SMART HEALTH MONITORING SYSTEM

Table of Contents

1.1 Purpose

The purpose of this project is to monitor the vital parameters of a patient using sensors remotely and to generate timely warning to the medical staff and Doctors in case of medical emergencies.

The purpose of the Android application (software) is to provide a clear, easy to use, and informative application for Doctor, Nurse, Admin and patient (users) to track and visualize their overall health.

1.2 Product Scope

This system can fetch real time data of a patient through the various sensors such as- ECG sensor, Temperature sensor and blood pressure sensor. The purpose of this project is to monitor real time parameters such as blood pressure, temperature and heartbeat of a patient to observe the health parameters. If the patient is feeling uneasy and has gone out of the ward for a walk and if doesn't come after 20 min, then this system will trigger an alarm or send notification to the nurse. Each bed will be having a sensor system allocated to it. A RFID reader Is also allocated to every bed, so in case a patient wants to switch the bed then that can be done using this RFID device.

1.3 Reference

This document is built upon the following references:

- IEEE Software Specification Requirement template
https://ieeexplore.ieee.org/document/8405471

2. OVERALL DESCRIPTION

2.1 PRODUCT PERSPECTIVE

In this proposed work the vital parameters such as temperature, ECG, humidity and heart beat readings which are monitored using node MCU. These sensors signals are send to node MCU. The data is being stored in real time database (i.e. Firebase). Here patients body temperature, ECG and heart rate is measured using respective sensors and it can be displayed on the screen of an app using Node MCU connected to a server. It enables continuous monitoring of the patient's health parameters by the doctor and nurse. Any abrupt increase or decrease in these parameter values can be detected at the earliest and hence necessary medications can be implemented by the doctor immediately. If the patient is feeling uneasy and has gone out of the ward for a walk and if doesn't come after 20 min, then this system will trigger an alarm or send notification to the nurse. Each bed will be having a sensor system allocated to it. A RFID reader Is also allocated to every bed, so in case a patient wants to switch the bed then that can be done using this RFID Tag.

Proposed system consists of following sensors and modules:

1. Node MCU Micro Controller
2. ECG sensor
3. Temperature sensor
4. Blood Pressure sensor
5. RFID Tag and Reader

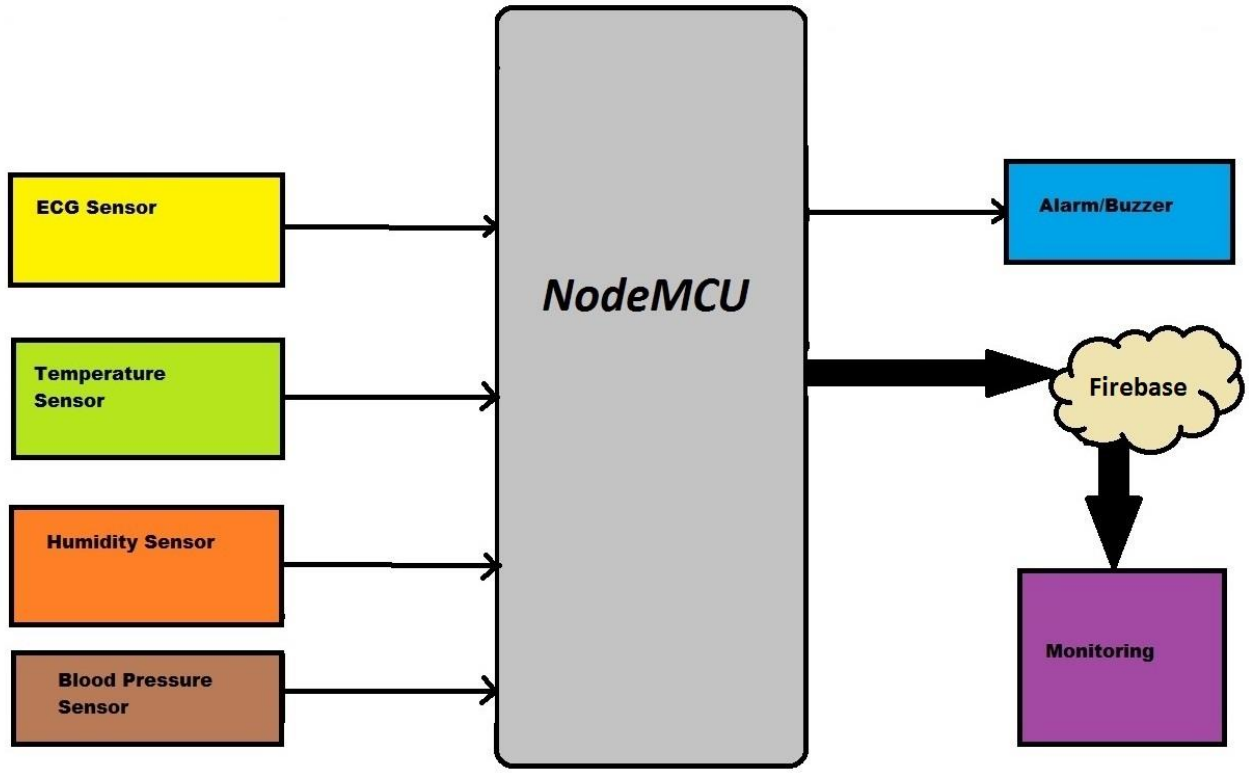


Fig: Block Diagram

2.2 PRODUCT FUNCTION

A brief summary of the major product functions and what the end-user may perform on the SHMS includes the following

Registering: - Here Admin will register the patient and provide him/her a patient ID.

Hospitalisation: - The admin will hospitalise the patient.

Allocation of sensors: - The patient will be allotted with bed , a RFID tag and sensors .

Monitoring: - With the help of sensors the real time health parameters of the patient will be gathered and send to the database. if the doctor is no present at the moment he can see the health .parameters using his mobile app.

Alarm generation: - If the patient's condition is critical then notification will be sent to the doctor and the nurse.

2.3 USER CLASS AND CHARACTERISTICS

This system can be used by the Doctor, Nurse, Admin and Patient/Guardian.

SYSTEM DESIGN

System design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. Overall product architecture, the subsystems that compose the product, and the way subsystems are allocated to processors are depicted using the System Design. UML is used to model system designs. Unified Modelling Language is a standard object-oriented analysis and design language. Use Case diagram which is a type of UML diagrams, of the application are shown below.

Use Case Diagram-

A Use Case Diagram consists of set of elements and the relationships between them. It depicts all the scenarios, regarding how our application interacts with users and other external systems to achieve the goals of application. The main components of a use case diagram include actors, use cases and their relationships. The use case is an external view of the system that represents some actions that the user performs to get a job done. Actors are the users who interact with the application.

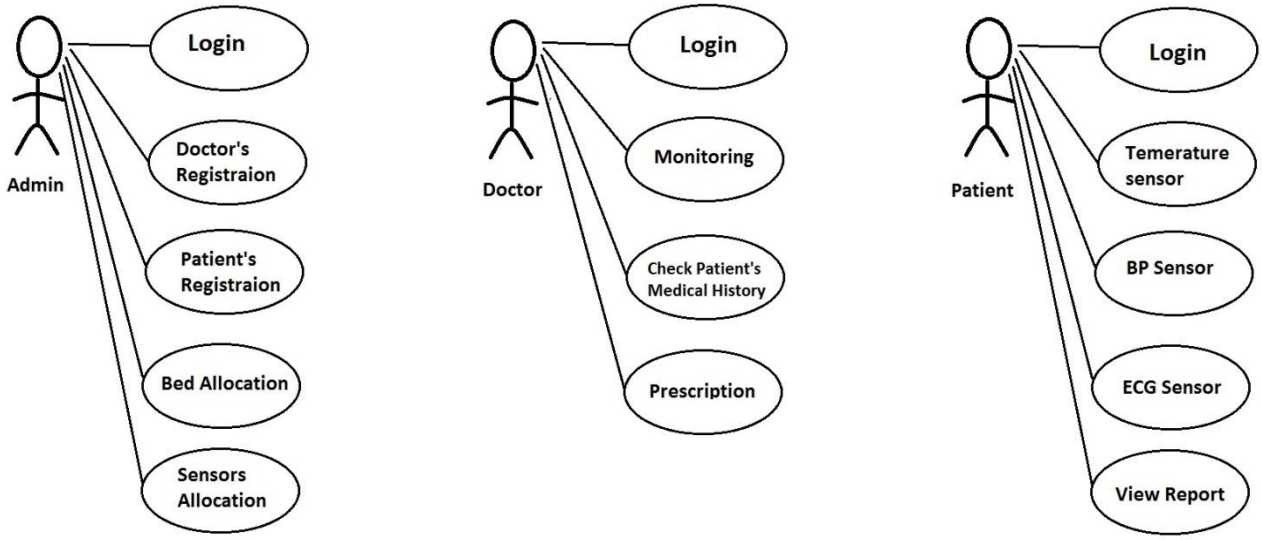


Fig: Use Case Diagram

(Here the actors are Admin, Doctor, patient.)

Use cases:

We have identified a set of use cases based on the functionalities and goals of the application.

- **Login**- This use case denotes a set of actions required for Subject to login into the application.
- **Registration** – This use case denotes a set of actions required for Subject to registering into the application.
- **Hospitalization** – Here the admin will hospitalize the patient.
- **Allocation of sensors** – In this case the patient will be allotted a bed and sensors.
- **Monitoring** – The health parameters of the patient will be monitored by the admin and doctor.
- **Report generation** – The admin will generate report.

2.4 Operating Environment

The application which is designed to view the Health Parameters is developed to use on mobile devices running Android OS 3.1 and above. Although it is not suggested, the application can run on an Android. In order for the user to obtain the application, it will be available for download via the Google Play Store and the user must have access to an internet connection. The health monitoring system will be available in a hospital.

2.5 Minimum Requirements

- Minimum android version required is 4.2.0
- The application must be designed in java.
- The application must be implemented by Android Studio.

2.6 Assumptions and Dependencies

There are certain assumptions that we have assumed that

- The user has access to an Android mobile device.
- The user has a valid internet connection to be able to access information.
- The user will be able to find the previous medical information of patient in the database.
- The sensors are working properly.
- The nurse or the staff knows how to handle and use the sensors.
- The electricity is provided 24 hours.
- There already exists a well designed website of the Hospital.

3. EXTERNAL INTERFACE REQUIREMENT

3.1 USER INTERFACE

This Project is designed to make sure that user interface pages are easily understandable and the navigation between pages is obvious. Below are list of web pages that user can navigate between and are shown in details.

Patient Registration-

The receptionist of the hospital will register the patient and fill all his/her details. After registration patient will be provided a unique id (Patient ID) which can be further used by the doctor to see the health parameters.

Login Page-

For Doctor: Here, doctor enters the patient's unique credentials (Patient Id). Once the credentials are verified, login page will be navigated to Patient vital monitoring page where doctor can view current vital readings of the patient. Here patient's unique credentials must be kept confidential by the doctor and Guardian to protect privacy of the patient data.

For Patient/Guardian: In this page, besides the doctor, patient/guardian can also see the patient's history (Reports).

Patient's Vital Monitoring Page-

After doctor login successfully, he will be able to view live patient's vital information which includes temperature, humidity, heartbeat, ECG etc. In order to protect privacy of the patient's data, data is encrypted while sending it to Firebase database server and is decrypted while relaying same data on web page.

In case device is not connected or any of the sensors or is not attached to patient, then all the readings or respective reading would be shown as zero in case of digital values. In case device is switched off then this page would display only last known readings that were stored in database.

Show History of Patient data-

Here, doctor can see history of patient vitals that has been recorded and stored in server in tabular form. This data can specifically be used by doctor to perform analysis on patient health condition to predict any irregularities in health conditions, to recommend change in medication or treatments etc. and can be used to recommend patient regular visits.

Prescription Page- This page allows doctor to give prescription to the patient based on his earlier reports and the new reports .

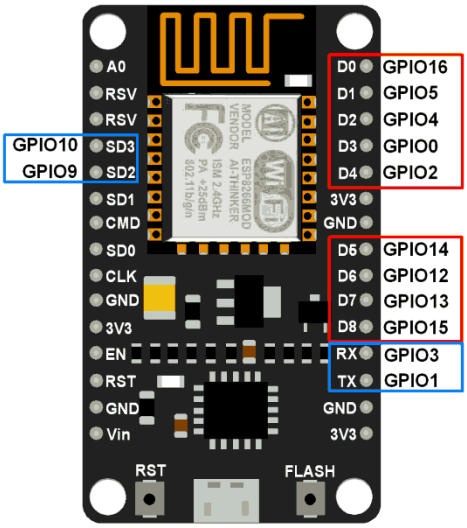
3.2 Hardware interfaces

- Node MCU
- Sensors: Temperature, ECG, Humidity, Blood Pressure
- An android phone(For Application)
- RFID: Tag & Reader

Node MCU:

NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the development kits. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson and SPIFFS.

Fig: NodeMCU



Pin Diagram (NodeMCU)

ECG:

ECG records the electrical activity generated by heart muscle depolarisations, which propagate in pulsating electrical waves towards the skin. Although the electricity amount is in fact very small, it can be picked up reliably with ECG electrodes attached to the skin. The full ECG setup comprises at least four electrodes which are placed on the chest or at the four extremities according to standard nomenclature (RA = right arm; LA = left arm; RL = right leg; LL = left leg). Of course, variations of this setup exist to allow more flexible and less intrusive recordings, for example, by attaching the electrodes to the forearms and legs. ECG electrodes are typically wet sensors, requiring the use of a conductive gel to increase conductivity between skin and electrodes.

Figure 5: ECG Sensor

Temperature Sensor:

Temperature sensor is a device which is designed specifically to measure the hotness or coldness of an object. LM35 is a precision IC temperature sensor with its output proportional to the temperature (in °C). With LM35, the temperature can be measured more accurately than with a thermistor. It also possesses low self-heating and does not cause more than 0.1 °C temperature rise in still air. The operating temperature range is from -55°C to 150°C.The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy.

Fig: Temperature Sensor

BP (Heartbeat) Sensor:

Heartbeat sensor provides a simple way to study the function of the heart which can be measured based on the principle of psycho-physiological signal used as a stimulus for the virtual-reality system. The amount of the blood in the finger changes with respect to time. The sensor shines a light lobe (a small very bright LED) through the ear and measures the light that gets transmitted to the Light Dependent Resistor. The amplified signal gets inverted and filtered, in the Circuit. In order to calculate the heart rate based on the blood flow to the fingertip, a heart-rate sensor is assembled with the help of LM358 OP-AMP for monitoring the heartbeat pulses.

Figure: Heartbeat Sensor

Humidity Sensor:

A humidity sensor (or hygrometer) senses, measures and reports both moisture and air temperature. The ratio of moisture in the air to the highest amount of moisture at a particular air temperature is called relative humidity. Relative humidity becomes an important factor when looking for comfort.

Fig: Humidity Sensor

RFID: Tag & Reader

Radio-Frequency Identification (RFID) is the use of radio waves to read and capture information stored on a tag attached to an object. A tag can be read from up to several feet away and does not need to be within direct line-of-sight of the reader to be tracked.

A passive tag is an RFID tag that does not contain a battery, the power is supplied by the reader. When radio waves from the reader are encountered by a passive rfid tag, the coiled antenna within the tag forms a magnetic field. The tag draws power from it, energizing the circuits in the tag.

A RFID reader is a device used to gather information from an RFID tag, which is used to track individual objects. Radio waves are used to transfer data from the tag to a reader.

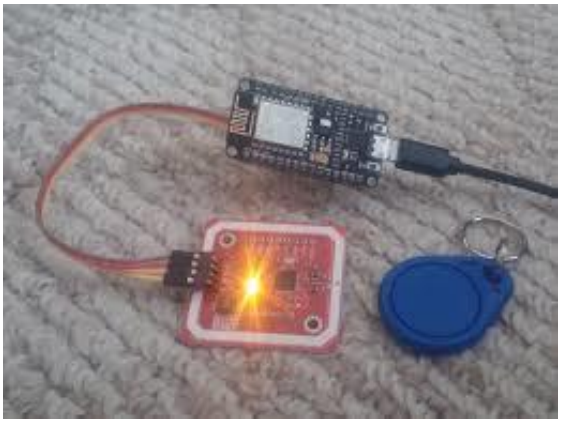


Fig: RFID Tag & reader

3.3 Software interfaces:

- Platform: IoT
- IDE: Arduino IDE
- Database: Google real time database (Firebase)
- Technologies used: Java, C, Android
- Android Application
- Application IDE: Android Studio

IoT (Internet of Things): The Internet of Things is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

Arduino IDE: The Arduino Integrated Development Environment is a cross-platform application that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards.

Firebase: Firebase is a mobile and web application development platform developed by Firebase, Inc. in 2011, then acquired by Google in 2014.

It provides a real-time database and backend as a service. The service provides application developers an API that allows application data to be synchronized across clients and stored on Firebase's cloud.

Android Studio: Android Studio is the official integrated development environment for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is purpose built for Android to accelerate your development and help you build the highest-quality apps for every Android device. It offers tools custom-tailored for Android developers, including rich code editing, debugging, testing, and profiling tools.

4. SYSTEM FEATURES

- Application must have a module for login using unique credentials of a patient for the doctor to monitor patient's vital data.
- Application must have a module for login using unique credentials of a patient for Guardian/Caretaker to see patient's vital data.
- RFID: Tag and reader.
- Notification Service.

5. Other Non-Functional Requirements

Non-functional requirements are not directly related to the functional behavior of the system.

- Web application must be user friendly, simple and interactive.
- The user interface is designed in such way that novice users with little knowledge of web should be able to access this application.
- Users are required to have some knowledge regarding Google maps.

5.1 Performance Requirements

The sensors should start working after 2 seconds. The sensor should work after 5 seconds as soon as the RFID reader reads the RFID tag.

5.2 Safety Requirements

Potential safety concerns associated with the use of the application include:

- End-users should consult with a doctor and other appropriate medical personnel before making significant medical decisions based on information provided and/or accessed via the SHMS.
- The developers of this software application product are not responsible or liable for any advice, course of treatment, diagnosis or any other information, or consequences as a result of information provided, accessed or disseminated via the SHMA.

5.3 Security Requirements

- The system should be secure enough so that personal health data may not be disclosed inappropriately or unauthorized.
- System Application will have a secure "Pass code" to protect the end-user details from unauthorized access.
- Developers will maintain the confidentiality of all user data knowledge.
- Health Monitoring System App will not affect any other applications or software installed on the end-user's mobile device nor will it cause any damage to the device or its internal components.

6. OTHER REQUIREMENTS

- The application should be able to keep a database for all user data.
- The application should be easy to use/user friendly
- The graphical data should be displayed in a clear and neat way.
- The application response should be quick. (i.e. clicking a button, calculations)
- The application should require minimal maintenance.
- The application should be reliable, with minimal bugs/crashes and downtime.

Appendix B: Analysis Models

Data Flow Diagram (DFD):

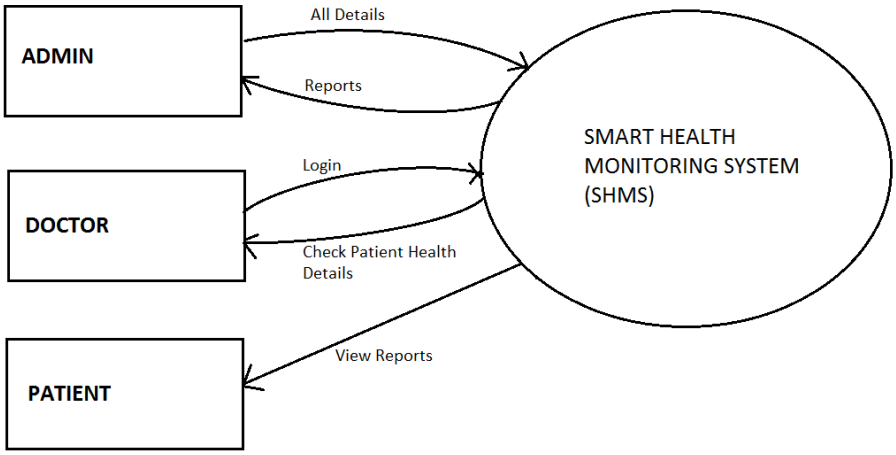


Fig: DFD (LEVEL-0)

Figures: Data Flow Diagram (Level-0)

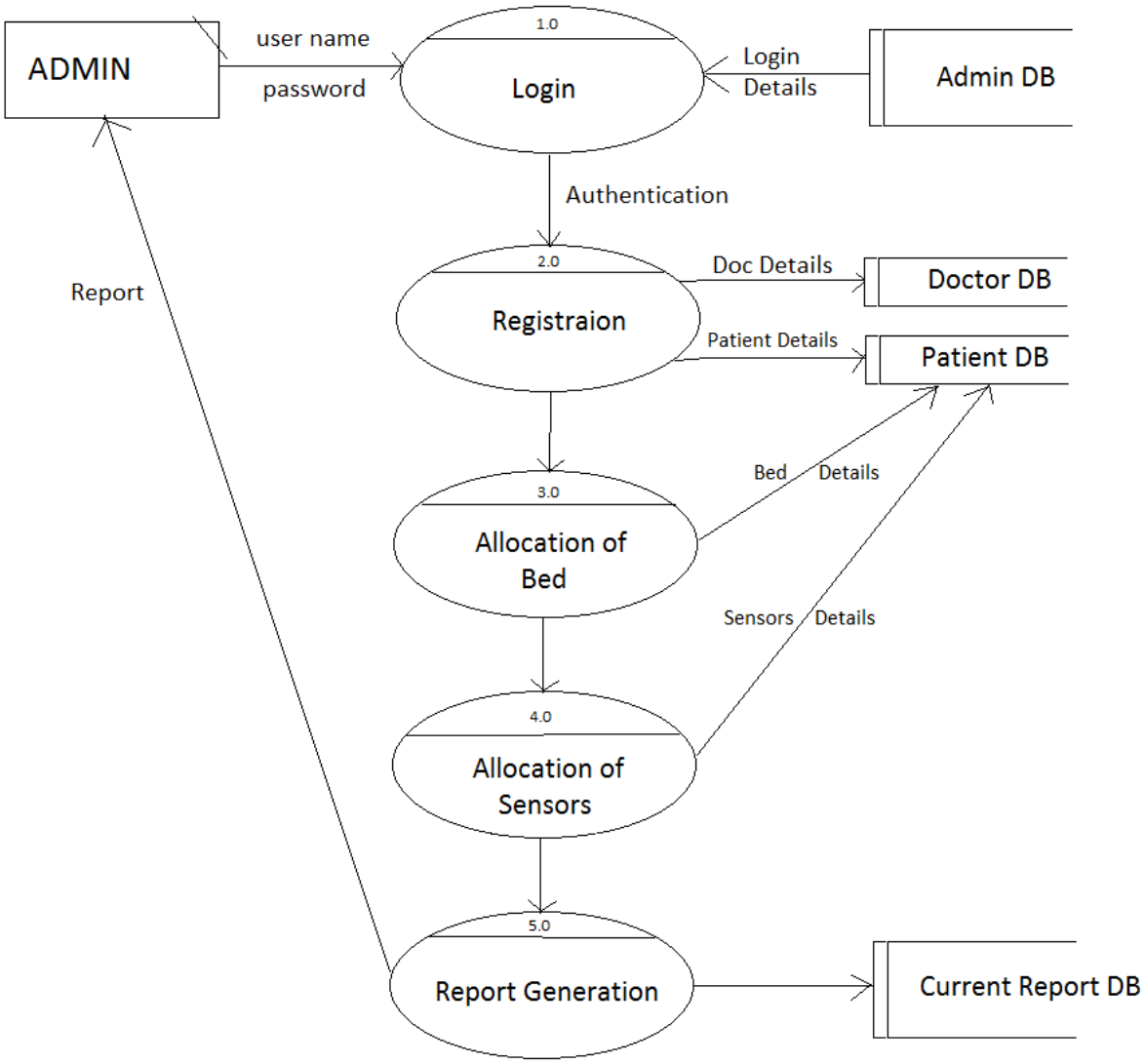


Fig: DFD (level-1)

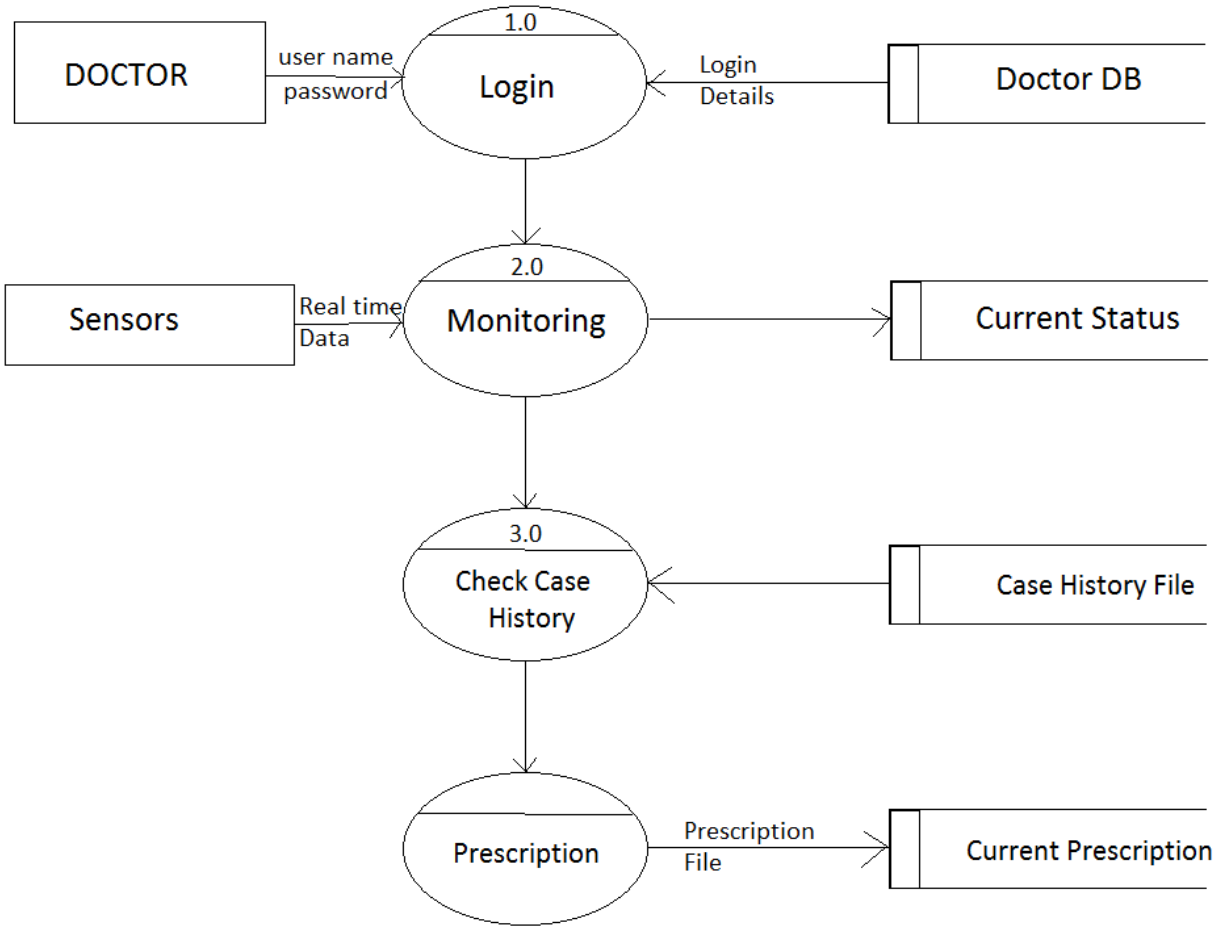


Fig: DFD (level-1)

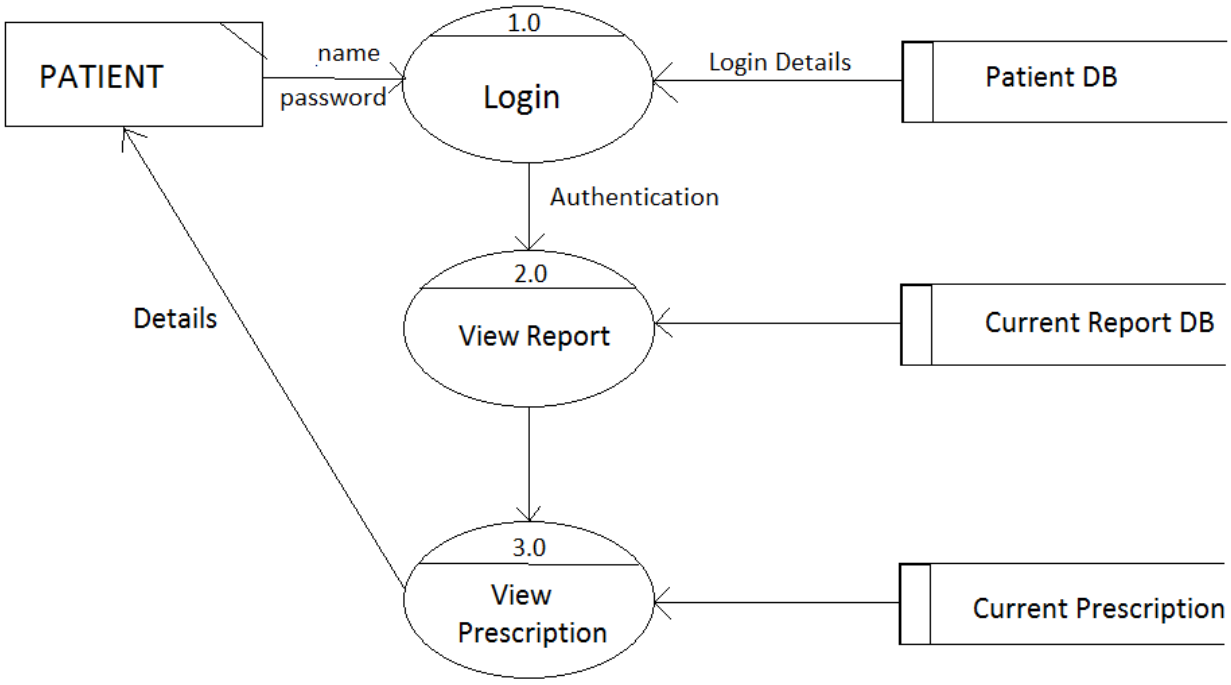


Fig: DFD (level-1)