

CHAPTER 1

INTRODUCTION

Jeevini i-NOS is a diabetic patients system app. It is designed and implemented to treat diabetic patients by using modern world technologies. Where it will help to the patients to get easy consultation and treatment from doctor. For this the patient must provide related information that is necessary for doctor. That includes calories, fats, carbohydrates and proteins of the patient in receiving required food consumption. Health parameters such as PPBS, FBS, HBA1C, RBS and other health status are used to collect data from patients. The patient's information about the meal they consumed. Based on the details that are provided by patients, the application will find the diabetic's stage. The report will be generated for the patients. Then doctor will treat the patients based on the report generated.

1.1 Literature review/survey

Existing System:

- **Glucose Buddy**

It is a self-managed smart phone application whose major function is to monitor and control Blood Glucose Self-Monitoring (SMBG). The data gathered by the smart phone app can be visually presented and emailed to a health care professional. Almost all participants found it pretty simple to access, utilize, and enter data. However, the most significant concern was the lack of help or training. You may manually enter your blood glucose readings as well as food consumption notes in the Glucose Buddy app. If your blood glucose meter is compatible, you can use it together with it. You may also input your carbohydrate intake, insulin doses, and physical activity. If you allow it, Apple Health Data might automatically include this. The graph tool allows you to keep track of your blood glucose levels over time. Downloadable and shareable reports are available. You may photograph your meal and then track your blood glucose levels in the meal IQ area. It will provide a score to your meal based on how it affected your blood glucose levels, allowing you to make smarter eating choices. You may share your workout and diet.

- **Healthify Me**

Healthify Me is a smart phone app that combines a personal fitness module, diet, and meal planning into one app. Weight loss diet regimens and personal trainers are available through Healthify Me, a health and fitness app. It keeps you fit on all devices with a simple Google Fit and Samsung Health connection. The app includes a hand washing meter, a sleep tracking system, an exercise tracker, a losing weight tracker, and a calorie counter. Healthify Me's no-equipment home workout videos for men and women offer full-body routines for crunches, fat accumulation, biceps, chest, forearms, shoulder, and quads. Stretching and breathing exercises are part of yoga. When it comes to fitness, it's a one-stop shop with personal trainers who listen to your needs and create a workout plan specifically for you. This diet software also has daily challenges that you may participate in with your friends to enhance your fitness and reduce weight. A supportive training group keeps your spirits up and guarantees that you stick to your diet and exercise routine. It is simple to lose weight. With health data, fitness monitors, and a personalized meal plan, this weight loss training app's calorie counter aids weight reduction and fitness.

- **Fitocracy**

It is the fitness tracking and monitoring software that provides users with a variety of workouts to help them modify their lives. This software helps you in a variety of ways by providing features and services such as fitness evaluation, dietary success, workable workouts, and constant motivation. This app's fitness assessment will guide you through a practical evaluation process so that you can develop a winning strategy. Fitocracy includes training routines and scenarios for everyone, whether you're a beginner or a seasoned athlete.

- **Health Kart**

Health Kart is a reputable online retailer of bodybuilding, sports, and health supplements. In the Health Kart App, we can look for the product you want. With the Health Kart App, we can filter our search by price, gender, and concern, enjoy secure checkout, recommend a friend, access fitness-related blogs and videos, and save money with Health Kart coupons and membership.

- **Evolut:**

Evolut is a fitness app for health, fitness, and wellness. You don't have to go to the gym to become in shape. With us at-home and on-the-go training system, you can get in great condition. We make exercises, yoga, and healthy eating entertaining and engaging with Evolut app, along with customized fitness plans tailored to your lifestyle demands. Whether you're searching for fitness, exercises, yoga, nutrition, a tracker, to lose weight, tone up, develop muscle, improve your flexibility, or increase your endurance, you've come to the right place. With programs ranging from beginner to expert, you can get leaner, fitter, and stronger. All types of workouts are available, including fitness, weight reduction, HIIT, functional training, cardio, strength training, and more for your perfect exercise fitness app, choose or even develop the right regimen.

- **Clinical Communication Applications:**

Voice calling, video calling, text messaging, email messaging, multimedia (text, picture, and video) texting, and conferences through the cellular phone service provider are all possible with smart phones. Patient care applications, like regular communications, are designed to make it easier for medical staff to communicate with one another. This study comprised five papers that discussed clinical communication utilizing cell phones. Two of them explored three smart-phone-based communication apps, and three of them illustrated the effect and improvement in clinical communication utilizing cellphones. In, a system based on mobile applications is proposed (Imteaj & Hossain, 2017) this app solely caters to hospitals and patients. Hence it does not allow immediate access to test findings or a direct interaction mechanism with healthcare professionals.

- **Remente:**

Small daily activities lead to life-changing behaviors. Goal-setting and personal improvement are made easier with our mental health tracker and self-care notebook. Remente serves as your life coach, offering a variety of self-help tools for self-improvement, healthy behaviors, and overall well-being. Begin building healthy habits right now to enhance your self-control and live a healthier lifestyle with more self-love, more self-help, and less stress, anxiety, and sadness. Daily video sessions with life coaches discussing various principles of self-care, and interactive tutorials to help users learn and practice mental health and well-being. Set good life goals to achieve personal growth and

growth. Remente acts as a life coach and provides goal-setting guidance and recommendations to help you develop a sustainable lifestyle that includes narcissism and healthy behavior. You must study self-help and intend to achieve your goals. The daily planner includes a smart and dynamic to-do list that organizes your day according to your life goals, as well as long-term goal activities and self-improvement tasks.

1.2 Challenges

- The developer's task is to examine current systems in order to learn about their weaknesses and to think about how to make his system stand out from the crowd.
- The most challenging thing for developers to figure out how to get own software seen by users in this competitive environment, as well as how to make it more functional with features and how to make it work for society.
- Security issues, which are yet another severe difficulty in mobile app development, are causing alarm among developers.
- Malware should not be present in software. Otherwise, it might cause software or hardware to malfunction, costing additional time and money to fix.

1.3 Objectives

The objectives of the project are as follows:

- The major goal is to treat the diabetics patients according to the stage that they are currently belong. And easily they can get the treatment and reports.
- The patient will approach the agents with some basic information like contact, health conditions. They will receive OTP for the registered mobile number.
- When the agent enters the patient health information in the system. The agent gets to know the diabetics stage and assigns doctor with confirmation of patient.
- Depending on their health condition doctor assist the diet and treatment get starts. And patient should be under care of doctor and food consultation department.

1.4 Problem definition

Everyone's health and manner of life has changed as a result of the epidemic. The suggested solution's goal is to design and build the Jeevini i-NOS Android health application. The aim of Jeevini i-NOS system is to eradicate diabetics through proper diet and routine. It aids to maintain the diabetic patients easily under the control of doctor and food consultation department. By providing proper assistance the diabetics can be reversed. The aim of this project is to design and develop API for “Customer Info” and “Doctor” modules that shall facilitate the operations of the application. These two API facilitates to find stage of diabetics, based on stage doctor can start treatment to the patient.

CHAPTER 2

PROPOSED SYSTEM

2.1 Description of proposed system with simple block diagram

After a preliminary inspection, it is suggested to automate the entire procedure, taking into consideration the flaws in the current system. **“JEEVINI i-NOS”** An android application has been created using Dart language, Flutter as frontend framework and backend in PHP language, Laravel framework.

This software gathers the basic information of the patients from agents based on health characteristics like FBS, HBA1C, PPBS, RBS, gender, and other health conditions. The patient provides information input for each component, such as the diabetes duration, number of tablets, insulin and its amount etc., and food consumed and the quantity consumed. On the given input the algorithm checks for the condition and finds the stage of diabetes. Further it is moved to doctor form, doctor finds the patients records using patients id. Doctor consults the patient. After confirmation of patient the treatment starts with trial subscription. The patients' dietary requirements based on the meal input provided by the patient. If a patient's discrepancy is discovered, the patient is given an increase-recommendation and a reduction-recommendation to increase or reduce a specific food item in order to correct the discrepancy. And if exercise is required that also has been provided with suitable video.

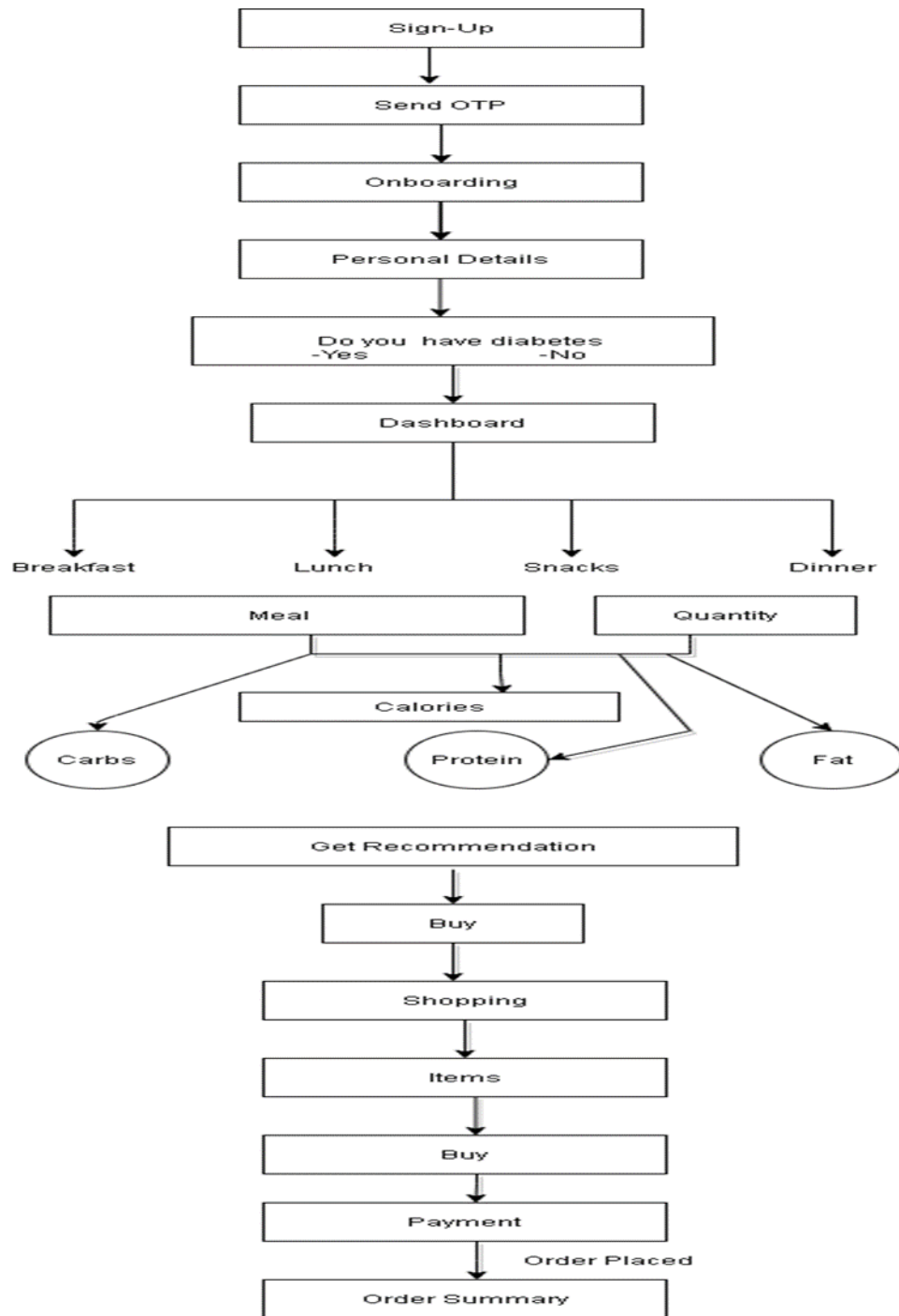


Fig 2.1: Block Diagram of Jeevini i-Nos

2.2 Description of Target users

- **Actors:** Patient
- **Registration:** The user enters his or her contact information and a two-factor authentication code is sent to the specified mobile number to validate the user.
- **Login:** After registering, the user can access the application.
User gives health metrics such as height, weight, age, and health status information.
- **Dashboard:** The user enters the meal type and quantity consumed, and the system calculates the calories consumed depending on the user's health characteristics and the meal type. Calories, carbohydrates, lipids, and protein are all calculated. And suggestions are made, such as what foods should be increased or decreased.
- **Shops:** The shops module allows users to buy any Jeevini products.

2.3 Advantages of Proposed System

Advantages of proposed system are as follows:

- Easily the patient can get report, he can find his improvement.
- Easy to find diabetic's stage.
- Doctor can easily monitor the patients.
- From food consultation department patient diet can be maintained easily.

If necessary, Jeevini product is suggested based on the recommendation.

2.4 Scope

The purpose of this system is to give customers recommendations based on their meal type, such as whether they are eating properly or if they need to modify their food consumption. This allows patients to consume the balanced meals in adequate quantities to maintain their health. It also displays a graph for each patient that depicts the required and ingested nutrients, which help the patient, recognize their current health situation.

CHAPTER 3

SOFTWARE REQUIREMENT SPECIFICATION

A product necessity particularly Software Requirement Specification (SRS) is a model that frames what the product will perform and how it will work. It likewise determines the highlights that the item should have to fulfill the necessities of all partners (business and clients).

3.1 Overview of SRS

A software requirement specification (SRS) is a detailed description of a software system that must be developed, which includes both functional and non-functional requirements. Use cases about how the user will interact with the software program could be included. All necessary requirements for project development are included in the software requirement specification document. We must have a thorough understanding of software systems in order to design them. To accomplish this, we must maintain constant contact with patients collect all requirement.

A proper SRS specifies how the Software Program will connect with all internal operations, modules, technology, as well as inter-program communication and the human user's interactions with a number of alternative scenarios. In order to minimize errors in system testing and intended outcomes, testers must comprehend every piece stated in this document. Reviewing or testing Software Requirement is highly suggested. Before you begin developing test cases or making any plans, you should review the specifications papers for testing purposes.

3.2 Requirement Specifications

The Requirement Specification is a process of examining, analyzing and generating the functional requirements and non-functional requirements. These two requirement specifications are used to build, develop, analyze and process the system. Requirement Specification is one of the important aspects of the system designing part. It helps in understanding the recorded and provided requirements very easily.

3.2.1 Functional requirements

The functional requirements of the project are used to create a working prototype: A one-time password (OTP) is necessary to verify a person's contact information.

To generate the recommendation report, the meal item and quantity must be entered into the system according to the meal type.

Login:

User enters Mobile number and Name for registration.

Input: Valid Mobile number.

Error handling: If valid mobile number is not entered then user is not registered.

Output: User is successfully registered.

Dashboard:

Once the user logs in dashboard is displayed where user can choose meal item and quantity

Input: User can choose consumed meal item and quantity from dropdown.

Error handling: User has to select food item otherwise report is not generated.

Output: Calorie, proteins and fats and is calculated once the user chooses valid meal item.

On boarding:

Patient enters the other personal details to get recommendation report.

Input: Enter details such as height, age, health status, gender, and weight.

Error handling: If the space is left empty error is prompted.

Output: Based on the details entered by user, recommendation report is generated.

Graphs:

Health of users is represented in graphical format.

Input: Food items and quantity consumed.

Output: Pictorial representation of minimum required nutrition and consumed nutrition in weekly and monthly basis.

Shops:

User purchase Jeevini product from store.

Input: User can choose item from store and add to cart.

Error handling: If a user clicks on specific item, information about the item should be displayed and user must be able to add the item in cart.

Output: Selected item must be displayed in cart.

3.2.2 Use case diagrams

The Jeevini i-NOS health application graphically depicted in this use case diagram of interactions between the system and the module in a proper manner. It is the process used in system analysis to discover, clarify, and organize Jeevini i-NOS system requirements.

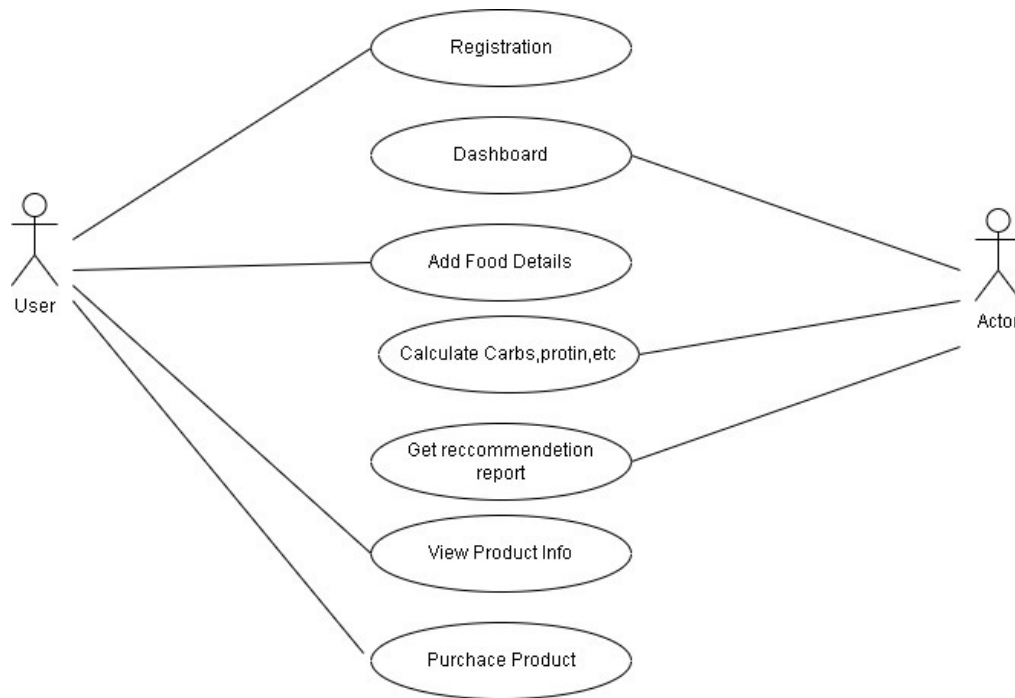


Fig 3.1: Use Case Diagram of user

3.2.3 Use case descriptions using scenarios

- User will register by entering Name and Password.
- System will display dashboard page to feed consumed items and quantity.
- User will enter other details such as age, gender, health status, height, and weight.
- Based on food consumed and health parameters system will generate recommendation report.
- User can purchase Jeevini products from store module considering their report.
- Cash on delivery and Online payment option is given to user to purchase the product.

3.2.4 Non-functional requirements

The system's non-functional requirements are a collection of the user's most pressing desires. In terms of the user interface, it is hoped that the current system would be improved. Security should be simple. The system must also be straight forward to operate from a technological aspect, based on the user credentials provided when the user first logs into system. Though these ideas will not be immediately discussed in terms of functional requirements, the design must be designed around them because they are just as important as functional requirements.

3.3.4.1 Performance requirements

- The loading time for user interface screens should not be more than three seconds.
- The log-in information must be authenticated within three seconds.
- Queries must return results in three seconds or less.
- Available 24*7.

3.3.4.2 Safety requirements

- Patient information is never shared with third parties.
- Personal information such as height, weight, gender, and health status are entered by the patient. This information is used purely to make the best possible advice.

3.3.4.3 Security requirements

- For the user's phone number, two-step authentication is used.
- An OTP is issued to the specified phone number.

3.3.4.4 Usability

The user can review the report at any moment and track his or her health using a pictorial representation.

3.3 Hardware and Software requirements

Hardware:

- **Operating System** : Windows 10 Home
- **Processor** : Core i5
- **Hard Disk** : Minimum 512 GB
- **RAM** : Minimum 8 GB

Software:

- **Front End** : Flutter
- **Backhand** : Laravel (PHP)
- **Server** : AWS Server

3.4 GUI of proposed system

The screenshot shows the 'Customer Form' interface. At the top, there are four circular icons with labels: 'call', 'offer rejected', 'call later', and 'call not received'. Below these is a date field set to '13-01-2023'. A 'Language' dropdown menu is set to 'patient's language', with an 'update language' button below it. The form includes input fields for 'Name', 'mobile number' (with a 10/10 character count), 'age' (with a 0/3 character count), 'Profession', and 'Address'. At the bottom, there is a 'Gender' section with radio buttons for 'Male' and 'Female', and a partially visible 'Height' field.

The screenshot shows the 'Customer Form' interface with medical details. It includes 'Height' fields for 'Feet' and 'Inches'. A 'Weight' field is followed by a '0/3' character count. The 'Diabetes duration' section has 'Years' and 'Months' fields, each with a '0/2' character count. A 'Recent blood sugar test report' section contains four input fields: 'FBS', 'PPBS', 'Hba1c', and 'RBS'. Below this are three questions with radio button answers: 'Is patient suffering from Thyroid problem?' (Yes/No), 'Is patient having burning foot sensation?' (Yes/No), and 'Is patient suffering from low or high blood pressure(BP)?' (Yes/No).

11:10 AM | 26.4KB/s

Customer Form

Hba1c

RBS

Is patient suffering from Thyroid problem?

☐ Yes ☐ No

Is patient having burning foot sensation?

☐ Yes ☐ No

Is patient suffering from low or high blood pressure(BP)?

☐ Yes ☐ No

Is patient taking any diabetes tablets?

☐ Yes ☐ No

Is patient taking insulin?

☐ Yes ☐ No

Remarks

Submit

11:10 AM | 31.6KB/s

Patients List

2371	CallFrom : 0945333584 dateTime : 2023-07-13 10:01:25	08:00:00	08:00:00
2370	CallFrom : 0781000000 dateTime : 2023-07-13 10:00:28	08:00:00	08:00:00
2369	CallFrom : 0945333773 dateTime : 2023-07-13 09:27:21	08:00:00	08:00:00
2368	CallFrom : 0945333584 dateTime : 2023-07-13 08:52:27	08:00:00	08:00:00
2367	CallFrom : 0945333584 dateTime : 2023-07-13 08:25:17	08:00:00	08:00:00
2366	CallFrom : 0945333584 dateTime : 2023-07-13 08:20:51	08:00:00	08:00:00
2365	CallFrom : 0945333584 dateTime : 2023-07-13 08:13:59	08:00:00	08:00:00
2364	CallFrom : 0945333584 dateTime : 2023-07-13 08:11:27	08:00:00	08:00:00
2363	CallFrom : 0945333584 dateTime : 2023-07-13 08:03:57	08:00:00	08:00:00
2362	CallFrom : 0945333584 dateTime : 2023-07-13 08:00:00	08:00:00	08:00:00

↑

CHAPTER 4

SYSTEM DESIGN

System design is a frameworks configuration method involved with characterizing a framework's system, item development, parts, connection points, and information to meet specific rules. Design is the method of applying systems theory to the creation of the product.

4.1 Architecture of the system

An architecture diagram is a visual representation of how software components are practically represented. It displays the software system's overall structure, and the relationships, constraints, and limits that exist between its many modules. In this diagram there are three modules: Login module, Management module, Dashboard Module

Login Module: It begins with sign-up, then retrieves the user's mobile number and performs two-step verification. Once the number has been verified as valid, it is entered into the patient database.

Management Module: Patient details such as age, gender, health status, height, and weight are extracted here, and if there is any health condition such as diabetes or obesity, the patient can also provide such information. These details are saved in a database.

Dashboard Module: The patient enters the consumed meal and quantity, which is kept in the meal item database; analysis is performed based on the meal item and health information; and finally, a recommendation is provided based on the diet plan database.

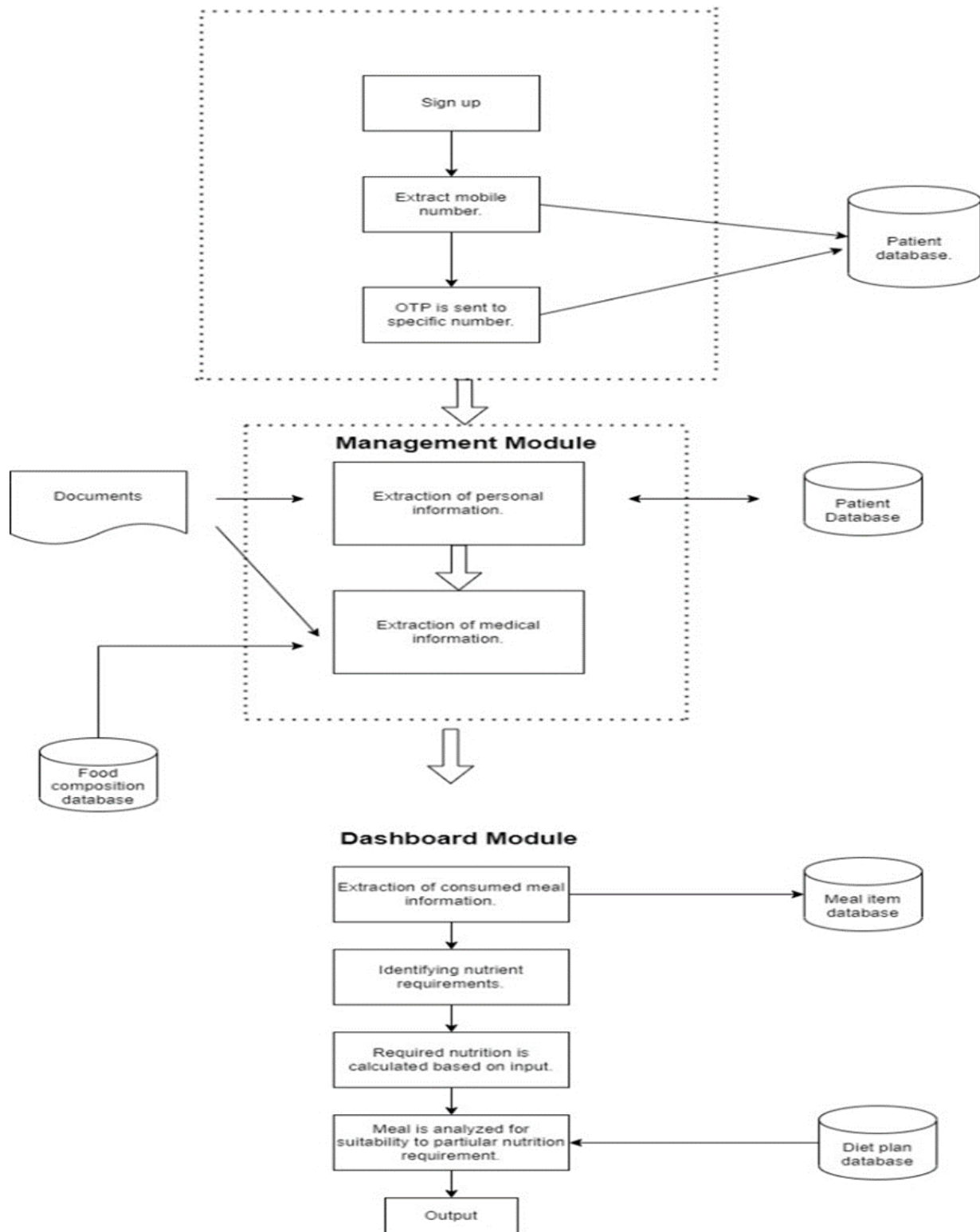


Fig 4.1: Architecture Diagram of Jeevini i-NOS

4.1 Dataflow Diagram

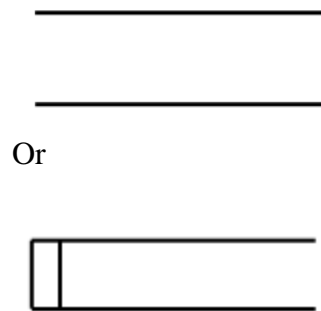
A data flow diagram (DFD) is a significant modeling technique for analyzing and constructing information processes. Data-flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. DFDs can also be used for the visualization of data processing (structured design). On a DFD, data items flow from an external data source or an internal data store to an Internal data store or an external data sink, via an internal process. A DFD provides no information about the timing or ordering of processes, or about whether processes will operate in sequence or in parallel. It is therefore quite different from a flowchart.

This shows the flow of control through an algorithm, allowing a reader to determine what operations will be performed, in what order, and under what circumstances, but not what kinds of data will be input to and output from the system, nor where the data will come from and go to, nor where the data will be stored (all of which are shown on a DFD). Data-flow diagrams provide the end user with a physical idea of where the data they input ultimately has an effect upon the structure of the whole system from order to dispatch to report. How any system is developed can be determined through a data-flow diagram.

With a data-flow diagram, users are able to visualize how the system will operate, what the system will accomplish, and how the system will be implemented. A designer usually draws a context-level DFD showing the relationship between the entities inside and outside of a system as one single step. This basic DFD can be then disintegrated to a lower-level diagram demonstrating smaller steps exhibiting details of the system that is being modeled. Numerous levels may be required to explain a complicated system. The different versions are Context Diagrams (Level 0), Partitioned Diagrams (single process only -- one level), functionally decomposed, leveled sets of Data Flow Diagrams.

Data Flow Diagrams Symbols

A DFD usually comprises of four components. These four components can be represented by four simple symbols. These symbols can be explained in detail as follows: External entities (source/destination of data) are represented by squares; Processes (input-processing-output) are represented by rectangles with rounded corners; Data Flows (Physical or electronic data) are represented by arrows; and finally, Data Stores (physical or electronic like XML files) are represented by open-ended rectangles.

Data store

A data store stores data passively for later access. A data store responds to requests to store and access data. It does not generate any operations. A data store allows values to be accessed in an order different from the order in which they were generated. Input flows indicate information or operations that modify the stored data such as adding or deleting elements or changing values. Output flows indicate information retrieved from the store; this information can be an entire value or a component of a value.

Data flow

A data flow moves data between processes or between processes and data stores. As such, it represents a data value at some point within a computation and an intermediate value within a computation if the flow is internal to the diagram. This value is not changed. The names of input and output flows can indicate their roles in the computation or the type of the value they move. Data names are preferably nouns. The name of a typical piece of data, the data aspect, is written alongside the arrow.

Rules for Creating DFD

- Processes should be named and numbered for easy reference. Each name should be representative of the process.
- The direction of flow is from top to bottom and from left to right. Data traditionally flow from the source (upper left corner) to the destination although they may flow back to a

source. One way to indicate this is to repeat the source symbol as a destination. Since it is used more than once in the DFD, it is marked with a short diagonal in the lower right corner.

- When a process is exploded into the lower-level details, they are numbered.
- The names of data stores, sources and destinations are written in capital letters. Process and data flow names have the first letter of each word capitalized.

Zero Level DFD

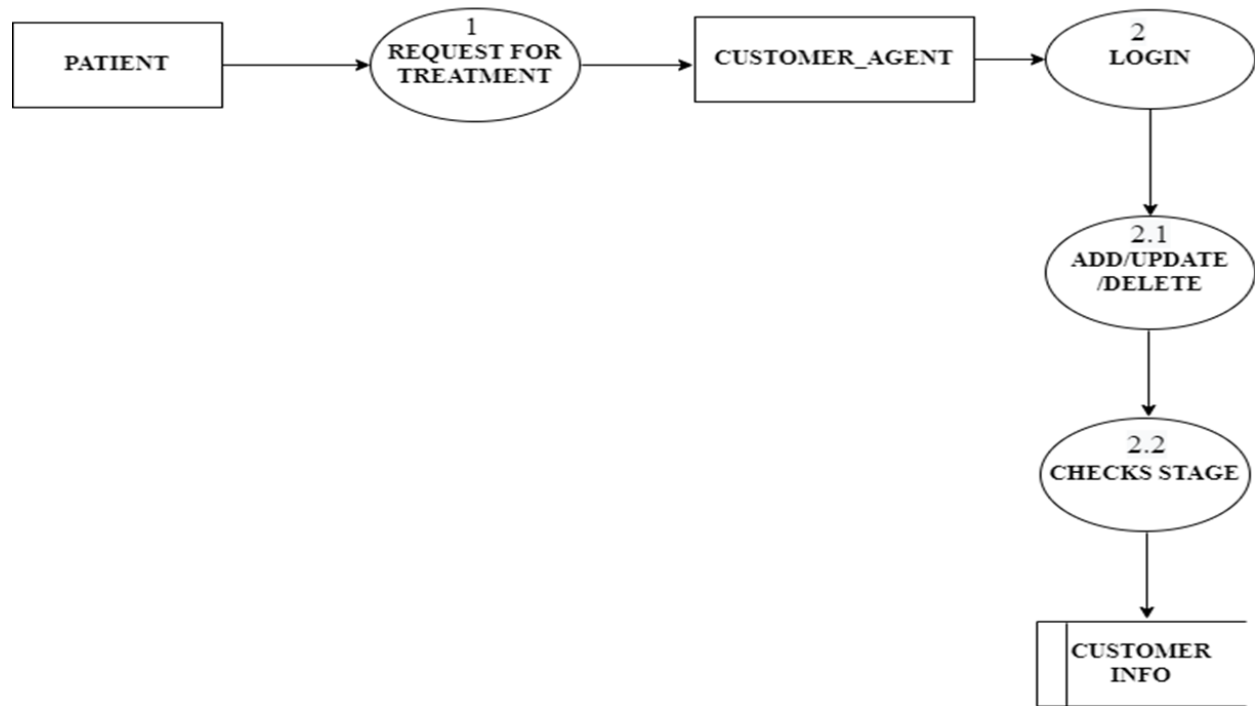


4.1 Zero level DFD for customer information

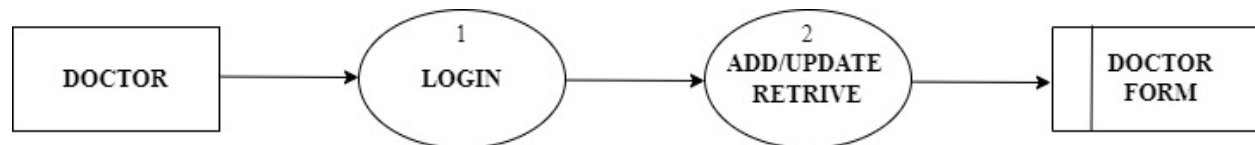
4.2 Level 1 DFD

The context diagram is split into numerous processes in 1-level DFD. At this level, we highlight the system's core functions and split down the high-level, 0-level DFD process into sub processes.

Level 1 DFD is displayed in Figure 4.2. The user module's work flow is described here. The Jeevini health application allows users to log in using their phone number. Once the contact number is entered, an OTP will be sent to the registered contact number. The user then enters the amount of food consumed. It estimates calories, carbohydrates, and proteins based on the amount of food consumed. A report of recommendations will be generated. It will make recommendations for the product based on the report.



4.2.1 level 1 DFD for customer information



4.2.2 level 1 DFD for doctor form

CHAPTER 5

IMPLEMENTATION

The project plan is to perform the actions during this phase that is implementation phase. The information of patient entered in dashboard page will be collected and sent to database to give the treatment and diet. Based on the recommendation report the Jeevini i-NOS product will be suggested.

5.1 Proposed Methodology

Nutrition apps for mobile devices (such as PDAs and smart phones) are becoming more widely available, making it easier for consumers to track their food intake for evaluation or review. Practice self-monitoring this review compiles and discusses studies on these technologies for tracking food intake in healthy people and individuals who are trying to lose weight. The purpose was to see how this technology compared to more traditional ways (eg, 24-hour dietary recall interviews and paper food diaries). Studies with healthy or weight-loss populations, usage of a nutrition app on a mobile device, and inclusion of at least one of three metrics, specifically the ability to record food intake compared to traditional techniques, were found in search databases between January 2000 and April 2011. Dietetics can benefit greatly from the usage of nutrition apps on mobile devices. Compared to previous approaches (typically a paper diary), self-monitoring and changes in food consumption and/or anthropometric measurements were frequently seen. Dietetics can benefit greatly from the usage of nutrition apps on mobile devices. Compared to previous approaches (typically a paper diary), self-monitoring and changes in food consumption and/or anthropometric measurements were frequently seen. Dietetics can benefit greatly from the usage of nutrition apps on mobile devices.

5.2 Modules

Patient:

The patient is the actor of the product. He provides information like diabetics duration, age, FBS, PPBS, HBA1C, RBS, number of tablets, insulin etc., which is necessary for finding condition of diabetics of patient.

Agent:

He is the person to communicate with patients. If any help or problem occurs, he is the point of contact in operating software. He also responsible for point of contact at the subscription time.

Source code of API's

Information from customers:

```
public function info_from_customers(Request $request)
{
    $CallSid=$request->input('CallSid');
    $PatientName=$request->input('PatientName');
    $mobile_no=$request->input('mobile_no');
    $tablets=$request->input('tablets');
    $isTablets=$request->input('isTablets');
    $noOfTabs=$request->input('noOfTabs');
    $isInsulin=$request->input('isInsulin');
    $Mg=$request->input('Mg');
    $times=$request->input('times');
    $age=$request->input('age');
    $gender=$request->input('gender');
    $city=$request->input('city');
    $Diabeties_duration=$request->input('Diabeties_duration');
    $FBS=$request->input('FBS');//A fasting blood sugar level
    $PPBS=$request->input('PPBS');//Blood Sugar Post Prandial (BSPP) test
    $HBA1C=$request->input('HBA1C');//A hemoglobin A1C (HbA1C) test is a
    blood test that shows what your average blood sugar level
    $RBS=$request->input('RBS');//Random blood sugar (RBS) measures blood
    glucose regardless of when you last ate.
```

```
$date=$request->input('date');
$stage=$request->null;
$BT1=false;
$BT2=false;
$BT3=false;
$BT4=false;
$BT5=false;
if(($FBS<=100)&&($HBA1C<=6.5)&&($PPBS<=200)||($RBS<=200))
{
    $BT1=true;
}
else if(($FBS<=150)&&($HBA1C<=7.5)&&($PPBS<=300)||($RBS<=300))
{
    $BT2=true;
}
else if(($FBS<=200)&&($HBA1C<=8.5)&&($PPBS<=400)||($RBS<=400))
{
    $BT3=true;
}
else if(($FBS<=300)&&($HBA1C<=9)&&($PPBS<=500)||($RBS<=500))
{
    $BT4=true;
}
else if(($FBS>300)&&($HBA1C>9)&&($PPBS>500)||($RBS>500))
{
    $BT5=true;
}
else{
    if($PPBS!='empty')
    {
        if($PPBS<=200)
        {
            $BT1=true;
        }
        else if($PPBS<=300)
```

```
{
    $BT2=true;
}
else if($PPBS<=400)
{
    $BT3=true;
}
else if($PPBS<=500)
{
    $BT4=true;
}
else
{
    $BT5=true;
}
}
if($PPBS=='empty' && $RBS!='empty')
{
    if($RBS<=200)
    {
        $BT1=true;
    }
    else if($RBS<=300)
    {
        $BT2=true;
    }
    else if($RBS<=400)
    {
        $BT4=true;
    }
    else if($RBS<=500)
    {
        $BT5=true;
    }
    else
```



```
{
    $BT5=true;
}
}
else if($PPBS=='empty' && $RBS=='empty' && $HBA1C!='empty')
{
    if($HBA1C<=6.5)
    {
        $BT1=true;
    }
    else if($HBA1C<=7.5)
    {
        $BT2=true;
    }
    else if($HBA1C<=8.5)
    {
        $BT3=true;
    }
    else if($HBA1C<=9)
    {
        $BT4=true;
    }
    else
    {
        $BT5=true;
    }
}
else if(($PPBS=='empty') && ($RBS=='empty') && ($FBS!='empty'))
{
    if($FBS<100)
    {
        $BT1=true;
    }
    else if($FBS<=150)
    {
```

```
        $BT2=true;
    }
    else if($FBS<=200)
    {
        $BT3=true;
    }
    else if($FBS<=300)
    {
        $BT4=true;
    }
    else{
        $BT5=true;
    }
}
}
if($isInsulin=='true')
{
    $stage=4;
}
else
{
    if($isTablets=='true')
    {
        if($noOfTabs>=3)
        {
            $stage=4;
        }
        else if($noOfTabs==2)
        {
            if($BT4==true||$BT5==true)
            {
                $stage=4;
            }
            else{
                $stage=3;
            }
        }
    }
}
```

```
    }
  }
  else if($noOfTabs==1)
  {
    foreach($times as $times)
    {
      $time=$times;
      if($time==1)
      {
        if($BT1=='true' || $BT2=='true')
        {
          $stage=2;
        }
        else if($BT3=='true')
        {
          $stage=3;
        }
        else if($BT4=='true' || $BT5=='true')
        {
          $stage=4;
        }
      }
    }
  }
  else
  {
    if($BT1=='true' || $BT2=='true')
    {
      $stage=3;
    }
    else if($BT4=='true' || $BT5=='true' || $BT3=='true')
    {
      $stage=4;
    }
  }
}
```

```
}
else
{
    if($BT1=='true')
    {
        $stage=0;
    }
    else if($BT2=='true')
    {
        $stage=1;
    }
    else if($BT3=='true')
    {
        $stage=2;
    }
    else if($BT4=='true')
    {
        $stage=3;
    }
    else if($BT5=='true')
    {
        $stage=4;
    }
    else {
        $stage=null;
    }
}

}

$status=\DB::table('info_from_customers')->insert(
    array(
        'CallSid'=>$CallSid,
        'PatientName'=>$PatientName,
        'mobile_no'=>"91".$mobile_no,
        'tablets'=>json_encode($tablets),
```

```
'Mg'=>json_encode($Mg),
'times'=>json_encode($times),
'age'=>$age,
'gender'=>$gender,
'city'=>$city,
'Diabeties_duration'=>$Diabeties_duration,
'FBS'=>$FBS,
'PPBS'=>$PPBS,
'HBA1C'=>$HBA1C,
'RBS'=>$RBS,
'date'=>$date,
'isTablets'=>$isTablets,
'noOfTabs'=>$noOfTabs,
'isInsulin'=>$isInsulin,
'stage'=>$stage
)
);

if($status)
{
    return[
        "status"=>"success",
    ];
}
else
{
    return[
        "status"=>"failed to insert",
    ];
}
}
```

Doctor form:

```
public function doctor_form(Request $request)
{

    $raw_data = $request->json()->all();
    $CallSid=$raw_data['CallSid'];

    $mobile_no = $raw_data['mobile_no'];

    $name = $raw_data['name'];
    $diabetes_stage = $raw_data['diabetes_stage'];
    if($diabetes_stage==1)
    {
        $new_med='Morning Feed,Before Food,Voglimore:1';
    }
    else if($diabetes_stage==2)
    {
        $new_med='Morning Feed,Before Food,Voglimore:2';
    }
    else if($diabetes_stage==3)
    {
        $new_med='Morning Feed,Before Food,Voglimore:3';
    }
    else
    {
        $new_med='Morning Feed,Before Food,Dapacan(10),Voglimore:4';
    }
    $stop_med = $raw_data['stop_med'];
    $continue_med=$raw_data['continue_med'];

    $reccBreakfast=$raw_data['reccBreakfast'];
    $reccLunch=$raw_data['reccLunch'];
    $reccSnacks=$raw_data['reccSnacks'];
    $reccDinner=$raw_data['reccDinner'];
    $insert_doctor_form = array
    (
        "CallSid"=>$CallSid,
        "Diabeties_duration"=>$raw_data["Diabeties_duration"],

        "mobile_no" => $raw_data['mobile_no'],
        "PatientName" => $raw_data['name'],
        "diabetes_stage" => $raw_data['diabetes_stage'],
        "stop_med" => $raw_data['stop_med'],
        "continue_med"=>$raw_data['continue_med'],
        "new_med" =>$new_med,
```

```
"reccBreakfast"=>$raw_data['reccBreakfast'],
"reccLunch"=>$raw_data['reccLunch'],
"reccSnacks"=>$raw_data['reccSnacks'],
"reccDinner"=>$raw_data['reccDinner'],
"remarks"=>'empty'
);
$submitted = array(
    "CallSid"=>$CallSid,
    "CallFrom" => $raw_data['mobile_no'],
);
$ID1 = \DB::table('doctor_form')->insert($insert_doctor_form);
return array('msg'=>"inserted successfully");
}
```

To get Patient Details:

```
public function patientDetails(Request $request)
{
    $id=$request->input('id');
    $getpatient=\DB::table('info_from_customers')->where('id',$id)->first();
    return $getpatient;
}
```

CHAPTER 6

TESTING

6.1 Details of Testing Methodologies

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation. Testing is done to see if the entire feature provided by the module are performing satisfactorily and to ensure that the process of testing is as realistic as possible. Testing part is done to major phases via unit level and module level testing. The idea of testing the software in a phased manner is to identify and isolate the bugs for any easy correction. Both phases are over and results meet with the user needs.

Test Approaches:

Black box Testing:

Black box testing is done to find:

- Incorrect or missing functions.
- Interface Errors.
- Errors in external database access.
- Performance error.
- Initialization and termination error.

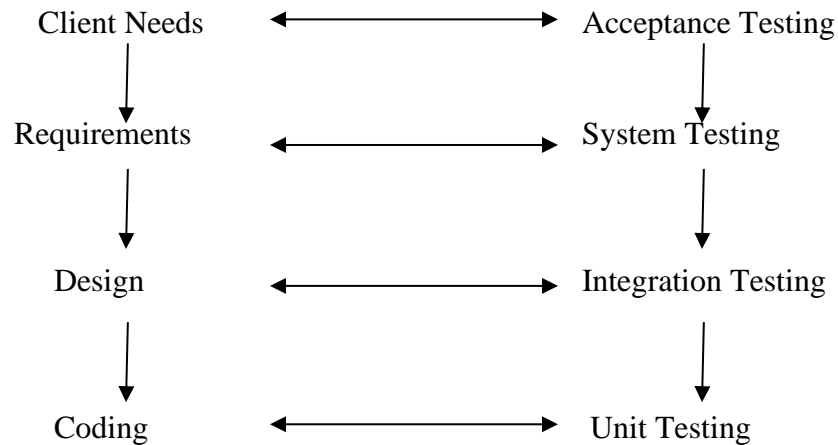
White box Testing:

White box testing is done to find out:

- Check whether all independent paths within a module have been exercised at least once.
- Exercise all logical decision on their true and false sides.
- Execute all loops at their boundaries and within their bounds.
- Exercise the internal data structure to ensure their validity.
- Ensure whether all the possible validity checks and validity lookups have been provided to validate data entry.

Testing Strategies:

Levels of testing:



The different testing is carried out on the reflects the effectiveness and efficiency of different phases of software development these test help to uncover the error in the corresponding phase.

- **Unit testing** is carried out to check the coding errors and program logic.
- **Integration testing** which also known as module testing uncovers the system design errors.
- **Performance testing** is performed to determine how fast some aspect of a system performs under a particular workload.
- **User Acceptance testing** is done to test whether the product developed needs the client needs and acceptable to him.

Unit Testing:

Unit testing involves, checking all the modules in the system individually against the specification produced during the design of the module and for their performance. Unit testing also involves code produced in the coding phase and hence the internal logic of the program. Each module is tested for different test cases design to check each specific combination of conditions handled by the program. Error handlers are included in each module for each event trap and handled the errors.

Unit testing is done by inputting proper input s in each page and checking whether the data is in correct format as used to backend. Each servlet is also checked whether each method is working properly to fulfill the requirement.

Integration testing:

Integration testing takes as its input modules that have been checked out by unit testing, groups them in larger aggregates, applies tests defined in an Integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing. The purpose of integration testing is to verify functional, performance and reliability requirements placed on major design items.

In our project we integrate all the .ASPX pages and run complete module and check for all conditions whether it's working properly.

System Testing:

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic. As a rule, system testing takes, as its input, all of the "integrated" software components that have successfully passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called *assemblages*) or between any of the *assemblages* and the hardware. System testing is a more limited type of testing. It seeks to detect defects both within the "inter-assemblages" and also within the system as a whole.

Security Testing:

This attempts to verify whether the protection mechanism is built into the system. In this testing the authentication of the users is checked and authorized users are allowed to access the database.

Performance Testing:

Performance testing can serve different purpose. It can demonstrate that the system meets the performance criteria. It can compare two systems to find which performs better, or it can measure what parts of the system or workload cause the system to perform badly. In the diagnostic case, software engineers use tools such as profilers to measure what parts of a device or Software contributes most to the poor performance.

Acceptance Testing

Acceptance testing can mean one of two things:

1. A smoke test is used as an acceptance test prior to introducing a new build to the main testing process, i.e., before integration or regression.
2. Acceptance testing performed by the customer, often in their lab environment on their own hardware, is known as user acceptance testing (UAT). Acceptance testing may be performed as part of the hand-off process between any two phases of development.

Test Cases:

Test Case	Input	Expected-Output	Actual-Output	Error handling	Result
1	RBS, PPBS, FBS, HBA1C Enter at least one of input must be required.	Provide the result based on algorithm. (Preferences as PPBS, HBA1C, FBS, RBS)	Find the correct stage.	If error happened in algorithm correct its condition and recheck with manually.	Get stage of diabetics.
2	Provide patient id.	All information about patient.	Get patients complete information.	Cross check with database.	Get information of patient with his id
3	Input for tablets.	Data inserted successfully.	Insert the data to customer_info database.	Provide data in JSON format.	Insert data smoothly into the database.

4	If the RBS<100, HBA1C<6.5, PPBS or RBS<200	It's the BT1 (BT is a variable)	It's BT1 value	If the value is not given properly check the code.	Pass BT1 value in algorithm.
5	If the RBS<150, HBA1C<7.5, PPBS or RBS<300	It's the BT2	It's BT2 value	If the value is not given properly check the code.	Pass BT2 value in algorithm.
6	If the RBS<200, HBA1C<8.5, PPBS or RBS<400	It's the BT3	It's BT3 value	If the value is not given properly check the code.	Pass BT3 value in algorithm.
7	If the RBS<200, HBA1C<8.5, PPBS or RBS<400	It's the BT4	It's BT3 value	If the value is not given properly check the code.	Pass BT3 value in algorithm.

CHAPTER 7

RESULTS

API f... / sta... / {{host}}info_from_customers?CallSid=40158&PatientName=Ashfaq&mobile_no=9620042745&isTablets=true&n...

Save

Send

POST {{host}}info_from_customers?CallSid=856412&PatientName=Pratap S&mobile_no=9875462416&isTablets=true&noOfTabs=2&isInsulin=true&numbe...

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/> CallSid	856412			
<input checked="" type="checkbox"/> PatientName	Pratap S			
<input checked="" type="checkbox"/> mobile_no	9875462416			
<input checked="" type="checkbox"/> isTablets	true			
<input checked="" type="checkbox"/> noOfTabs	2			
<input checked="" type="checkbox"/> isInsulin	true			
<input checked="" type="checkbox"/> numberOfInsulin	empty			
<input checked="" type="checkbox"/> tablets	["nad","nad"]			
<input checked="" type="checkbox"/> Mg	["500","500"]			
<input checked="" type="checkbox"/> times	["1","1"]			

Body Cookies Headers (10) Test Results

Status: 200 OK Time: 234 ms Size: 328 B Save as Example

Pretty Raw Preview Visualize JSON

```
1
2
3
{"status": "success"}
```

id	CallSid	PatientName	mobile_no	tablets	isTablets	noOfTabs	isInsulin	Mg	times	age	gender	city	Diabetes_duration	FBS
18	564596	Vatsala	919798564852	["nad","nad"]	true	2	false	["500","500"]	["1","1"]	29	Female	Chikkaballapur	5	empty
19	854695	Vasudev Bhajantri	919596964690	["nad","nad"]	true	2	false	["500","500"]	["1","1"]	35	Male	Channasandra	6	85
20	856493	Praveen	919594265120	["nad","nad"]	true	1	false	["500","500"]	["1","1"]	38	Male	Davanagere	5	82
21	856493	Praveen	919594265120	["nad","nad"]	true	1	false	["500","500"]	["1","1"]	38	Male	Davanagere	5	82
22	856493	Praveen	919594265120	["nad","nad"]	true	1	false	["500","500"]	["1","1"]	38	Male	Davanagere	5	82
23	856493	Praveen	919594265120	["nad","nad"]	true	1	false	["500","500"]	["1","1"]	38	Male	Davanagere	5	82
24	856493	Praveenkumar S M	9195942651556	["nad","nad"]	true	2	false	["500","500"]	["1","1"]	40	Male	Davanagere	5	82
26	856412	Pratap S	919875462416	["nad","nad"]	true	2	true	["500","500"]	["1","1"]	28	Male	Mysore	1	305
32	884412	Pushapa	919764152487	["nad","nad"]	true	2	false	["500","500"]	["1","1"]	54	Female	Rajajinagar	5	empty

CURILUX DIABETICS APP

PPBS	HBA1C	RBS	date	stage	created_at	updated_at
empty	empty	empty	24-11-2019	3	NULL	NULL
150	5.5	140	24-11-2019	3	NULL	NULL
empty	empty	empty	24-08-2019	3	NULL	NULL
empty	empty	empty	24-08-2019	3	NULL	NULL
empty	empty	empty	24-08-2019	3	NULL	NULL
empty	empty	empty	24-08-2021	3	NULL	NULL
506	9.3	empty	20-05-2023	4	NULL	NULL
142	6.5	empty	14-03-2023	3	NULL	NULL

API for miniproject / stageapifinal / doctor_form

Save

Send

POST

{{host}}doctor_form

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Cookies

Beautify

```
1 {
2   "CallSid": "564596",
3   "mobile_no": "9798564852",
4   "name": "Vatsala",
5   "diabetes_stage": 3,
6   "new_med": "Anything",
7   "Diabeties_duration": 1,
8   "stop_med": "Stop all the previous diabetes medicines",
9   "continue_med": "Continue all the previous medicines except old dia",
10  "reccBreakfast": "Morning Feed --- 1 scoop in half glass of warm mil",
11  "reccLunch": "1 Chapathi or Roti , 1 vegetable baaji , 1 pulses ",
12  "reccSnacks": "1 small glass sugarless Tea",
13  "reccDinner": "1 Chapathi or Roti , 1 pulses baaji 1 Apple",
14  "remarks": "Empty"
15 }
```

Body

Cookies

Headers (10)

Test Results

Status: 200 OK

Time: 352 ms

Size: 339 B

Save as Example

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "msg": "inserted successfully"
3 }
```

CURILUX DIABETICS APP

	id	CallSid	mobile_no	PatientName	diabetes_stage	Diabetes_duration	stop_med	new_med	continue_med	reccBreakfast	reccLunch	reccSnacks	reccDinner	remarks
							diabetes medicines	Food,Voglimore 2	except old dia	glass of warm mil	baaji , 1 pulses	sugarless Tea	pulses baaji , 1 glass milk	
	4	3512456	8565456317	Raju	3	2	Stop all the previous diabetes medicines	Morning Feed,Before Food,Voglimore 3	Continue all the previous medicines except old dia	Morning Feed - 1 scoop in half glass of warm mil	1 Chapathi or Roti , 1 vegetable baaji , 1 pulses	1 small glass sugarless Tea	1 Chapathi or Roti , 1 pulses baaji , 1 glass milk	empty
	5	3512442	9955404127	Raajesh	2	2	Stop all the previous diabetes medicines	Morning Feed,Before Food,Voglimore 2	Continue all the previous medicines except old dia	Morning Feed - 1 scoop in half glass of warm mil	1 Chapathi or Roti , 1 vegetable baaji , 1 pulses	1 small glass sugarless Tea	1 Chapathi or Roti , 1 pulses baaji , 1 glass milk	empty
	6	524561	8264953175	Lavanya	2	1	Stop all the previous diabetes medicines	Morning Feed,Before Food,Voglimore 2	Continue all the previous medicines except old dia	Morning Feed - 1 scoop in half glass of warm mil	1 Chapathi or Roti , 1 vegetable baaji , 1 pulses	1 small glass sugarless Tea	1 Chapathi or Roti , 1 pulses baaji	empty
	7	564596	9798564852	Vatsala	3	1	Stop all the previous diabetes medicines	Morning Feed,Before Food,Voglimore 3	Continue all the previous medicines except old dia	Morning Feed - 1 scoop in half glass of warm mil	1 Chapathi or Roti , 1 vegetable baaji , 1 pulses	1 small glass sugarless Tea	1 Chapathi or Roti , 1 pulses baaji 1 Apple	empty

API for miniproject / stageapifinal / patientdetails

POST `{{host}}patientDetails?id=26` Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description
<input checked="" type="checkbox"/> id	26	

Body Cookies Headers (10) Test Results Status: 200 OK Time: 235 ms Size: 743 B Save as Example

Pretty Raw Preview Visualize JSON

```

2  "id": 26,
3  "CallSid": "856412",
4  "PatientName": "Pratap S",
5  "mobile_no": "919875462416",
6  "tablets": "\"[\\\"nad\\\"\\\",\\\"nad\\\"\\\"]\"",
7  "isTablets": "true",
8  "noOfTabs": "2",
9  "isInsulin": "true",
10 "Mg": "\"[\\\"500\\\"\\\",\\\"500\\\"\\\"]\"",
11 "times": "\"[\\\"1\\\"\\\",\\\"1\\\"\\\"]\"",
12 "age": "28",
13 "gender": "Male",
14 "city": "Mysore",
15 "Diabetes_duration": "1",
16 "FBS": "305",
17 "PPBS": "506",
18 "HBA1C": "9.3",
19 "RBS": "empty",
20 "date": "20-05-2023",
21 "stage": 4,
22 "created_at": null,

```

CONCLUSION

Jeevini i-NOS App is designed to treat the diabetes patient by using modern technologies. The doctor can treat the patients through this app. In addition, the dietitians are also used to support the diet plans of patient. It uses some of medical information from the patient. Based on the inputs of patient the stage of diabetic can be found. This will reduce time required by the doctor to treat the patient and also patient can get easy assistance. The patient data is securely maintained in system. The patients are managed easily. The assistances are given to patients based on their condition. Diet plans and improvement of patients can be provided to doctor. So, lot of paper work and prescription providing is reduced.

FUTURE SCOPE

- This system may also feature a teleconsultation option, in which patients can directly contact a doctor and obtain advice on their problems.
- This application allows users to manage not just consultations but also medical test appointments.
- Artificial Intelligence can calculate and recommend calorie intake, fat intake, and carbohydrate intake. Also create a meal recommendation system so that users may eat the foods that are recommended to them in order to keep their health.

REFERENCES/BIBLIOGRAPHY

1.Laravel Course

<https://www.udemy.com/course/php-with-laravel-for-beginners-become-a-master-in-laravel/>

2. Laravel (PHP)

<https://laravel.com/docs/7.x>

3. Alternate Applications for reference

<https://www.topbestalternatives.com/ketodiet/>

4. Google Scholars

https://ieeexplore.ieee.org/abstract/document/9515229?casa_token=X1Em_39W-1sAAAAA:tt2j75ARta2SrD1nV3NpaHZdaGM8qrXFVYUSf4U4MXlofjO0dLbQMhral0iHZRhQx3T9xBBcW0U

6. Centre for Disease Control and Prevention. National Diabetes Statistics Report (2017).

Estimates of diabetes and its burden in the United States. 2017, [online] Available:

https://effectivehealthcare.ahrq.gov/sites/default/files/pdf/technical-brief-31_mobile-health-applications-for-self-management-of-diabetes.pdf.

7. "Track calories in Indian curries with Healthify Me". Daily News and Analysis. 1 May 2015. <https://en.wikipedia.org/wiki/HealthifyMe>

8. Research2Guidance. mHealth App Developer Economics. 2017.

<https://research2guidance.com/diabetes-management-solutions-still-considered-the-number-one-therapy-field-preference-for-app-developers/>. Accessed 4 Apr 2017.

9. Aungst T. Evaluation of Glucose Buddy app as diabetes monitoring tool for patients and clinicians.

10. [online] Available: <https://www.commonwealthfund.org/blog/2019/hospital-price-transparency-making-it-useful-patients>.

11. <https://www.perfecto.io/blog/what-is-flutter-> Flutter technologies used.

12. <https://www.cloudways.com/blog/connect-mysql-with-php/>: PHP and Database Information

APPENDIX

Technology Used

In this “JEEVINI i-ONS” used flutter for front end and laravel framework for back end. PHP and MySQL used for implementing backend. In this project sending OTP to registered mobile number. For sending mobile number we used third party twilio platform.

Laravel

Laravel is a development platform with a syntax that is both expressive and elegant. While we sweat the minutiae, a web framework gives structure and a place to start for your applications, helps to focus on developing something extraordinary. Laravel offers a great developer experience while also including powerful capabilities like substantial middleware, a creative database interface layer, queues and scheduled operations, unit and integration and system testing, and more.

When creating a web application, you can use a range of tools and frameworks. We believe, however, that Laravel is the ideal solution for building full-stack web apps. Laravel sail is a command-line tool that allows you to interact with Laravel’s default Docker configuration. If you’re new to Docker and want to construct a Laravel app with PHP, MySQL, and Redis, Sail is a wonderful place to start. Laravel is a PHP framework based on the MVC model.

M: A database model is a class that manipulates databases. For example, if an application has users, we’ll have a user model which interacts with a database to query the user table. We will have user’s table if we have user’s model. From the example, we can deduce that the model will include a table for that specific model.

V: A view is an HTML document management class. Everything we see on the software in the browser is the view or representation.

C: The controller serves as link between the model and view. The class that obtains the data from the model and sends it to the view is known as controller.

Laravel is an open-source PHP framework. It also includes essential aspects of PHP frameworks like CodeIgniter and Yii, as well as other programming languages like Ruby on Rails.

PHP

The Hyper Text Pre-processor (HTTP) is a scripting language that excels at web development. It was created by Rasmus Lerdorf, a Danish-Canadian programmer in 1994. A PHP interpreter implemented as a module is commonly used to process PHP code on the web server. The outcome of PHP code being interpreted and executed on a web server-which might be any type of data such as produced HTML or binary image data-would make up the entirety or a portion of an HTTP response. Outside of the web, PHP can be used for a variety of activities such as standalone graphical programs and robotic drone control. PHP code can also be run directly from the command prompt.

PHP is a server-side programming language for creating dynamic web applications that leverage the MySQL database.

- It is in control of the dynamic content, database, and session tracking on the website.
- PHP has the ability to create sessions.
- It can read and write to the cookie's variable.
- It aids in the encryption and validation of data.
- PHP supports a variety of protocols, including HTTP, POP3, SNMP, LDAP, IMAP and others.
- PHP allows you to limit user access to specific areas of your website.
- One of the main reasons why PHP is the best language to learn is because it is simple to install and configure.
- PHP is capable of handling forms, such as collecting input from user via forms, saving it to a database, and returning valuable information to the user consider the registration form.

Database MySQL

MySQL is currently the most widely used database management system software. Oracle Corporation supports this open-source database software. When compare to SQL Server and Oracle Database, it is a fast, scalable, and simple-to-use database management system. It's commonly combined with PHP scripts to construct capable server-side or internet enterprise applications.

MySQL is a widely used relational database management system (RDBMS) that is used by both small and large enterprises. MySQL is developed, marketed, and supported by MySQL AB, a Swedish company. For a multitude of reasons, MySQL is becoming increasingly popular.

- MySQL is a powerful program in and of itself. It is capable of handling significant portion of the functionality of the most expensive and powerful database products.
- MySQL makes use of the well-known data language as standard.
- MySQL is a fast system that handles large data sets well.
- MySQL is especially well-suited to PHP, the most widely used web development language.
- MySQL is capable of handling large databases with tables containing up to 50 million rows. A table's default file size restriction is 4GB; also, you can increase it to a theoretical maximum of 8 million gigabytes.