DIGITAL PRESCRIPTION

Project Report

SUPERVISOR:

Dr. Vibha Gaur

SUBMITTED BY:

Gaurav Lohkna	Z-1219
Daya Shankar	Z-1216
Ramesh C. Pant	Z-1243
Arvind Kumar	Z-1212

2020

Department of Computer Science
ACHARYA NARENDRA DEV COLLEGE, UNIVERSITY OF DELHI.

ACKNOWLEDGMENT

Apart from the efforts of the team, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.
The completion of any inter-disciplinary project depends upon cooperation, co-ordination and combined efforts of several sources of knowledge.

We are eternally grateful to our **Dr. Vibha Gaur** for her even willingness to give us valuable advice and direction under whom we executed this project. Her constant guidance and willingness to share her vast knowledge made us understand this project and its manifestations in great depths and helped us to complete the assigned tasks.

We are highly thankful to our project's internal guide **Dr. Vandita Grover**, and **Dr. Baljeet Kaur** whose invaluable guidance helped us understand the project better. Although there may be many who remain unacknowledged in this humble note of gratitude, there are none who remain unappreciated.

Gaurav Lohkna   Daya Shankar   Ramesh C. Pant   Arvind Kumar

ACHARYA NARENDRA DEV COLLEGE
(University of Delhi)

CERTIFICATE

This is to certify that Software Engineering project report entitled "Digital Prescription" is the work carried out by **Gaurav Lohkna**, **Daya Shankar**, **Ramesh C. Pant** and **Arvind Kumar** students of BSc(H) Computer Science IV Sem, Acharya Narendra Dev College, University of Delhi under the supervision of **Dr. Vibha Gaur**.

This report has not been submitted to any other organization/institution for award any other degree/diploma.

Gaurav Lohkna Daya Shankar Ramesh C. Pant Arvind Kumar

Supervisor
(Dr.Vibha Gaur)

Principal
(Dr. Ravi Toteja)

CONTENTS

PROBLEM STATEMENT

Doctors’ handwriting can only be understandable either by the doctors themselves or by the pharmacist.
And sometimes the doctors write the prescription in a hurry or say too sloppy that it becomes hard to understand the prescription-even by the pharmacist- and this leads to inconsistency, and an unwanted prescribed drug can be given to the person just by the wrong understanding of the medication’s name and this may lead to the death of that person. To eliminate the risk of handwriting misunderstanding, we have come across an idea as follows :

Designing an app to write formatted prescriptions based on dictation from a doctor. The app will be able to send the prescription to the patient directly on his phone and email id. A Doctor should be able to dictate his prescription to the patient while talking to his phone or PC running windows.

The Chemist should be able to either preview or hear the prescription as prescribed or signed by the doctor with a digital signature on the prescription.

PROCESS MODEL

A software process model is an abstraction of the actual process, which is being described. It can also be defined as a simplified representation of a software process. Each model represents a process from a specific perspective.

In this project of ours, we are using the EVOLUTIONARY MODEL.

Evolutionary Model:[1]

The evolutionary model is a combination of the Iterative and Incremental model of the software development life cycle. Delivering your system in a big bang release, delivering it in an incremental process over time is the action done in this model. Some initial requirements and architecture envisioning need to be done.
It is better for software products that have their feature sets redefined during development because of user feedback and other factors. The Evolutionary development model divides the development cycle into smaller, incremental waterfall models in which users are able to get access to the product at the end of each cycle.

Feedback is provided by the users on the product for the planning stage of the next cycle and the development team responds, often by changing the product, plan or process. Therefore, the software product evolves with time.

Our project is falling under the prototype evolutionary model because there will be many phases and parameters before we deliver the final product to the client or the stakeholders and because our project/product needs to be accurate to the stakeholders and if there is any mistake, error or flaw then it may lead to the wrong prescription to the patients which are not tolerable in the medical markets. All the models have the disadvantage that the duration of time from the start of the project to the delivery time of a solution is very high. The evolutionary model solves this problem in a different approach.

Advantages:

- In an evolutionary model, a user gets a chance to experiment with a partially developed system.
- It reduces the error because the core modules get tested thoroughly.

Disadvantages:

- Sometimes it is hard to divide the problem into several versions that would be acceptable to the customer which can be incrementally implemented and delivered.

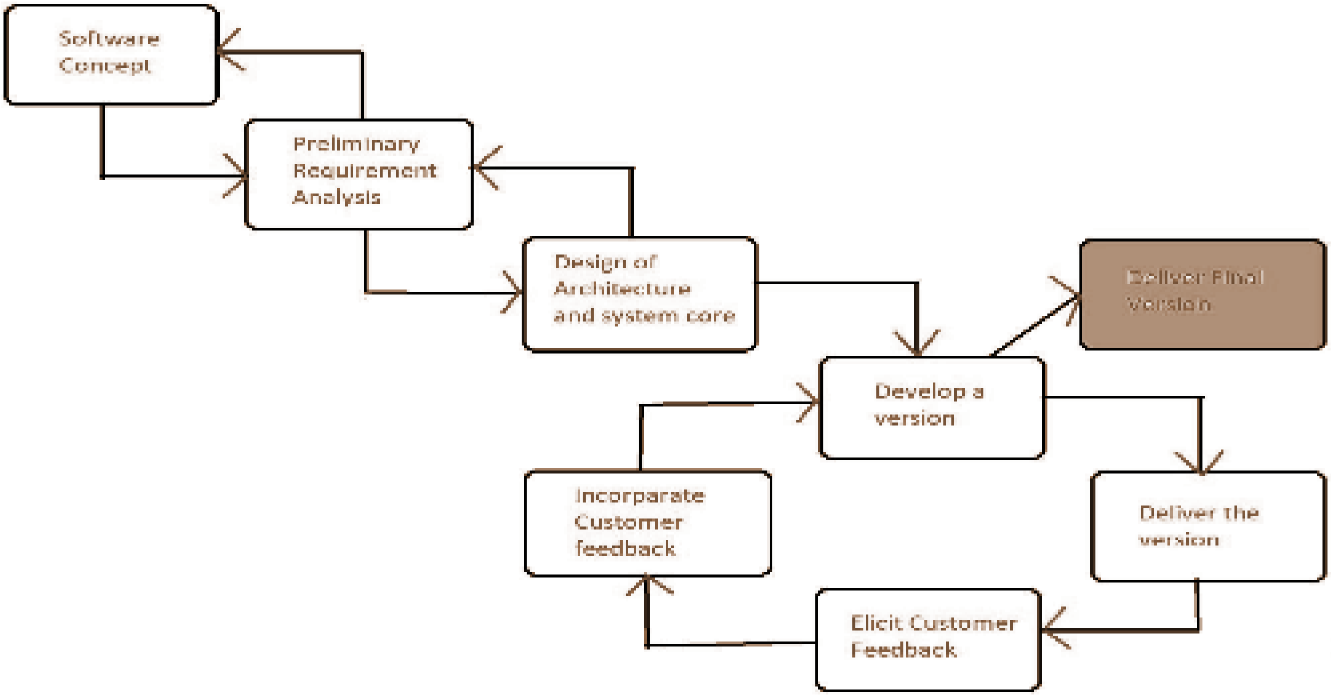

Software Concept
Preliminary Requirement Analysis
Design of Architecture and system core
Deliver Final Version
Develop a version
Incorparate Customer feedback
Deliver the version
Elicit Customer Feedback

image source [1] Diagram of *Evolutionary model.*

Reasons, why our team is using an Evolutionary model, are as follows:

- We are a small team, encompassing only 4 members.
- We know that our software team is not very trained and skilled, so we keep on learning and keep on adding new functions or modules.
- We are short on time and we have to deliver this product within 4-5 months, so we need not waste our time doing things that can be done later on - in the next version maybe.
- Our team found the Evolutionary model a little bit more flexible and less expensive to adjust the requirements of the client/stakeholder.
- And the most important reason to use this model is that the software will be generated quickly during the software life cycle.

Requirements of the stakeholders:

The keywords that are needed to be tagged as follows:
Name, symptoms, diagnosis, prescription, advice, etc.

- For all keywords, data should be captured discreetly.
- The doctor should be able to edit/delete by voice or hand any entry that has been made.
- The doctor should be able to preview the prescription.
- The patient should get the PDF document either directly or as a link in SMS/WhatsApp.

Annually the death rate due to the wrong prescription leads to around 250 thousand [1] deaths in the US alone.

SOFTWARE REQUIREMENTS SPECIFICATION(SRS)

The requirements for a system are the descriptions of the services that a system should provide and the constraints on its operation. These requirements reflect the needs of customers for a system that serves a certain purpose such as controlling a device, placing an order, or finding information. The process of finding out, analyzing, documenting and checking these services and constraints is called requirements engineering (RE).

A software requirements specification (SRS) is a document that is created when a detailed description of all aspects of the software to be built must be specified before the project is to commence. Software system requirements are often classified as functional or non-functional requirements:

3.1 OVERALL DESCRIPTION

The Digital Prescription is intended to give the prescribed prescription to the patient using a formatted voice input from the doctor and then the doctor can either preview or share the prescription to the patient through the provided option and hence the patient will be able to pay the total cost of the prescription through the provided payment gateways.

3.1.1 PRODUCT FUNCTIONS

- Registration for both the doctors and the patients.
- Login and forget password mechanisms for both the patients and doctors.
- The doctors will then provide a prescription using formatted voice-input.
- The doctors, after the completion of the prescription, can either edit or share it.
- The patients will receive the prescribed prescription from the doctors.
- The patients will pay the cost of the prescription.

3.1.2 USER CHARACTERISTICS

- Both doctors and patients must be familiar with app usage and the internet.
- The doctors must know the correct pronunciation of the prescription.
- The patients must have knowledge of online payment methods.

3.1.3 GENERAL CONSTRAINTS

- Easy to use and easy to understand User Interface, so that no training should be required for any type of user.
- An active internet connection is required to use the software, as the software will be using the google voice recognition module.

3.1.4 ASSUMPTIONS AND DEPENDENCIES

- A proper working microphone or any compatible device, as recognizing the voices is the backbone of our software.
- The patients and doctors must be logged in to use the software.
- The doctor and patient must have proper knowledge of other related applications to view and preview the prescription.
- The device or platform should be compatible with the software.
- The users must have given all the permission to use the software like recording audio and saving into the device.

3.2 EXTERNAL INTERFACE REQUIREMENTS

This explains the detailed description of all the input and output processes of the software. There will be a proper description of software and hardware requirements for the software. In this, we'll also explain the interface between the user and software and hardware.

3.2.1 USER INTERFACE

- The software is a graphical user interface(GUI) for the ease of use of the users.
- The first time users will have to register themselves using the registration page or if they have an account registered then simply login.
- Every patient should have a profile page where they can edit their personal details as well as medical details.
- The software's GUI will consist of menus, drop-down lists, buttons, and icons for ease of use.

3.2.2 HARDWARE INTERFACES

The software will require no additional hardware interfaces other than a microphone in a case where there is no in-built microphone in the user's device.

3.2.3 SOFTWARE INTERFACES

- The software will require the use of other applications's interaction for proper functioning.
- A PDF file viewer will be required to review the prescription.
- A payment gateway to transfer and accept the fee.
- It will require add-on software or module i.e. Google voice recognition for taking audio inputs.

3.3 FUNCTIONAL REQUIREMENTS

These are statements of services the system should provide, how the system should react to particular inputs, and how the system should behave in particular situations. In some cases, the functional requirements may also explicitly state what the system should not do.

3.3.1 FUNCTIONAL REQUIREMENTS(DOCTOR)

FR 1: Signup(Doctor)

The doctor will Sign up and will fill The Details Given Below:

- Name
- Degree
- Expertise
- Address
- Contact Details(Phone no, Email)

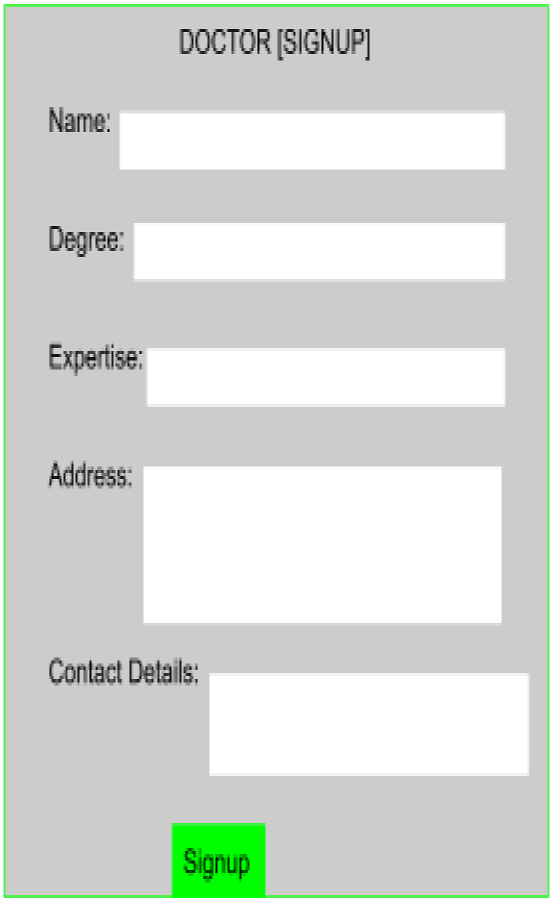


Fig: 3.1.1 Registration page for the Doctors.

FR 2: Login

The doctor will Enter the Email and Password.

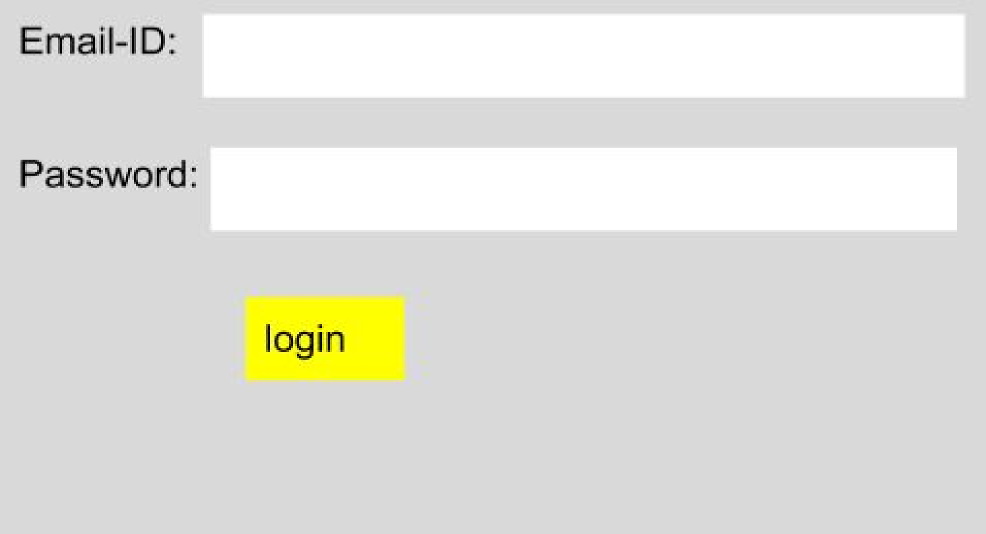


Fig: 3.1.2 Login page for the Doctor.

FR 3: Forgot Password

If the user(doctor)Forgot his Password then mail verification will be sent for resetting Password.

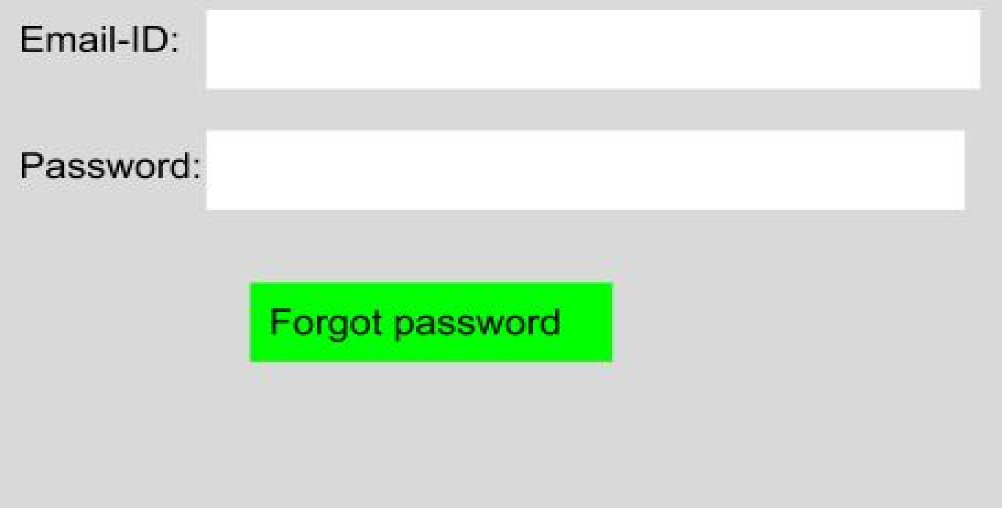


Fig:3.1.3 Resetting the password.

FR 4: Home Page

The doctor can see all the details like the history of patients' medical records etc.

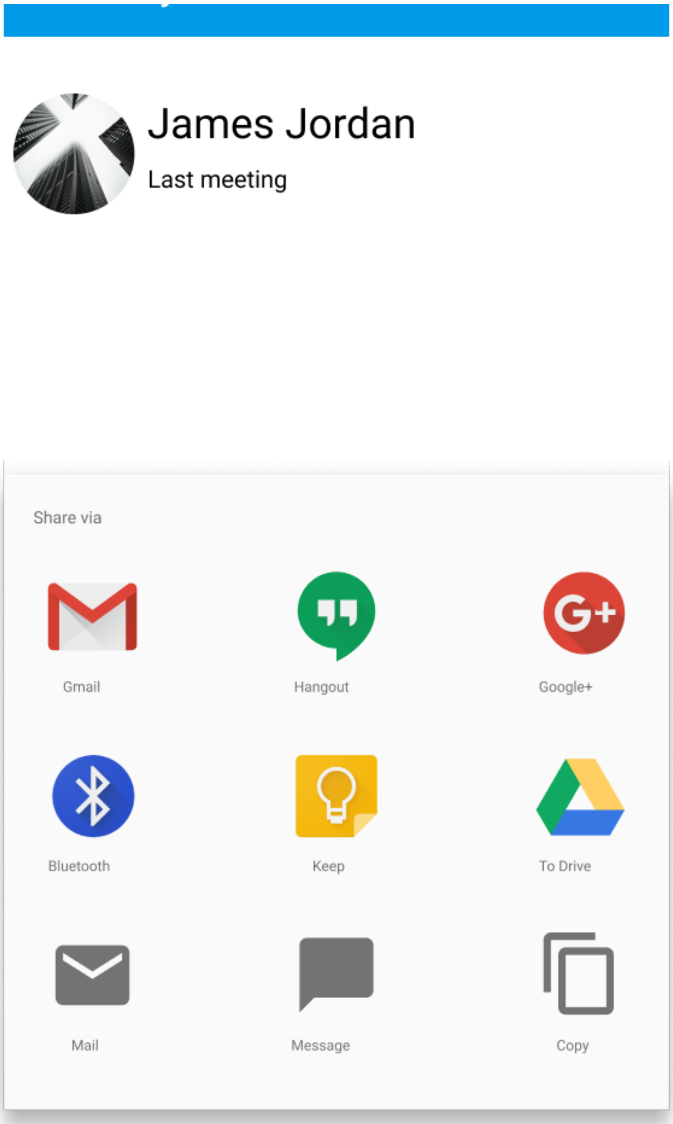
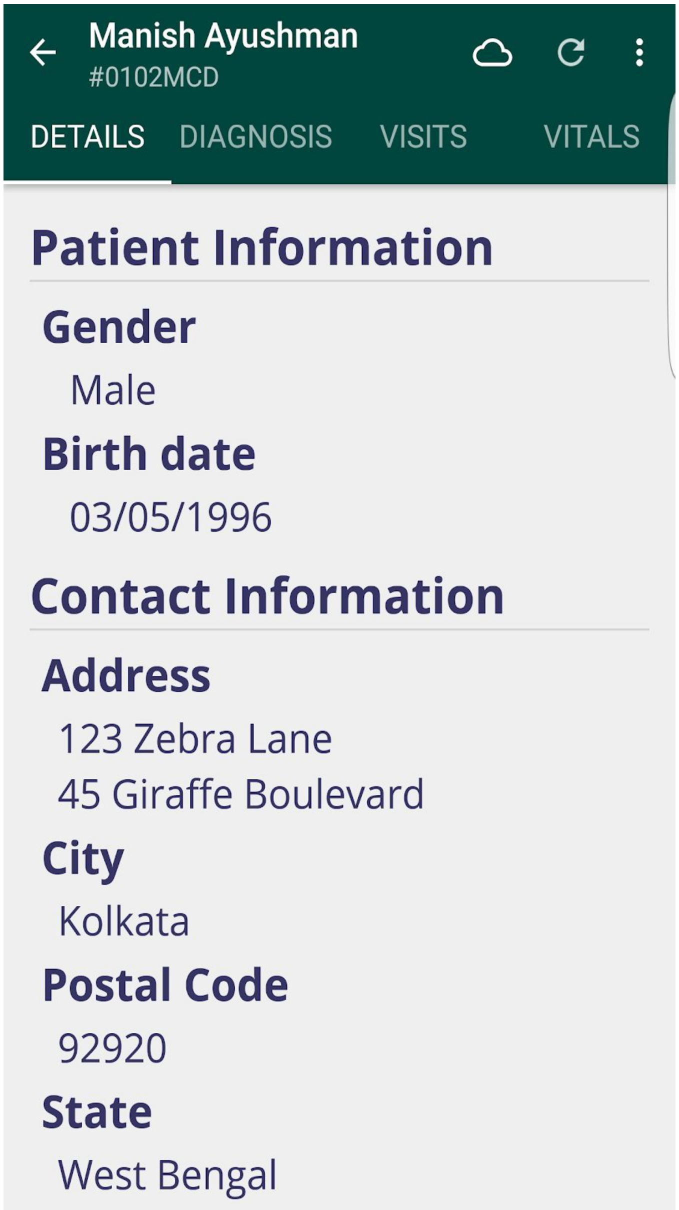


Fig:3.1.4 The homepage for the Doctor.

FR 5: Patient Details

The doctor can see current Patient Details like Name, Age, Diagnosis based on the token number.



Patient Information

Gender

Male

Birth date

03/05/1996

Contact Information

Address

123 Zebra Lane
45 Giraffe Boulevard

City

Kolkata

Postal Code

92920

State

West Bengal

Fig:3.1.5 The patient details page.

FR 6: Diagnosis Sheet and Prescription Sheet

The recorder will auto-fill The details Of the Patient with the voice of the doctor.

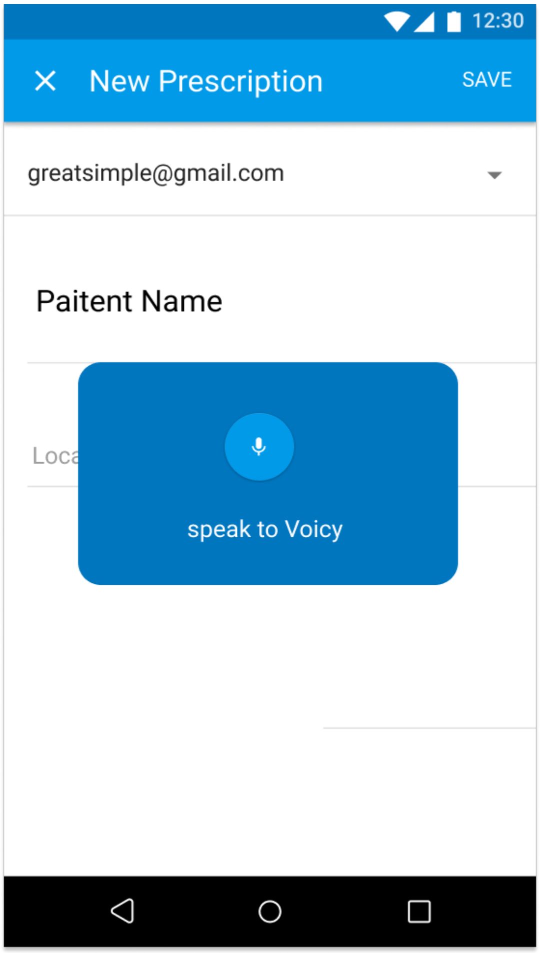


Fig:3.1.6 Prescription sheet and voice recognizer

FR 7: Preview of Sheet

On Completion The Doctor can Preview The Sheet and Can also edit it.

FR 8: Sharing Prescription

The doctor will share The Sheet (email, WhatsApp) to Patient (and it will automatically get reflected in patient’s app)

3.3.2 FUNCTIONAL REQUIREMENTS(PATIENT)

FR 1: Signup(Patient)

The patient will Sign up and will fill The Details Given Below:

- Name
- Age
- Sex
- Address
- Contact Details(phone no and Email-ID)

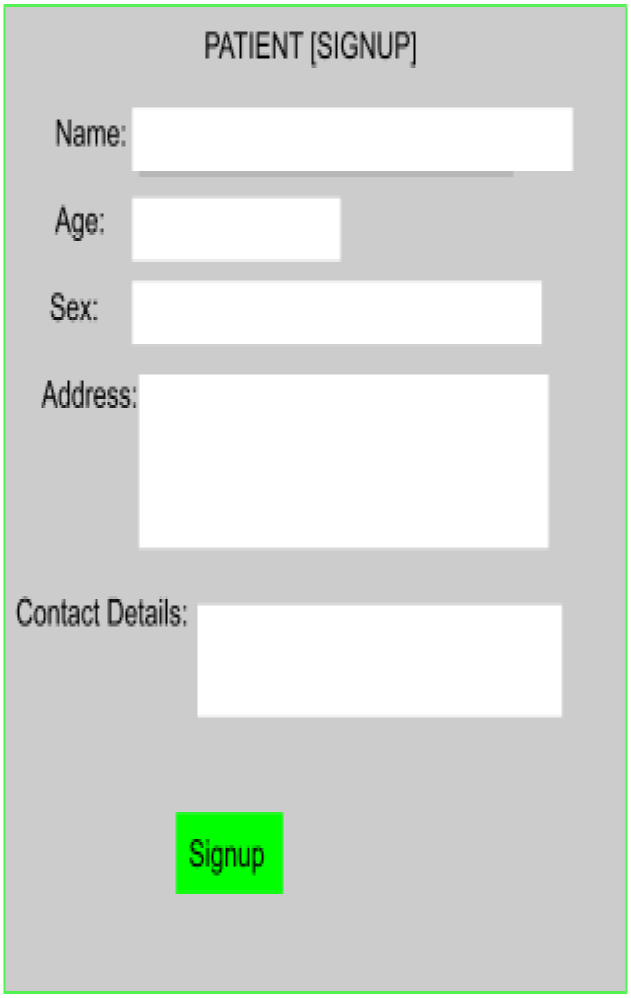


Fig:3.2.1 Registration for the Doctor.

FR 2: Login

The patient will Enter Email and Password.

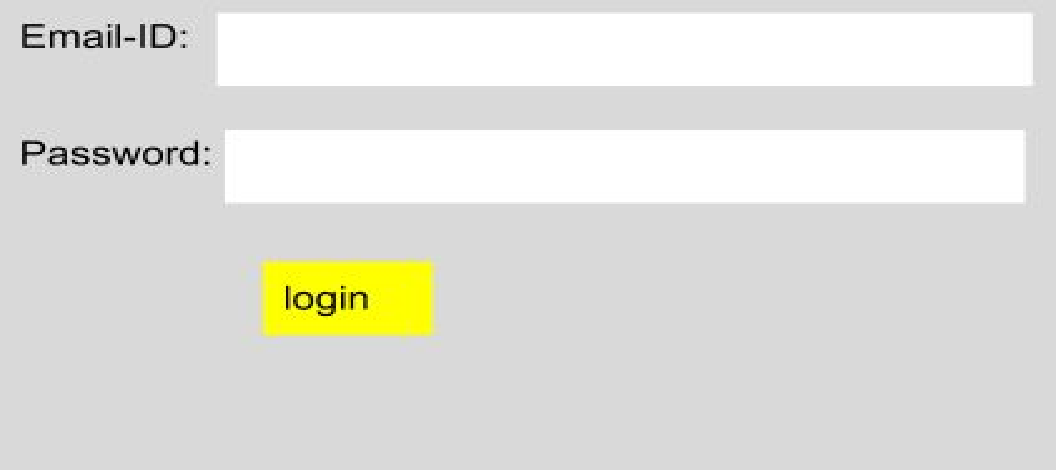


Fig:3.2.2 The login screen for the patient.

FR 3: Forgot Password

If a user Forgot his Password then mail verification will be sent for Setting a new Password.

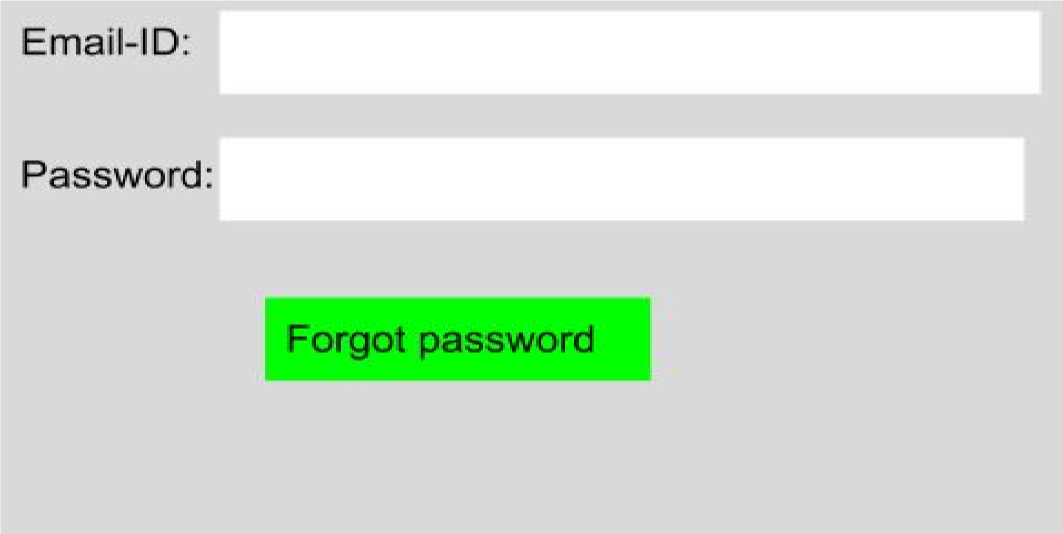
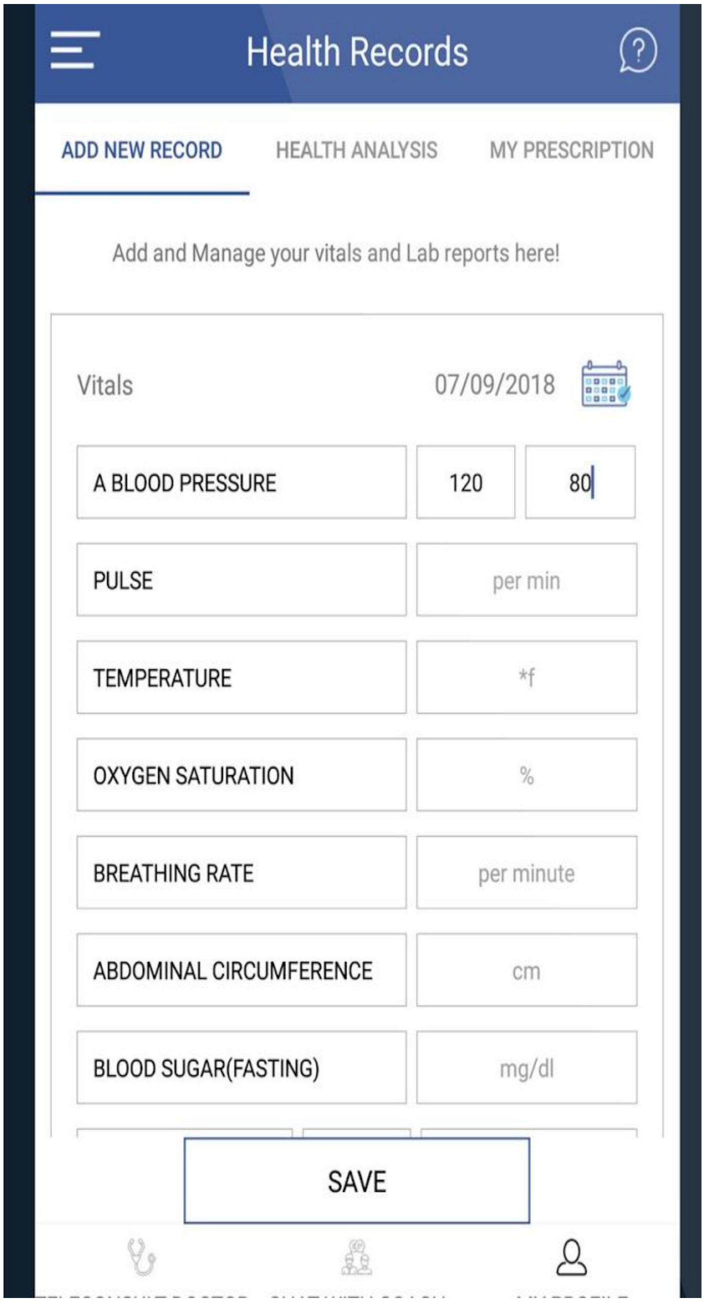


Fig:3.2.3 Resetting the password

FR 4: Home Page

The patient can see all the details, History of the previous check-ups, health analysis prescriptions.



Add and Manage your vitals and Lab reports here!

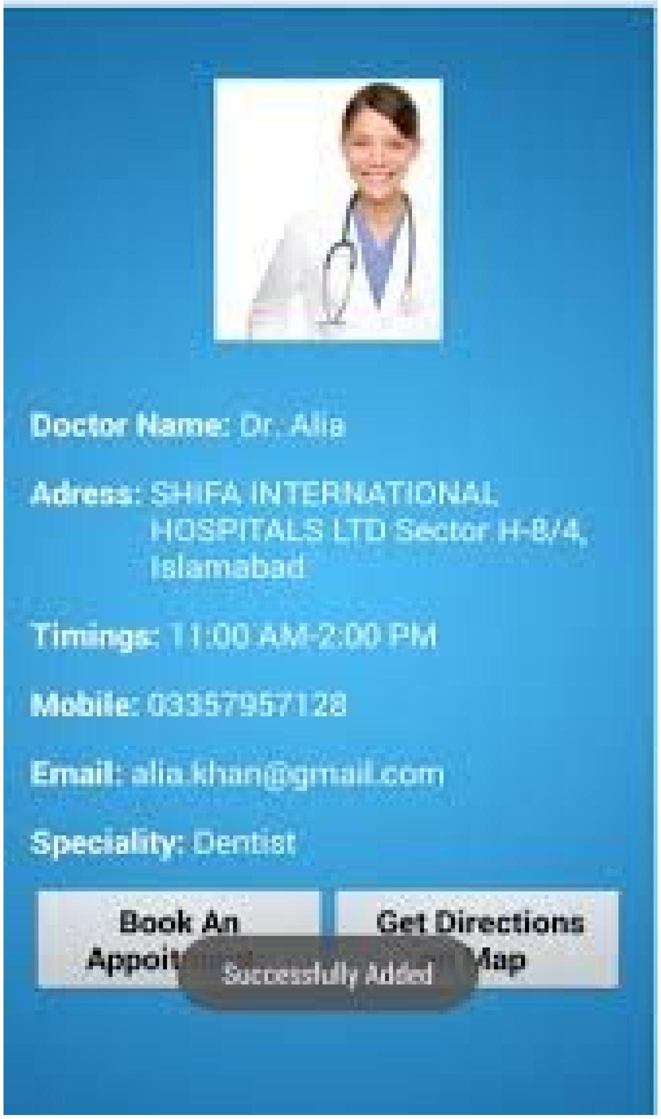
Vitals 07/09/2018

A BLOOD PRESSURE	120	80
PULSE	per min	
TEMPERATURE	*f	
OXYGEN SATURATION	%	
BREATHING RATE	per minute	
ABDOMINAL CIRCUMFERENCE	cm	
BLOOD SUGAR(FASTING)	mg/dl	

Fig:3.2.4 The patient’s homepage screen.

FR 5: Doctor Details

The patient can see Doctor Details like Name, speciality.



Doctor Name: Dr. Alia

Adress: SHIFA INTERNATIONAL HOSPITALS LTD Sector H-8/4, Islamabad

Timings: 11:00 AM-2:00 PM

Mobile: 03357957128

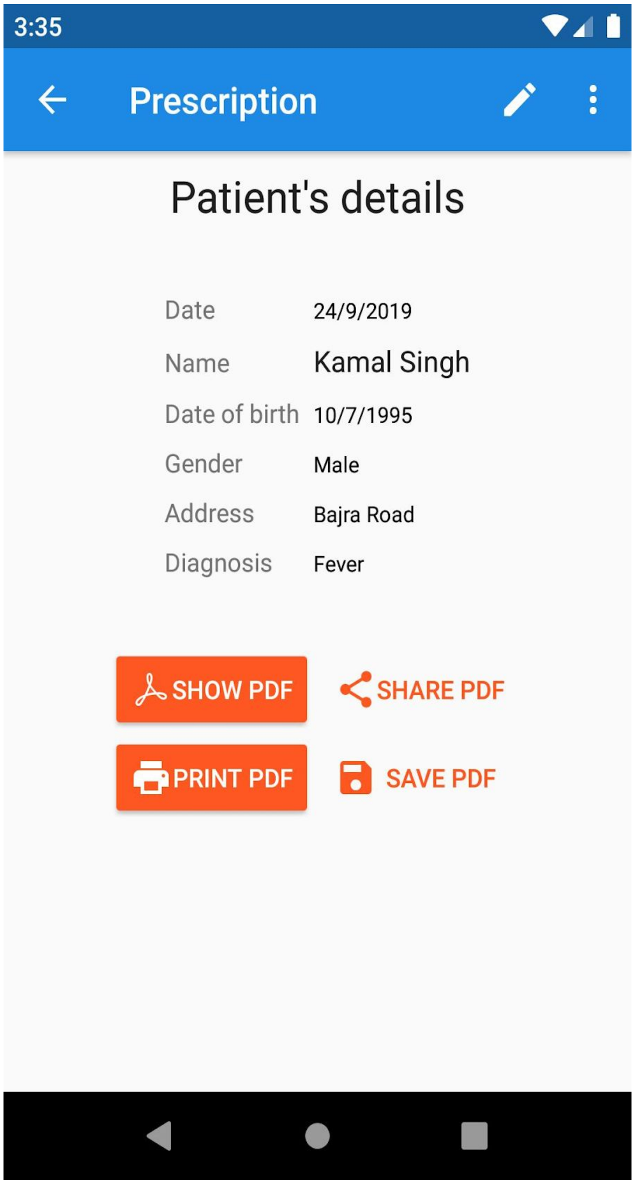
Email: alia.khan@gmail.com

Speciality: Dentist

Fig:3.2.5 Doctor details screen for the patients

FR 6: Receiving Sheet

On Completion, The Patient will Receive The Sheet and Can Preview it but cannot edit it.



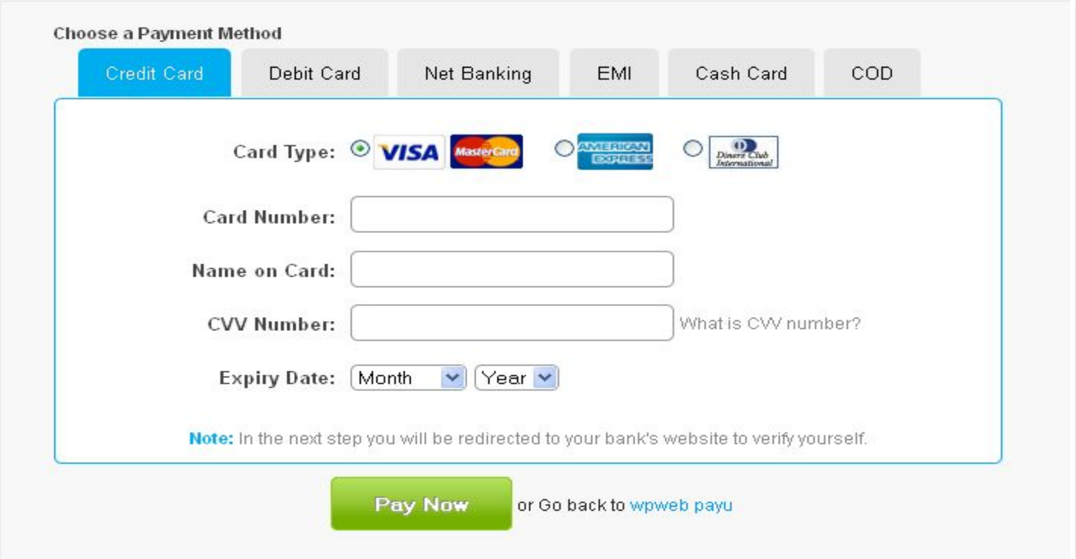
Patient's details

Date	24/9/2019
Name	Kamal Singh
Date of birth	10/7/1995
Gender	Male
Address	Bajra Road
Diagnosis	Fever

Fig:3.2.6 The prescription receiving sheet

FR 7: Payment Options

After previewing the prescription the patient can choose between many payment options at the payment gateway.



Card Type: ● VISA MasterCard ○ AMERICAN EXPRESS ○ Diners Club International

Card Number:

Name on Card:

CVV Number:

Expiry Date: Month Year

Note: In the next step you will be redirected to your bank's website to verify yourself.

Fig:3.2.7 The payment gateway screen.

3.4 PERFORMANCE REQUIREMENTS

Performance requirements define how well the system performs certain functions under specific conditions. Examples are speed of response, throughput, execution time, and storage capacity. The service levels comprising performance requirements are often based on supporting end-user tasks. Like most quality attributes, performance requirements are key elements when designing and testing the product. Performance requirements need to be considered along with other types of quality attributes.

- The system must be interactive and the delays involved must be less. So, in every action response of the system, there are no immediate delays.
- The audio capturing quality must be excellent so that no wrong prescription will be given.
- The speed of recognizing the words should match the speed of the doctor's speech.

3.5 DESIGN CONSTRAINTS

Design constraints are those constraints that are imposed on the solution space or ESS white box. These constraints are typically imposed by the customer, by the development organization, or by external regulations. The constraints may be imposed on the hardware, software, data, operational procedures, interfaces, or any other part of the system. Design constraints can have a significant impact on the design and should be validated prior to imposing them on the solution

Design constraints of our system will be:-

- Internet connection is the mandatory need for the access of the software, as the software requires the use of the google library.
- The system will work properly in the mobile device(android and IOS)as compared to the PC version.
- The software will be single user not multi user or account.
- The system may require credentials, personal information, and health information.

ESTIMATIONS

Estimation of the size of the software is an essential part of Software Project Management. It helps the project manager to further predict the effort and time which will be needed to build the project. Various measures are used in project size estimation. Some of these are:

- Lines of Code
- Number of entities in ER diagram
- Total number of processes in a detailed data flow diagram
- Function points

We will basically follow the *function points* approach to calculate the size of the project and its estimation.

4.1 FUNCTION POINTS

Function Point Analysis (FPA) is a method or set of rules of Functional Size Measurement. It assesses the functionality delivered to its users, based on the user's external view of the functional requirements. It measures the logical view of an application, not the physically implemented view or the internal technical view.

Objectives of FPA:

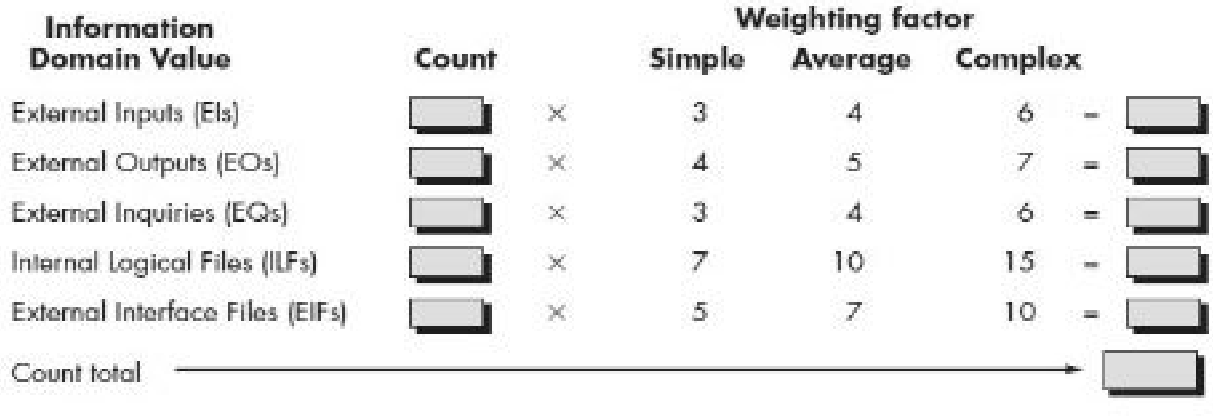
1. The objective of FPA is to measure functionality that the user requests and receives.
2. The objective of FPA is to measure software development and maintenance independently of the technology used for implementation.
3. It should be simple enough to minimize the overhead of the measurement process.
4. It should be a consistent measure among various projects and organizations.

Types of FPA:

1. ***Transactional Functional Type –***
    - **(i) External Input (EI):** EI processes data or control information that comes from outside the application's boundary. The EI is an elementary process.
    - **(ii) External Output (EO):** EO is an elementary process that generates data or control information sent outside the application's boundary.
    - **(iii) External Inquiries (EQ):** EQ is an elementary process made up of an input-output combination that results in data retrieval.
2. ***Data Functional Type –***
    - **(i) Internal Logical File (ILF):** A user identifiable group of logically related data or control information maintained within the boundary of the application.
    - **(ii) External Interface File (EIF):** A group of user recognizable logically related data allusion to the software but maintained within the boundary of another software.

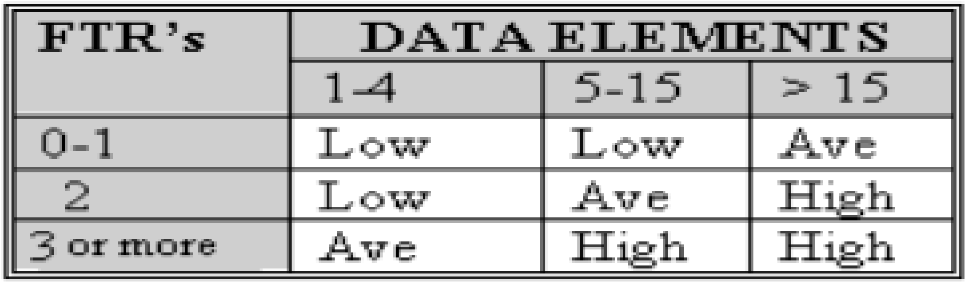
Formula

$$FP = \text{count total} \times [0.65 + 0.01 \times \Sigma (F_i)]$$



Information Domain Value	Count		Weighting factor				
			Simple	Average	Complex		
External Inputs (EIs)	□	×	3	4	6	=	□
External Outputs (EOs)	□	×	4	5	7	=	□
External Inquiries (EQs)	□	×	3	4	6	=	□
Internal Logical Files (ILFs)	□	×	7	10	15	=	□
External Interface Files (EIFs)	□	×	5	7	10	=	□
Count total						→	□

SIZE ESTIMATION FOR THIS PROJECT:



FTR's	DATA ELEMENTS		
	1-4	5-15	> 15
0-1	Low	Low	Ave
2	Low	Ave	High
3 or more	Ave	High	High

EXTERNAL INPUTS:

Total external inputs = 5 + 5 + 4 = 14

low complexity : 5
average complexity : 9

EXTERNAL OUTPUTS :

Total external outputs =1+1+1 = 3

low complexity :1 average complexity:1 high complexity:1

INTERNAL LOGICAL FILES:

Total internal logical files = 1 + 1 = 2

All of these are of low complexity

EXTERNAL INQUIRIES:

Total external inquiries = 2 + 3 = 5

All are average complexity: 5

*UFP = (5*3 + 9*4) + 1*4 + 1*5 + 1*7 + 2*7 + 5*4*

UFP = 101

Considering all the functions have an average influence.

FP = 101(0.65 + 0.01* 15 *3)*
FP = 111.1 = 111

So, FUNCTION POINT COUNT = 111

4.2 EFFORTS

In software development, effort estimation is the process of predicting the most realistic amount of effort (expressed in terms of person-hours or money) required to develop or maintain software based on incomplete, uncertain and noisy input.

No. of screens in the project: 17
No. of reports: 4
No. of 3GL components: 0

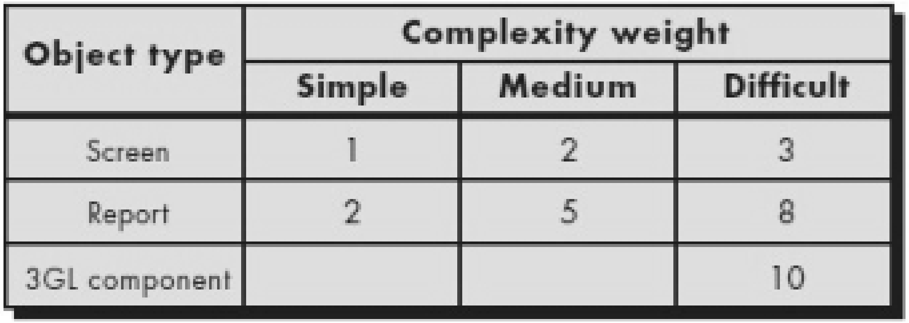
Developer's experience/capability: Very Low

The complexity of screens: 9 simple, 5 medium, 3 difficult

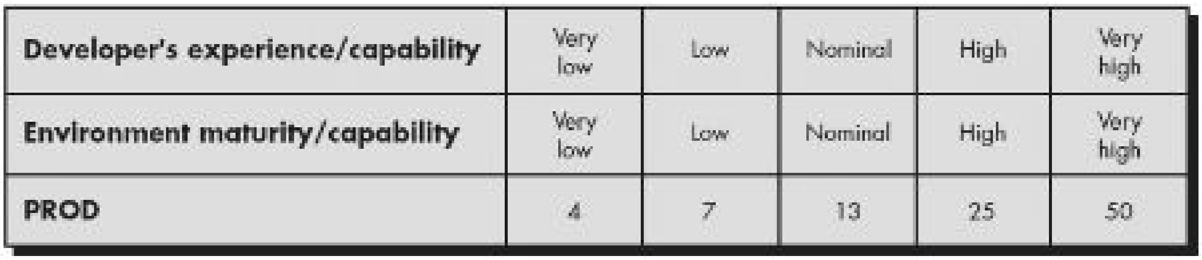
The complexity of reports: 4 simple

% reuse: 55

Taking the values from the given tables:



Object type	Complexity weight		
	Simple	Medium	Difficult
Screen	1	2	3
Report	2	5	8
3GL component			10



Developer's experience/capability	Very low	Low	Nominal	High	Very high
Environment maturity/capability	Very low	Low	Nominal	High	Very high
PROD	4	7	13	25	50

*Object points = [9*1 + 5*2 + 3*3 + 4*2 + 0*10] = 36*

NOP = (Object points)[(100- %reuse)/100] = 36 *0.45 = 16.2*

PROD = 4

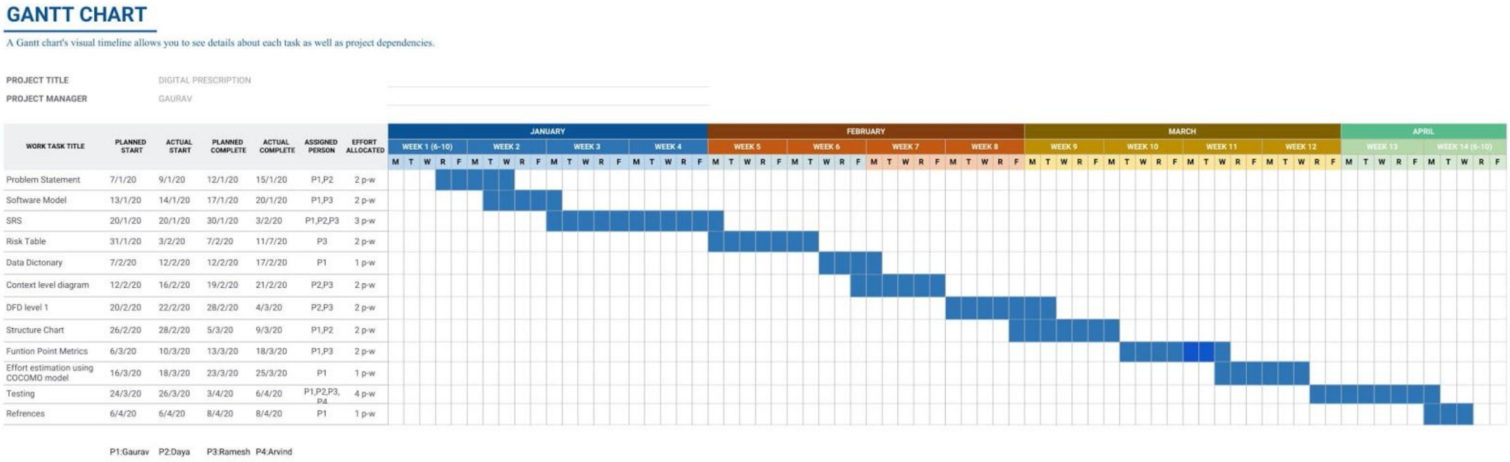
estimated efforts = NOP / PROD = 16.2/4 **= 4.05 person months**

SCHEDULING

Scheduling in project management is the listing of activities, deliverables, and milestones within a project. A schedule also usually includes the planned start and finish date, duration, and resources assigned to each activity. Effective project scheduling is a critical component of successful time management.

Gantt Chart:

Gantt chart is a type of bar chart that is used for illustrating project schedules. Gantt charts can be used in any project that involves effort, resources, milestones, and deliveries.



GANTT CHART

A Gantt chart's visual timeline allows you to see details about each task as well as project dependencies.

PROJECT TITLE DIGITAL PRESCRIPTION

PROJECT MANAGER GAURAV

WORK TASK TITLE	PLANNED START	ACTUAL START	PLANNED COMPLETE	ACTUAL COMPLETE	ASSIGNED PERSON	EFFORT ALLOCATED
Problem Statement	7/1/20	9/1/20	12/1/20	15/1/20	P1,P2	2 p-w
Software Model	13/1/20	14/1/20	17/1/20	20/1/20	P1,P3	2 p-w
SRS	20/1/20	20/1/20	30/1/20	3/2/20	P1,P2,P3	3 p-w
Risk Table	31/1/20	3/2/20	7/2/20	11/7/20	P3	2 p-w
Data Dictonary	7/2/20	12/2/20	12/2/20	17/2/20	P1	1 p-w
Context level diagram	12/2/20	16/2/20	19/2/20	21/2/20	P2,P3	2 p-w
DFD level 1	20/2/20	22/2/20	28/2/20	4/3/20	P2,P3	2 p-w
Structure Chart	26/2/20	28/2/20	5/3/20	9/3/20	P1,P2	2 p-w
Funtion Point Metrics	6/3/20	10/3/20	13/3/20	18/3/20	P1,P3	2 p-w
Effort estimation using COCOMO model	16/3/20	18/3/20	23/3/20	25/3/20	P1	1 p-w
Testing	24/3/20	26/3/20	3/4/20	6/4/20	P1,P2,P3, P4	4 p-w
Refrences	6/4/20	6/4/20	8/4/20	8/4/20	P1	1 p-w

P1:Gaurav P2:Daya P3:Ramesh P4:Arvind

Fig:5.1 Gantt Chart / Work table

RISK MANAGEMENT

A software project can be concerned with a large variety of risks. In order to be adept to systematically identify the significant risks which might affect a software project, it is essential to classify risks into different classes.

There are three main classifications of risks which can affect a software project:

- **Project risks:** Project risks concern different forms of budgetary, schedule, personnel, resource, and customer-related problems.
- **Technical risks:** Technical risks concern potential method, implementation, interfacing, testing, and maintenance issues.
- **Business risks:** This type of risk contains risks of building an excellent product that no one needs, losing budgetary or personnel commitments, etc.

RISK TABLE

1—Negligible
2—Marginal
3—Critical
4--Catastrophic

Risk	Category	Probability	Impact
Inexperienced Employee	ST	30 %	2

Procedural(Team May Not Work as Planned)	PS	20%	2
Delivery-deadline(High Competition)	BU	20 %	3
Client May Refuse To Cooperate (Upper Authority may Change)	BU	30 %	2
The sales team may not be Able to Sell the Product	BU	25 %	2
Funding may be lost	PS	10%	2

Doctors and Patient May Find it Complicated to use	PS	20%	2

DATA FLOW MODELING

Data-flow based modeling, often referred to as the structured analysis technique, uses function-based decomposition while modeling the problem. It focuses on the functions performed in the problem domain and the data consumed and produced by these functions. It is a top-down refinement approach, which was originally called structured analysis and specification, and was proposed for producing the specifications.

A data flow model is a diagrammatic representation of the flow and exchange of information within a system. Data flow models are used to graphically represent the flow of data in an information system by describing the processes involved in transferring data from input to file storage and reports generation. A data flow model may also be known as a data flow diagram (DFD).

7.1 DATA FLOW DIAGRAM(DFD)

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both.

It shows how data enters and leaves the system, what changes the information, and where data is stored.

7.1.1 CONTEXT LEVEL DFD

It is also known as the fundamental system model, or context diagram represents the entire software requirement as a single bubble with input and output data denoted by incoming and outgoing arrows.

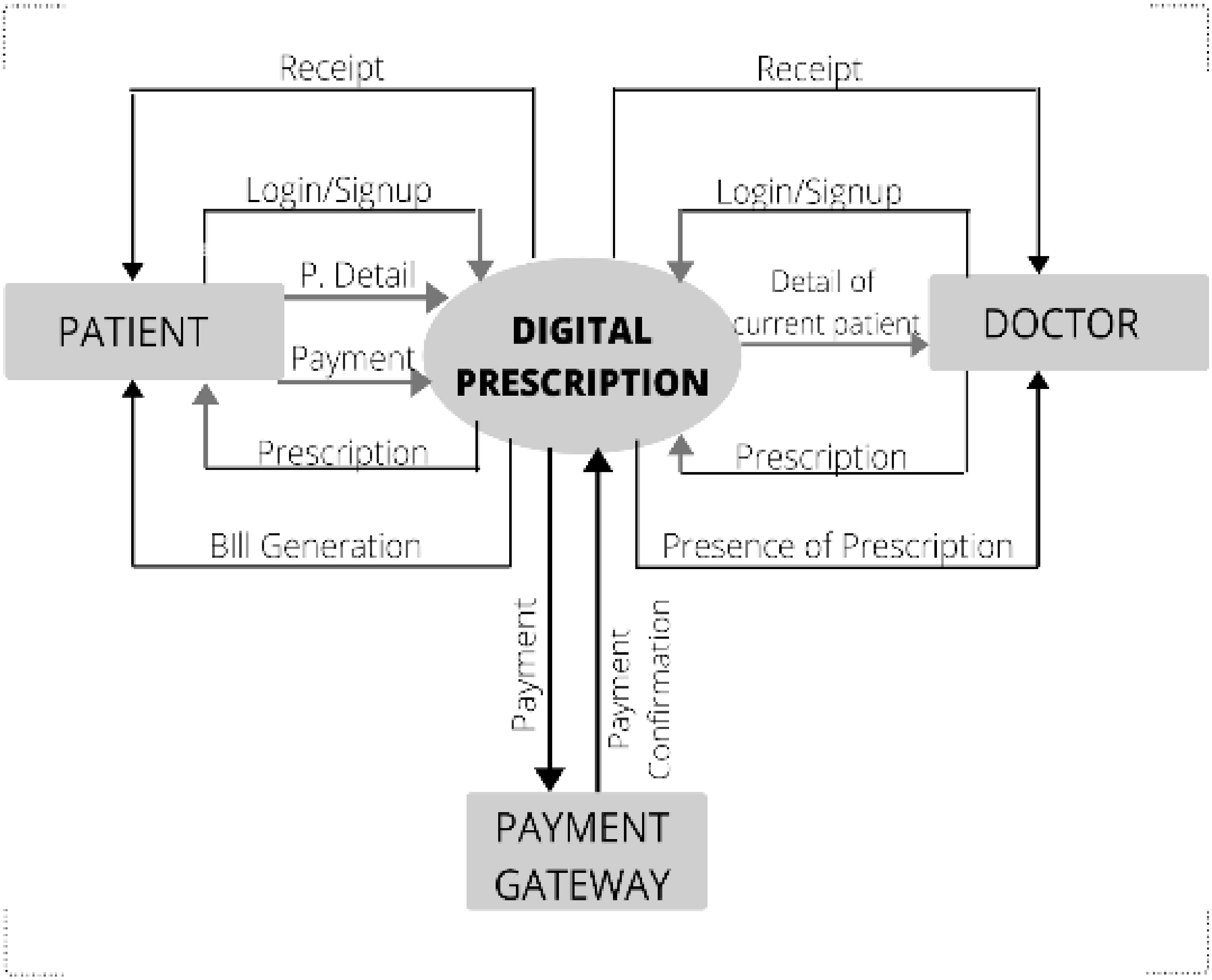


Fig:7.7.1 Context level DFD for Digital Prescription

7.1.2 LEVEL-1 DFD

In 1-level DFD, a context diagram is decomposed into multiple bubbles/processes. In this level, we highlight the main objectives of the system and break down the high-level process of 0-level DFD into subprocesses.

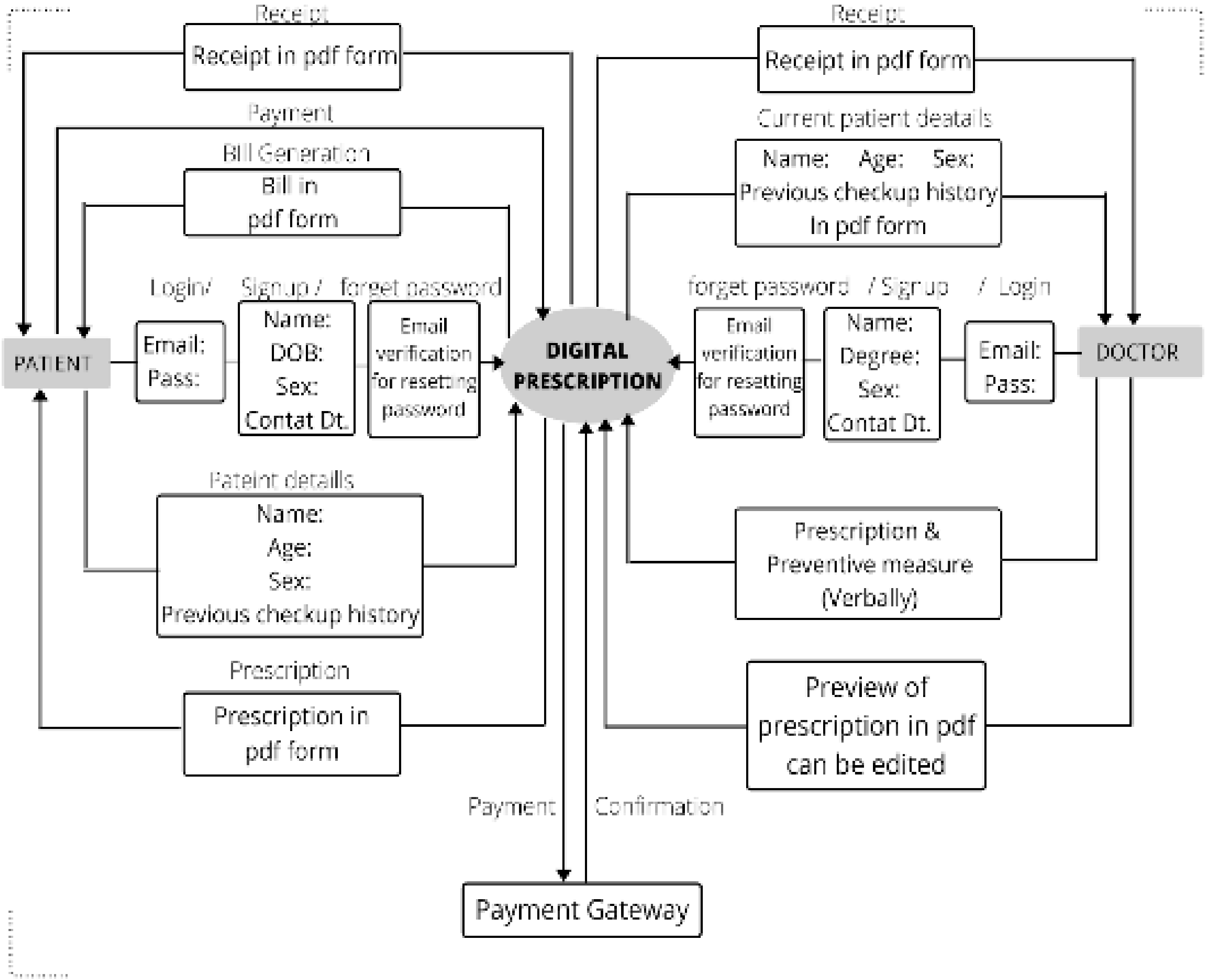


Fig:7.1.2 1- level DFD for Digital Prescription

7.2 DATA DICTIONARY

A data dictionary contains metadata i.e data about the database. The data dictionary is very important as it contains information such as what is in the database, who is allowed to access it, where is the database physically stored, etc.

Patient Signup = Firstname + Lastname + Password + Email + Sex

Doctor Signup = Firstname + Lastname + Password + Email + Speciality

Login = Email-ID + Password

Prescription Format = Patient’s information + voice input + preview

Bill Format = Patient’s information + doctor’s prescribed prescription + amount in Rupees

Receipt Generation = Bill’s Information + patient’s details + Email-ID + payment method

ITEMS	FIELD	DESCRIPTION	REMARKS
First Name	A – Z | a – z |*	User will provide his/her first name	It should contain at least 3 and at most 10 characters.
Last Name	A – Z | a – z |*	User will provide his/her last name.	It should contain at least 3 and at most 10 characters.
Password	[A-Za-z0-9@_]{8,16}$	It will be the password for the user account. It will contain capital and small alphabets, numerals and specified special characters (‘@’,’_’).	It should be at least 8 characters long containing alphabets, digits and special characters (‘@’,’_’).
Email-ID	^([a-zA-Z0-9_\-\.]+)@([a-zA-Z0-9_\-\.]+)\.([a-zA-Z]{2,5})$	users will provide their valid email ID.	the Email-ID must be valid and working.
Sex	A – Z | a – z |*	user will provide their gender i.e, either male, female or other	it shouldn't contain abbreviation words.

Speciality	A – Z | a – z |*	This field will be used by the doctor for describing their speciality in their medical field like surgeon etc.	it shouldn't contain an abbreviation except Dr. .
Amount	[0 - 9]	This will provide the amount of the total fee and prescription of the doctor.	amount should be in INR.

DESIGN

Software design is a process to transform user requirements into some suitable form, which helps the programmer in software coding and implementation.

For assessing user requirements, an SRS (Software Requirement Specification) document is created whereas, for coding and implementation, there is a need for more specific and detailed requirements in software terms. The output of this process can directly be used for implementation in programming languages.

8.1 DESIGN DESCRIPTION

Software design provides a design plan that describes the elements of a system, how they fit, and work together to fulfil the requirement of the system. The objectives of having a design plan are as follows:-

- To negotiate system requirements, and to set expectations with customers, marketing, and management personnel.
- Act as a blueprint during the development process.
- Guide the implementation tasks, including detailed design, coding, integration, and testing.

8.2 ARCHITECTURAL DESIGN

Architectural design can be defined as “the process of defining a collection of hardware and software components and their interfaces to establish the framework for the development of a computer system”. Architectural design serves as a blueprint for a system. It provides an abstraction to manage the system complexity and establish a communication and coordination mechanism among components. The primary goal of the architectural design is to identify requirements that affect the structure of the application.

Further, it involves a set of significant decisions about the organization related to software development and each of these decisions can have a considerable impact on quality, maintainability, performance, and the overall success of the final product. These decisions comprise of −

- Selection of structural elements and their interfaces by which the system is composed.
- Behaviour as specified in collaborations among those elements.
- Composition of these structural and behavioural elements into large subsystems.
- Architectural decisions align with business objectives.
- Architectural styles guide the organization

8.3 STRUCTURAL DESIGN

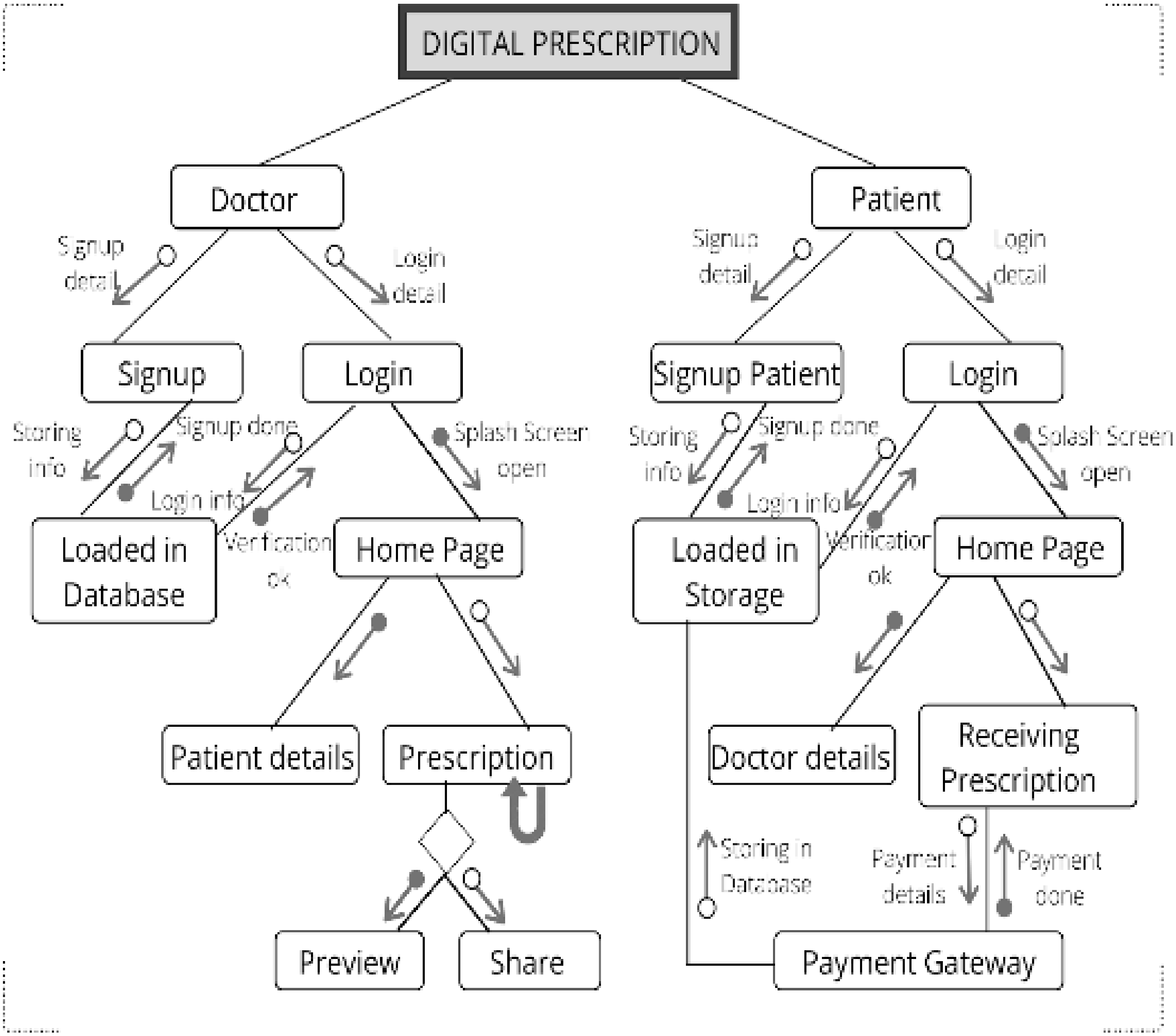
The concept of the structure of a program has at the heart of the structured design method. During design, structured design methodology aims to control and influence the structure of the final program. The aim is to design a system so that programs implementing the design would have a hierarchical structure, with functionally cohesive modules and as few interconnections between modules as possible. The overall strategy is to identify the input and output streams and the primary transformations that have to be performed to produce the output. High-level modules are then created to perform these major activities, which are later refined.

Structural design mainly focuses on following 3 points: System, Process & Technology. There are four major steps in this strategy:

- Restate the problem as a data flow diagram
- Identify the input and output data elements
- First-level factoring
- Factoring of input, output, and transform branches

8.3.1 Structural Chart:

The structure chart is a chart derived from the Data Flow Diagram. It represents the system in more detail than DFD. It breaks down the entire system into the lowest functional modules, describes functions and sub-functions of each module of the system to a greater detail than DFD.



8.4 DATA DESIGN

Database design is the organization of data according to a database model. The designer determines what data must be stored and how the data elements interrelate. With this information, they can begin to fit the data to the database model. Database management system manages the data accordingly. Database design involves classifying data and identifying interrelationships

Data design for both the patient and the doctor.

Data Name	Data Type	Length	Remarks
Password	alphanumeric	min:8 - max:16	not NULL
Email-ID	alphanumeric	var(20)	not NULL , UNIQUE

8.5 PROCEDURAL DESIGN

The procedural design is a software design process that converts and translates structural elements into procedural explanations. Software Procedural Design starts straight after data design and architectural design that uses mainly control commands applied to the predefined data such as:

- **Sequences**: serve to achieve the processing steps in order that is essential in the specification of any algorithm.
- **Conditions:** provide facilities for achieving selected processing according to some logical statement.
- **Repetitions:** serve to achieve looping during the computation process.

These three commands are implemented as ready programming language constructs.

8.5.1 Module: login

```
// Output 'user-id'
Input user_id
// Output 'password'
Input password
if(user clicks on 'login')
    search user id and password in the database
    if(user_id and password match that in the database)
        Redirect to user homepage
    Else    Output 'Invalid user id or password'
    Endif
Endif
```

TESTING

Testing is a process of executing a program with the aim of finding error. To make our software perform well it should be error free.If testing is done successfully it will remove all the errors from the software.

9.1 TESTING APPROACHES

BLACK BOX TESTING -

- In Black-box testing, a tester doesn't have any information about the internal working of the software system.
- Black box testing is a high level of testing that focuses on the behavior of the software. It involves testing from an external or end-user perspective.
- Black box testing can be applied to virtually every level of software testing: unit, integration, system, and acceptance.

WHITE BOX TESTING-

- White-box testing is a testing technique that checks the internal functioning of the system.
- In this method, testing is based on coverage of code statements, branches, paths or conditions.
- White-Box testing is considered as low-level testing. It is also called a glass box, transparent box, clear box or code base testing.
- The white-box Testing method assumes that the path of the logic in a unit or program is known.

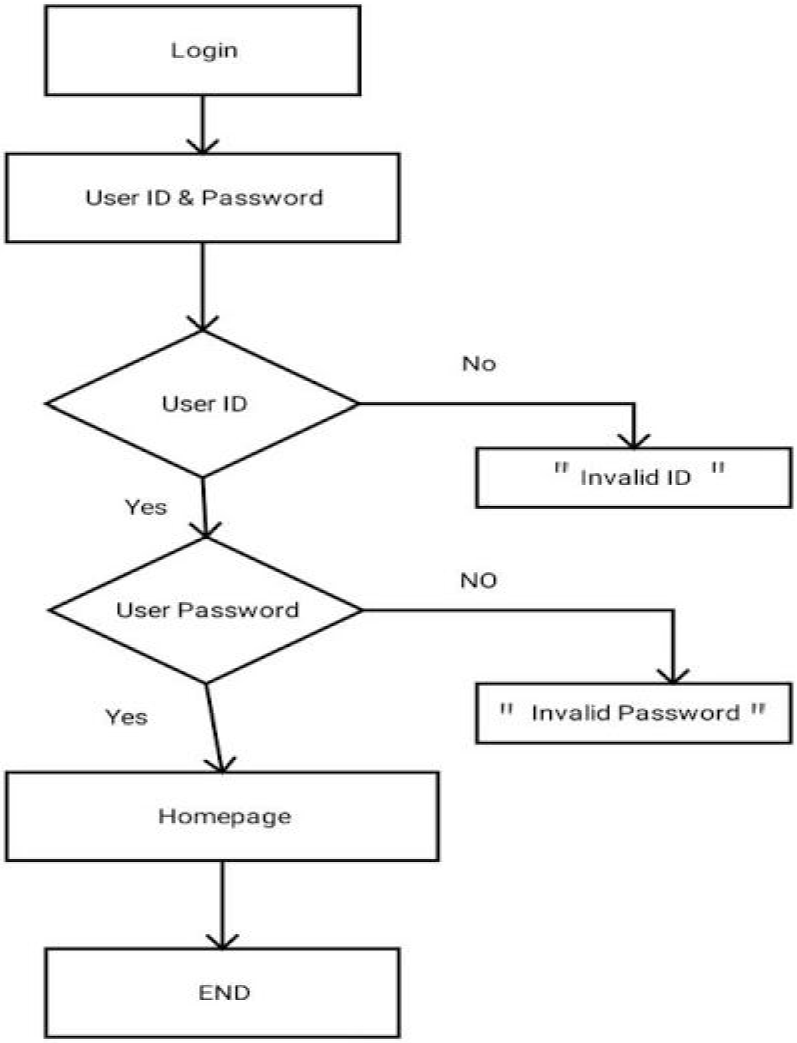
9.2 CYCLOMATIC COMPLEXITY:

Cyclomatic complexity is a source code complexity measurement that is being correlated to a number of coding errors. It is calculated by developing a Control Flow Graph of the code that measures the number of linearly-independent paths through a program module.

Lower the Program's cyclomatic complexity, lower the risk to modify and easier to understand. It can be represented using the below formula:

*Cyclomatic complexity = E - N + 2*P*
where,
E = number of edges in the flow graph.
N = number of nodes in the flow graph.
P = number of nodes that have exit points

MODULE LOGIN



Here;

$E = 7$
$N = 8$
$P = 1$

So,
Cyclomatic complexity = $E - N + 2*P$
= $7 - 8 + 2*1$
= 3

9.3 TEST SUITE

Test suite is a container that has a set of tests which helps testers in executing and reporting the test execution status. It can take any of the three states namely Active, In Progress and completed.

A Test case can be added to multiple test suites and test plans. After creating a test plan, test suites are created which in turn can have any number of tests.

Table for test cases:

Test Scenario Id	Test Scenario Description	Test Case Id Test	Test Case Description	Test Steps	Precondition	Test Data	Post Condition	Expected Result	Actual Result	Status
TS_DP_001	Verify The Functionality	TC_DP_Login_01	Enter A Valid User-	Enter Valid UserName	Valid Url	Username:rpant453@gmail.com	User Should	Successful	Successful	Pass
	of Digital Prescription		Name and Password	Enter Valid Password		Password: Ramesh@123	Able to See The	Login	Login	
	Login Page			Click On Login			Homepage			
TS_DP_002	Verify The Functionality	TC_DP_Login_02	Enter A Valid User-	Enter Valid UserName	Valid Url	Username:rpant453@gmail.com	Error Message	A Popup Box	A Popup Box	Pass
	of Digital Prescription		Name and Invalid	Enter InValid Password		Password:12345678	Invalid	with msg	with msg	
	Login Page		Password	Click On Login			Password	'Invalid	'Invalid	
								Password"	Password"	
TS_DP_003	Verify The Functionality	TC_DP_Login_03	Enter an Invalid User	Enter invalid User	Valid Url	Username:rpant@gmail.com	Error Message	A Popup Box	A Popup Box	Pass
	of Digital Prescription		Name and Valid	name		Password:Ramesh@123	Invalid	with msg	with msg	
	Login Page		Password	Enter Valid User Pass			Username	'Invalid	"Invalid	
				word				Username"	Username"	
				Click on Login						
TS_DP_004	Verify The Functionality	TC_DP_Login_04	Enter an Invalid User	Enter invalid User	Valid Url	Username:rpant@gmail.com	Error Message	A Popup Box	A Popup Box	Pass
	of Digital Prescription		Name and Valid	name		Password:12345678	Invalid User	with msg	with msg	
	Login Page		Password	Enter inValid User Pass			name and	"Invalid User	"Invalid User	
				word			Password	name and	name and	
				Click on Login				Password"	Password"	

REFERENCES

Software Engineering: A Practitioner's Approach

Book by Roger S. Pressman

Software Project Management in Practice

Book by Pankaj Jalote

1. **Tutorialspoint:**

   https://www.tutorialspoint.com/index.htm

2. **GeeksforGeeks:**

   https://www.geeksforgeeks.org/

3. **Wikipedia:**

   https://www.wikipedia.org/

4. **Creately:**

   https://creately.com/