

MILESTONE 2

We are using Trello to plan and track the progress of our project.

Use Case

Name – A clear verb/noun or actor/verb/noun descriptor that communicates the scope of the use case.

1. All-in-one temperature, brightness, humidity, altitude data collection device
2. Application Display

Brief Description – A brief paragraph of text describing the scope of the use case.

1. The Arduino device can be placed anywhere to collect data on the specific conditions of where it is placed.
2. The application receives the data from the device and displays it to whoever wishes to monitor those conditions remotely.

Actors – A list of the types of users who can engage in the activities described in the use case. Actor names should not correspond to job titles.

1. Actors: Any users that want to know the conditions at a specific place and time e.g. scientists monitoring lab conditions.
2. Actors: Any users that want to know the conditions at a specific place and time while away from that location (remotely).

Preconditions – Anything the solution can assume to be true when the use case begins.

1. Circuits are set up correctly with Arduino to collect and process data accurately.
2. Accurate data has been collected and displays effectively on the user's remote device through the database.

Basic Flow – The set of steps the actors take to accomplish the goal of the use case. A clear description of what the system does in response to each user action.

1. The device should be placed where conditions are wished to be monitored by the user.
2. Once the user will clicks on the application a screen will come up displaying the time, date, temperature (in either fahrenheit or celsius), humidity, sunlight intensity, wind speeds, altitude, etc.

Exception Flows – The things that can happen that prevent the user from achieving their goal, such as providing an incorrect username and password.

1. Failure of Arduino, database, or android syncage.
2. Failure of application launch.

Postconditions – Anything that must be true when the use case is

1. Circuits are set up correctly with Arduino to collect and process data accurately.
2. The Application opens and displays temperature, brightness, humidity, altitude data collection device

Functional Requirement:

Design an interface that is appealing and usable
Deliver understandable information.

Non-functional Requirement:

Collect accurate weather data from our Arduino sensors that feeds accurately to our database
Build stable connection between Arduino, database, and application.

Project Management Tool:

Trello(Team name: "JARRZ" SOFTWARE DEV)

Project Plan:

We will have 3 sprints for both of the groups working on the app and the Arduino. The first sprint is one and half week long(Due Nov 6). The second sprint is three weeks long(Due Nov 27). Third sprint will be two weeks long(Due Dec 4). We will be working on setting up the Arduino, user interface of our app and our database during our first sprint. During our second sprint, we will focus on feeding data from our sensors to the app and the database. The third, and the final sprint will mostly be polishing of our product such as testing, maintenance, debugging and making sure the sensors work and feed data properly.

Scrum Discussion:

During the meeting, we found out the potential problem of our original idea about the weather application. Originally, we planned to use the data collected from Arduino and notify the user when the weather will change and what to wear. The biggest problem of this idea is that Arduino cannot predict the future weather condition. Instead of making a weather app, we are going to create app for Arduino device that can collect data of temp, humidity, wind, etc. User can use our device and app anywhere at anytime.