DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
THE UNIVERSITY OF TEXAS AT ARLINGTON

ARCHITECTURAL DESIGN SPECIFICATION
CSE 4316: SENIOR DESIGN I
SPRING 2020

TEAM INGENUITY
STATE FARM FIT

CHELSEA MAY
WEI SHI
DEVI TRIPATHY
SUDEEP BHADEL

Revision History

Revision	Date	Author(s)	Description
0.1	04.11.2020	CM, WS, DT, SB	document creation
1.0	04.12.2020	CM, WS, DT, SB	complete draft
1.1	05.10.2020	WS	V2 update

CONTENTS

List of Figures

List of Tables

1 Introduction

Upon completion, our product will provide State Farm customers with a mobile application to track their health information in order to calculate a corresponding fitness level. This corresponding fitness level will then be used to potentially provide life insurance customers with a reduced life insurance rate. The person's heart rate, sleep, exercise, driving habits, and location will be tracked by our product.

1.1 Purpose and Use

Our mobile application will be similar to a fitness app and will extract data from a Fitbit in order to track the user's health information. The user should be able to log-in, view their health information, as well as their current fitness score. The health information that will be recorded will include the person's heart rate, steps, sleep, driving habits, and how often they exercise/calories burned. Our application will also have GPS capabilities in order to track the user's location.

1.2 Intended Audience

The intended audience of our product is the customers of State Farm. Specifically, the customers that are paying for life insurance at State Farm.

1.3 Features & Functions

As in Figure 1, the user will first see a Get Started screen with "State Farm Fit" title at the top. Also shown in Figure 1 the user will be able to view their health information such as their heart rate, steps, sleep, calories burned, etc. Our application will also have GPS capabilities in order to track the user's location. The users will also be able to view their current fitness score which will be used to determine whether or not the user gets a discount.

1.4 Product Interfaces

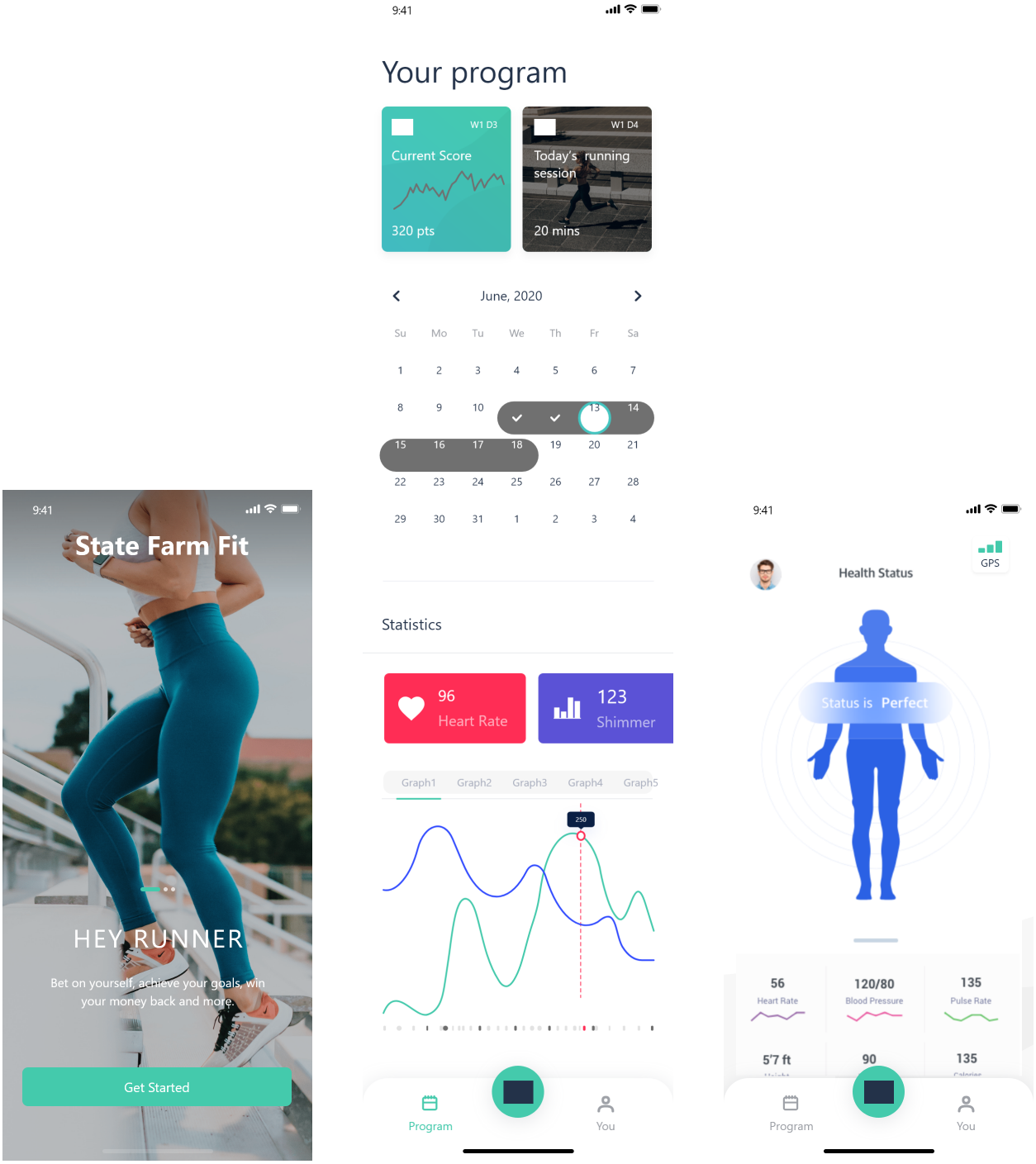

Figure 1: Potential product interfaces

2 System Overview

This section should describe the overall structure of your software system. Think of it as the strategy for how you will build the system. An architectural "layer" is the top-level logical view, or an abstraction, of your design. Layers should be composed of related elements of similar capabilities, and should be highly independent of other layers, but should have very clearly defined interfaces and interactions with other layers. Each layer should be identified individually and should be unique as to its function and purpose within the system. This section should also contain the high-level block diagram of the layers, as shown in the example below, as well as detailed descriptions of the functions of each layer.

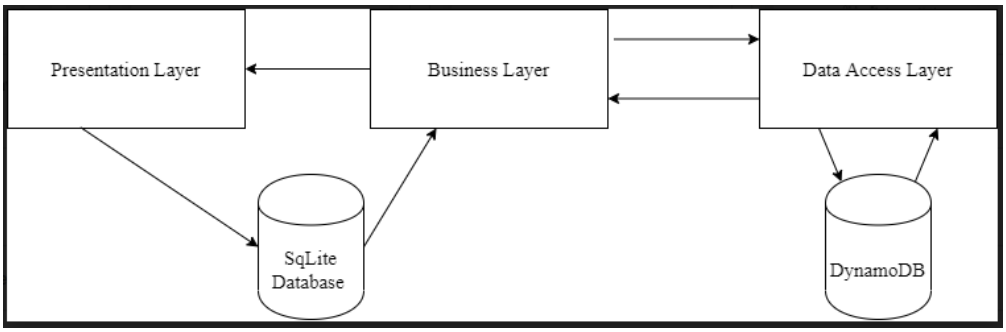

Figure 2: A simple architectural layer diagram

2.1 Presentation Layer Description

The presentation layer will include the android interface that the user would interact with as well as a web UI for an administrator to interact with. The android UI will be on the user's phone and will include a log-in screen, screens for the user to view their health data, and a screen for the user to view their fitness score. The presentation layer will also be responsible for capturing the health data from the wearable device, then sending that data to the SqLite database. Another component of the presentation will be the Web UI. The Web UI will be used by system administrators. With the Web UI administrators will be able to log-in and access user data.

2.2 Business Layer Description

The business layer will include AWS, from which we will be using many different services. The business layer will also be where the data analysis is done. There will be a Route53 subsystem, this service will be used to manage all incoming and outgoing traffic requests and will act as DNS Manager. The VPC service will allow a private network to be built, where all of the cloud computing resources will be allocated. A load balancer subsystem will also be present to allow AWS ELB to be spawned up into multiple instances and easily route traffic through them. Next, we will have an EC2 subsystem, EC2 is a service provided by AWS for aquiring computing power. AWS EC2 will be used to host our Wordpress site and necessary files. The API Gateway subsystem will source a primary point of contact to our VPC through HTTP/HTTPS traffic routed via Route53. The Web App Service will be responsible to allow multiple entry points for users and will also allow admins to check the system status and data analytic. The final subsystem in the business layer is the deep learning subsystem. The deep learning subsystem will be responsible for analyzing the data. There will be three folds to this stack: a CNN based classification Model, a BI directional LSTM for forecasting, and a ML based method for fraud detection.

2.3 Data Access Layer Description

The data access layer will store the data that will be collected and provide access to the data. This layer will be composed of 4 subsystems: User, Administrator, Raw data, and Analyzed data. It will interact with both the business layer and the DynamoDB. It will receive data from the business layer and also send data to be stored in the DynamoDB. The user subsystem will be where the user credentials are verified. The administrator subsystem will be where the administrators credentials are verified. The raw data subsystem will simply communicate the raw data it receives from the business layer to the online database, DynamoDB, so that the administrator can view it. Lastly, the analyzed data subsystem is present to communicate the analyzed data from the business layer to the online database, DynamoDB.

3 SUBSYSTEM DEFINITIONS & DATA FLOW

The data flow diagram shown below gives us clear picture of the flow of data from capturing it to processing and saving it. It is a 3-tier application architecture with presentation layer, business layer and data access layer serving as three main layers of this system architecture. The Presentation layer has three subsystems: capture, Android UI, and Web UI. Capture subsystem will run on a mobile device to capture important data from user though Fitbit API, include heart rate, location, activities, and sleep data. The raw data will be stored at a local database SQlite, which then used by the Android UI and Web UI to let the costumer view their health data. Then, the business layer, where the main logic of the system reside is composed of Amazon Web Service (AWS), this is used to host both the app and Web UI. The subsystems are EC2, API Gateway, Web app service, and deep learning. Here, EC2 will allows us to get necessary computing resources required to run our business logic. API gateway will will source a primary point of contact to our VPC through HTTP/HTTPS traffic routed via Route53. Web app service will be responsible to allow multiple entry points for users and will also allow Admins to check system status and data analytic. Deep learning stack will be responsible for analysing the data. Finally, the data access layer comes into play where the database, DynamoDB needed to be accessed for read or write purpose. This layer has 4 subsystems: user, administrator, raw data analyzed data. User and administrator subsystem is where the credentials of user/administrator gets verified and gives certain privilege to data access based on user type. Raw data and analyzed data is simply used for communicating data to or from database.

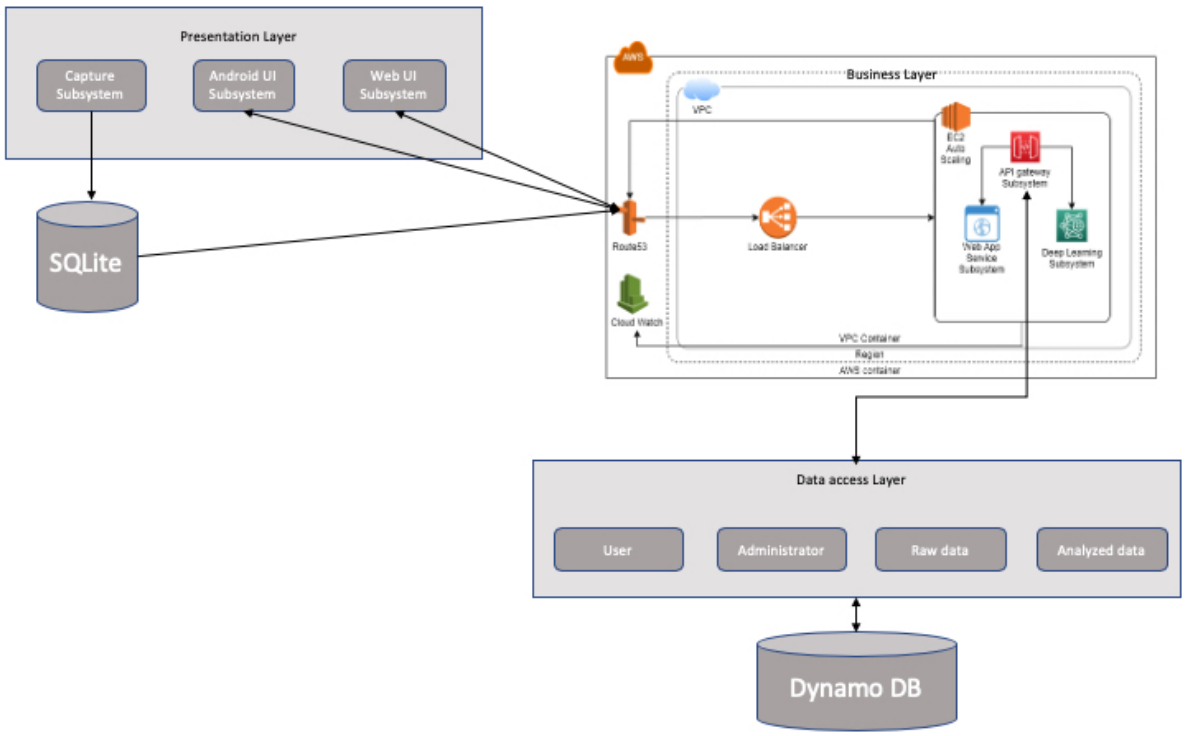

Figure 3: data flow diagram

4 Presentation Layer Subsystems

Presentation layer includes three isolated subsystems which run on a android device or an end system. The capture subsystem response for collecting user Location, heart rate, activities, and sleep data from a Fitbit smart watch. The android user-interface subsystem will allow user to Login, register , and modify their account, get a quote from the insurance company, and review his or her activities on their mobile device when the device is online. They also are allowed use a website to access the data.

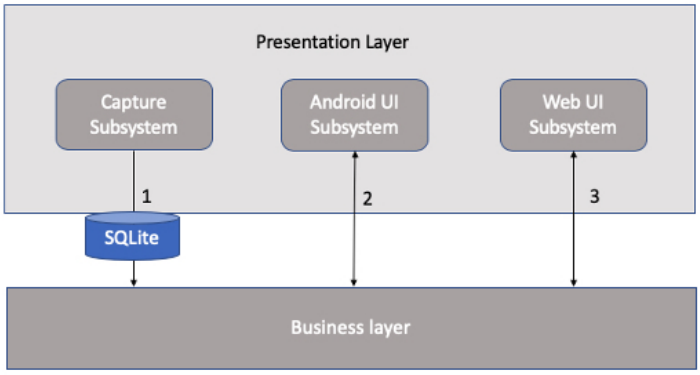

Figure 4: Presentation layer

4.1 Capture Subsystem

Capture subsystem will run on a mobile device to capture important data from user though Fitbit API, include heart rate, location, activities, and sleep data. The raw data will be stored at a local database SQlite, a subsystem in the business layer will upload and update the data between local database and Dynamo DB.

4.1.1 Assumptions

Assumption 1, user will use a Fitbit smart watch.
Assumption 2, we will use SQLite as local database.
Assumption 3, a subsystem in the business layer will synchronize data between the local database and online database.

4.1.2 Responsibilities

The subsystem responsible for collecting user health data from user's smart watch such as heart rate, location, activities, and sleep. the data will be stored at a local database.

4.1.3 Interfaces

The input is user health data. The output is same. the data will be stored at a local database.

ID	Description	Inputs	Outputs
#01	Description of the interface/bus	User health data	user health data

Table 2: Subsystem interfaces

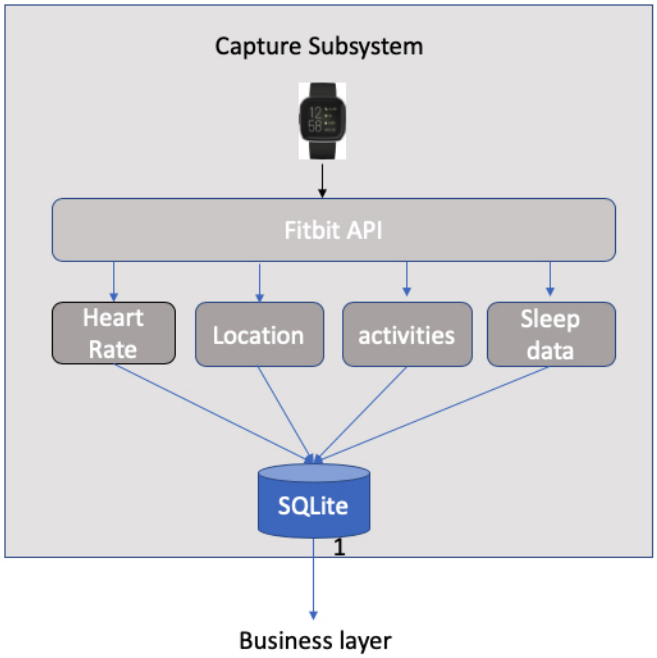

Figure 5: Capture Subsystem

4.2 Android UI Subsystem

Android UI subsystem will run on a mobile device to allow user to review their health data, manage their account, and acquire a quote from the insurance company. The quote will based on the regular quote and the user's fitness level in the database.

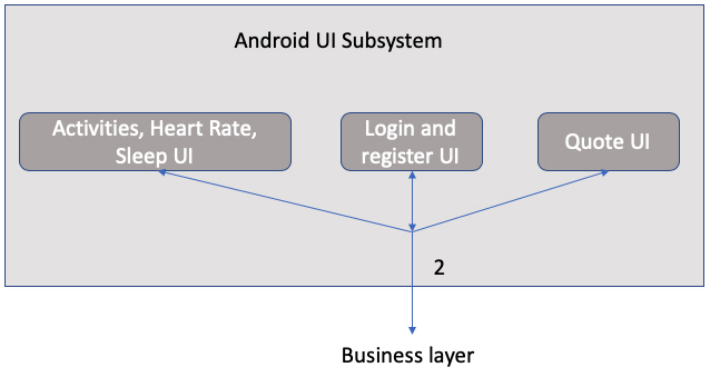

Figure 6: Android UI Subsystem

4.2.1 Assumptions

Assumption 1, user is allowed use this subsystem when the android device is online and user should login the application.
Assumption 2, we will use SQLite as local database.
Assumption 3, a subsystem in the business layer will synchronize data between the local database and

online database.

4.2.2 Responsibilities

The subsystem response for three user-interfaces, user health data view, user account management view, and quote view. The subsystem will access the local database as a cache. If user want to access the past data which will be stored in the AWS, a subsystem in the business layer will download the data to local database.

4.2.3 Interfaces

The input is user health data, account information. The output is a rate base on the information.

ID	Description	Inputs	Outputs
#02	Description of the interface/bus	user data and info	a rate

Table 3: Subsystem interfaces

4.3 Web UI Subsystem

Web UI subsystem will be hosted on AWS and be accessible to system administrators. From the web UI administrators will be able to view user health information and insurance rates, as well as their own account information.

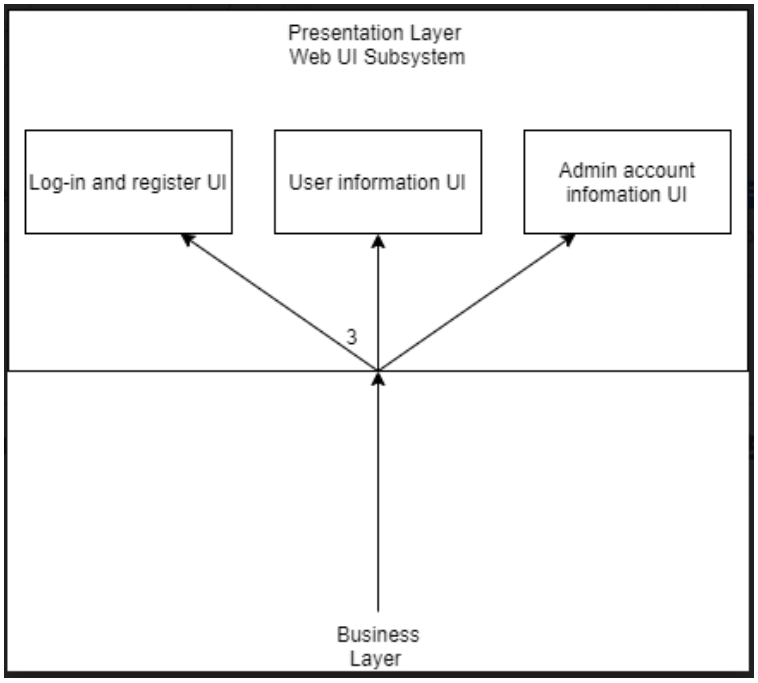

Figure 7: Web UI Subsystem

4.3.1 Assumptions

Assumption 1, administrator should be able to log into system to view user information. Assumption 2, the web UI will retrieve user data from AWS database Assumption 3, administrators should be able to view user health data in order to manage the data.

4.3.2 RESPONSIBILITIES

The subsystem is responsible for four user-interfaces for the system administrators to view. The user health data, user account information, user life insurance rate, and the administrators own account information. The subsystem will access the AWS database in order to display the data.

4.3.3 INTERFACES

ID	Description	Inputs	Outputs
#01	Administrator log-in screen	Username and password	Go to screen which will show user health information, account information, and insurance rate.
#02	Administrator account information screen	Username and password	Screen which will allow them to change username, password, security questions, etc.

5 Data Access Layer Subsystems

The main objective of data access layer is to store data and to provide access to stored data. This layer is composed with 4 subsystems: User, Administrator, Raw data and Analyzed data, and a database, DynamoDB which is used for managing read or write access.

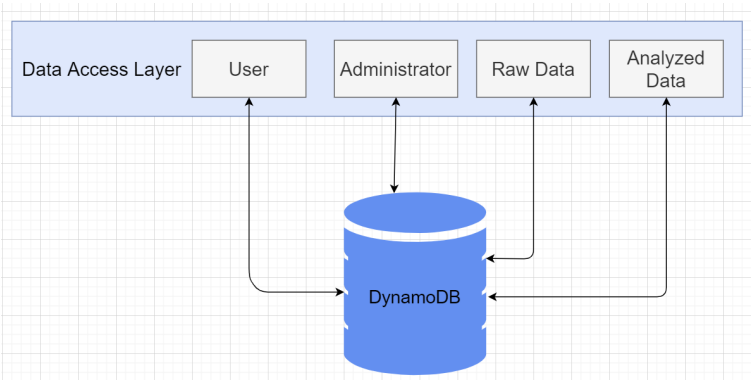

Figure 8: Data Access Layer

5.1 User

User layer is where the credentials i.e. user id and password of the user is verified. If any of the credentials doesn't matched to one that is previously saved in the database, captured while user sign up then , the user is not allowed access to the system.

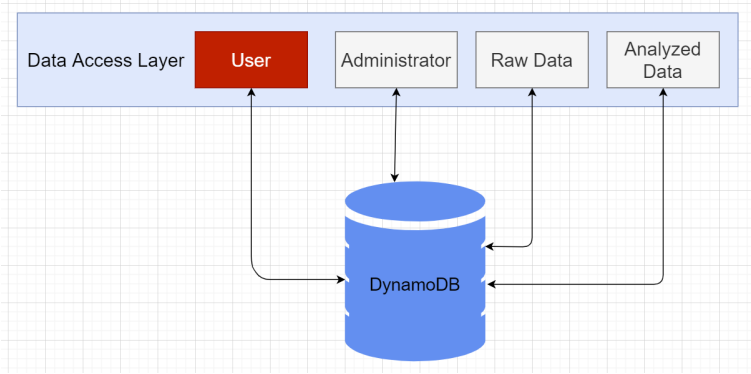

Figure 9: User Subsystem

5.1.1 Assumptions

Assumption 1, user has proper connection in android device.
Assumption 2, a subsystem in business layer help in data transfer between presentation to data access layer.
Assumption 3, proper connection setup with the database, DynamoDB

5.1.2 Responsibilities

The subsystem is responsible for verifying if the credentials provided during sign in process matched the stored credentials.

5.1.3 Interfaces

ID	Description	Inputs	Outputs
#01	Data Layer	User Id Password	Boolean

Table 4: User Interfaces

5.2 Administrator

This subsystem gets the admin login credentials with the help of Web UI Subsystem in the presentation layer, then a subsystem in business layer transfers data to data access layer from which it first verifies the admin credentials and then gives admin access to the database data.

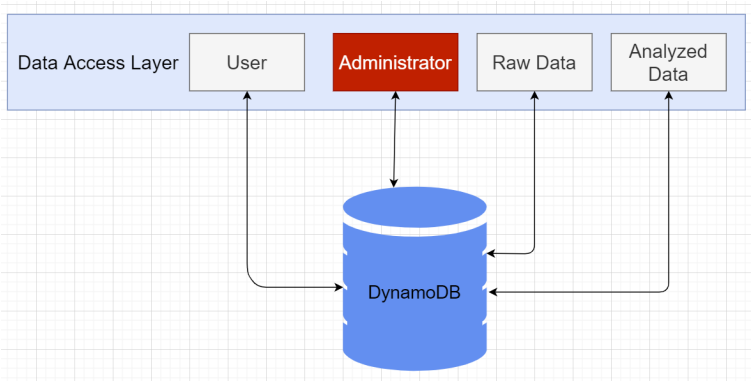

Figure 10: Administrator Subsystem

5.2.1 Assumptions

Assumption 1, existence of Web UI from which admin can login
Assumption 2, existence of subsystem in Business Layer responsible for transfer of data between Presentation and Data access layer
Assumption 3, proper connection setup with the database, DynamoDB

5.2.2 Responsibilities

The subsystem is responsible for verifying if the admin credentials provided during sign in process matched the stored credentials and lets the admin access the database data.

5.2.3 Interfaces

ID	Description	Inputs	Outputs
#01	Administrator Login UI	User ID password	Boolean
#02	Data Layer	N/A	Dataset

Table 5: Administrator interfaces

5.3 Raw Data

This layer simply communicates the raw data captured from the fitbit watch in presentation layer, transferred through the business layer to the online database, DynamoDB, which then can be viewed by the administrator.

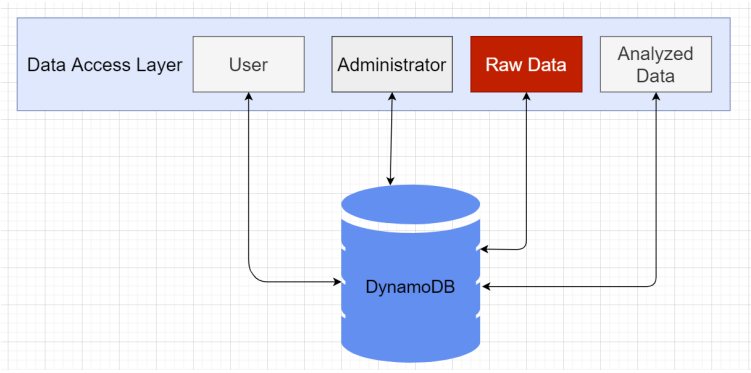

Figure 11: Raw Data Subsystem

5.3.1 Assumptions

Assumption 1, user uses fitbit watch
Assumption 2, subsystem in business layer synchronize data between local database and online database
Assumption 3, proper connection setup with the databases
Assumption 4, captured raw data should not be manipulated

5.3.2 Responsibilities

The main purpose of this subsystem is to communicate the raw data captured by the fitbit watch to the DynamoDB.

5.3.3 Interfaces

ID	Description	Inputs	Outputs
#01	Data Layer	captured health data	health data

Table 6: Raw Data Interfaces

5.4 Analyzed Data

The main purpose of this layer is to communicate the analyzed data such as, fraud activity,fitness score and risk factor from a subsystem of the business layer, AWS to the online database, DynamoDB.

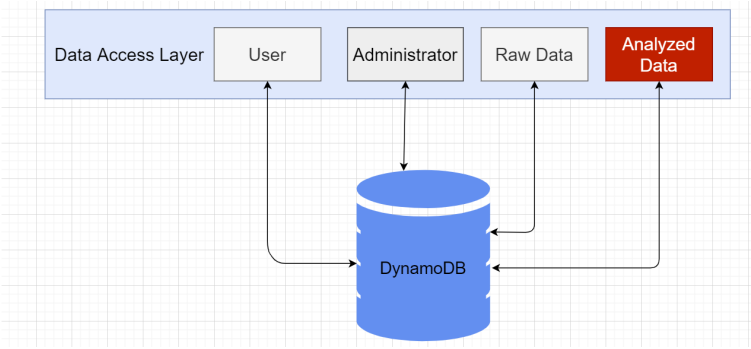

Figure 12: Analyzed Data Subsystem

5.4.1 Assumptions

Assumption 1, a subsystem in business layer which is responsible for producing analyzed data and then transferring to Data access layer
Assumption 2, proper connection setup with the AWS and DynamoDB

5.4.2 Responsibilities

This subsystem is responsible for communicating the analyzed data such as, fraud activity,fitness score and risk factor from a subsystem of the business layer, AWS to the online database, DynamoDB.

5.4.3 Interfaces

ID	Description	Inputs	Outputs
#01	Data Layer	Analyzed User Dataset	Analyzed User Dataset

Table 7: Analyzed Data interfaces

6 Business Layer

The main objective of Business layer is to host business logic and allow users to access data in a multiple way. Business layer constitutes of many components amongst them core subsystems are EC2 instance, API gateway, Web App service and Deep learning. These subsystems encapsulates the entire businesses logic and needs.

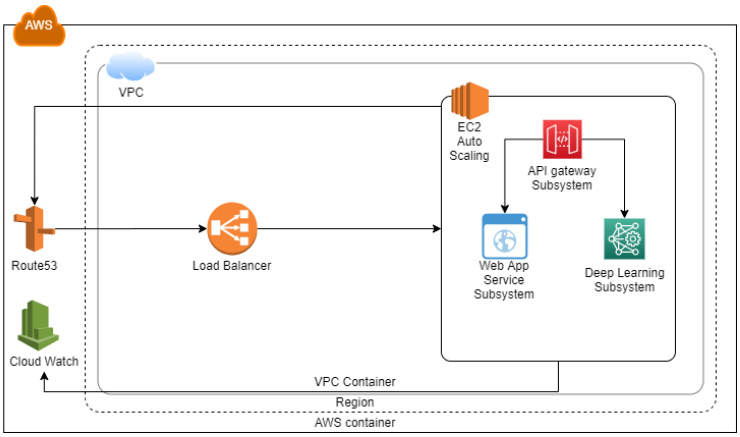

Figure 13: Business Layer

6.1 EC2

EC2 instances are virtual machines that will allows us to get necessary computing resources required to run our business logic. AWS EC2 will host our Business Logic which will have its own several components.

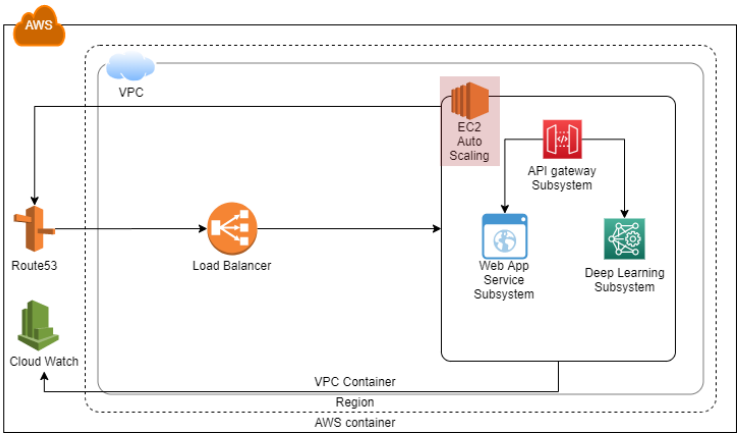

Figure 14: Business Layer

6.1.1 Assumptions

Auto scaling required by sponsor.
AWS ELB will be used to create AWS EC2 instance
AWS ELB will be used for the auto scaling

6.1.2 Responsibilities

EC2 instance is responsible for hosting the site, running docker environment scale automatically as needs grow.

6.1.3 EC2 Interfaces

ID	Description	Inputs	Outputs
#1	Route53	HTTP/HTTPS request	HTTP/HTTPS response

Table 8: EC2 interfaces

6.2 API Gateway

This will source a primary point of contact to our VPC through HTTP/HTTPS traffic routed via Route53. This service will be responsible for handling all requests send to the server.

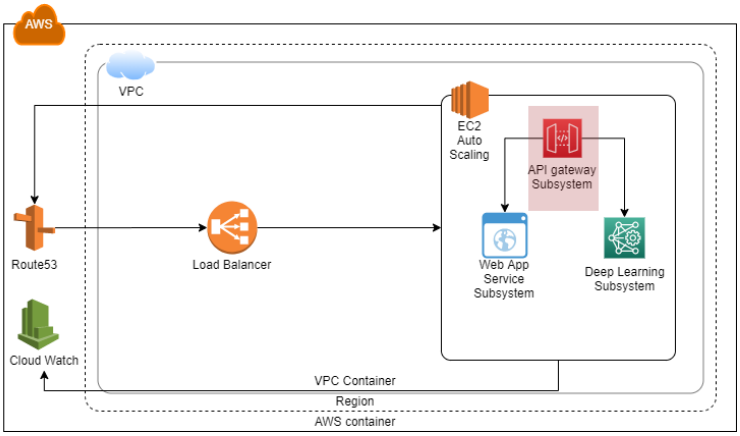

Figure 15: Business Layer

6.2.1 Assumptions

API gateway will be based on REST architecture
API will be hosted on EC2 instance

6.2.2 Responsibilities

API gateway will be the nervous system of the entire stack. It will be responsible for handling all request resolved from Route53. This will interface with Android application and web app service to handle all business operation such as Login, Sign up, data collection, score update etc. The API is responsible to interface with Data access layer to fetch and update database. API will also handle all deep learning calls and scheduled jobs.

6.2.3 API Gateway Interfaces

ID	Description	Inputs	Outputs
#1	Android interface	JSON	JSON
#2	Web Services interface	JSON	JSON
#3	Deep Learning interface	JSON	JSON
#4	Data Access interface	JSON	JSON

Table 9: API Gateway interfaces

6.3 Web App Service

This will be responsible to allow multiple entry points for users and will also allow Admins to check system status and data analytic.

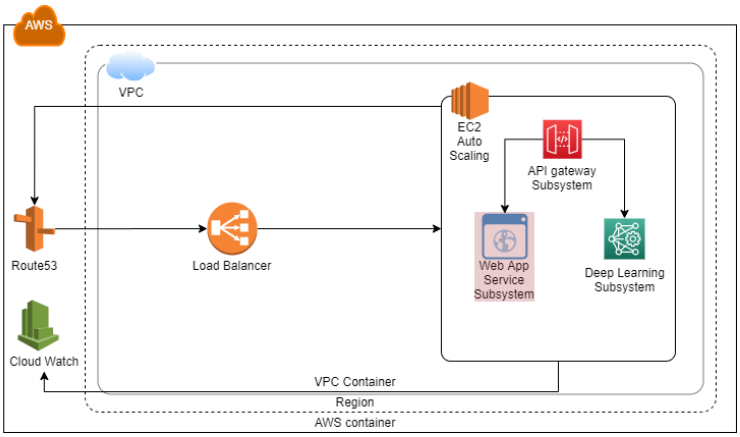

Figure 16: Business Layer

6.3.1 Assumptions

Admin access is required
Multiple user entry point is required

6.3.2 Responsibilities

This will be the core logic of web app that will be used by both users and admins to access various section of the app. The user would be able to login and access their health statistics and get review their data collection permissions. Admin will be able to access system status, data analytic on all users. The web service will be a progressive web app allowing user the versatility and functionality of a native app.

6.3.3 Web App Interfaces

ID	Description	Inputs	Outputs
#1	API gateway interface	JSON	JSON
#xx	WebUI interface	JSON	HTML

Table 10: Web App interfaces

6.4 Data Analysis Subsystem

This stack will be responsible for analysing the data. Currently there are three folds to this stack, a CNN based classification Model, a BI directional LSTM for forecasting, a ML based method for fraud detection.

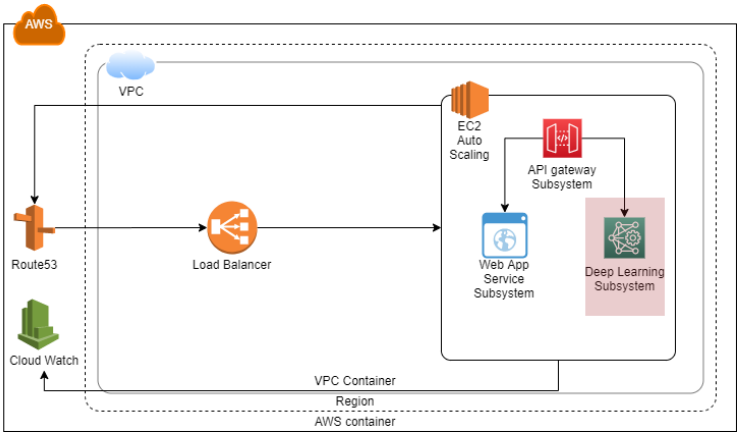

Figure 17: Business Layer

6.4.1 Assumptions

The main logic for deep learning will lay here.
Deep Learning will be used for Data Analysis.
CNN is viable for time series classification.
LSTM is viable for time series forecasting.

6.4.2 Responsibilities

This subsystem constitutes of several components. LSTM based model will be used to forecast user's trend into the future and will be feed into CNN based model for classification which will be used to categorize users in different risk factor category. Result will be an risk factor sore derived from confidence matrix of CNN classification, this will be used in a non-linear function to adjust the user's overall score.

6.4.3 Data Analysis Interfaces

ID	Description	Inputs	Outputs
#1	LSTM interface with API	Raw user data	Forecast user data
#2	CNN interface with API	User score time series User Data trend forecast data	Confidence score

Table 11: Data Analysis interfaces

REFERENCES