PROJECT ON

BLOCKCHAIN HYPERLEDGER IN MEDICAL APPLICATION

SUBMITTED TO:

DR. DURGANSH SHARMA

SUBMITTED BY:

TEAM FOXTROT

S. No.	NAME	SAP ID	ROLL NO	BATCH
1.	RIBHU RISHI	500071293	44	B2
2.	SANSKAR DUA	500066533	48	B2
3.	ROHAN RATHORE	500071555	84	B2
4.	NITIN KUMAR	500071821	88	B2

INDEX

S No.	Content	Page No.
1.	Abstract	3
2.	Possible Implementation in Military Applications	4
3.	Objectives	5
4.	Minimum Hardware Requirement	6
5.	Agile Manifesto	7-8
6.	Software Characteristics i. Functionality ii. Reliability iii. Efficiency iv. Usability v. Maintainability vi. Portability	9-11
7.	Scrum Development	12-15
8.	Entity Relationship Diagram	16
9.	Data Flow Diagram	17-18
10.	PERT Chart	19
11.	ETVX (Entry criteria Task Verification and eXit Criteria)	20-21
12.	Outcome	22

ABSTRACT

Blockchain is a technology that is used as a basic structure for operation and control of modern digital transactions. It is a representation of indexing an entry in the ledger, and recording a transaction-taking place. All the transactions that take place contain a digital signature to ensure its authenticity, so that no one could temper with the data present in Blockchain. This technology allows digital information to be distributed and shared, but not to be copied. This in turn means that each individual possesses a defined portion of data that can only have one owner.

Any new information or data on Blockchain is added into the database, which is stored at multiple locations and updated at regular intervals. It makes the data to be public and verifiable. Since there is no data centralization, it is extremely hard to bruteforce, as the information exists simultaneously in millions of places at an instance.

POSSIBLE IMPLEMENTATION IN MILITARY APPLICATIONS

1. Preventing data theft.
2. Secure Government and Battlefield Messaging.
3. Tracking Defence shipments/contracts in real-time.
4. Military Manufacturing and processing.
5. Cyber Warfare Preparedness.
6. Maintaining health record of defence personnel.
7. Coordinated control over sophisticated weapons and ordinances.

OBJECTIVES

Following are the objectives of this project:

1. Implementing this project in order to facilitate a transparent and multi-user controlled database.
2. Bringing up a notion of a single data record in order to avoid any redundancy and ambiguity.
3. Distributed Peer-to-peer connection allows sharing of data with all the users at the same time.
4. Data once sent to a Blockchain network, cannot be deleted or removed from all the systems, preventing deletion of data from one source.
5. It will improve the overall data collection by removing any duplicates from the database.
6. Attacking on distributed storage is far more complex and difficult than when compared to centralized databases.
7. It allows users to interact with each other directly without any necessity of a mediating third party.

MINIMUM HARDWARE REQUIREMENT

Following are the Min. Hardware requirement for a member node system.

1. Disk Space: <=200 GB
2. Download: 250 MB/day (8 GB/Month)
3. Upload: 5 GB/day (150 GB/Month)
4. Memory (RAM): 2GB
5. Operating System: Windows, Mac OS X, Linux

AGILE MANIFESTO

The four Manifestoes of Agile Systems

In this project, we have tried to include the four Manifestoes of Agile System, as it will help in increasing the overall efficiency during development and therefore a better output will be generated.

Following are the four Manifestoes of Agile Systems:

1. **Individuals and Interactions** over processes and tools.
2. **Working software** over comprehensive documentation.
3. **Customer Collaboration** over contract negotiation.
4. **Responding to change** over following a plan.

We will be implementing the Manifestoes in the following manner.

Individuals and interactions over processes and tools**:**

In this project, we will be implementing the concept of interaction with each individual working on this project over every process and the tools we will be using. Moreover, decisions made during this project will be according to every individual's perspective.

Working software over comprehensive documentation**:**

In this project, we will be developing a working software rather than only preparing a comprehensive documentation. As we know, that documentation have a limited practical usability and even do not provide the customer with a first-hand perspective of how the actual software operates and functions. Therefore, we are stressing on building a working software in order to get the best out of user's requirements and feedback.

Customer collaboration over contract negotiation**:**

As we know, customer collaboration is the main framework on which this whole concept of building software relies. We will be focusing mainly on customer collaboration rather than stressing on contract negotiation.

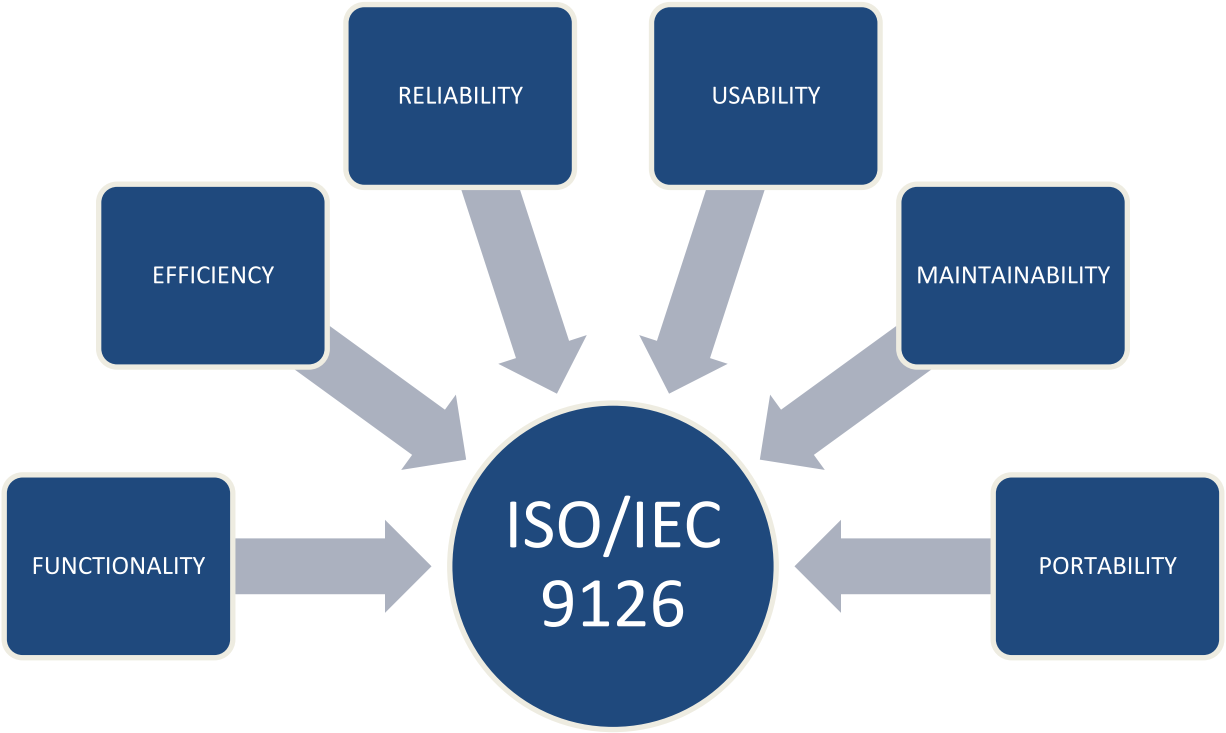
Responding to change over following a plan**:**

After we receive feedback, we will be implementing the same in our project in order to meet the customers' requirements and satisfaction which is a matter if great importance. The implementation of changes to be made shall be done in a planned manner.

Software Characteristics

Software Characteristics contains the following:

1. Functionality
2. Reliability
3. Usability
4. Efficiency
5. Maintainability
6. Portability



Functionality

- Suitability – Is the software suitable for client necessity
- Accuracy – Accuracy to meet the requirements of client
- Interoperability – Ability to perform on different types of machines/computers
- Compliance – To meet the standard instruction
- Security – To withstand any threat and keeping the records safe

Reliability

- Recoverability – To recover the data and stored record in case of unexpected failure.
- Fault Tolerance – To continue its normal operation in case of system or hardware fault/failure.
- Maturity – It is the ability to deal and adjust with forthcoming features.

Efficiency

- In Time – Utilization of time that is to be spent in order to prepare a fully functional.
- In Resource – This is the total resource provided by a team in order to develop a fully functional software.

Usability

- Understandability – To make a usable software that the user is able to interact with.
- Learnability – The project interface allow users to quickly understand and learn the way to use it and thus allowing them to work on it with ease.
- Operability – It must be able to be operated by a user.

Maintainability

- Testability – Use the software, and then testing it on the benchmark for all possible inputs.
- Stability – The software must be stable for all the inputs a user can provide.
- Changeability – It allows a simplified procedure in order to make any change in the software after testing.
- Operability – This step involves the overall check of how well the software works while operating.

Portability

- Adaptability – Adaptable to systems
- Installability – It can be installed on systems of the user.
- Replicability – It is able to be replaced with a newer version and is compatible to run an older version.

SCRUM DEVELOPMENT

Scrum is one of the most important methodology of Agile Systems. It is a framework for developing complex products and systems. People usually thinks that any agile model especially scrum is for small projects but contrary to this believe, scrum is used for developing complex systems and models. It is grounded in empirical process and control theory. Empirical means to do something and adapt it make changes and again adapt it. It follows iterative and incremental approach.

Functionality and workflow of Scrum

All it starts with product vision. The stakeholders, customers come up with the vision of the product. Based on the high-level vision, there is something called product backlog, which is nothing but series of features, which the product inherits. Product backlog undergoes many increments and decrements. From product backlog, the scrum team pulls a manageable chunk, which is of high priority into something called sprint backlog. This sprint backlog will be executed, designing, testing, coding of all the features for a particular period, come out with the working products and keep the product piece by piece and integrate building the model with the vision.

Scrum roles

The roles in scrum are quite different from the traditional software methods. Defined roles helps the individuals perform their tasks efficiently. In scrum, there are three roles, product owner, and development team and scrum master.

- **Product Owner:** The product owner represents all the possible stakeholders and the customer in each case. It has the responsibility of delivering the maximum possible value to the customer. The product owner does not follow the traditional methods of getting things done rather s/he prioritizes things on the basis or dependencies and importance. A scrum team should have only one product owner. This role is different from that of the scrum master. The product owner focuses on the business part of a product and acts as an intermediate between the stakeholders and the team. The product owner is no dictator but a consensus maker. Therefore, a good product owner should be able to understand the needs of the business, the reasons behind those needs and communicate them to the stakeholders and the concerned team.
- **Development Team:** The development team is responsible for delivering optimum quality work with each passing sprint. The sprint goal describes the work a team should do and the quality it must deliver at the end of each sprint. The team has from three to nine members who carry out all tasks required to build the product increments. Although a team consists of people from various backgrounds all of them are generally, referred to as

developers. It is a self-organizing team and the product owner or the scrum master manages interactions outside the team.

- **Scrum Master:** Scrum master is responsible for facilitating a scrum, ensure that there are no hassles on the way of quality product or deliverable being delivered. The scrum master acts as a mediator between the team and any hassles on the path of work and is not necessarily the team leader. The scrum master ensures that the Scrum framework is followed. The scrum master encourages the team to improve. The role has also been referred to as a team facilitator or servant-leader to reflect the dual aspects.

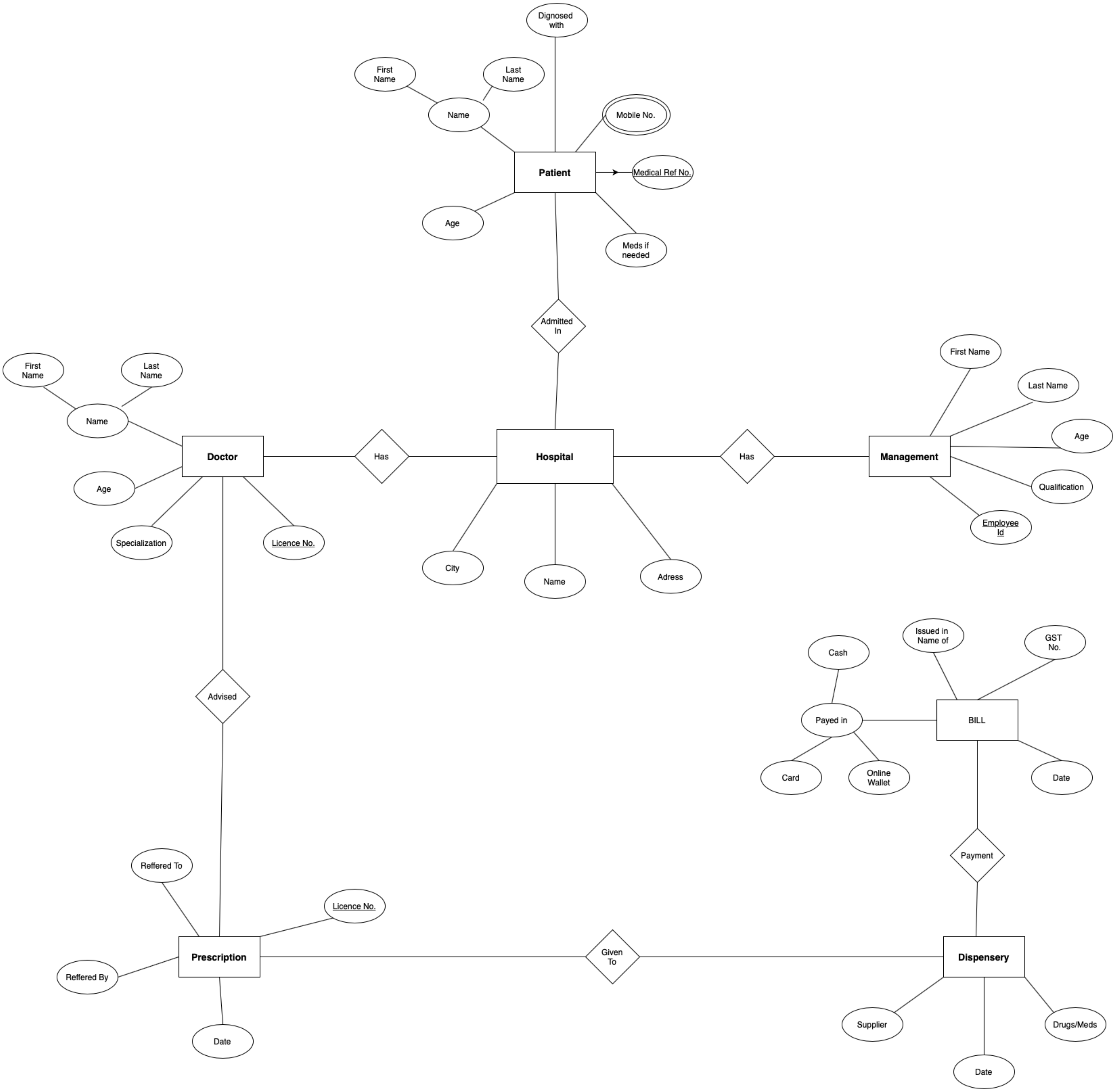
Induction of Scrum Development Methodology

We have divided out projects into different sprints. As this development model based upon dividing the overall development into small sections called sprints. These sprints contains a sectional phase, which will be completely developed and then held aside. As these sprints reach their final stages and finalized at the end, they are compiled together to form a fully functional software.

Advantages of using Scrum methodology

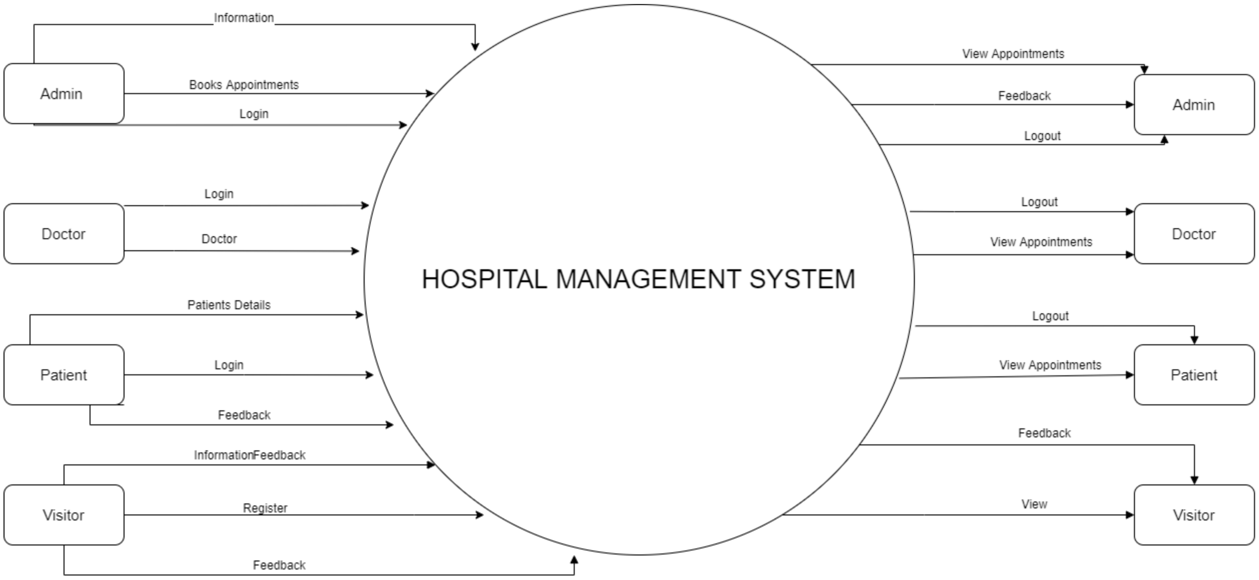
1. Fast moving, innovative developments can be quickly coded and tested using this method, as a mistake can be easily rectified.
2. Due to short sprints and constant feedback, it becomes easier to cope with the changes.
3. It is easier to deliver a quality product in a scheduled time.
4. Daily scrums do more than keep workers accountable to their assignments. It is also a way for a company to maintain their transparency with their clients.
5. Because this methodology requires daily check-ins for progress reports, there is always feedback offered at the team and individual level. This helps to make the project better in the end.

E-R Diagram

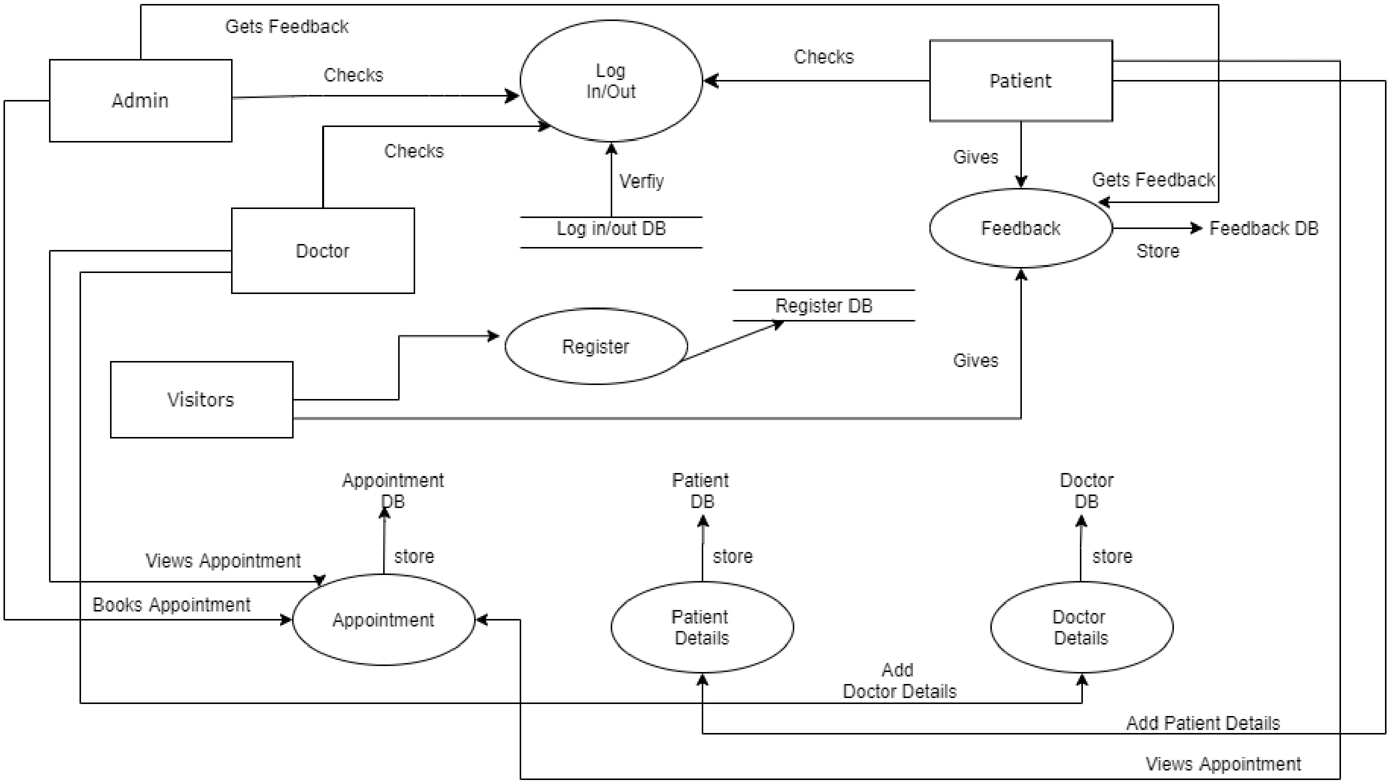

Dignosed with
First Name
Last Name
Name
Mobile No.
Patient
Medical Ref No.
Age
Meds if needed
Admitted In
First Name
Last Name
Name
Doctor
Age
Specialization
Licence No.
Has
Hospital
City
Name
Adress
Has
Management
First Name
Last Name
Age
Qualification
Employee Id
Issued in Name of
Cash
GST No.
Payed in
BILL
Card
Online Wallet
Date
Advised
Payment
Reffered To
Licence No.
Prescription
Reffered By
Date
Given To
Dispensery
Supplier
Date
Drugs/Meds

DATA FLOW DIAGRAM

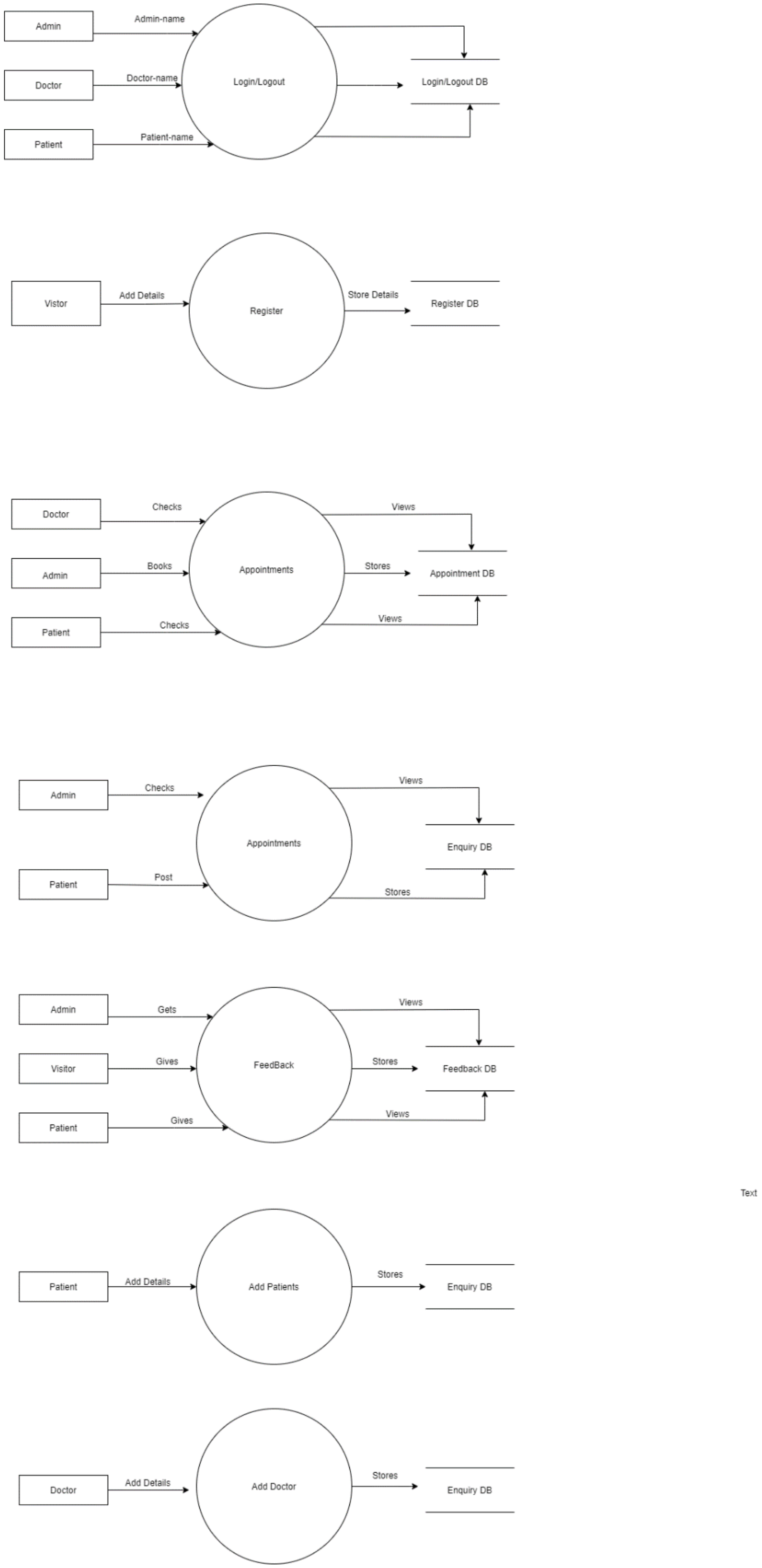
LEVEL 0



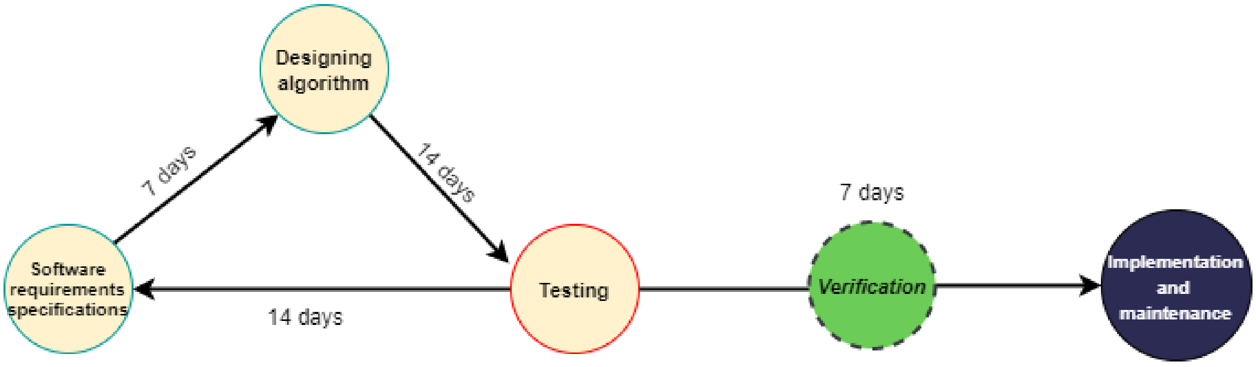
LEVEL 1



LEVEL 2


Admin
Admin-name
Doctor
Doctor-name
Patient
Patient-name
Login/Logout
Login/Logout DB
Vistor
Add Details
Register
Store Details
Register DB
Doctor
Checks
Admin
Books
Patient
Checks
Appointments
Views
Stores
Views
Appointment DB
Admin
Checks
Patient
Post
Appointments
Views
Stores
Enquiry DB
Admin
Gets
Visitor
Gives
Patient
Gives
FeedBack
Views
Stores
Views
Feedback DB
Text
Patient
Add Details
Add Patients
Stores
Enquiry DB
Doctor
Add Details
Add Doctor
Stores
Enquiry DB

PERT CHART


Designing algorithm
7 days
14 days
Software requirements specifications
14 days
Testing
7 days
Verification
Implementation and maintenance

PERT (Project Evaluation and Review Technique) chart is a tool, which shows project as network diagram. It was developed in 1950 for US navy. It reduces both, the time and the cost requirement for completing a project.

Steps involved in making pert chart:

1. Identify the specific activities and milestones.

2. Determine the proper sequence of the activities.

3. Construct a network diagram.

4. Calculate the estimate time required for each event or activity.

5. Determine the critical path.

6. Update the project as per the pert chart.

Entry criteria **T**ask **V**erification e**X**it criteria

System requirement specification (SRS):

- **Entry:** There are no entry criteria for this; it simply forwards it to the next phase
- **Task:** To state the minimum system requirement to the requesting party. And to take permission from the client to check their system specification
- **Verification:** To verify the system specification, if the permission was granted then crosscheck the client system specification then forward the user to the next phase. If the permission was not granted then the user proceeds towards the exit.
- **Exit:** If the process was verified then forward towards the next process. Else Whole process ends.

Designing Algorithm:

- **Entry:** If the system supports the Designing algorithm and is readable by the same, then it proceeds to the next phase
- **Task:** The system will then read and compile the given algorithm
- **Verification:** Compiled code will be provided to the user & if the feedback is positive from user side than it proceeds to next phase otherwise it sends the info about the faulty code to the management system and then proceeds to exit.
- **Exit:** Once the developer designs a suitable algorithm according to the SRS received, it is further sent for the testing.

Testing:

- **Entry:** If the system has a proper testing kit, then it moves to the next phase, or it provides a suggestion for some testing kit or directs it to the exit phase
- **Task:** The testing kit gets in the play and approves or finds out bugs in the algorithms and codes.
- **Verification:** If the testing kit gives a green signal then it, proceeds to the next phase, else it makes a fault report and send it to management.
- **Exit:** Moves to the next process if the code is right, else, the fault code is sent to management.

Management and Implementation:

- **Entry:** All the data of the System passes through this, so there is no restriction of entry in this phase.
- **Task:** If all the working is all good then the system passes the daily work report without any error. However, if the Maintenance system receives a fault report then it passes it to the next phase.
- **Verification:** Here the system verifies that the fault report sent are from what department and then forward it to the technical personnel of the department in charge.
- **Exit:** The faults are sent to the tech team when the problem is solved, a notification is sent to the user by whom the problem was encountered. Moreover, in case no fault report is found the system generates daily work report.

OUTCOME

Following are the outcome of this project:

1. Decentralization: Since the data on Blockchain is distributed among all the participating users, it makes data more secure and recoverable.
2. Resilience: It will improve the capability of data to be secure in case of a massive attack.
3. Transparency: This technology allows all the peers to view the detail of transactions taking place on this network.
4. Security: Implementation of Blockchain Technology will actively increase security in maintain a record as no one node controls the data as in case of centralized databases.