

Table of Contents

Table of Contents	1
Introduction	2
1.1 Overview	2
1.2 Business Context	2
1.3 Glossary	2
2. General Description	3
2.1 Product / System Functions	3
2.2 User Characteristics & Objectives	3
2.3 Operational Scenarios	4
2.4 Constraints	4
3. Functional Requirements	6
3.1 User Login	6
3.2 User Registration	6
3.3 Notification Creation	6
3.4 Notification Edit	7
3.5 Notification Deletion	7
3.6 Data Export	7
3.7 Prediction	7
3.8 Device Connection	8
3.9 Device Disconnection	8
4. System Architecture	9
Figure 4.1	9
5. High Level Design	10
Figure 5.1	10
6. Preliminary Schedule	11

Introduction

1.1 Overview

Our idea is to provide users with the ability to monitor indoor air quality information through the use of a Raspberry Pi and associated sensors.

It will allow public users to see data on multiple metrics while allowing administrators to automatically monitor data and send notifications once a metric has passed a threshold. Ideally it will also include data visualisation through graphing and predictive analytics through the processing of the data provided by the sensors.

1.2 Business Context

There are a few ways that this can be implemented for a profit. An initial subscription based service can be charged in order to access the systems, it can be marketed as an environmentally/sustainable product as it can be used to monitor the temperature and humidity of each room in order to know where and when to turn off heating in buildings with hundreds of rooms.

A constraint can also be put on the cooldown between notifications so users may pay to increase the amount, same can be applied to the amount of devices that can make up a network.

1.3 Glossary

- HTML5: Abbreviation for Hypertext Markup Language Revision 5, this is used to create the base website for the application.
- CSS: Cascading Style Sheets, used for styling the base website
- Django: Open Source python based web framework that includes templating.
- React: Open Source frontend Javascript library used to build user interfaces.
- SQLite: Database engine based on the C language.
- Raspberry Pi: Small single board computers popular for its modularity and performance relative to its size.

2. General Description

2.1 Product / System Functions

The preliminary list for necessary functions for this Air Quality Monitor are, but not limited to:

- Login - Login to User Account
- Register - Allows users to create an account
- Create Notification- Allows the creation of notifications.
- Edit Notification- Allows the user to to edit the notification parameters.
- Delete Notification- Removes the selected notification.
- Export Data - Exports the selected historic data.
- Prediction - Allows the user to see what the system predicts will be the next measurement within the next hour.
- Connect Device - Allows the user to connect another device to provide data once verified.
- Disconnect Device - Allows administrators to delete devices from the network.

2.2 User Characteristics & Objectives

The system will be hosted online, therefore the general target audience would be anyone with an internet accessible device. The website will be user friendly in order to allow for access for people of various levels of digital literacy.

Each user will however have different levels of access to the system.

- Guest: These are unregistered users or users that have not signed into the website. They are limited to previewing publicly available data set by the administrator.
- Registered Users: This group are users that have an account with the website, they are allowed to view the publicly available data set by the administrator however with the added permissions to add notifications, see the predictive analysis results.
- Administrators: The maintainers of the website, with elevated permissions, have access to all the previous permissions however with the bonus of being able to connect and remove devices from the network and hide certain devices from general users.

2.3 Operational Scenarios

Scenario 1: User Registration

User opens up the website, clicks onto the register button and inputs a valid username, email and password. A verification number is sent to the email address which the user inputs back into the website and an account is created.

Scenario 2: Notification creation

User logs into the website, navigates to the notification section or the specific sensor node's page. They then press notification creation and the system asks the user which parameter they would like it to monitor from a drop down menu and what parameter should be exceeded for it to send a notification. Users are then asked if they'd like an email notification or pop up notification on the OS they are using.

Scenario 3: Device Addition

Administrator logs into the website and is greeted by the control panel page. Presses the add device button and is then prompted to start up the Raspberry Pi and run the program on it and follow the instructions displayed. It will prompt the user to enter the address the server is on and a brief handshake will occur to check if the Raspberry Pi is compatible. A verification code will be displayed and shown to the user to input on the website in order to link the device.

2.4 Constraints

Time Constraints

In this project, we will be using hardware we haven't used over the course of the programme, so we'd need some time to research and write the code for the sensors to return data.

Time constraints will also apply to the calibration of the sensors as the positioning may cause inaccurate readouts.

Financial Constraints

Due to the nature of the system being dependent on hardware, in order to actually fully test if the system is scalable, investments would be required in order to source the main board and it's associated sensors.

Network Constraint

The core part of this project will be its network, due to the fact that the raspberry pi requires a network in order to send its data to the server we're implementing, slow or unstable internet will be a problem. It is also unsure as of this moment whether or not network security will be an issue on the DCU network.

3. Functional Requirements

3.1 User Login

Name	Login
Description	Allows users with existing username and password to log in.
Criticality	Highly important- (users will not be able to create notifications without logging in.)
Technical Issues	User information must be secure, so the database needs to be protected.
Dependencies with other Requirements	User Registration

3.2 User Registration

Name	Register
Description	Allows the users to sign up to the website.
Criticality	Highly important - (users need an account to access more parts of the site.)
Technical Issues	User information must be secure, so the database needs to be protected.
Dependencies with other Requirements	N/A

3.3 Notification Creation

Name	Create Notification
Description	Allows the user to set notifications to be sent to the OS or email.
Criticality	Critical- (one of the main parts of the application that helps with the sustainability aspect.)
Technical Issues	N/A
Dependencies with other Requirements	N/A

3.4 Notification Edit

Name	Edit Notification
Description	Allows users to edit the parameters for their notification such as email or data to watch.
Criticality	Important - (Allows the user to edit, not critical however it will be inconvenient if not implemented.)
Technical Issues	N/A
Dependencies with other Requirements	Create Notification

3.5 Notification Deletion

Name	Delete Notification
Description	Allows users to remove existing notifications they've created.
Criticality	Critical - (Users must have the ability to remove notifications or else they will be spammed.)
Technical Issues	N/A
Dependencies with other Requirements	Create Notification

3.6 Data Export

Name	Export Data
Description	Exports the data selected from selected node(s) in various formats if requested.
Criticality	Important - (Not critical however will be a quality of life feature.)
Technical Issues	N/A
Dependencies with other Requirements	N/A

3.7 Prediction

Name	Prediction
------	------------

Description	Allows the user to select the node to predict the predicted values for the next hour.
Criticality	Vital- (Another important part of the system, this allows for data processing to occur.)
Technical Issues	N/A
Dependencies with other Requirements	N/A

3.8 Device Connection

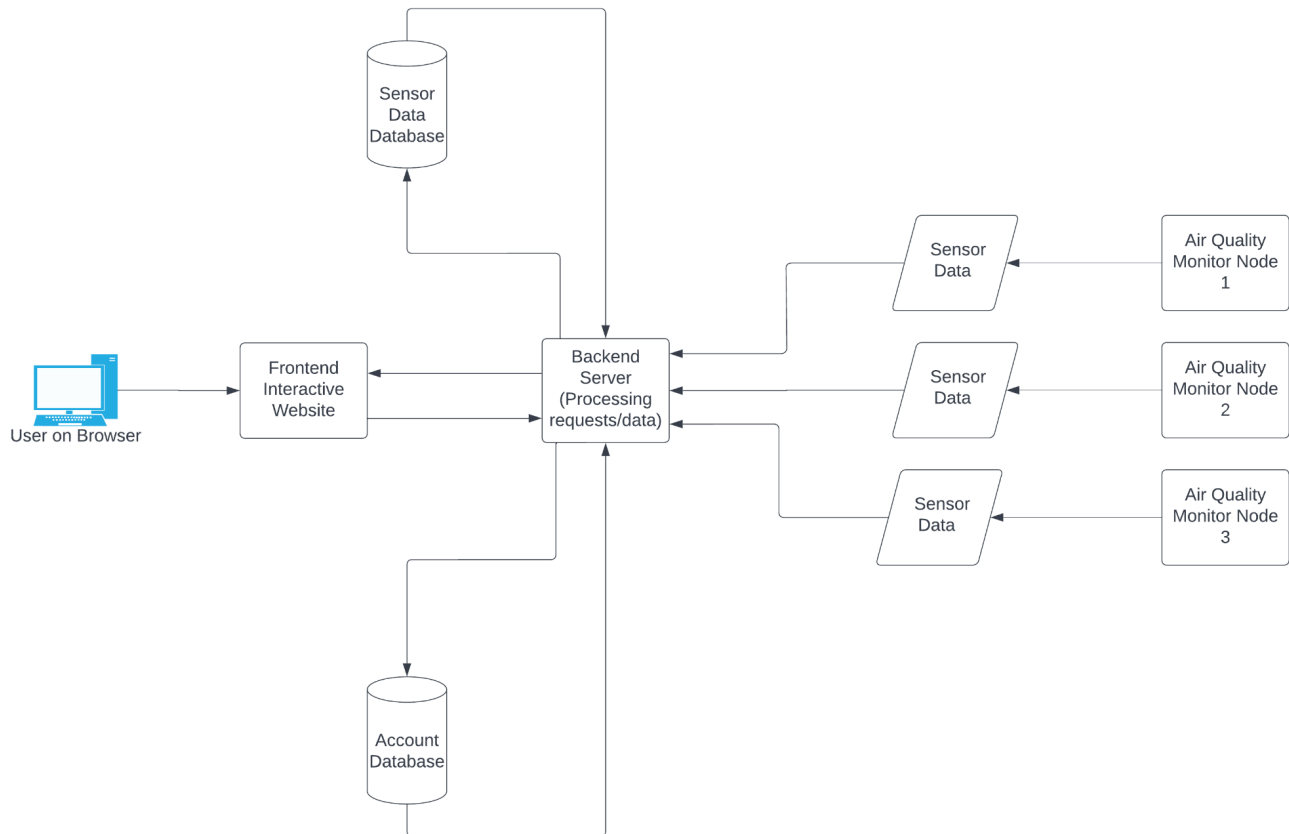
Name	Connect Device
Description	Allows administrators to connect another device to act as a node for the air quality monitoring network.
Criticality	Vital - (Administrators need the ability to add extra devices in order to increase the scalability.)
Technical Issues	N/A
Dependencies with other Requirements	N/A

3.9 Device Disconnection

Name	Disconnect Device
Description	Allows administrators to disconnect another device from the network in case of faults or migrations.
Criticality	Vital - (A vital function which gives the administrator the ability to remove devices without affecting others.)
Technical Issues	N/A
Dependencies with other Requirements	Connect Device

4. System Architecture

Figure 4.1



The figure above illustrates our system architecture.

The user connects via browser into the website.

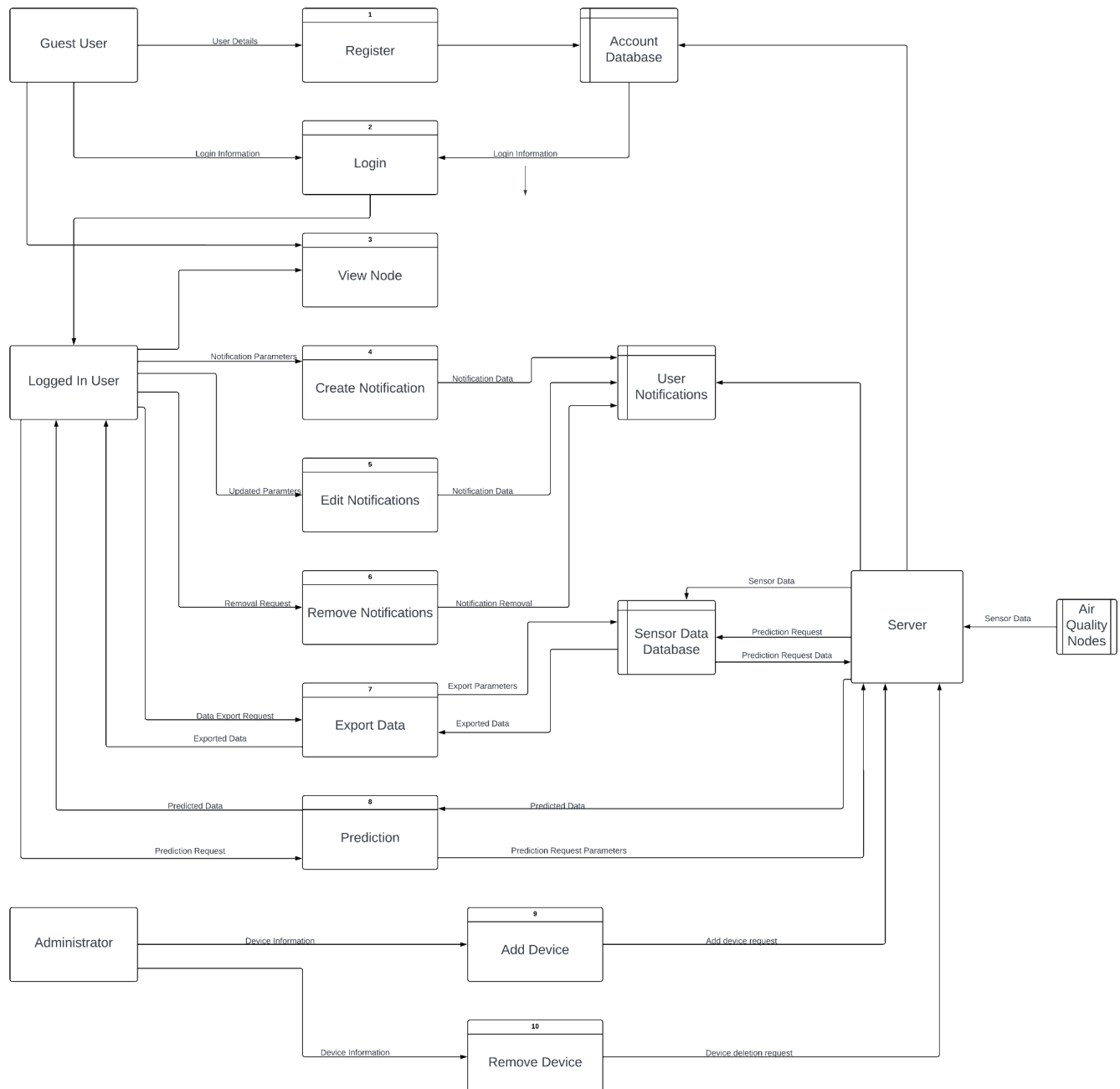
The frontend will be built using HTML, CSS and ReactJS. It will receive information and data from the backend server and it will be used to display the information that the user requested or update the pages that users have opened.

Two databases are included in architecture to manage the accounts created and the sensor data collected.

Multiple Air Quality nodes consisting of sensors and a raspberry pi's will report sensor data back to the backend to be stored in the database, however due to **financial constraints**, only 1 will be shown in the implementation and demonstration.

5. High Level Design

Figure 5.1



In this figure above, it shows how the program would flow through a Data Flow Diagram (DFD), it shows how the user and systems interact with each other and how data flows through the program through different parts of the system to another.

6. Preliminary Schedule

[illegible]