appointments or between doctors and medical records.

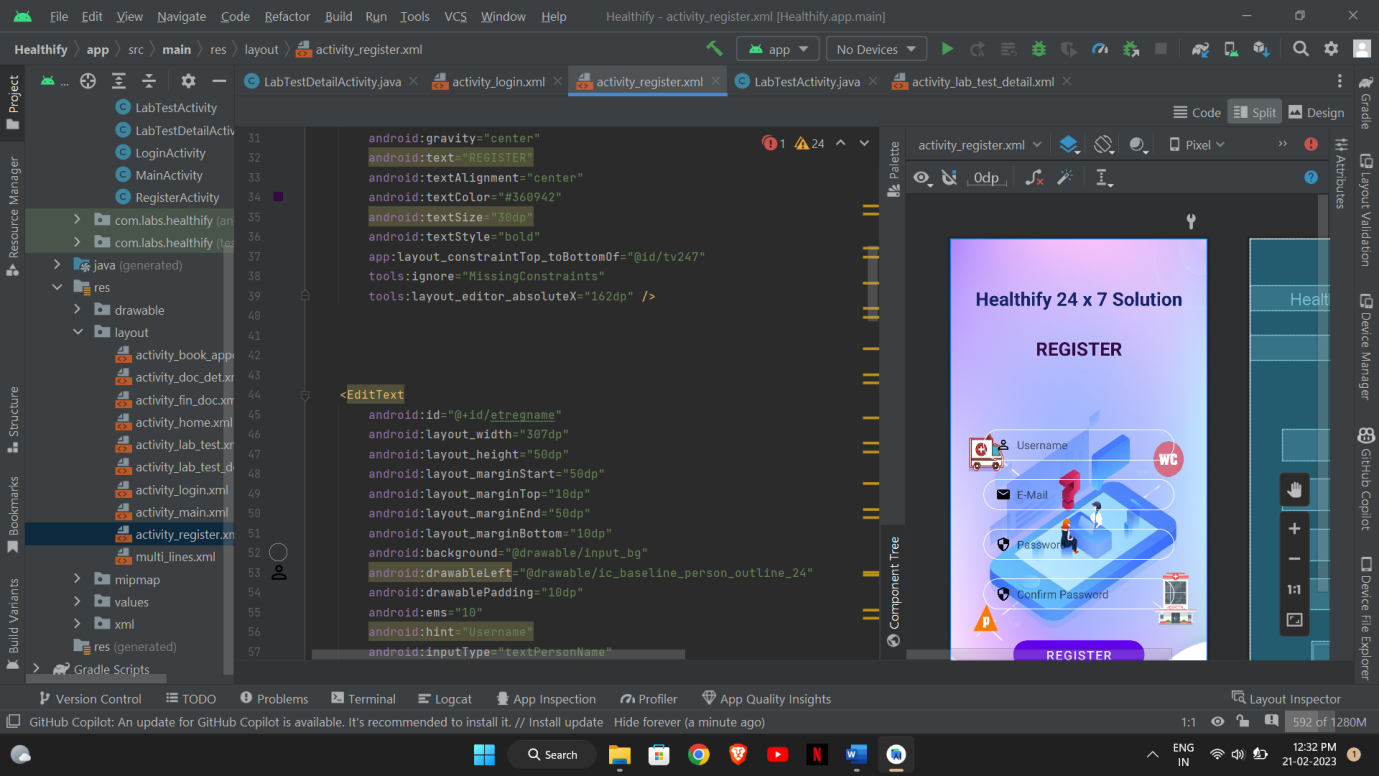
The data dictionary is a crucial tool for ensuring the consistency, accuracy, and integrity of data in the hospital management system. It can also be used by designers, developers, and stakeholders to understand the requirements of the system and to ensure that all necessary data elements have been considered.

In summary, the data dictionary provides a comprehensive view of all the data elements in a hospital management system, and is an essential tool for ensuring the quality and accuracy of data in the system.

1. **Form Design**
   1. **Development Phase -1**

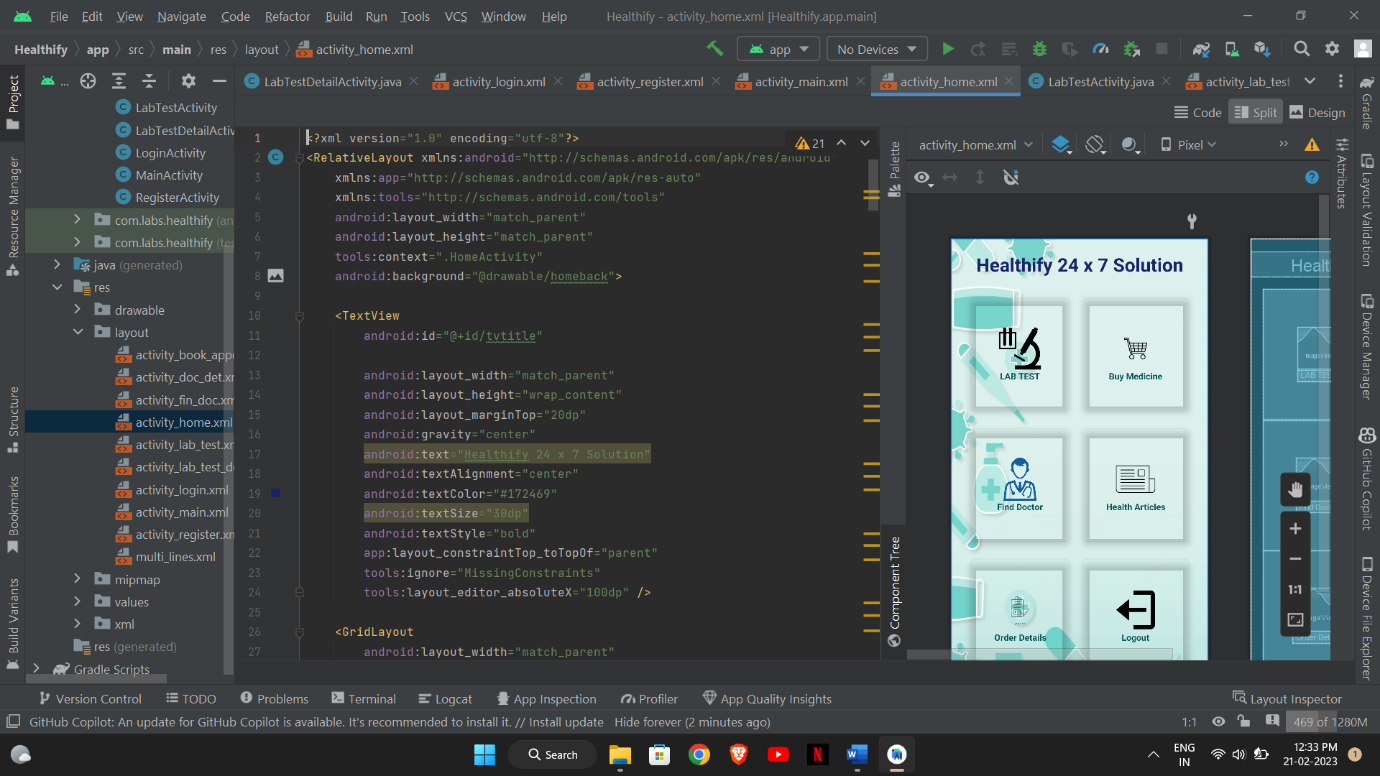


****

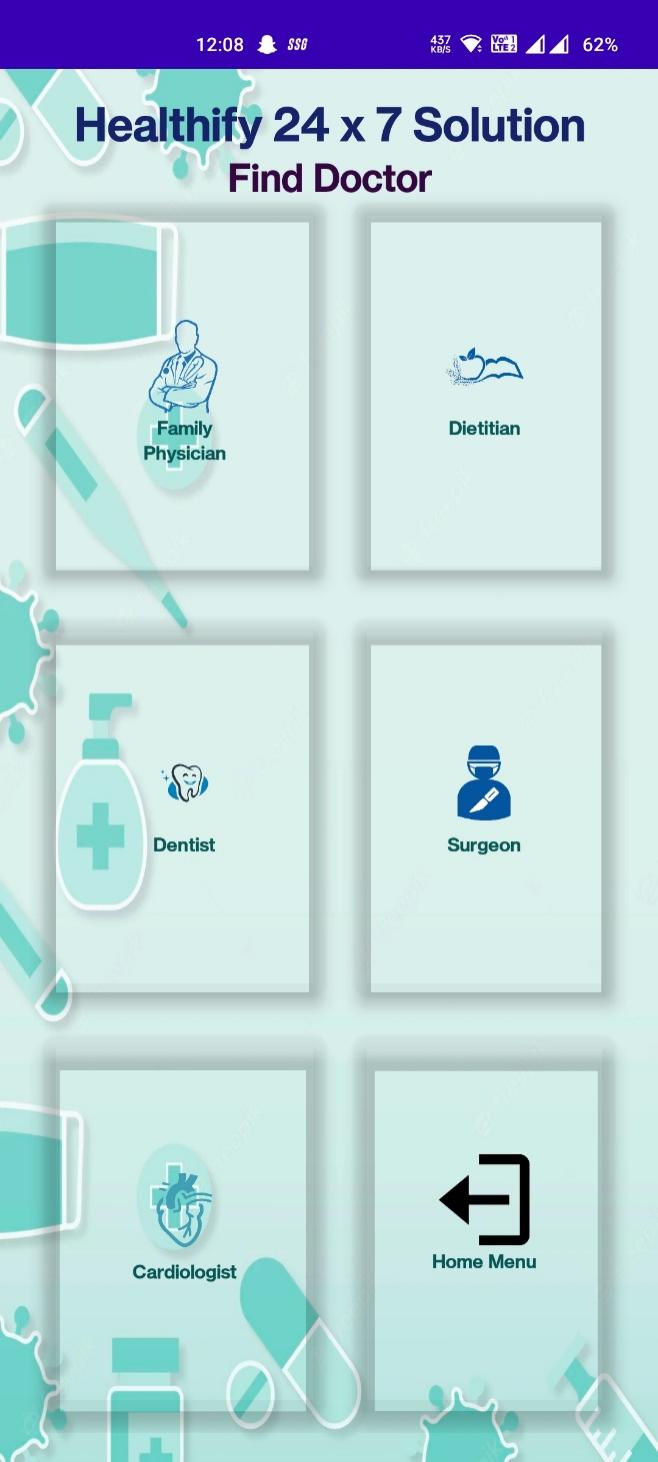
****

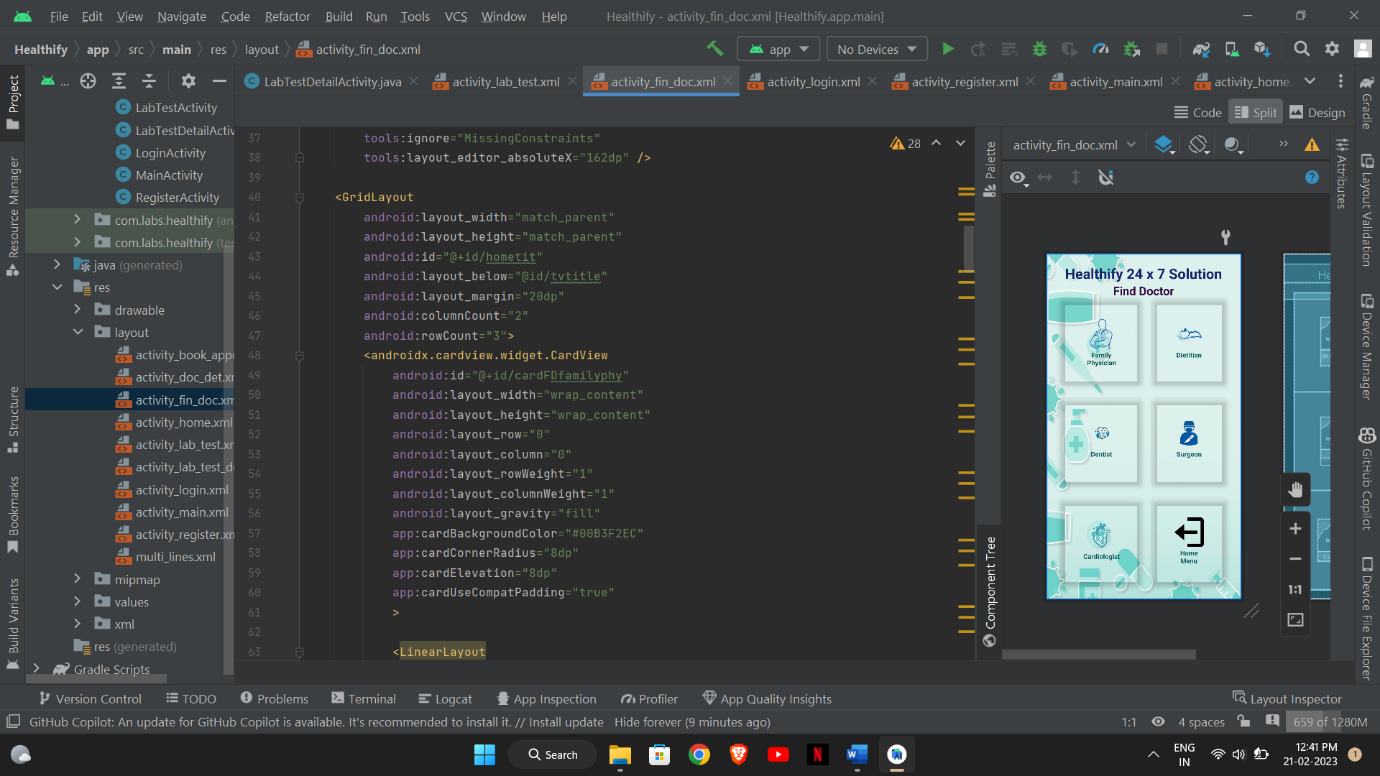
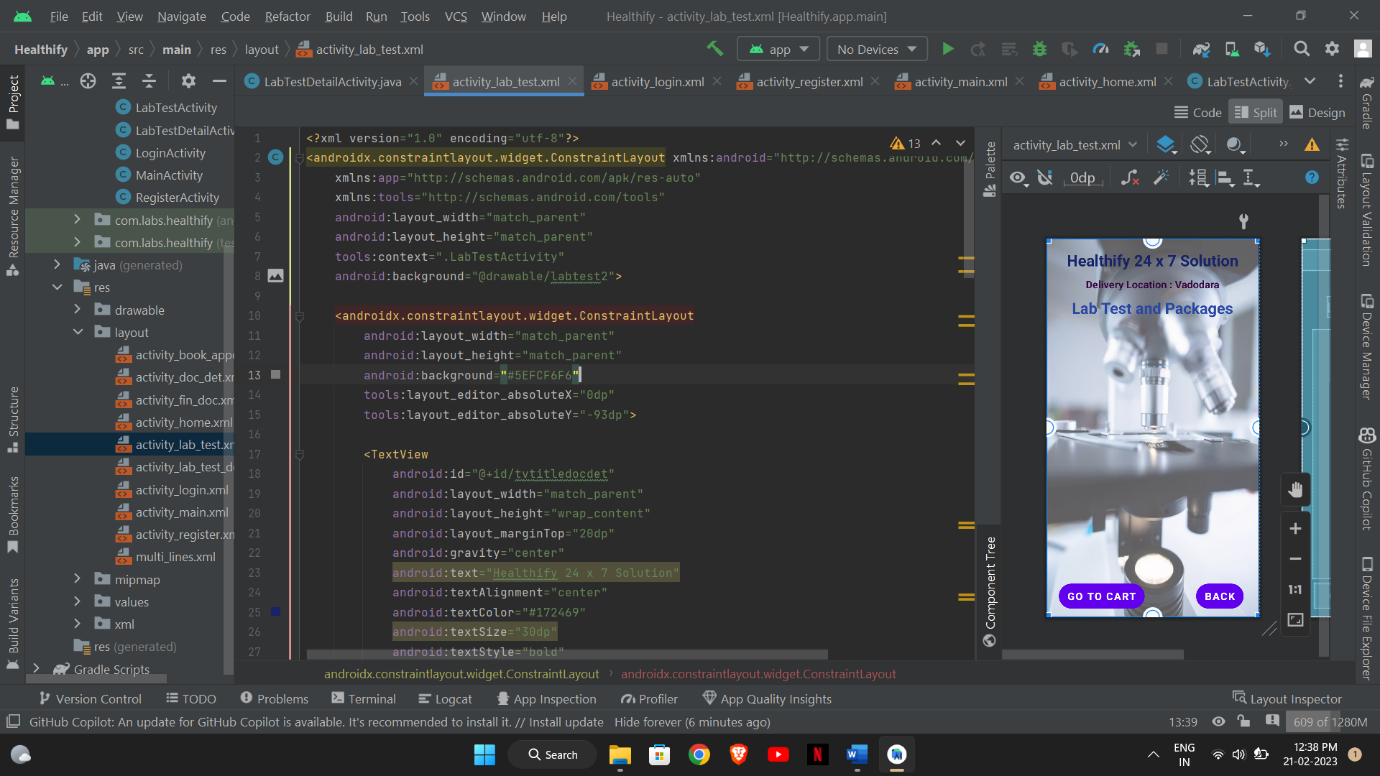
* 1. **Development Phase -2**



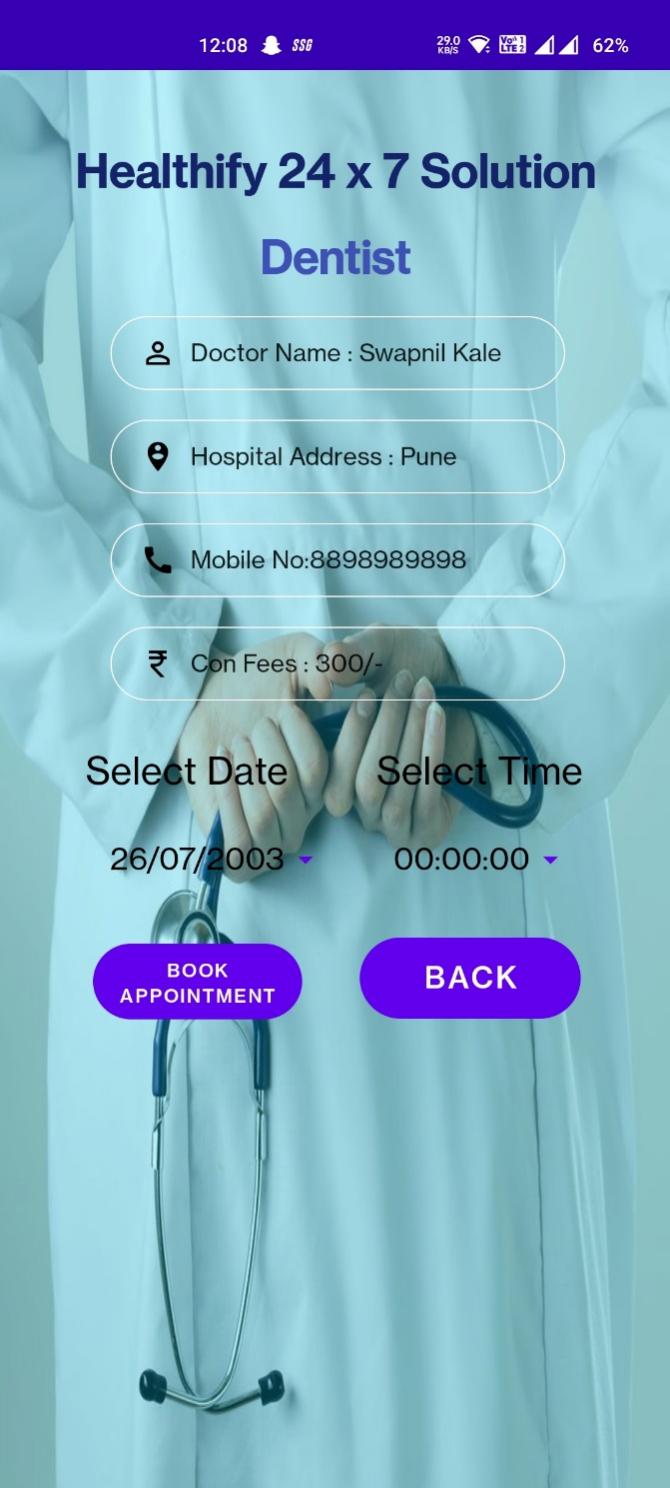


* 1. **Development Phase -3**



****

* 1. **Development Phase -4**



**12. What is testing?**

**Software Testing** is a method to check whether the actual software product matches expected requirements and to ensure that software product is[Defect](https://www.guru99.com/defect-management-process.html)free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, gaps, or missing requirements in contrast to actual requirements.

Some prefer saying Software testing definition as a [White Box](https://www.guru99.com/white-box-testing.html) and [Black Box Testing](https://www.guru99.com/black-box-testing.html). In simple terms, Software Testing means the Verification of Application Under Test (AUT). This Software Testing course introduces testing software to the audience and justifies the importance of software testing.

**12.1. Importance and types of testing**

The following are important reasons why software testing techniques should be incorporated into application development:

* **Identifies defects early.** Developing complex applications can leave room for errors. Software testing is imperative, as it identifies any issues and defects with the written code so they can be fixed before the software product is delivered.
* **Improves product quality.** When it comes to customer appeal, delivering a quality product is an important metric to consider. An exceptional product can only be delivered if it's tested effectively before launch. Software testing helps the product pass quality assurance ([QA](https://www.techtarget.com/searchsoftwarequality/definition/quality-assurance)) and meet the criteria and specifications defined by the users.
* **Increases customer trust and satisfaction.**Testing a product throughout its development lifecycle builds customer trust and satisfaction, as it provides visibility into the product's strong and weak points. By the time customers receive the product, it has been tried and tested multiple times and delivers on quality.
* **Detects security vulnerabilities.** Insecure application code can leave vulnerabilities that attackers can exploit. Since most applications are online today, they can be a leading vector for [cyber-attacks](https://www.techtarget.com/searchsecurity/definition/cyber-attack) and should be tested thoroughly during various stages of application development.
* **Helps with scalability.** A type of non-functional software testing process, [scalability](https://www.techtarget.com/searchdatacenter/definition/scalability) testing is done to gauge how well an application scales with increasing workloads, such as user traffic, data volume and transaction counts. It can also identify the point where an application might stop functioning and the reasons behind it, which may include meeting or exceeding a certain threshold, such as the total number of concurrent app users.
* **Saves money.**Software development issues that go unnoticed due to a lack of software testing can haunt organizations later with a bigger price tag. After the application launches, it can be more difficult to trace and resolve the issues, as software patching is generally more expensive than testing during the development stages.

**The following are the main types of software testing methodologies:**

* Integration testing. This groups together two or more [modules](https://www.techtarget.com/whatis/definition/module) of an application to ensure they function collectively. This type of testing also reveals interface, communication, and data flow defects between modules.
* Unit testing. Typically conducted during the application development phase, the purpose of unit testing is to ensure that each individual unit or component performs as expected. This is a type of [white box testing](https://www.techtarget.com/searchsoftwarequality/definition/white-box) and test automation tools -- such as NUnit, JUnit and xUnit -- are typically used to execute these tests.
* Functional testing. This entails checking functions against functional requirements. A common way to conduct functional testing is by using the [black box testing](https://www.techtarget.com/searchsoftwarequality/definition/black-box)
* Security testing. This ensures the software is free of potential vulnerabilities, known flaws and security loopholes that might affect the user system and data. Security testing is generally conducted through [penetration testing](https://www.techtarget.com/searchsecurity/definition/penetration-testing).
* Performance testing. This tests the performance and speed of an application under a given workload.
* Regression testing. This verifies whether adding new features causes a decline in the functionality of an application.
* Stress testing. This assesses the strength of software by testing how much load it can take before reaching a breaking point. This is a type of non-functional test.
* Acceptance testing. This evaluates the entire system against the desired requirements and ensures the project is complete.

**13. Future Enhancement**

* **Following features are to be implemented in future:**

1. **Live doctor check-up via video call.**

Due to emergency, sometimes doctors are not available in clinics, so via the facility of video calls doctor can give treatment to patient.

1. **Complete payment Mechanism.**

There should be a complete payment module will be implemented so that a patient can pay in any online payment method according to their convenience.

1. **Real-Time Medicine/Test order updates.**

The customer/patient can see the updates of the medicine/ test updates by the notifications and will state that at what status are they currently.

1. **Updates for Regular Health Check-ups.**

There should be module which can inform the patient about their check-ups by which patient gets their check-ups done more conveniently.

1. **References & Bibliography**

Website:

1. w3schools.com
2. javapoint.com
3. developers.android.com
4. youtube.com

Resources:

1. Guide – Prof Manish Joshi
2. Teacher – Prof Priyavat Dwivedi

Books:

1. Android Wireless Application Development by Lauren Darcey and Shane Conder, Pearson Education, 2nd ed. (2011)
2. Professional Android 2 Application Development by Reto Meier, Wiley India Pvt Ltd (2011)
3. Beginning Android, by Mark L Murphy, Wiley India Pvt Ltd (2009)
4. Pro Android, by Sayed Y Hashimi and Satya Komatineni, Wiley India Pvt Ltd (2009)