

# REPORT ON HEART MONITORING SYSTEM

BISHRUTI SIKU - 181404

## INTRODUCTION

Heart Monitoring System is a console application which helps its users to track the status of their heart. This application obtains the current health status from the user and based on the information obtained, it automatically sets up goals, suggests activities to achieve that goal and recommends appropriate food for them. In addition to that, whenever the user logs into their account, they are always welcomed by a 'Thought of the Day' which is a quotation message.

## FEATURES

Currently this application supports Admin View and User View.

### Admin View

Admin can view list of all the users. In addition to that, they can also check personal information and the current health profile of any user they want. Additionally, they can also delete the user.

### User View

- **New User**

If the user is new to the application, then they can create their profile by entering their personal details and current health information. After creating their profile, they will be welcomed with a 'Thought of the Day' and based on their health information, they would be assigned with goals and suggested with some activities to achieve the goal. They will also be recommended with the food that is good for their health.

- **Existing User**

If the user is existing, then they can view their profile where they are welcomed with a 'Thought of the Day' and can see their goals and suggested activities and the new recipe that they can try out. Furthermore, they can also edit their personal information, create a new measurement details, view existing measurement details, view a particular measurement, and update it if they want.

## SYSTEM ARCHITECTURE

### 1. External Adapter Service

External Adapter Service is responsible to interact with external services. In fact, this service creates a link between our application and Edamam - Eat Better! and Quotes on Design.

- **Edamam - Eat Better!**

External Adapter Service obtains the information about the type of food that we are looking for, from Edamam - Eat Better. For instance, if we supply the types like high protein, low carbohydrate,

balanced diet, low fat, absolutely low fat then it will return the food which has the respective quality. The communication between these services are in REST with JSON objects.

- **Quotes on Design**

External Adapter Service obtains quotes which we use in our application as 'Thought of the Day' from Quotes on Design. The communication between these services are in REST and uses JSON objects.

After obtaining necessary data, this service pass them to the Storage Service.

## **2. Data Service**

Data Service deals with all the data related requests in the application. This service is based on SQLite database and provide data related operations to the client services which is Storage Service in this application. This service uses SOAP Protocol to communicate with other services.

## **3. Storage Service**

Storage Service serves as a junction where all the required data for the application is found. It obtains the suggested Food and Quote from External Adapter Service with which it communicates in REST with JSON objects. It also acquires data related to User, their Health Profile and Health Measure History from Data Service with which it communicates with SOAP Protocol. In addition to that, it scrubs the obtained data such that only the data required by the upper layers which are the Business Logic Layer and Process Centric Layers are passed to them as JSON object through RESTful APIs.

## **4. Business Logic Service**

Business Logic Service implements all the logics that are used in this application. It obtains data from Storage Service through JSON object in RESTful APIs, manipulates them to obtain meaningful response and pass it to the User Interface Service as a JSON object in RESTful API. The chief logic that is implemented in this service is to automatically recommend appropriate Goal, Activities, Food with respect to Blood Pressure and BMI of the user. Some of the logics in the application is shown in the tables below.

Systolic Blood Pressure	Diastolic Blood Pressure	Result
< 70	< 40	Very Low
70 - 90	40 - 60	Low
90 - 120	60 - 80	Normal
120 - 140	80 - 90	Pre High
140 - 190	90 - 100	High
180 >	100 >	Very High

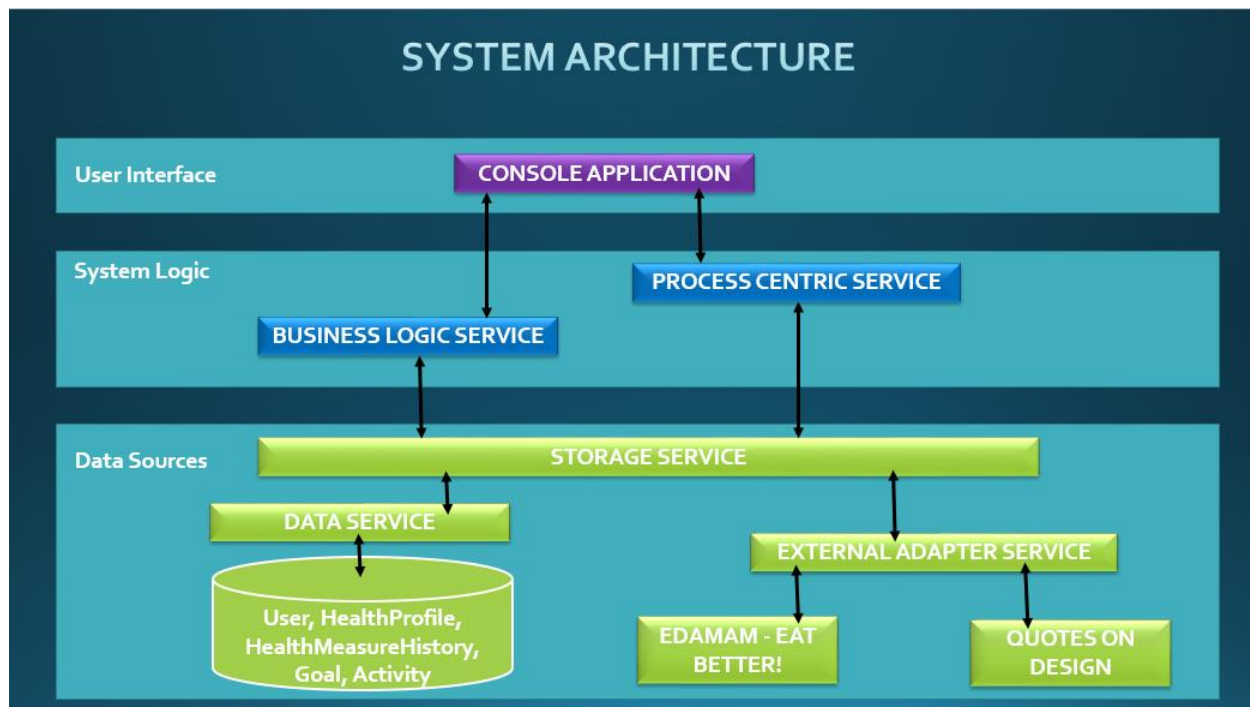
BMI	Result
< 18.5	Under Weight
18.5 – 24.9	Healthy
25 – 29.9	Over Weight
30 - 35	Obese
35 >	Very Obese

## **5. Process Centric Service**

Process Centric Service handles all the requests that changes the state of this application. It obtains data from User Interface Service as a JSON object through REST and pass it to the Storage Service in the same format after performing operations that change the state of the application.

## **6. User Interface Service**

User Interface Service provides interface where the user interacts with the application. This service passes JSON object through RESTful APIs to interact with Business Logic Layer where it gets all the User related data and Process Centric Layer where it sends all data.



## FURTHER ENHANCEMENTS

- User Interface

Users these days prefer to use either smartphone or tablets to use applications like this. So this application can be extended by developing a web page.

- Authentication

As this application has User View and Admin View, Role Based Access Control (RBAC) can be implemented to restrict system access to authorized users.

- Implementation of Graph while showing Measure History

Since a user can view the history of a particular measure type, graph can be implemented to make the user experience even better as it is easy to track changes in graph.

- Push Notifications

Accuracy of this application depends on the information provided by the users. Thus, push notifications and reminders can be sent to the users to ask them to add their new health profile in certain interval of time.

## CONCLUSION

In a nutshell, this application obtains user information and processes it to suggest Goal, Activities and Food. For the implementation of these features, we use five internal web services, two external web services and one console application. The interaction between these services are either with JSON objects which are passed through RESTful APIs or in SOAP.