SCC 230

Design Report

Group 5

Aditya Abhijit Khan, Blade Eastham, David Mellon, Kian Tomkins, Samuel Chee Lok To

Related stuff

Amazon Routines:

* Can be scheduled for times/events/user inputs
* Can output fixed events/current time
* Created from fixed template and drop down menus

Limitations

* Can only wait for 5 sec increments
* Cannot use multiple input types for the same routine
* Cannot compare data(no if commands)
* Cannot loop
* Accounts are per household(problems with many people having the password)
* Requires a centralised server(Amazon)
* Too many IOT can lead to slower connections for main devices

BBC Microbit:

* Can be used as a microcontroller
* Can output variable results based on inputs
* Can take built in inputs from:
  + Built in compass
  + Built in gravity sensor
  + pins
* Cannot take input from:
  + temperature sensor
  + Microphone
* Built in outputs:
  + pins(can output pulses/high or low voltage)
  + 5x5 display
  + Speaker
  + Radio signals(simple)
* Not built in outputs:
  + Motor
* 256 Kb flash memory
* 16Kb RAM
* Can be written in JS, Python or drag+drop

Scratch was here.

Intention

Our system intends to allow users to automate interactions between Interconnected (IoT) Devices at a high level. This should help remove the technical aspect from developing networked systems. Taking inspiration from other high-level development platforms, such as Scratch and Micro:Bit, we plan to give users a code-block based development platform that interacts with IoT devices through APIs.

Motivation

As more devices are added to the internet of things and controlling them becomes more commonplace we decided to look into more user friendly methods of automating them.

We discovered that alexa routines does not allow much versatility when creating automated routines. This results in inefficiencies when trying to automate IOT interactions.

Vision

Our project intends to reduce the entry level barrier on developing networked systems. We would like to address the challenges in using Amazon's automation by abstracting the code behind the systems to a level that the users can interact with, without any prior knowledge. This should reduce the technical knowledge required and hopefully expand the number of the people using IoT devices and may pique interest into development as a whole.

Values

Increasing accessibility of technology to a wider audience.

Related Work

After deciding to create an app that will interact with IOT devices we decided to do some research into existing apps that do the same. We looked into the Amazon Alexa app as the most well known app to interact with IOT devices.

Some problems we found was that while the app only has a user friendly design it is limited in what can be accomplished in the “routines”. These “routines” do not allow for looping or for comparing data while the routine is running, this means that while it is possible to set up simple actions, setting up any true automation is impossible.

We decided to look into how best to create a user friendly method of creating code with more complex capabilities. After some thought the idea of producing a block code system. After doing some research into BBC microbit makecode we found that while the microbit is able to run any code produced in its block code interface it is unable to connect to the internet in order to control IOT devices. The closest we have been able to find is running the code on the microbit and having all outputs done by the output pins.

We looked into scratch as the most well known block code system. We found that it has more functions related to sprites than the BBC microbit however any use of external hardware was not a feature that was built in.

Requirements

| **ID** | **Feature** | **User Story** | **Description** |
| --- | --- | --- | --- |
| 1.0 | Sign up | As a **potential user** I want to sign up | Allow **user** to create an account |
| 2.0 | Login | As a **user** I want an account to keep my details | Allow **user** to login to their own account |
| 3 | Create Hub | As a **user** I want to create a home to use link my devices | Allow **user** to create a **hub** to link devices to |
| 4 | Register devices | As a **user** I want to register devices to the house | Allow **user** to connect multiple devices to a **hub** |
| 4.1 | See device status | As a **user** I want to be able to see which devices are currently connected to my hub | Allow **user** to see all previously connected devices and their current status |
| 5 | Design Schedule | As a **user** I want to create a theoretical design / schedule | Allow **user** to make a schedule which can be hooked up to virtual devices (as a temporary palcement). |
| 6 | Assign Schedule | As a **user** I want to utilise my designed schedule and put it on a house/hub | Allow **user** to assign a design schedule to **hub** and assign devices to replace the temp virtual devices, It will then be stored and activated on the **hub** if it all checks out |
| 7 | Create schedule | As a **user** I want to create a schedule that can be loaded instantly to a hub | Allow **user** to create a schedule that is assigned to the **hub** on completion |
| 8 | Delete schedule | As a **user** I want to delete a schedule that I have previously made | Allow **user** to delete a schedule from their account |
| 9 | Export schedule | As a **user** I want to turn my schedules into a shareable template | Allow **user** to turn their schedule into a shareable format |
| 10 | Import Schedule | As a **user** I want to be be able to import templates from others | Allow **user** to import a file to add a schedule into the account |
| 11.0 | Hub invite | as a **user** I want to add others to the hub | Allow **user** to add other **users** to the **hub** |
| 11.1 | Select Roles | as a **user** I want to be able to select specific roles for others when inviting them to the **hub** | Allow **user** to select roles when creating an invite for others |
| 12.0 | Hub join request | as a **user** I want to be able to request to join a **hub** | Allow **user** to send a **hub** join request |
| 13.0 | Accept/Deny join request | as a **hub owner** I want to be able to accept or deny a **user** request to join my **hub** | Allow **hub owner** to accept or deny a **hub** join request |
| 14.0 | Manage Roles | As a **Hub admin** I want to manage roles of other accounts on the hub | allow **hub admin** to change other **user**'s roles |
| 15.0 | Remove schedule | As a **user** with appropriate role i want to be able to remove an inactive or unwanted schedule from the **hub** | Allow **user** (perm oriented) to remove a schedule from the **hub** |
| 16.0 | Modify Settings | As a **user** I want to modify my own settings | Allow **user** to modify account settings |
| 17.0 | Toggle schedule activation | As a **user** with appropriate role want to be able to activate/deactivate a schedule on the **hub** | Allow **user** with appropriate role to activate and deactivate schedules on **hub** |
| 18.0 | Request schedule activation | As a **user** having lower role I want to be able to request owner for activation/deactivation of my schedule on the **hub** | Allow **user** with lower role to request **hub owner** activate and deactivate schedules on the **hub** |
| 19.0 | Accept/Deny schedule activation | As a **hub owner** I want to be able to accept or deny any request for activation/deactivation of a schedule on my **hub** | Allow **hub owner** to accept or deny requests to activate and deactivate schedules on my **hub** |
| 20.0 | Scalability | As a **user** I want to be able to access the system regardless of other users. | Write the system to be able to support several **users** concurrently |
| 21.0 | Availability | As a **user** I want to be able to access the system at any time | Write the system so that it is active 24/7 for the majority of each year |
| 22.0 | Robustness | As a **user** I do not want to encounter errors | Write code to handle all erroneous inputs |
| 23.0 | User Serviceability | As a **user** I want to be able to fix mistakes that I have made | Include options to undo, require **user** confirmation for big changes, add a help page. |
| 24.0 | Accessibility | I want to be able to effectively use the system as an **Impaired User** | Ensure the UI is colorblind-friendly, easy to read, and can be easily navigated without the use of a cursor. |
| 25.0 | Device Usability | As a **user**, I wouuld like to be able to access the site across multiple devices | The system could allow for access across multiple devices |
| 26.0 | Paired Programming | As a **user**, I would like to have **AI** assistance for code, similar to autocorrect, based on the project's title. | The system should implement a simple **AI** to suggest the next steps in a program, based on the title and description |
| 27.0 | Open Source | As a **user**, I would like to be able to access and upload schedule templates to a shared **database**. | The system should allow **users** to send working schedules to a shared pool |
| 27.1 | Positive Recommendations | As a **user**, I would only like to be recommended relevant and high-quality schedules from the shared **database** | The system should allow **users** to like & dislike schedules which will then be used to suggest schedules |
| 28 | Password reset | As a **user** I would want to be able to reset my password in the case i forget it | System should allow **user** to reset the password with email or phone number |

Use case diagrams

Use Case Description

#### **Name: Design Schedule**

**Scope:** Scheduling system  
**Primary Actor:** User  
**Secondary Actor:** Database  
**Preconditions:**

* User is logged in
* User has internet connection

**Main Success Scenarios:**

* 1- User presses design schedule
* 2- New draft schedule created (with default trigger)
* 3- User selects trigger
* 4- User designs schedule with available nodes
* 5- User clicks save schedule
* 6- Schedule input are verified
* 7- Schedule saved onto database
* 8- Schedule assigned to user's account

**Alternatives:**

* 5a. User clicks draft toggle
  + 6, 7, 8
  + 9- Set draft to not true on system
* 5b. user clicks assign to house
  + 6,7,8,9
  + 10- Schedule is assigned to valid house
  + run create schedule

**Exceptions:**5b - User has no house, not ran  
6b - User has invalid input

* User prompted to reinput values  
  6c - Malicous inputs detected
* run Malicous activity handling

**Postconditions:**1a. User has new shcedule  
1b. User has new shedule assigned to house

#### **Name: Add Device**

**Scope:** Hub  
**Primary Actor:** User  
**Secondary Actor:** Database  
**Preconditions:**

* User has hub on account
* User logged in
* User has admin+ permissions on hub
* User on hub page
* User has a device that can connect
* Device is capapble f conneccting to hub (wired or wireless)

**Main Success Scenarios:**

* 1- User clicks add device
* 2- User is taken to add device page
* 3- User inputs needed details
* 4- User tests connections
* 5- Device is connected to the hub
* 6- User clicks save
* 7- Device is registered to hub
* 8- Device saved to hub DB

**Alternatives:**7b. Device details not matching and doesnt connect to hub - back to 3

**Exceptions:  
Postconditions:**User has new device connected to the hub

Software Architecture

User Tests

| **ID** | **Description** | **How will test be ran** | **expected outputs** | **Results to be recorded** |
| --- | --- | --- | --- | --- |
| 1 | The user should be able to create an account from out login page -> sign up page | We will create a few fake accounts with the email verification turned off, We will then try and sign into each of those accounts | Database shall now have a few accounts on the system, we shouldve been able to login to all accounts | If accounts are able to login, if not why, also checking that details are correct |
| 2 | The user should be able to try and reset their password | Will try use the forgot password option and try and recover the account | System should allow us to reset our password, if 2FA is enabled the 2FA app will have to be utilised to be able to reset it | Whether account can reset password, IF can reset W/O 2FA it is failure |
| 3 | user should be able to logout of their account and be brought to the main login page | Will attempt to logout of page by clicking on account -> logout | The user should be brough back to the main login page with no reminence of the account left | Will check the cookies or whatever system was used to keep the account logged in |
| 4 | User should be able to delete their account from the account details / settings page | Will go to the settings and delete the account, there should be no reminance of their details | The account should be off the DB upon deletion, anything that they own or are attatched to their account will be removed (if they own a house with others in the ownershipp shall be transferred) | Will check for reminence of the account |
| 5 | User should be able to request to join a hub | Will input a code / address for the hub that they wish to join | There will be a request to be accepted/denied on the hub that they had requested to join | Wether the account is displayed on the requests of the hub |

\*Additional user tests have been designed but have not been copied to save space on the report document