



# SRS WALKTHROUGH

THE GLOBAL TECH FACE-OFF

# AGENDA

- Introduction to TechWiz 3
- Project Categories
- Technologies for the Contest at a Quick Glance
- Software Requirements Specification - Website
- Software Requirements Specification - Web Application
- Software Requirements Specification – Mobile App
- Software Requirements Specification - IoT Solutions



# WHAT IS TechWiz 3?



**A Technology-based competition conducted globally across all Aptech centers.**



**Tests the software knowledge, proficiency, time & team management skills of students.**



**Gives students an opportunity to showcase their talent, Win Cash prizes and more.**



**Evaluation by Aptech Head Office and industry experts.**

# PROJECT CATEGORIES

Project categories include the development of apps/websites, which a team of 4 to 6 students will develop or create as per themes assigned by Aptech HO. The categories are as follows:

**Category 1**



**Website Design and Development**

**Category 2**



**Web Application Development**

**Category 3**



**Mobile App Development**

**Category 4**



**Internet of Things (IoT) Solutions Development**

# CATEGORY 1

## WEBSITE DESIGN & DEVELOPMENT

- With digital medium taking the world by storm, Website Design and Development are one of the most in-demand skills for I.T. students.
- Since users spend a vast segment of their time and attention on the internet, they expect flawless and user-friendly online experiences .
- In this category, students can use a combination of HTML5, CSS3, JavaScript, AngularJS, jQuery, Bootstrap, and so on to create their projects.



# CATEGORY 2

## WEB APPLICATION DEVELOPMENT



- The rise in web applications has led to a digital revolution hence Web Application Development is chosen as a competing category.
- A Web application is an application program stored on a remote server and delivered over the Internet through a browser interface. Web Application runs within the user's Web browser at the client site. This information is severed from the server.
- Students can use technologies like Servlets, JSP, Spring, ASP.NET, ASP MVC and Core, PHP, Laravel framework, and Python for their projects.



# CATEGORY 3

# MOBILE APP DEVELOPMENT

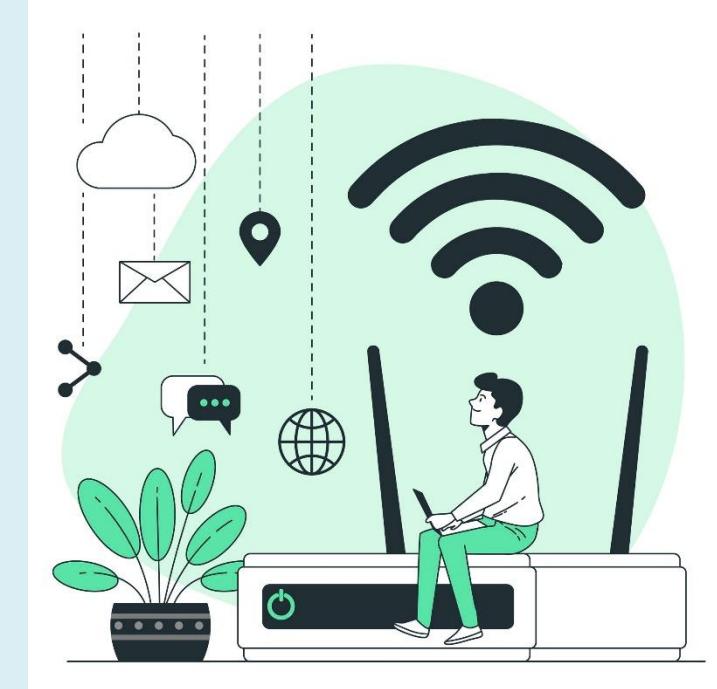
- Mobile Apps rule the world of digital assets. There are numerous mobile apps that are helping us lead a comfortable and connected life.
- A mobile app is a software application developed specifically for use on small, wireless computing devices, such as smartphones and tablets, rather than desktop or laptop computers
- Mobile apps are categorized according to whether they are web-based or native apps or hybrid apps, which are created specifically for a given platform.
- For TechWiz 3, students can use Android with SQL Lite or Firebase to build their project



## CATEGORY 4

# INTERNET OF THINGS (IoT) SOLUTIONS DEVELOPMENT

- Internet of Things technologies are taking over everyday domains like transportation, homes, appliances, health and lifestyle and more. There are numerous IoT solutions that are helping us lead a comfortable and connected life.
- An IoT app is a software application developed specifically for use on IoT devices such as Raspberry Pi.
- Students can use Python, OpenCV, and other technologies and libraries for their project.



# Software Requirements Specification

# TECHNOLOGIES AT A QUICK GLANCE

Category	Tool/Software
Web Design and Development	<p><b>Frontend:</b> HTML5, CSS 3, Bootstrap , JavaScript, jQuery, AngularJS, Angular 9 (optional), XML</p> <p><b>Data Store:</b> JSON files or TXT files</p>
Web Application Development	<p><b>Frontend:</b> HTML5, CSS 3, Bootstrap , JavaScript, jQuery, AngularJS, Angular 9 (optional), XML</p> <p><b>Backend:</b> Java 8 or higher, J2EE or Jakarta EE OR C# 7.2 or higher, with ASP.NET MVC, ASP.NET MVC Core OR PHP 7.2 or higher with Laravel 5 or higher OR Python 3 or higher with Flask</p> <p><b>Database:</b> MySQL 5.7 or higher/SQL Server 2016 or higher</p>

# TECHNOLOGIES AT A QUICK GLANCE

Category	Tool/Software
Mobile App Development	<b>Technology:</b> Android 9 or higher with Java <b>IDE:</b> Android Studio <b>Database:</b> SQLite or Firebase
IoT App Development	<b>Hardware and Software:</b> Raspberry Pi, Sensors, OpenCV, Python, and other libraries

# CATEGORY 1

## WEBSITE DESIGN & DEVELOPMENT

### SRS SAMPLE



*Software Requirements Specification  
Version 1.0*

**Fit4Life**

**Domain:** Fitness & Lifestyle

**Category:** Website Design and Development

## 1.0 Need for the Website

Modern fast-paced lifestyles have led to deteriorating health conditions. People are now contracting ailments far earlier and spending more on hospitals and institutionalized healthcare. Heart diseases, back aches, inflamed joints, diabetes, and so on are becoming too common and are being seen even in younger people.

To counter this, one should adopt a fitter and healthier lifestyle, that will have a good balance of nutrition, exercise, activity, and rest.

### 1.1 Proposed Solution

The proposed Website will be titled Fit4Life and should help individuals with fitness programs and diet plans.

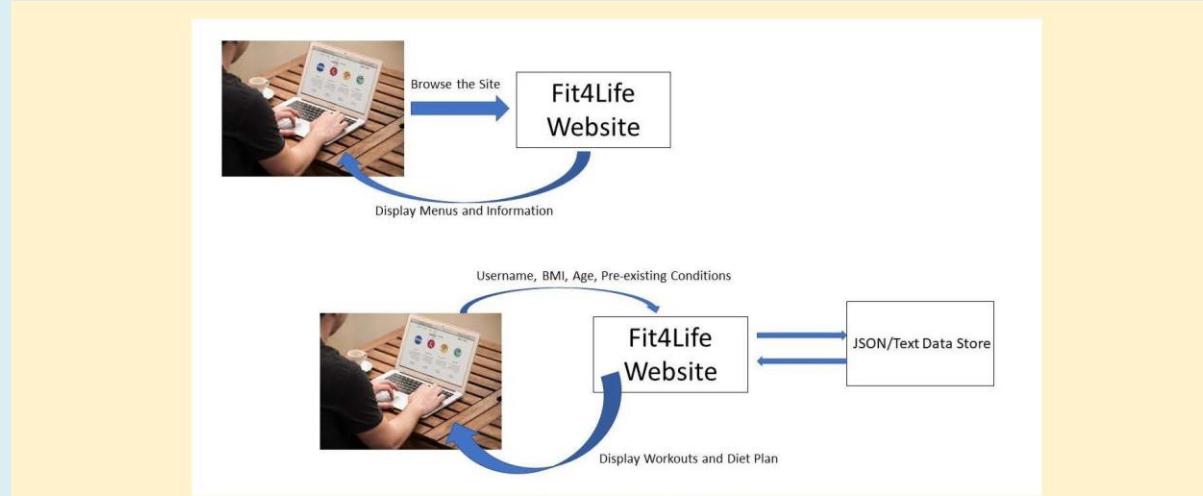
You have been given a contract to design and build this Website.

### 1.2 Purpose of this Document

The purpose of this document is to present a detailed description of the Fitness and Lifestyle Website, titled Fit4Life.

Fit4Life guides individuals on various fitness programs and helps them choose most suitable diet that will align with their fitness goals.

This document explains the purpose and features of the Website, the interfaces of the Website, what the Website will do, and the constraints under which it must operate. This document is intended for both stakeholders and developers of the Website and will be proposed to the client for approval.



*Broad View of the Portal*

### 1.3 Scope of Project

This Web portal will be a responsive and visually appealing Website to be used by individuals. This portal will be designed to provide health and fitness-based information such as diet plans and workout programs.

### 1.4 Constraints

The Web portal will not have any facility to store information on the server. Information can be fetched from pre-populated JSON or TXT files and displayed, however, information cannot be written to the files from within the portal.

### 1.5 Functional Requirements

The portal will be designed as a Single-Page-Application and responsive Website with a set of pages and menus that represent choice of activities to be performed. The pages, menus, and other visual elements must be designed in a visually appealing manner with attractive fonts, colors, and animations. All of these should also be laid out in a responsive manner.

Following are the functional requirements of the portal:

- i. **Welcome Message:** The home page should accept first name from the user and display a personalized welcome message. At the top corner, the user's first name should be displayed for the entire duration that the portal is loaded.
- ii. **Find a Diet Plan:** Based on given Body Mass Index (BMI), age, and pre-existing conditions, a suitable diet plan should be suggested to the user.
- iii. **Find Workouts:** Based on given Body Mass Index (BMI), age, and pre-existing conditions, suitable workouts in the form of images and videos should be suggested to the user.
- iv. **View Resources:** This menu option upon clicking should display book suggestions, other Website links, and video channels that the user can reference in order to improve their fitness. Information will be retrieved from a pre-populated JSON file and displayed. (Hint: Use AngularJS directives, filters, services, controllers, and other features to implement this).
- v. **About Us and Contact Us:** This menu option should display Email id, address, and contact number of the organization who is developing the system.

Over and above this, the portal should implement the following functionalities:

- Display a continuous scrolling ticker at the bottom of the page with current date, time, and location (hint: Use geolocation features of HTML5).
- Display a visitor count at the top right corner of the page beside a logo image.
- The menu options should change color on hover and also after clicking.
- Fade in and fade out options can be used for the menus.

## 1.6 Non-Functional Requirements

There are several non-functional requirements that should be fulfilled by the system.

The system should be:

Safe to use: The system should not result in any malicious downloads or unnecessary file downloads.

Accessible: The system should have clear and legible fonts, user-interface elements, and navigation elements.

User-friendly: The system should be easy to navigate with clear menus and other elements and easy to understand.

Operability: The system should operate in a reliably efficient manner.

Performance: The system should demonstrate high value of performance through speed and throughput. In simple terms, the system should be fast to load and page redirection should be smooth.

Capacity: The system should support large number of users.

Availability: The system should be available 24/7 with minimum down time.

Compatibility: The system should be compatible with latest browsers.

## 1.7 Interface Requirements

### 1.7.1 Hardware

---

Intel Core i3/i5 Processor or higher

8 GB RAM or above

Color SVGA

500 GB Hard Disk space

Mouse

Keyboard

### 1.7.2 Software

---

Technologies to be used:

1. Frontend: HTML5, CSS3, Bootstrap , JavaScript, jQuery, AngularJS, Angular9 (optional), XML
2. Data Store: JSON files or TXT files

## 1.8 Project Deliverables

You need to design and build the project and submit it along with a complete project report that includes:

- Problem Definition
- Design specifications
- Diagrams such as flowcharts for various activities, Data Flow Diagrams, and so on
- Source Code
- Test Data Used in the Project
- Project Installation Instructions (if any)

The consolidated project will be submitted as a zip file with a ReadMe.doc file listing assumptions (if any) made at your end and JSON/TXT files containing test data.

Over and above the given specifications, you can apply your creativity and logic to improve the portal.

~~~ End of Document ~~~

# CATEGORY 2

## WEB APPLICATION DEVELOPMENT

SRS SAMPLE

# Software Requirements Specification

Version 1.0

## FitnessBuddy

Domain: Fitness and Lifestyle

Category: Web Application Development



**1.0**

## Background and Need for the System

Modern fast-paced lifestyles have led to deteriorating health conditions. People are now contracting ailments far earlier and spending more on hospitals and institutionalized healthcare. Heart diseases, backaches, inflamed joints, diabetes, and so on are becoming too common and are being seen even in younger people.



To counter this, one should adopt a fitter and healthier lifestyle, that will have a good balance of nutrition, exercise, activity, and rest.

Statistics have shown that regular exercise can reduce your risk of major illnesses, such as heart disease, stroke, type 2 diabetes and cancer by up to 50% and lower your risk of early death by up to 30%. Hence, a Web application is needed that will help people to monitor and track their health.

**1.1**

## Proposed Solution



The proposed solution is a Web application geared to assist users in improving their health and fitness. The Web application will enable users to register themselves, login to the application, look for diet plans and workouts, and set and track their fitness regularly. Users can also browse resources to improve their fitness and diets. Users can submit their feedback regarding the application using a feedback form.

1.2

## Purpose of this Document

The purpose of this document is to present a detailed description of the Web application titled **FitnessBuddy**.

FitnessBuddy guides individuals on various fitness programs and helps them track progress of their fitness goals.

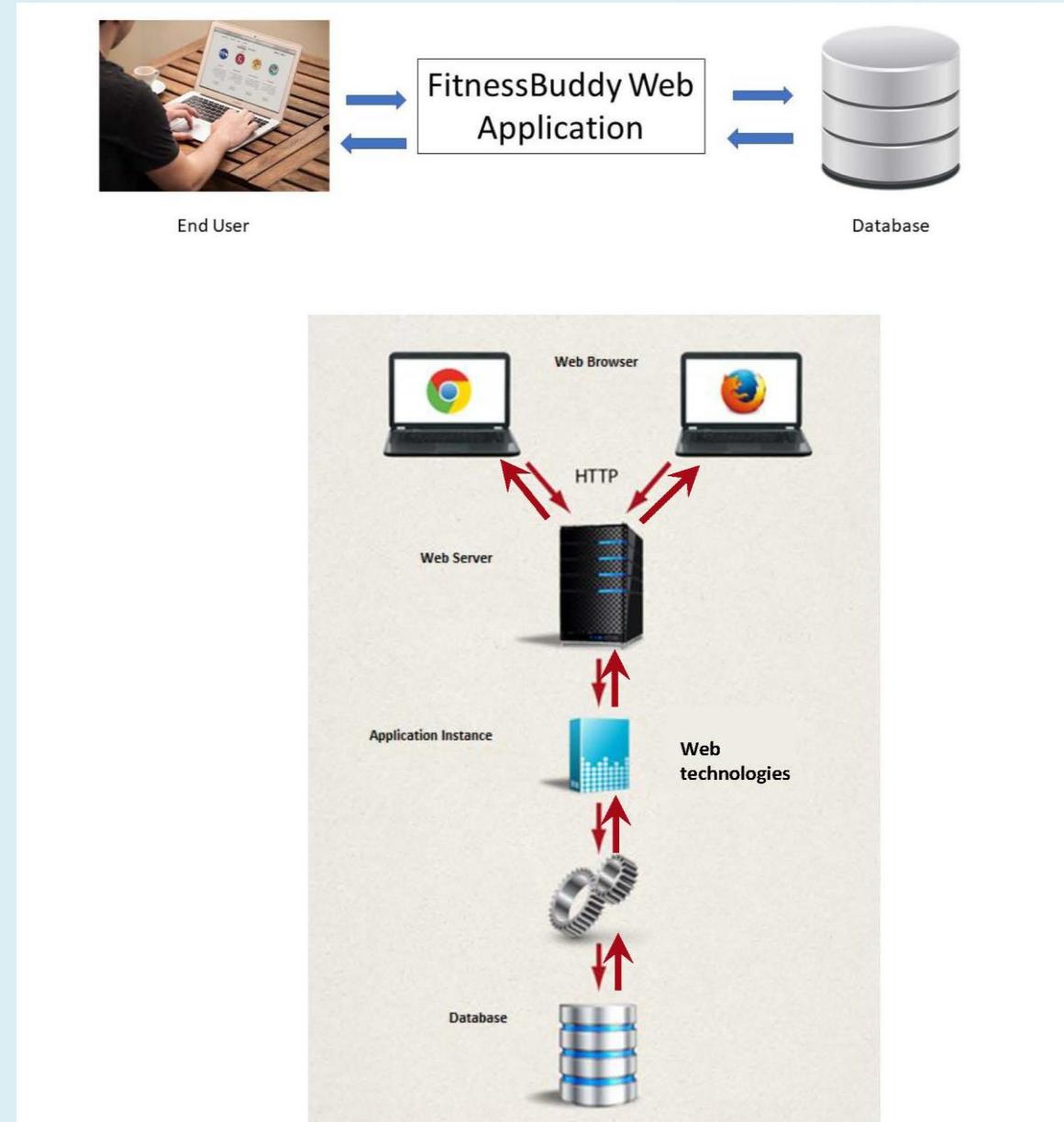
This document explains the purpose and features of the Web application, its interfaces, what the Web application will do, and the constraints under which it must operate. This document is intended for both stakeholders and developers of the Web application and will be proposed to the client for approval.



1.3

## Scope of the Project

This Web portal will be a responsive and visually appealing Web application to be used by individuals. This portal will be designed to provide health and fitness based information such as diet plans and workout programs and help track progress of their fitness goals.



## 1.4 Functional Requirements

Application will be designed with a set of forms/pages with menus representing choice of activities to be performed.

Following are the functional requirements of the system:

- **Home Page:** It will display general purpose information pages through menus such as Login, Registration, Diet Plans, Workout, About Us, Contact Us, Fitness Resources, and so on.
- **Register:** It will enable new users to register themselves with FitnessBuddy. At the time of registration, users need to provide Name, Email ID, Contact Number, and Username, and then, select their Password.  
  
Appropriate error-checking must be done on the fields of the form to ensure correct data. For example, email id can be checked to see if it is of appropriate format. (Hint: Use client-side validation).
- **Login:** It will allow successfully registered users to login to the FitnessBuddy application and access various features of the application through menus or other options such as Account Setting (Create, Update, and Delete Profile). A profile can include detailed information about the user, such as vital statistics (height and weight), age, pre-existing conditions, and so on.
- **Account Settings:** Users will be able to manage their accounts by using options such as Create, Update, and Delete Profile.

- **Find Diet Plan:** Using this option, users can search for suitable diet plan based on their profile details. For example, if a person has diabetes, a diet with less carbohydrates and sugar will be recommended. If a person is anaemic, an iron-rich diet will be recommended.
- **Find Workout Program:** Using this option, users will be able to search for appropriate workout exercises based on their body type and profile details. Workout programs may be displayed in the form of text instructions, images, and videos.
- **Set Fitness Goal:** Through this option, users can set their target weight/BMI and deadline date by which they hope to accomplish their fitness goal.
- **Track Fitness Goal:** This option will track the progress by comparing current date and current weight/BMI against the targeted values and update the user about the progress. Visual tools such as progress bars can be used if possible to denote the progress.
- **View Resources:** This option will display helpful resources for additional exercises and diets through book suggestions.
- **View Workout Videos:** This option will display helpful workout videos and YouTube channels, and so on.
- **Generate Reports:** FitnessBuddy will be able to generate and display different kinds of reports such as progress achieved so far, top ten popular diets, top ten popular workouts, and so on.
- **Submit Feedback:** Users can submit their feedback regarding the application using a feedback form.
- **Notifications:** Users can receive timely reminders and notifications from the application.
- **Calculate BMI:** Given the values of height and weight of the user, the application should calculate and display the BMI value.

**1.5**

## Non-Functional Requirements

There are several non-functional requirements that should be fulfilled by the system.

The system should be:



- **Safe to use:** The system should not result in any malicious downloads or unnecessary file downloads.
- **Accessible:** The system should have clear and legible fonts, user-interface elements, and navigation elements.
- **User-friendly:** The system should be easy to navigate with clear menus and other elements and easy to understand.
- **Operability:** The system should operate in a reliably efficient manner.
- **Performance:** The system should demonstrate high value of performance through speed and throughput. In simple terms, the system should be fast to load and page redirection should be smooth.
- **Security:** The system should implement adequate security measures such as authentication. For example, only registered users can access certain features.
- **Capacity:** The system should support large number of users.
- **Availability:** The system should be available 24/7 with minimum downtime.
- **Compatibility:** The system should be compatible with latest browsers.



## 1.6 Interface Requirements

### Hardware

- Intel Core i3/i5 Processor or higher
- 8 GB RAM or above
- Color SVGA
- 500 GB Hard Disk space
- Mouse
- Keyboard



### Software

**Frontend:** HTML5, CSS 3, Bootstrap, JavaScript, jQuery, AJAX, XML

**Client & Server:** Java 8 or higher, Java EE 7

OR

Visual Studio IDE, C# 7.2 or higher, ASP.NET MVC

OR

PHP 7.2 or higher version with Laravel Framework (optional)

OR

Python 3 or higher and Flask

**Database:** MySQL 5.7 or higher/ SQL Server 2016 or higher

### Database Design

**Data Dictionary:** User, UserProfile, PreExistingConditions, FitnessGoals, and so on

Based on the given specifications, you will define suitable entities, attributes for these entities, and identify relationships between the entities.

For example, some entities along with their attributes can be identified as follows:

| User:            |
|------------------|
| 1. User ID       |
| 2. Name          |
| 3. Email ID      |
| 4. Contact Phone |
| 5. Username      |
| 6. Password      |

| UserProfile:                          |
|---------------------------------------|
| 1. User ID                            |
| 2. Gender                             |
| 3. Height                             |
| 4. Weight                             |
| 5. Age                                |
| 6. BP                                 |
| 7. BMI (calculated field)             |
| 8. PreExisting Condition Status (Y/N) |

| PreExistingConditions: |
|------------------------|
| 1. User ID             |
| 2. Condition ID        |
| 3. Ailment Type        |
| 4. Description         |
| 5. Since Year          |
| 6. Medications (Y/N)   |

Similarly, you can define other entities and also relationships between entities and methods representing activities on the entities.

**Note:** These are just examples, you need not adhere to these structures and can design your own table structure with more or less columns.

## 1.7

# Project Deliverables

You will need to design and build the project and submit it along with a complete project report that includes:

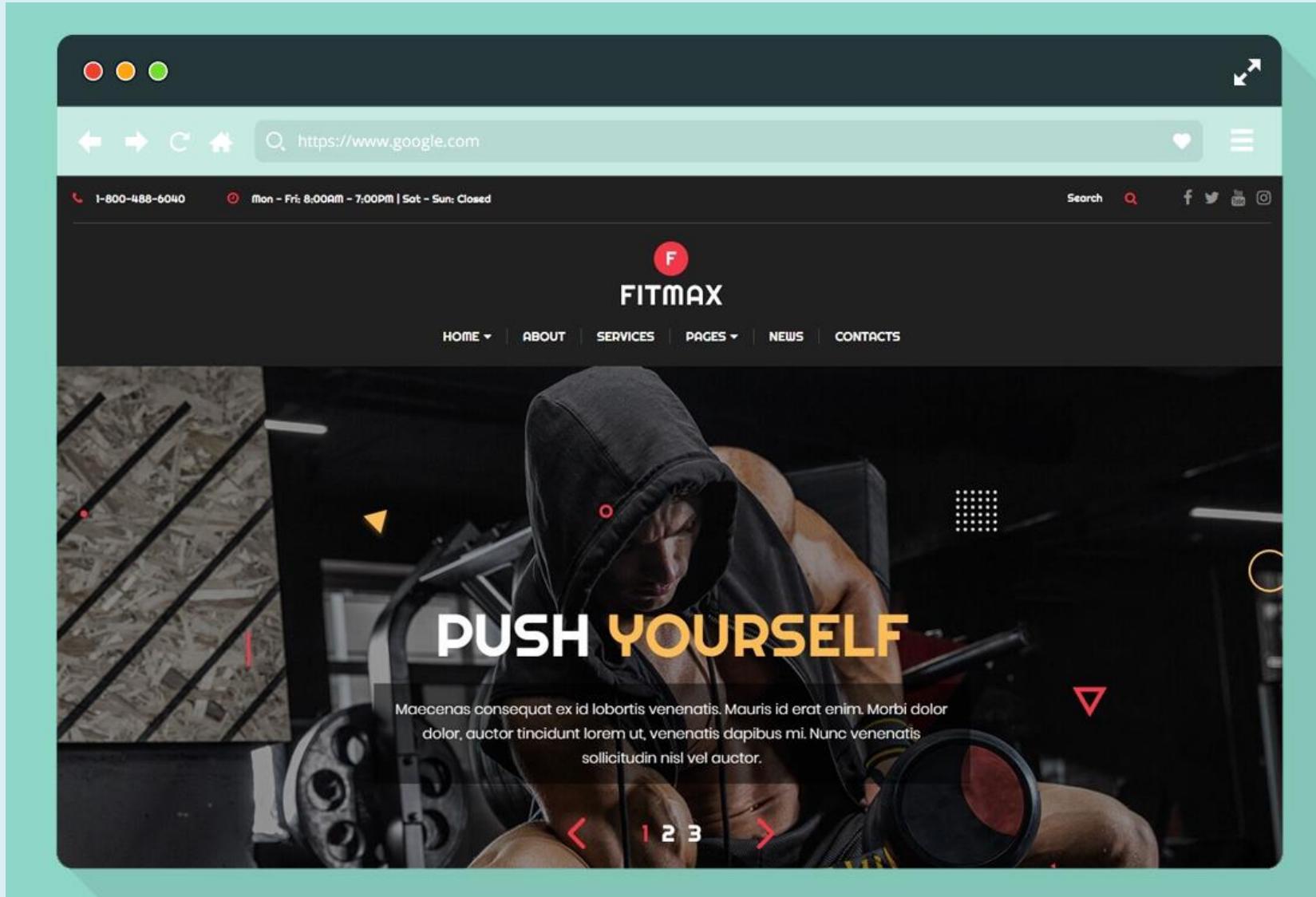
- Problem Definition
- Design specifications
- Diagrams such as flowcharts for various activities, Data Flow Diagrams, and so on
- Database Design
- Source Code
- Test Data Used in the Project
- Project Installation Instructions (if any)

The consolidated project will be submitted as a zip file with a ReadMe.doc file listing assumptions (if any) made at your end and SQL scripts files (.sql) containing database and table definitions.

Over and above the given specifications, you can apply your creativity and logic to improve the system.

**Sitemap:** To understand the flow of FitnessBuddy Web application, you will have to create a Sitemap and add it to the home page of your application.

# Example: Random Fitness Web Application from Internet



# CATEGORY 3

## MOBILE APP DEVELOPMENT

### SRS SAMPLE

# Software Requirements Specification

Version 1.0

## Healthify

Domain: Fitness and Lifestyle  
Category: Mobile Application



**1.0**

## Background and Need for the System



Modern fast-paced lifestyles have led to deteriorating health conditions. People are now contracting ailments far earlier and spending more on hospitals and institutionalized healthcare. Heart diseases, backaches, inflamed joints, diabetes, and so on are becoming too common and are being seen even in younger people.



To counter this, one should adopt a fitter and healthier lifestyle, that will have a good balance of nutrition, exercise, activity, and rest.



Statistics have shown that regular exercise can reduce your risk of major illnesses, such as heart disease, stroke, type 2 diabetes and cancer by up to 50% and lower your risk of early death by up to 30%.

The number of smartphone users worldwide today surpasses 3.8 billion and is forecast to further grow by several hundred million in the next few years. Total number of Android smartphones is estimated to be three billion worldwide.

Considering these facts, a mobile application that will help people to monitor and track their health and fitness would be a much welcome solution.

**1.1**

## Proposed Solution



The proposed solution is a mobile application/app geared to assist users in improving their health and fitness. The mobile app will enable users to register themselves, login to the application, look for diet plans and workouts, and set and track their fitness goals regularly. Users can also browse resources to improve their fitness and diets. Users can submit their feedback regarding the application using a feedback form. Notifications will be sent to the users' phones regarding reminders, goal progress, and so on.



**1.2**

## Purpose of this Document

The purpose of this document is to present a detailed description of the mobile application titled **Healthify**.

**Healthify** guides individuals on various fitness programs and helps them track progress of their fitness goals.

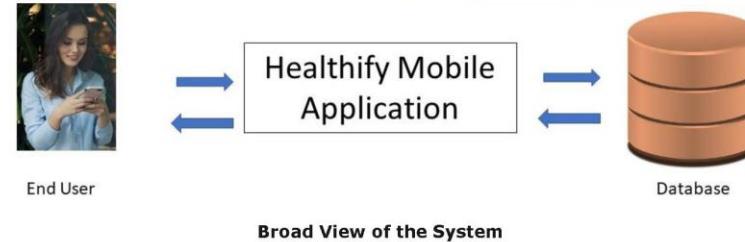
This document explains the purpose and features of the mobile app, its interfaces, what the app will do, and the constraints under which it must operate. This document is intended for both stakeholders and developers of the app and will be proposed to the client for approval.

**1.3**

## Scope of the Project

This mobile app will be a responsive and visually appealing application to be used by individuals. This portal will be designed to provide health and fitness based information such as diet plans and workout programs and help track progress of their fitness goals. Constraints for the project is that it must be a free app since implementing payment gateways is beyond scope of the project.





**Broad View of the System**

1.4

## Functional Requirements

Application will be designed with a set of forms/screens, Navigation, and Fragments with menus representing choice of activities to be performed.

Following are the functional requirements of the system:

- **Home screen:** It will display general purpose information screens through menus such as Login, Registration, Diet Plans, Workout, About Us, Contact Us, Fitness Resources, and so on.
- **Register:** It will enable new users to register themselves with Healthify. At the time of registration, users need to provide Name, Email ID, Contact Number, and Username, and then, select their Password.

Appropriate error-checking must be done on the fields of the form to ensure correct data. For example, email id can be checked to see if it is of appropriate format. (Hint: Use client-side validation).

- **Login:** It will allow successfully registered users to login to the Healthify application and access various features of the application through menus or other options such as Account Setting (Create, Update, and Delete Profile). A profile can include detailed information about the user, such as vital statistics (height and weight), age, pre-existing conditions, and so on.





- **Account Settings:** Users will be able to manage their accounts by using options such as Create, Update, and Delete Profile.
- **Find Diet Plan:** Using this option, users can search for suitable diet plan based on their profile details. For example, if a person has diabetes, a diet with less carbohydrates and sugar will be recommended. If a person is anaemic, an iron-rich diet will be recommended.
-   
**Find Workout Program:** Using this option, users will be able to search for appropriate workout exercises based on their body type and profile details. Workout programs may be displayed in the form of text instructions, images, and videos.
- **Set Fitness Goal:** Through this option, users can set their target weight/BMI and deadline date by which they hope to accomplish their fitness goal.
- **Track Fitness Goal:** This option will track the progress by comparing current date and current weight/BMI against the targeted values and update the user about the progress. Visual tools such as progress bars can be used if possible to denote the progress.
- **View Resources:** This option will display helpful resources for additional exercises and diets through book suggestions, YouTube channels, and so on.



- **Generate Reports:** Healthify will be able to generate and display different kinds of reports such as progress achieved so far, top ten popular diets, top ten popular workouts, and so on.
- **Submit Feedback:** Users can submit their feedback regarding the application using a feedback form.
- **Calculate BMI:** Given the values of height and weight of the user, the app should calculate and display the BMI value.
- **Notification:** These are notifications regarding reminders for diet, workouts, progress of fitness goals and so on.

For example, sample notifications can look like this:

- ❖ "Congratulations, you have lost 2 pounds this week!"
- ❖ "Reminder: You must increase your protein level today."
- ❖ "Reminder: You have a yoga workout scheduled for 3pm."



## 1.5

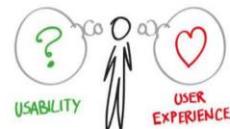
## Non-Functional Requirements

There are several non-functional requirements that should be fulfilled by the system.

The system should be:



- **Safe to use:** The system should not result in any malicious downloads or unnecessary file downloads.
- **Accessible:** The system should have clear and legible fonts, user-interface elements, and navigation elements.
- **User-friendly:** The system should be easy to navigate with clear menus and other elements and easy to understand.
- **Operability:** The system should operate in a reliably efficient manner.
- **Performance:** The system should demonstrate high value of performance through speed and throughput. In simple terms, the system should be fast to load and screen redirection should be smooth.
- **Security:** The system should implement adequate security measures such as authentication. For example, only registered users can access certain features.
- **Capacity:** The system should support large number of users.
- **Availability:** The system should be available 24/7 with minimum downtime.
- **Compatibility:** The system should be compatible with latest browsers.



**1.6**

## Interface Requirements

### Hardware

- Intel Core i3/i5 Processor or higher
- 8 GB RAM or above
- Color SVGA
- 500 GB Hard Disk space
- Mouse
- Keyboard
- Android Supported Smart Phone

### Software

**Programming Software & IDE:**

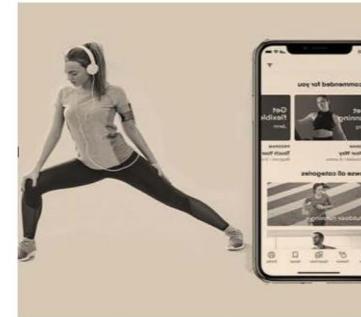
Android Studio IDE with Android 9 or higher  
with Java

**Database:** SQL Lite

### Database Design

**Data Dictionary:** : User, UserProfile, PreExistingConditions,  
FitnessGoals, and so on

Based on the given specifications, you will define suitable entities,  
attributes for these entities, and identify relationships between the  
entities.



For example, some entities along with their attributes can be identified as follows:

| User:            |
|------------------|
| 1. User ID       |
| 2. Name          |
| 3. Email ID      |
| 4. Contact Phone |
| 5. Username      |
| 6. Password      |

| UserProfile:                          |
|---------------------------------------|
| 1. User ID                            |
| 2. Gender                             |
| 3. Height                             |
| 4. Weight                             |
| 5. Age                                |
| 6. BP                                 |
| 7. BMI (calculated field)             |
| 8. PreExisting Condition Status (Y/N) |

| PreExisting Conditions: |
|-------------------------|
| 1. User ID              |
| 2. Condition ID         |
| 3. Ailment Type         |
| 4. Description          |
| 5. Since Year           |
| 6. Medications (Y/N)    |

Similarly, you can define other entities and also relationships between entities and methods representing activities on the entities.

**Note:** These are just examples, you need not adhere to these structures and can design your own table structure with more or less columns.

## 1.7

# Project Deliverables

You will need to design and build the project and submit it along with a complete project report that includes:

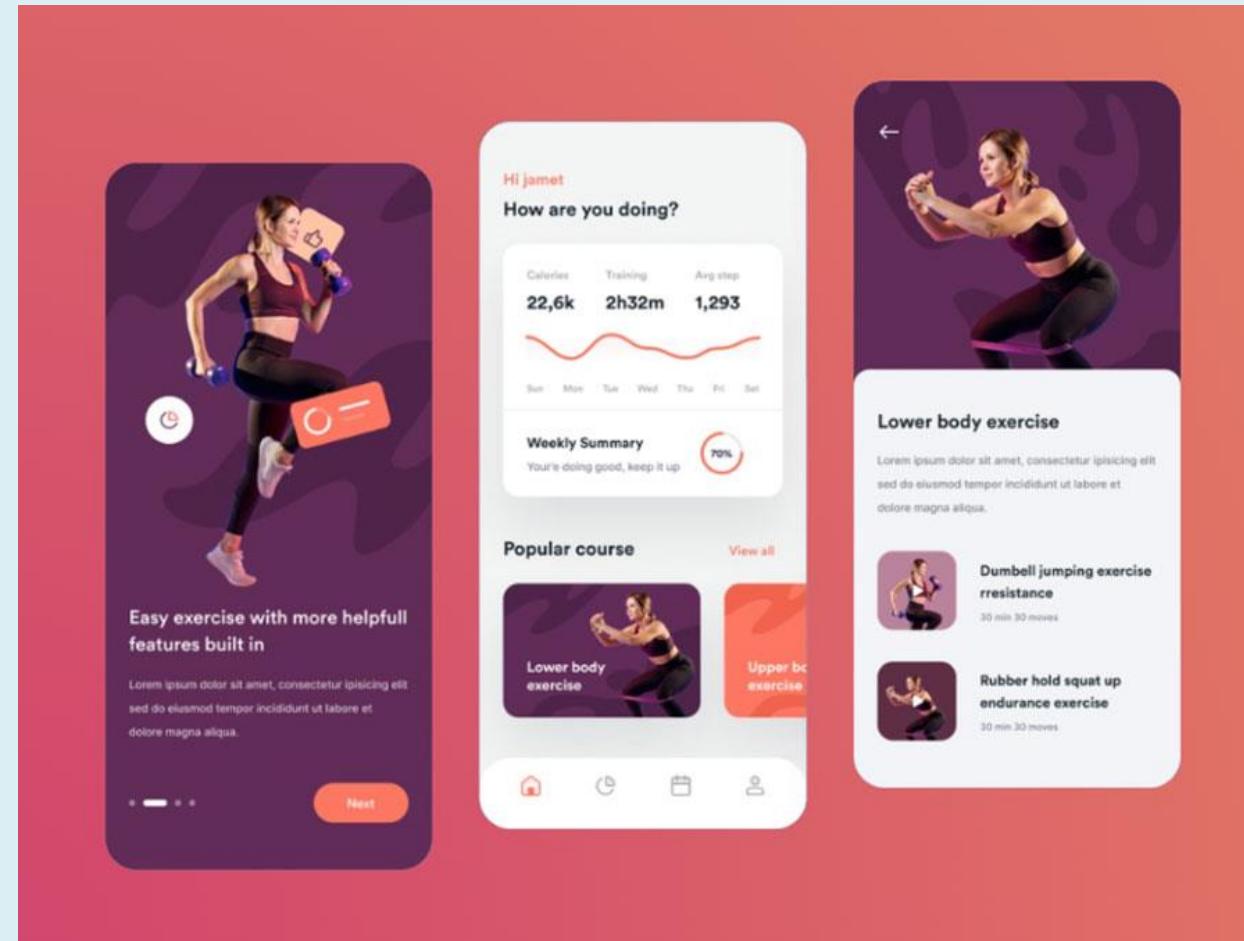
- Problem Definition
- Design specifications
- Diagrams such as flowcharts for various activities, Data Flow Diagrams, and so on
- Database Design
- Source Code
- Test Data Used in the Project
- Project Installation Instructions (if any)

The consolidated project will be submitted as a zip file with a ReadMe.doc file listing assumptions (if any) made at your end and SQL scripts files (.sql) containing database and table definitions. The ReadMe.doc may also list any additional software libraries or SDKs that have been used to build the application.

Over and above the given specifications, you can apply your creativity and logic to improve the system.

**Help:** To understand the flow and usage of Healthify Mobile Application, you will have to provide complete guideline document.

## Example: Random Fitness Mobile Application from Internet



# CATEGORY 4

## IoT SOLUTIONS

### SRS SAMPLE



*Software Requirements Specification  
Version 1.0*

Air Quality  
Tester

Category: IoT Solution/ Application



## 1.1 Background and Need for the System

Over past 20 years, pollution has become a worldwide public health issue and is counted amongst the most serious environmental hazards. The present measures are not able to tackle the rising levels of pollution effectively. Therefore, in order to completely monitor and keep a check on air pollution, we need to design a complete solution comprising a device and monitoring system to monitor air quality throughout the surrounding area.

You, being a lead-embedded Engineer at a Government Agency, have been assigned the task of designing and developing a pilot air quality monitoring system. The pilot monitoring system will facilitate in the monitoring of harmful gases present in the air around a small area of five kilometers diameter. Once the pilot is successful, in future, the complete system can be implemented in other states and districts.

The pilot system will have to do only three tasks:

- Monitor the amount of harmful gases present in the air
- Produce beep sound and send notification on the server if the amount of harmful gases exceeds the normal level
- Turn ON the air purifiers automatically when levels of harmful gases are above normal levels

## 1.2 Proposed Solution

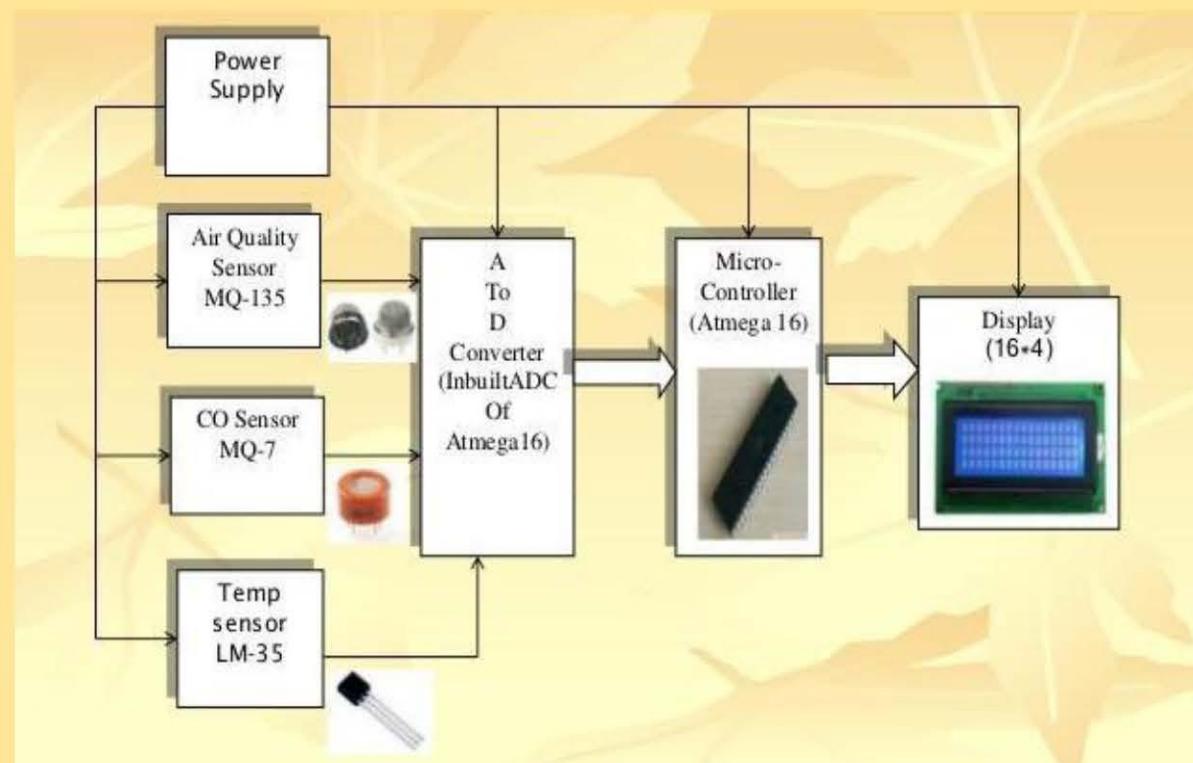
Assume that the Government Agency for which you are working wants you to design such a hardware and software which is robust. That is, you must use hardware such as PIC, ARM, RASPBERRY PI, and so on, and software such as Python and OpenCV for programming.

Always keep in mind that the dashboard you will be creating for this project must be vulnerable to cyber-attacks, which could be done using MongoDB as it is already secured by default. The data you access on MongoDB is already secured and encrypted end to end. In addition, the hardware you are going to design must be so tough that it cannot be broken easily as it is going to be deployed at public places. Therefore, you must use hard acrylic or metal for outer covering of circuitry. Also, check the sensors fixing, as it is in more reach to the public. So, keep the sensors in such a box that it cannot be accessed from outside.

The system is to be developed using Raspberry Pi and Python libraries.

Following functionalities are required to be integrated into the system:

- Display in the device at the installation site for public viewing
- Dashboard for real-time data analysis
- Automated Solution



*Sample Architecture of the Application*



## 1.3 Purpose of the Document

The purpose of this document is to present a detailed description of the IOT-based system interface with Raspberry Pi titled 'Air Quality Tester'.

This document explains the purpose and features of the IOT-based system and the constraints under which it must operate. This document is intended for both stakeholders and developers of the system.

## 1.4 Scope of Project

Microcontroller or System on Chip (SOCs) play a vital role in sending data to the server and storing the data inside the device. Apart from that, we can also use SD cards for further addition of memory blocks.

In higher-level operations, we need certain database such as SQLite inside our SOCs such as Raspberry Pi; however, it is not mandatory on the board. We should have a database on the server front, which suffices our need, but a database on device is useful for validation and security.

## 1.5 Functional Requirements

The pilot system will have to do only three tasks:

- Monitor the amount of harmful gases present in the air
- Produce beep sound and send notification on the server if the amount of harmful gases exceeds the normal level
- Turn ON the air purifiers automatically when levels of harmful gases are above normal levels

Only one person, the administrator, is required to operate the system as it is based on IoT.

### **Administrator**

An Administrator is a person who will be getting all the data feeds in real-time on the server.

The Administrator can perform the following operations:

- View location-wise extent of pollution
- Perform required actions during an emergency
- Track the functioning of devices



In the IoT-based Air Quality Monitoring System, inputs would be data received by sensors, working of esp8266 (Wi-Fi module), displaying the data on LCD, and so on.

Example of outputs would include data collected by sensors, sending data to the server in real-time, sending notifications to concerned departments, and so on.

Overview of processes involved in the system:

- Installation of the device
- Connecting it to the server
- Sending data to the server
- Sending notifications to the concerned departments
- Turning ON and OFF the air purifiers

## 1.6 Non-Functional Requirements

There are several non-functional requirements that should be fulfilled by the system.

The system should be:

**Flexibility:** The system should be flexible to add more sensors or allowing mobile number/ email id to send alert notifications to users.

**Usability:** The system should have user-friendly interface to deliver a clear messages and operate in a reliably efficient manner.

**Performance:** The system response time depends on the type of sensors used. The system should give accurate results with the use of sophisticated sensors.

**Capacity:** The system should support a database to validate a large number of data.

## 1.7 Interface Requirements

Hardware and Software  
Raspberry Pi 3 and Python 3 or higher versions.

Over and above these, you may use other hardware and software components as necessary.



## 1.8 Project Deliverables

You will design and build the project and submit it along with a complete project report that includes:

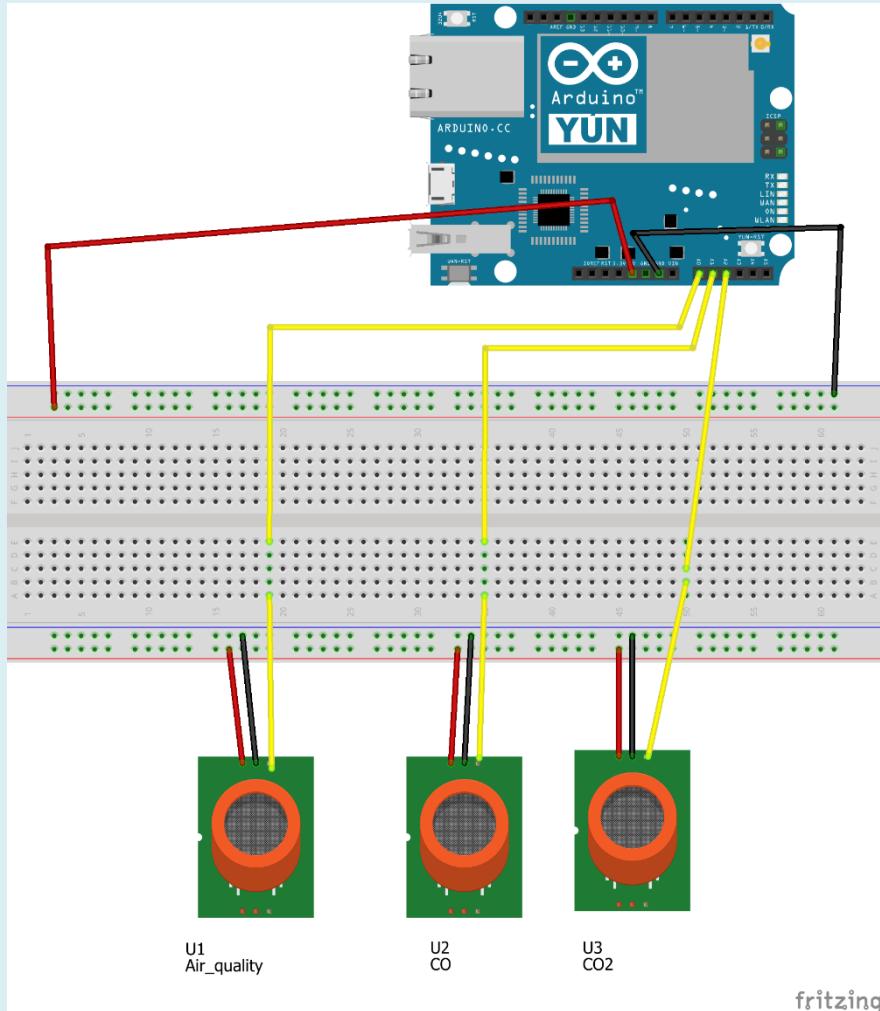
- Problem Definition
- Design specifications
- Diagrams such as flowcharts for various activities, Data Flow Diagrams, and so on
- Source Code
- Test Data Used in the Project
- Project Installation Instructions
- Database Design

The consolidated project must be submitted as a zip file with a ReadMe.doc file listing assumptions (if any) made at your end, details of the Pin configuration of Raspberry Pi, sensors used, other components used, and technical specifications of Raspberry Pi. Documentation is a very important of the project hence, all crucial aspects of the project must be documented properly.

In addition, you must submit a video clip showing the actual working of the system.

Over and above the given specifications, you can apply your creativity and logic to improve the system.

## Examples From Internet for IoT Air Quality System



# CENTER'S RESPONSIBILITY



- On the beginning day of the competition day (10<sup>th</sup> Aug) the first thing, a center must do is to give a detailed explanation to each batch regarding the SRS.
- Instruct students to take time to perform a detailed analysis of project requirements. This is a key factor in the success of every project. If analysis is properly in place, your project will be a success.
- Daily, two meetings to be conducted in the morning with each team by mentors. Purpose of this will be to monitor overall progress of the project. The same evening, mentors should conduct day-end review to check the project and identify gaps (if any). Measures should be taken to fill the gaps (if any).

# FUN ACTIVITIES



- Have fun activities during the competition, record the moments
- Ice Breaker Games
- Talent Shows
- Share the videos with HO



# THANK YOU