



REPORT ONE

Group 15: Mahmoud Bachir (Leader), Aaron Christie, Smruthi Srikumar, Gunbir Singh, Brendan Li, Kyle Ross, Jimmy Wen, Karthikey Thapliyal



AUTOHOME

<https://github.com/mahmoudbachir/homeautomation>
<https://ruautohome.weebly.com/>

Table of Contents

Contribution Breakdown Table	2
CSR Narrative	4
System Requirements	8
Original Graphic Mockups.....	9
Glossary.....	10
Internal Stakeholders.....	11
External Stakeholders	11
Actors and Goals	12
Use Cases	13
Use Case Diagrams.....	15
Traceability Matrix (Use Cases).....	16
Fully Dressed Descriptions for Important Use Cases	17
System Sequence Diagrams:	21
User Interface Preliminary Design/Navigation:	25
User Interface: User Effort Estimation.....	28
Domain Concepts	31
Domain Models.....	34
System Operation Contracts	42
Traceability Matrix: Domain Concepts.....	47
Plan of Work	48
References	49

Contribution Breakdown Table

	Mahmoud	Aaron	Smruthi	Jimmy	Brendan	Gunbir	Kartikey	Kyle
Project Management	100%							
CSR: Problem Statement	50%			50%				
CSR: Glossary	36%		16%	32%		16%		
Non-func reqs	25%		50%		25%			
Func. Req's	25%		25%	25%	25%			
UI Reqs		50%	50%					
Stakeholders + actors				100%				
Use Case Casual Descriptions	100%							
Use Case Full Description	20%							80%
Use Case Diagrams					100%			
Traceability Matrix - use cases						100%		
System Sequence Diagrams							100%	
UI Prelim Design		75%	25%					
UI Effort Estimate		25%	75%					
Domain Concepts		20%	20%	20%		20%		20%

Domain Assoc.		25%	25%			25%		25%
Domain Attrib.		25%	25%			25%		25%
Domain Contracts					100%			
Domain Traceability Matrix						100%		
Plan of Work	50%			50%				

Table 1: Individual Member Contributions

CSR Narrative

In the past few years I've found myself gradually more and more tired as I go through my day. I don't have a particularly stressful or physically laborious job, but rather I feel that my issue stems from the fact that there are several menial tasks that need to be completed around the house, several times over. Some examples of such tasks include changing the light settings around the house (turning it on/off, turning up/down the brightness, changing the color to create an ambience in the house), locking the doors and being sure that they are truly locked, and also, preparing an ambience and mood in the house before I arrive home (as my guests may arrive only moments after I do). Now stop, don't say anything, I know what you're thinking, you think I should hire a maid. Of course, that is certainly an option, however, that is not the best choice that I have. Hiring other people isn't the best course of action as they may be unreliable and untrustworthy, I couldn't see myself leaving them in my house without my supervision. I want my house to be secure and inviting a stranger into my house to take care of it simply does not make sense to me. Furthermore, it is also not good for my image in front of friends and coworkers, and they may look down upon hiring a maid. Instead of hiring a maid, I have a better idea.

The other day it occurred to me, 'Instead of hiring human workers to maintain and secure my house, I should have an electronic solution implemented!'. Machines are much more reliable and trustworthy than humans. Their behavior is programmed, and concurrently, they are predictable. A system can be devised where I can use my phone, laptop, or even my voice to give commands, and they will be executed these commands quickly and without question. Thereby, I want a system to automate my home, I will refer to it henceforth as autoHome.

The vision that I currently have for autoHome contains 5 main classifications of features. These features include: remote control using a mobile application, light control, audible alerts, music control and voice-controlled activation of the system from within the house. In my opinion, the most important feature of autoHome is the remote home control feature. This will be implemented with a mobile application. The application should have a secure internet connection to the home and the user should be able to connect to his/her house by simply logging in to a personal account. The account should only allow a predefined maximum number

of people to log in at once, you can add more at any time if you'd like. This should serve a dual purpose of preventing user conflicts and increasing account security (this way it is apparent when an unauthorized user accesses an account). The application should prevent a user from logging in without further verification of identification if the user fails to login after seven consecutive attempts. In the event that someone has attempted to access account more than seven times, then the application should also send an alert to the user's mobile device. Once the user has logged in, he or she should be able to monitor and control various connected devices around the house. The main device features that I currently envision for the user to use within the application include; viewing the status of each door lock around the house, the brightness of lights in each room, the colors of lights in rooms with RGB LED lights, the playing of music in each room, the volume of music in each room, and the connection status of the application to devices in the house via Bluetooth.

As an extra measure of security, buzzers should be installed next to each door. These buzzers will sound whenever a door is opened. The purpose of these buzzers is to alert any residents inside the house when another person enters or exits the house. Buzzers are installed next to each door so that residents can also have a sense of which door was opened. Additionally, this audible security measure is user-friendly to the visually impaired.

Another extremely appealing feature would be light control. First and foremost, I believe that users should be able to easily control the activation (on) and deactivation (off) of lights in each room of the house. You should be able to control the lights directly from the mobile/web application. This would easily and conveniently solve the issue of forgetting to turn off lights on the way out or being able to turn off my bedroom lights once I'm already in bed. Also, a potential use of this function could include remotely turning on lights when not in the house (giving impression that someone is home) to ward off potential burglars.

Two more features regarding the automation of light include controlling brightness, and the control of the bulb. While these features aren't necessarily related to home security, they are extremely versatile. The brightness and color of a light can create an ambience in the house and can influence the mood of people. These effects can be utilized in several situations such as functions (parties), meetings, studying, etc.

In addition to the automation of lights, I also want to be able to control audio devices in the house through my mobile device. I should be able to control playing/pausing music, as well as controlling the volume of the music. The application should be able to sync audio between the mobile device and the audio devices around the home via Bluetooth.

While users are inside the house, there are situations where it would be more cumbersome to unlock their phones, load the application, and ONLY THEN would they be able to control devices in the house. I think that instead of exclusively using the mobile application to control the home automation, the users should also be able to activate and deactivate voice control of the home automation system. Users should be able to set up a custom key phrase that will unlock or lock voice control. Additionally, the system should be able to take specific phrases as input and then execute the corresponding command. For example, suppose a user has already set up the key phrase, “monkey one two three” as his or her key phrase. The user should then be able to use the key phrase whenever he or she is in the house to activate the home automation system’s input reception. The user should then be able to give commands, such as “turn off all lights downstairs,” followed by “play next song.” The system should then turn off all lights on the first floor and skip to the next song on the user’s current playlist. Until the user repeats the key phrase, “monkey one two three,” the system should continue to listen for commands. After the user repeats the key phrase, the system should no longer listen for commands and voice control should be deactivated. If any person tries to give the system commands while voice control is deactivated, then the system would ignore those commands.

Those are the main features that I currently have in mind for a user to have access to. However, I understand that if multiple users are to be able to use this application, then users need to have profiles specifically for themselves and their homes, and they should have a username and password to log in to these profiles. Thereby I think it is necessary that user information is stored in a secure database. The most important information is obviously the username and password, as this gives you access to someone’s user account and thereby, control over their home. This will ensure that a user can only establish a secure connection to his/her specific home only. It is of the utmost importance that users are unable to gain unauthorized access to other homes.

Security is an important feature that is crucial to this electronic solution. I mentioned above that users should be alerted if there are multiple failed login attempts to an account (specifically 7 failed attempts) as well as being able to set a maximum number of users on a account. However, I think that there are additional safeguards that can be taken to combat this. I believe that there should be additional verification of user's identities, and in order to establish this, there must be verification questions (recovery questions) that a user must answer to gain access to an account. The system will have to store these questions and their answers in a database. The system should also record the time/date that these failed login attempts were made.

While I greatly value the convenience that an automated home system would give me, it is imperative that this system is secure. Nobody should have access to any account that they are not authorized to use, only me and other authorized users should be able login and use this system. It is my hope that autoHome will be reliable and convenient, since it is my intention that it is utilized on a daily basis. Additionally, in the future I expect ownership of an automated home system to become a standard of living. The features previously described are applicable in a wide variety of situations, which gives the product a large target audience. I am excited to see the future of autoHome and hope that it can be developed without any issues.

System Requirements

Identifier	Functional Requirements	Priority
REQ-1	The system shall be able to control various light fixtures around the house (on, off, dim)	5
REQ-2	The system shall be able to alert the user when a door opens or closes with a audible signal	5
REQ-4	The user should be able to control their home audio system (volume adjust, play/pause/skip)	2
REQ-5	The user should be able to control the system with their voice through the use of a key-phrase	1
REQ-7	The user should be able to easily add new items to be controlled with the system	2
Identifier	Non-Functional Requirements	Priority
REQ-3	The user should be able to monitor their system with their mobile phone through an application	4
REQ-6	The user will be able to save user profiles for different members in the household	1

Table 2: Functional + Non-Functional Requirements

Identifier	On-Screen Requirements	Priority
REQ-1	The application should allow activation of the voice control system	4
REQ-2	The application should provide audio/visual feedback when controlling a system	3
REQ-3	The application should have buttons to allow manual adjustment of various systems	5
REQ-4	The application should allow the user to see the status of the systems they have connected	5
REQ-5	The application should have a responsive and easy to use interface	2
REQ-6	The application should provide feedback on system alerts and status via notifications, when application is not running / running in background	3

Table 3: On-Screen (UI) Requirements

priority is ranked on a scale of 1-5, 1 being the highest and 5 being the lowest

Original Graphic Mockups

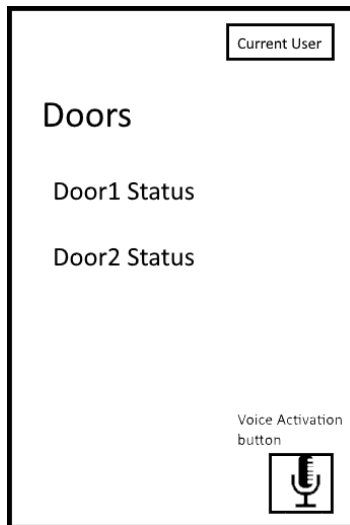


Figure 1: Door UI

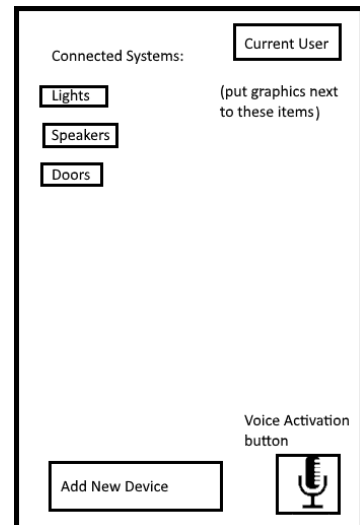


Figure 2: Devices UI

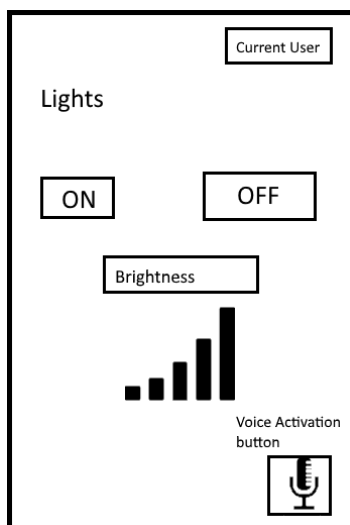


Figure 3: Lights UI

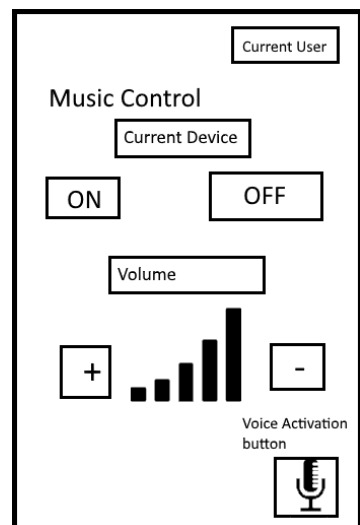


Figure 4: Music UI

Glossary

User - A person who intends to implement home automation in the home automation system.

User profiles – Profiles that store different settings for different users of the same system.

Key Phrase – Phrase that activates the voice control system

Home Automation – Automation of the home or household activities. This may include lighting, doors, speaker systems, etc.

Home Automation System – The system of devices that seamlessly connects all aspects of home automation in a form the user can easily understand and use.

Voice Activation – System that allows users to use voice to manipulate home system

Voice Activation Device – Device that listens to user's voice, and implements the user's intended change to the system

RGB – Letters that represent red, green and blue. Each of these three colors usually holds a value between 0 and 255.

LED – An acronym for light-emitting diodes. Light emitting diodes are two-lead semiconductor light sources.

Mobile Device – A device meant for mobile use; tablets, smartphones, and other android-capable devices.

Application – The primary program the user will be interacting with to control various parts of the home automation system

Device Status – Status of the user's devices. Lets the user know which devices are off/on, or locked/unlocked

Mobile Application – Application used on a mobile device

Bluetooth – Wireless protocol enabling mobile device to interaction with the home automation system

Arduino – Open-source electronic prototyping platform enabling users to create interactive electronic objects.

Microcontroller – Remote controller that will allow the user to communicate with the devices from a remote location.

12V Relay-Relay – an electromagnetic switch which can be turn on and off by an applying the voltage across its contacts. In this project used a 12V 4-channel relay

Relay driver – ULN2003 – Relay safely driven by ULN2003 IC. Protect microcontroller from relay kick back using integrated clamping diodes. Has 7 high current Darlington arrays each containing 7 open collector Darlington pairs with common emitters.

Internal Stakeholders

- **Developers (internal):** The individuals or groups of individuals who design, code and physically construct the components of autoHome. They have the right to access all parts of the product. The interests of internal developers include convenience, affordability, profitability, functionality and applicability.
- **Angel investor:** The individual or individuals who invest money in entrepreneurs starting a business. In this case, they would be investing in the internal developers of autoHome. Angel investors often become board members and guide the entrepreneurial developers through the initial phases of a startup business. Angel investors are usually interested in the success of the product or company, a share of the company's profits and equity ownership interest.
- **Venture Capitalist:** A firm that invests a large sum of money in a company in return for a percentage of equity and significant influence. Equity is the value of shares issued by a company, so the venture capitalist firm that invests in the company owns a percentage of the company. Venture capitalists are interested in profit, influence and stakes in the company.

External Stakeholders

- **Developers (external):** Individuals or groups of individuals wishing to utilize the software design, hardware design or code already developed for autoHome. External developers may be interested in personal profit, gaining a competitive advantage or educational benefits.
- **Customers:** Families, individuals or small businesses who want to use autoHome in their homes or in small buildings. The interests of customers include functionality, effectiveness, convenience, affordability and visual appeal.
- **Competitors:** Companies or organizations that make products or provide services with similar purposes as autoHome. These parties would be interested in reverse engineering autoHome

to gain a competitive advantage, buying rights to technology used in autoHome or buying out the company that produces autoHome altogether in order to eliminate competition

Actors and Goals

Actor	Category	Goals
User	Initiating	<ul style="list-style-type: none"> Control devices around his or her own home or building Monitor the statuses of devices around his or her own home or building
Device sensors	Initiating	<ul style="list-style-type: none"> Report either measured or binary values to the central system
autoHome System	Initiating	<ul style="list-style-type: none"> Send signals to devices to turn them on or off Send signals to devices to control their output levels Store data when signals are received
User Web Interface	Initiating	<ul style="list-style-type: none"> Send messages to user's respective autoHome system Display device status to user
autoHome System	Participating	<ul style="list-style-type: none"> Receive signals from devices around the building Listen for signals from devices
User Web Interface	Participating	<ul style="list-style-type: none"> Receive signals from user Receive signals from autoHome system
Database	Participating	<ul style="list-style-type: none"> Store data pertaining to device access and device access attempts Store data pertaining to status of each device

Table 4: Actors and Goals

Use Cases

Use Case Name	REQ's	Actor	Actor's Goals
Login (UC-1)	REQ-8	User	To login to personal account
Logout (UC-2)	REQ-8	User	To logout of personal account
Add-User (UC-3)	REQ-8	User	To create a new user account and allow access to home devices
Remove-User (UC-4)	REQ-8	User	To remove a user account and disable access to home devices
Access-History (UC-5)	REQ-10	Database	Stores all access attempts in log with login time/location (using ip address)
Lock-Monitor (UC-6)	REQ-2	Sensors	To monitor the door for possible intrusions
Add/Remove-User (UC-3, UC-4)	REQ-8	Database	To create/remove user profiles to database and subsequently enable/disable all user's home devices
Monitor-Device (UC-7)	REQ-3	User Web Interface	To monitor all devices statuses (light is on/off, music is paused/playing, door is locked/unlocked)
Monitor-Devices (UC-7)	REQ-3	User	To be able to monitor all device statuses from the mobile app, or the web app
Add-Device (UC-8)	REQ-7	User	To be able to add new devices (lights, speakers, etc.) easily
Add-Device (UC-8)	REQ-7	Database	To add new devices to database for each user
Device-Status (UC-9)	REQ-3	Database	To update the status of each device every interval of time
Read-Device-Status (UC-10)	REQ-3	User Web Interface	To take device status info from the database and display to the user in the appropriate section (status updates accordingly)
Light-Off (UC-11)	REQ-1	autoHome System	To turn off the light bulb
Light-On (UC-12)	REQ-1	autoHome System	To turn on the light bulb
Music-Play (UC-13)	REQ-1	autoHome System	To start playing music on music device (speaker, etc.)
Music-Stop (UC-14)	REQ-1	autoHome System	To stop playing music on music device
Light-On/Off (UC-11, UC-12)	REQ-1	User	To be able to turn on/off the light bulb from app
Music-Play/Stop (UC-13, UC-14)	REQ-1	User	To be able to play/pause music on device from app

Connect-Device (UC-15)	REQ-9	autoHome System	To be able to connect a new device to a user's autoHome (Bluetooth speaker, WiFi devices, etc.)
Add-Device (UC-16)	REQ-9	Database	To add a new device to a user's autoHome account
Music-Control (UC-17)	REQ-4	User	To be able to increase/decrease volume
Music-Control (UC-17)	REQ-4	autoHome System	To be able to adjust the volume on the music playing device

Table 5: Casual Use Case Descriptions

Use Case Diagrams

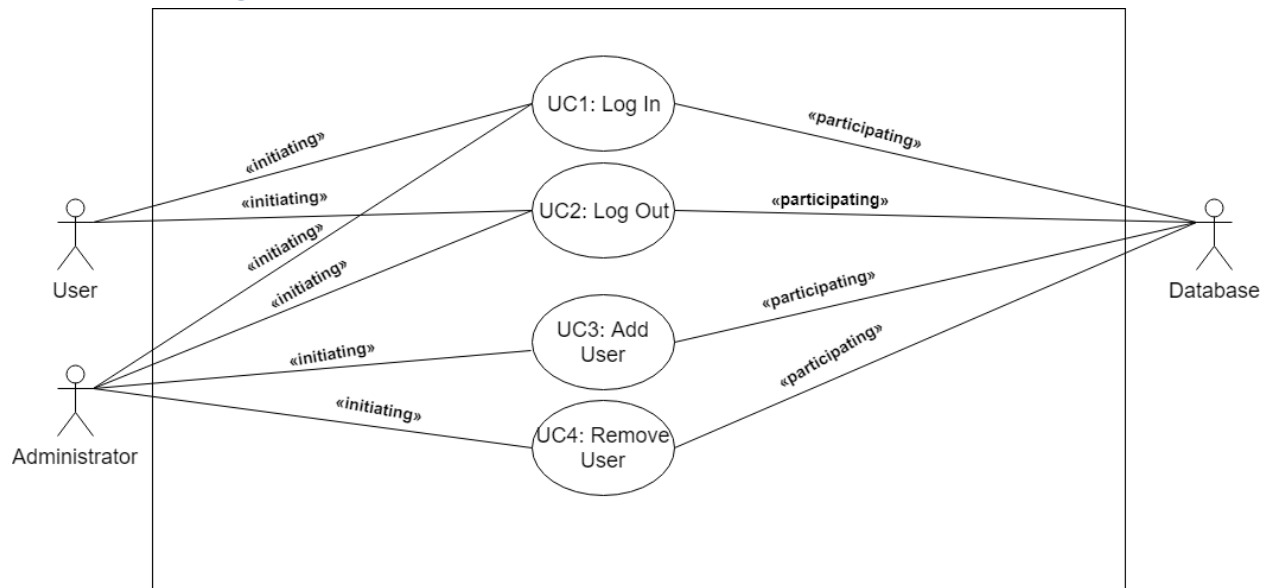


Figure 5: Account Management

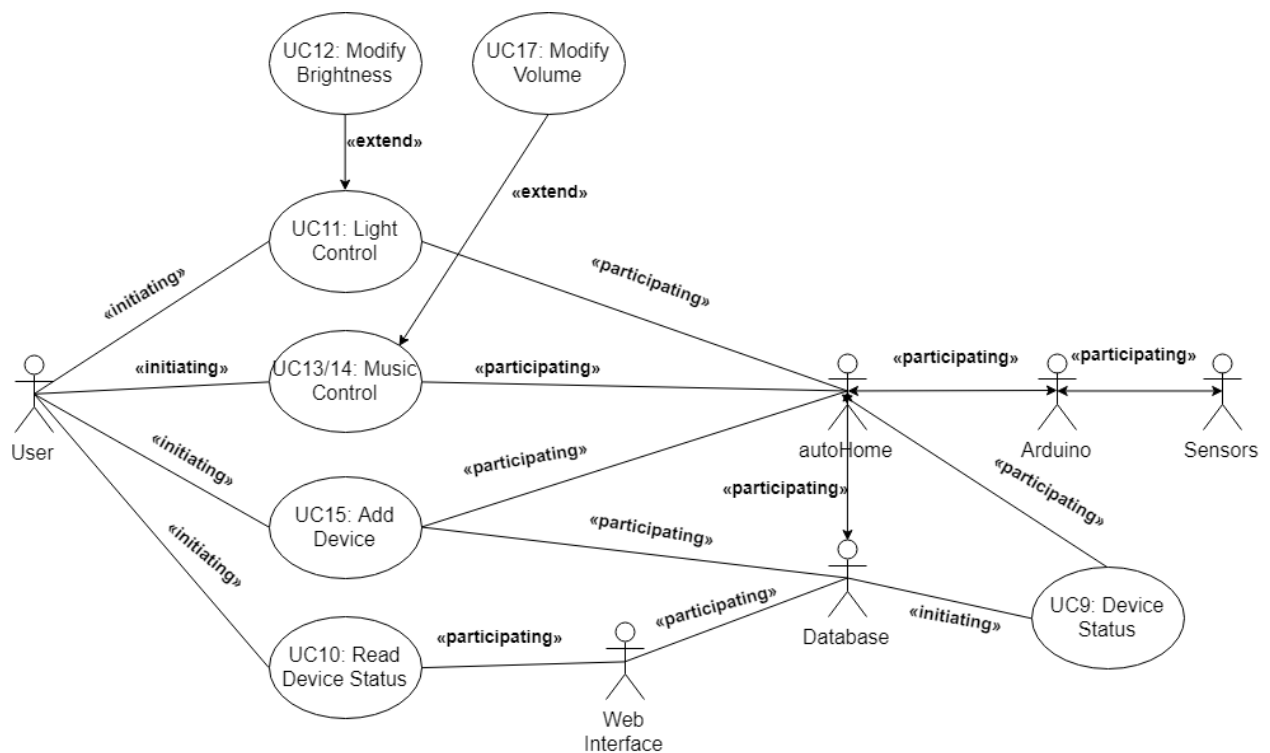


Figure 6: Device Management

Traceability Matrix (Use Cases)

REQ	P	UC	UC	UC	UC	UC	UC	UC	UC	UC	UC1	UC1	UC1	UC1	UC1	UC1	UC1	UC1
	W	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7
REQ1	5											x	x	x	x			
REQ2	5						x											
REQ3	4							x		x	x							
REQ4	2																	x
REQ5	1																	
REQ6	1																	
REQ7	2								x							x	x	
REQ8	4	x	x	x	x													
REQ10	1					x												
Max	P W	4	4	4	4	1	5	4	2	4	4	5	5	5	5	2	2	2
Total	P W	4	4	4	4	1	5	4	2	4	4	5	5	5	5	2	2	2

Table 6: Traceability Matrix

Fully Dressed Descriptions for Important Use Cases

Use Case UC-1:	Unlock
Related Requirements:	REQ-8
Initiating actor:	User
Participating Actors:	User, Database, User Web Interface
Preconditions:	UC-3
Postconditions:	User is logged into personal account and has access to personal settings as well as control over Autohome system.

Table 7: UC-1

Flow of Events for Main Success Scenario:

1. User opens User Web Interface and is prompted with login screen.
2. User inputs correct username and password that has been added into the database previously.
3. User Web Interface sends inputted data to database

Following this, alternate scenarios:

- A. When username/password is inputted correctly and found within database
 - a) User Web Interface displays main page.
- B. When username/password is inputted correctly or found within database
 - a) User Web Interface displays error and reverts to login screen.

Use Case UC-3:	Add-User
Related Requirements:	REQ-8
Initiating actor:	User
Participating Actors:	User, Database
Preconditions:	User must have app installed
Postconditions:	User has new account in database that can store user's personal settings profile

Table 8: UC-3

Flow of Events for Main Success Scenario:

1. User selects to create account on User Web Interface.
2. User creation page is displayed. User is prompted to input relevant settings information, as well as a unique username/password.
3. User inputs all relevant information correctly and prompts User Web Interface to register new account.
4. User Web Interface sends required information to Database.

Following this, alternate scenarios:

- a. When information is correctly inputted and username is unique:
 1. User profile is updated into database.
 2. App displays confirmation page that the new user is now added into Database.
 3. User is prompted with login screen.
- b. When correctly inputted or username has been registered
 1. App displays error
 2. Interface reverts back to user creation page.

Use Case UC-8:	Add-Device
Related Requirements:	REQ-7
Initiating actor:	User
Participating Actors:	User, autoHome Devices, User Web Interface, Database, Arduino
Preconditions:	UC-1
Postconditions:	User has added device to and is able to configure device.

Table 9: UC-8

Flow of Events for Main Success Scenario:

1. User prompts User Web Interface to add new device
2. User Web prompts Arduino to detect all devices near it.
3. Arduino sends all possible connectable autoHome Devices to User Web Interface.
4. User Web Interface displays all connectable devices to user. (If none, User Web Interface displays “no devices found” and reverts to main page)
5. User selects Device/s they would like to add and then confirms.
6. Device/s is added to Web User Interface.
7. User Web Interface adds device to database(UC-16)
8. User Web Interface displays connected devices as well as actions for those devices (UC-7,UC-11, UC-12, UC-13, UC-14, etc.).

Use Case UC-9:	Device-Status
Related Requirements:	REQ-3
Initiating actor:	User Web Interface
Participating Actors:	User Web Interface, Arduino, User
Preconditions:	UC-8
Postconditions:	Change of status of any devices is updated and displayed to User if a change is detected.

Table 10: UC-9

Flow of Events for Main Success Scenario:

1. Web interface prompts Arduino to monitor currently connected devices.
2. Arduino pings autoHome System and Device sensors to check if any changes have occurred.
3. autoHome System sends Arduino current status data
4. Arduino notifies User Web Interface of possible changes

Following this, alternate scenarios:

- A. If changes occurred:
 - a. Arduino sends back changes to User Web interface
 - b. User Web interface notifies changes to User (UC-10)
 - c. User Web interfaces returns to step 1. (UC-9 will being going at regular intervals)
- B. If no changes occurred:
 - a. User Web interfaces returns to step 1. (UC-9 will being going at regular intervals)

System Sequence Diagrams:

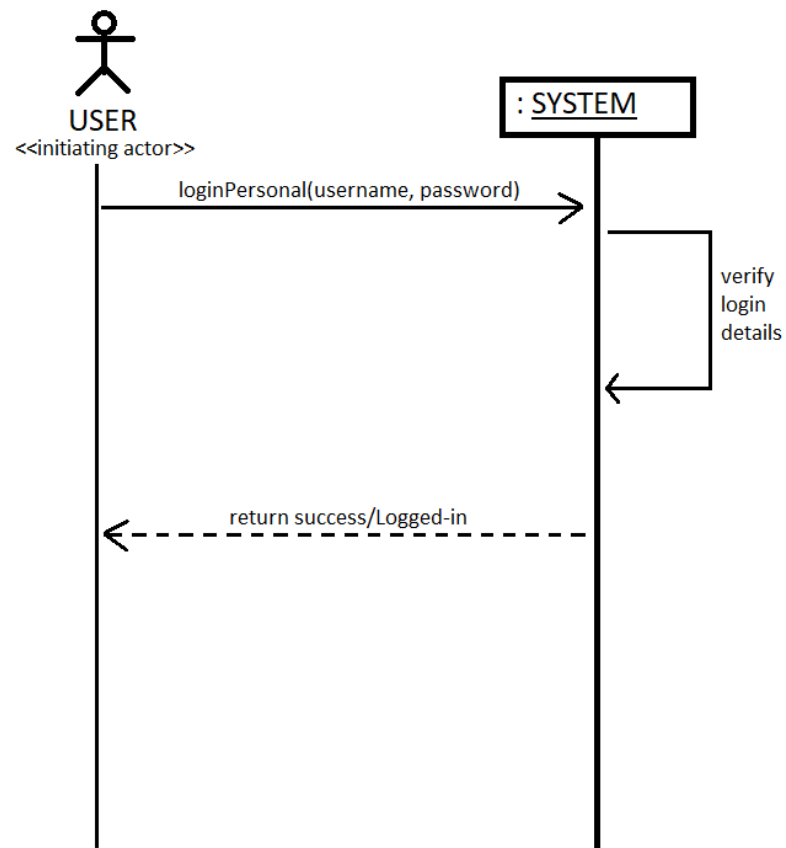


Figure 7: Login (UC-1):

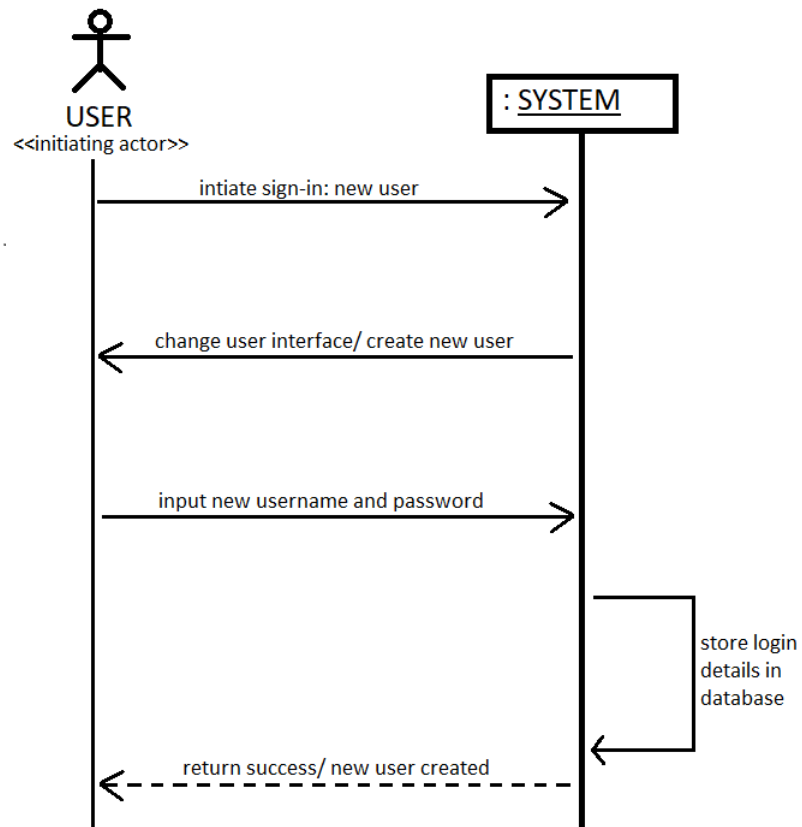


Figure 8: Add-User (UC-3):

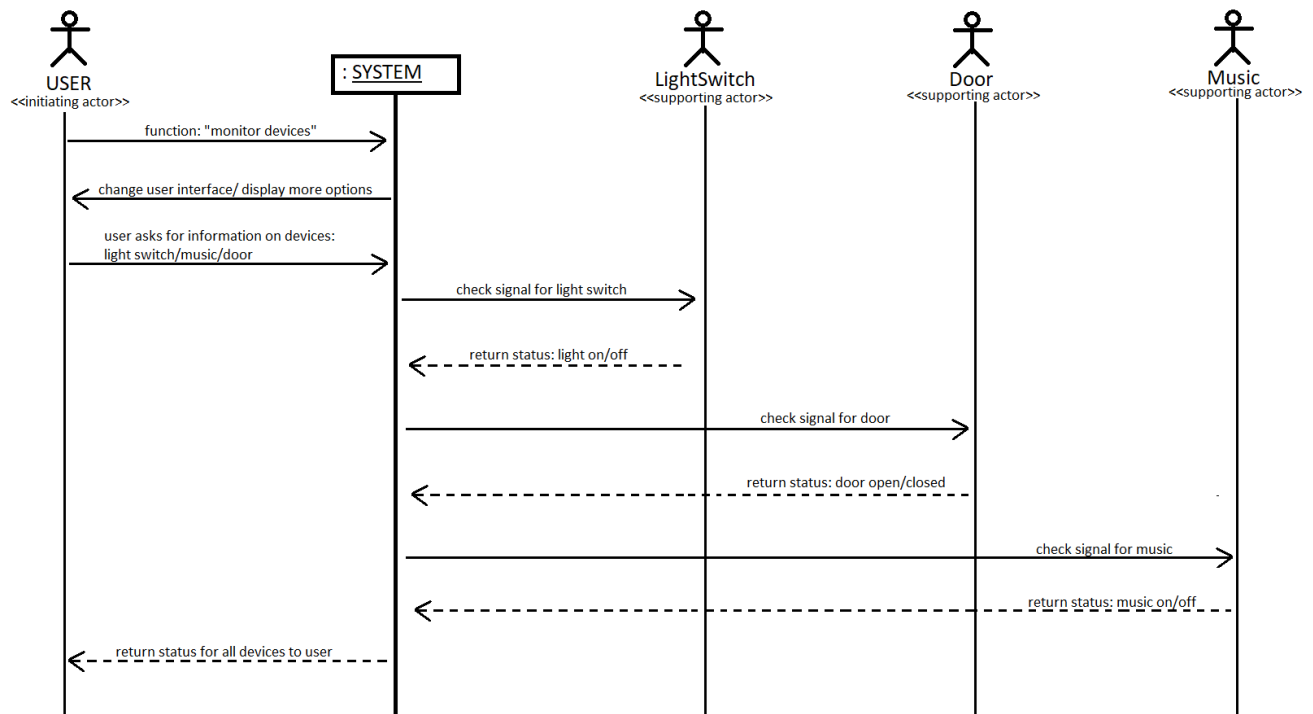


Figure 9: Monitor-Devices (UC-7):

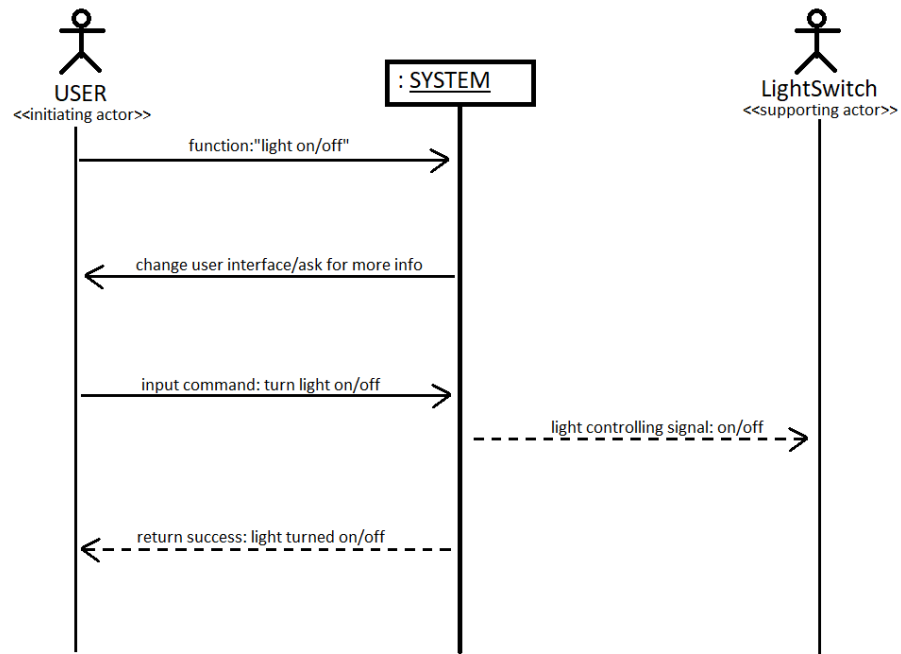


Figure 10: Light-On/Off (UC-11, UC-12):

User Interface Preliminary Design/Navigation:

1.) Sign in

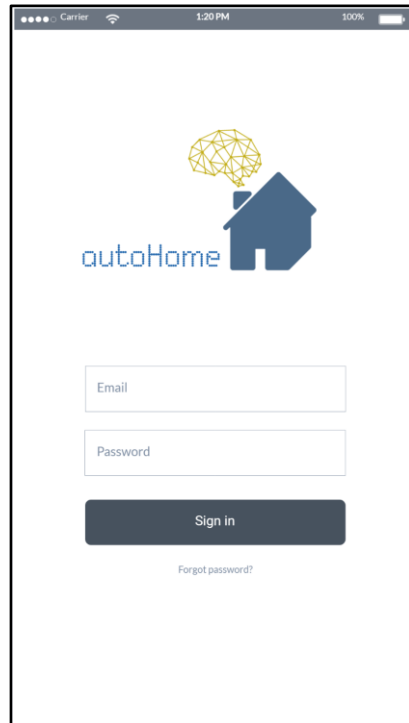


Figure 11: Login Screen

- User enters email and password, and then taps “Sign in” button in order to login to their user profile

2.) Home Screen

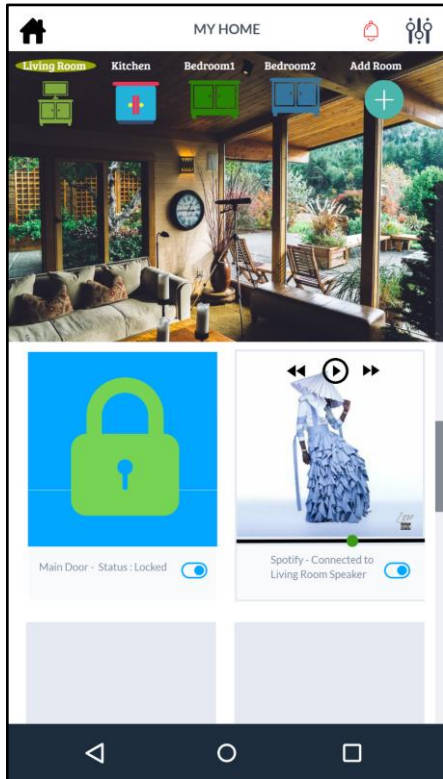


Figure 12: Living Room Tab

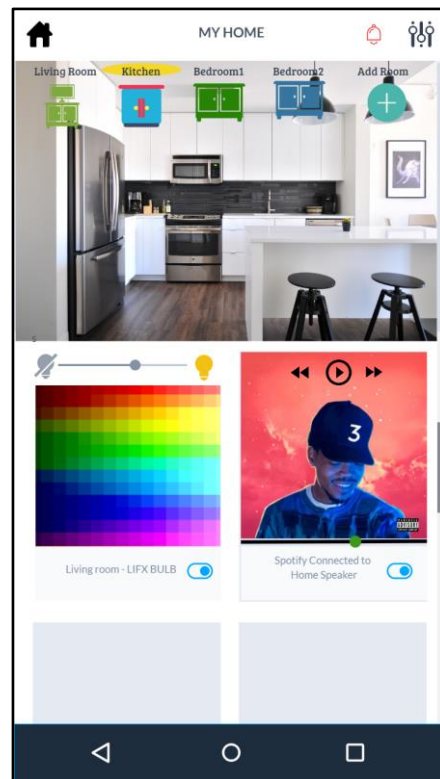


Figure 13: Kitchen Tab

- Organized by Rooms (Tabs), selected room gets highlighted and picture changes
- Displays all connected devices and controls for the devices
- In the living room, the user has connected Spotify and the main door
 - Can choose to turn off alarm or the music right from the home screen
 - Can adjust music volume / change songs
- In the Kitchen, the user has connected Spotify and a Lif-X bulb
 - S/he can turn on/off the lights as well as adjust the brightness and color of bulbs
- Can choose to add new devices by clicking the empty slots

3.) Main Menu

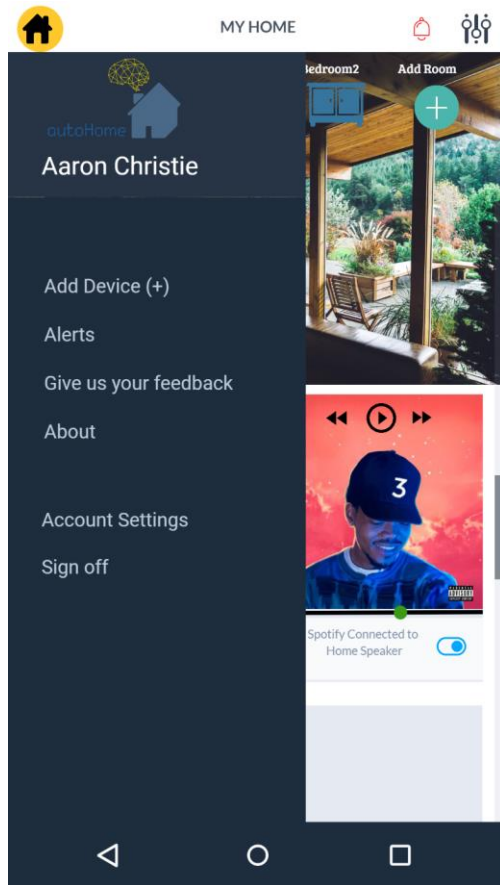


Figure 14: Menu

- ** Yellow circle indicated selected button **
- Home button (top left) brings you to the **Home Screen** unless currently on home screen. If on **Home Screen** and Home Button pressed, opens the Main Menu
- Add Devices (+) allows user to add new device
- Alerts (underneath Add Device button) contains all system alerts, past & present

User Interface: User Effort Estimation

Lights:

Turn on:

Click “lights” button (1) → click “on” (1)

- Two clicks total

Turn off:

Click “lights” button (1) → click “off” (1)

- Two clicks total

Change brightness:

Click “lights” button (1) → click “brightness” (1) → click up arrow → click down arrow

- Minimum 3 clicks total

Voice Activation:

Click “lights” button (1) → click “voice activation” (1) → use voice commands to control light

- Two clicks total

Music:

Turn on speakers:

Click “Speakers” button (1) → click “on” (1)

- Two clicks total

Turn off speakers:

Click “Speakers” button (1) → click “off” (1)

- Two clicks total

Change brightness:

Click “Speakers” button (1) → click “volume” (1) → click up arrow → click down arrow

- Minimum 3 clicks total

Voice Activation:

Click “Speakers” button (1) → click “voice activation” (1) → use voice commands to control light

- Two clicks total

Doors:

Check Door Status:

Click “Doors” button (1) → “Door Status” (1)

- 2 clicks total

Lock door:

Click “doors” button (1) → click “unlock” (1)

- 2 clicks total

Unlock door:

Click “doors” button (1) → click “lock” (1)

- 2 clicks total

Main Screen:

Add new device:

Click “Add new device” (1) → input device info (1) → click “add” (1)

- Minimum 3 clicks

Log in:

Enter email in “email” field (1) → enter password in “password” field (1) → “sign in”(1)

- Minimum 3 clicks

Log in (forgot password):

Enter email in “email” field (1) → click “forgot password” (1) → enter email again (1) → open mail app (1) → choose inbox(1) → open password reset email (1) → click reset link (1) → enter new password (1) → return to autoHome app → Enter email in “email” field (1) → enter password in “password” field (1) → “sign in”(1)

- Minimum 12 clicks

Log out:

Click “log out” at top corner (1)

- Minimum 1 click

Device statuses:

Click on any given device from home screen (1) → “device status” (1)

- Minimum 2 clicks

Add User:

“Settings” icon top right (1) → “devices” (1) → “add device” (1) → enter device information (min 1 click)

- Min 4 clicks

Domain Concepts

Concept Definitions

In order to develop the concept definitions, we must first revisit the description of the use cases to identify interactions between actors and the system. We will then identify the internal concepts and classify them into 'Know' and 'Do' concepts.

Boundary Concepts

We begin by analyzing specific actors and their interactions with the system. In our case, the specific actors are human users and the system includes the software and hardware. The first set of responsibilities is as follows:

- R1: Login to user account (UserLogin)
- R2: Logout of user account (UserLogout)
- R3: Add user account (AddUser)
- R4: Remove user account (RemoveUser)
- R5: Add a device to the system (AddDevice)
- R6: View device status (ViewStatus)
- R7: Turn audio device on (AudioOn)
- R8: Turn audio device off (AudioOff)
- R9: Control audio volume (VolumeControl)
- R10: Turn lights on (LightOn)
- R11: Turn lights off (LightOff)
- R12: Control light brightness (BrightControl)
- R13: Turn on voice control with a specific key phrase (VoiceUnlock)
- R14: Turn off voice control with a specific key phrase (VoiceLock)
- R15: Create user account (CreateUser)
- R16: Delete user account (DeleteUser)
- R17: Send alerts to a specific user's mobile device when a number of consecutive failed login attempts are made in a short amount of time (AlertUser)
- R18: Provide audio and visual feedback when user touches system controls (UserFeedback)

Internal Concepts

Now that interactions between users and the system have been identified, we analyze interactions between components of the system. These internal interactions can be split into 'Know' and 'Do' concepts. The set of 'Know' responsibilities are as follows:

- R19: Keep record of device statuses in database (RecordStatus)

- R20: Keep database of users and the respective login information of each user (StoreUserLogin)
- R21: Monitor lock status of locks around the home (MonitorLock)
- R22: Keep a record of home access attempts (RecordAccessAttempts)

The set of 'Do' responsibilities are as follows:

- R23: Add devices to database (AddDatabaseDevice)
- R24: Remove devices from database (RemoveDatabaseDevice)
- R25: Retrieve information about a device from the database (GetDeviceInfo)
- R26: Display device information on a screen in the appropriate section (DisplayDeviceInfo)
- R27: Update device status in database (UpdateStatus)
- R28: Send and receive signals between Arduino and server (SignalTransaction)
- R29: Sound buzzer when door opens (BuzzerOn)
- R30: Silence buzzer when door closes (BuzzerOff)

Many of the aforementioned responsibilities can be classified as part of a larger concept.

Concepts for the home automation system will include the following:

- AccountAccess (R1, R2)
- AccountManagement (R3, R4, R15, R16)
- DeviceManagement (R5, R6, R28)
- DeviceControl (R7, R8, R9, R10, R11, R12, R13, R14, R28, R29, R30)
- UserMachineInteraction (R17, R18, R26)
- DatabaseManagement (R19, R20, R21, R22, R23, R24, R25, R27)

Each responsibility was broken down such that it could be implemented with a single program. The responsibilities were then categorized in a single concept. It is important to note that we assigned R28 to two concepts, while all remaining responsibilities were mapped to a concept only once. R28 is an important link between the hardware and software interface, which is why it needs to be mapped twice. This discussion of concepts is summarized in the following figure.

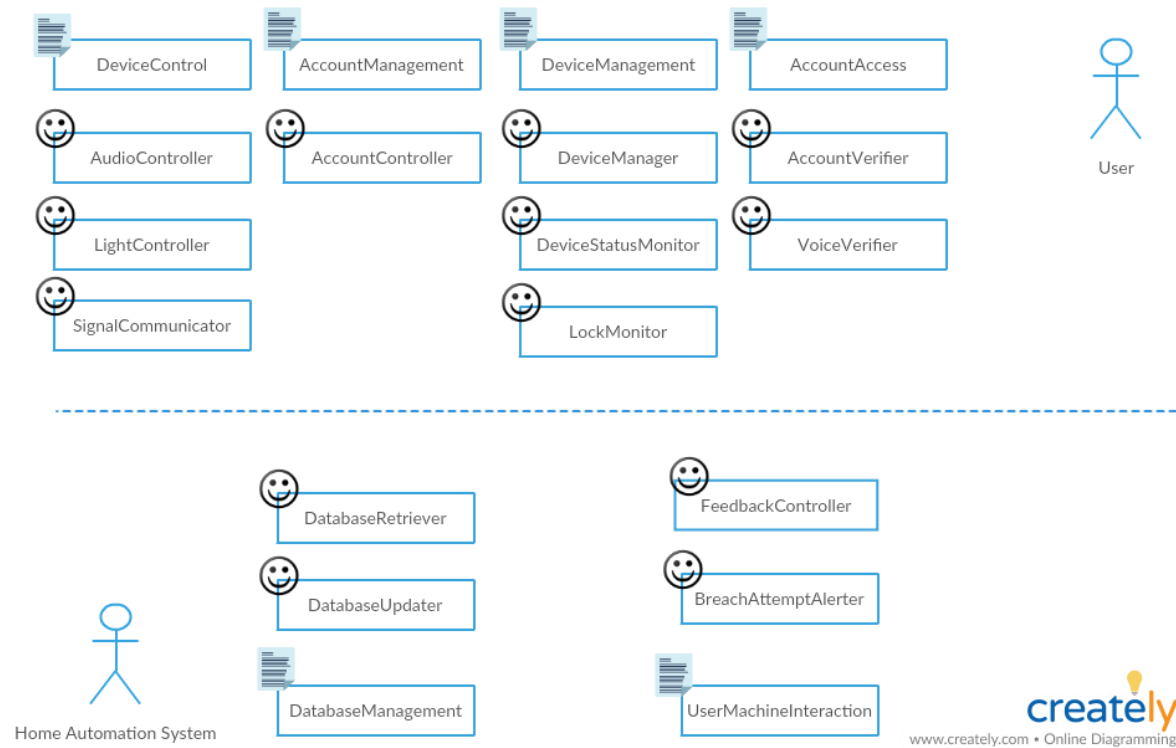


Figure 16: Domain Concepts

Domain Models

UC-1: Unlock Domain Model

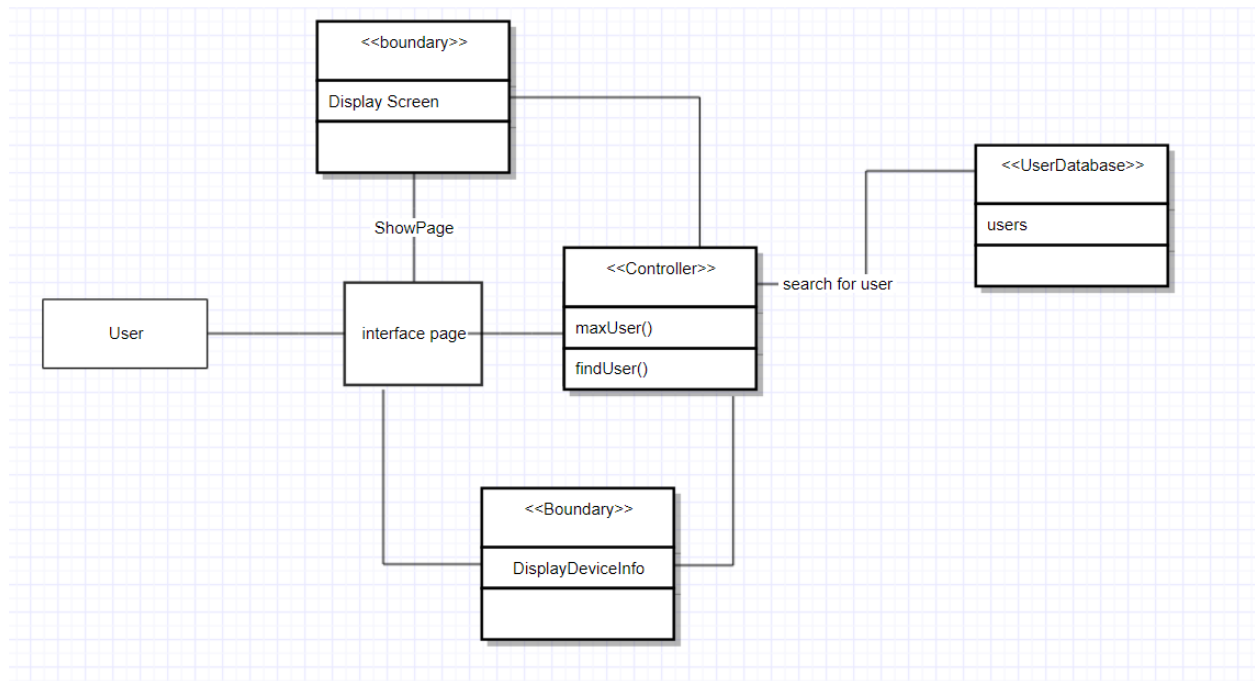


Figure 17: Domain Model for UC-1: Unlock

Concept pair	Association description:	Association Name:
Interface page <-> DeviceControl	Sends request to add-device to DeviceControl. Then, if device is found, prompts user to connect device.	UserMachineInteraction
DeviceControl <-> DisplayScreen	Display the data received on the user screen	ShowPage
DeviceControl<-> UserDatabase	Search for the user in the database	UserDetails
DeviceControl<-> AddDatabaseDevice	DeviceControl sends device data to AddDatabaseDevice; prepares it to be uploaded into Database	Convey device data

AddDatabaseDevice <-> Database	Inputs device data into Database	DatabaseManagement
DeviceControl <-> DisplayDeviceInfo	DeviceControl conveys what type of page should be sent to user (success, confirmation, error, etc.)	Write page
DisplayDeviceInfo <-> Interface page	Shows page to user	UpdateStatus

Table 10: Associations for Domain Model (UC-1)

Concept	Attributes	Attribute Description
DisplayScreen	ShowPage	Display the login page to the user
DeviceControl	SearchUser	Search if the user is present in the database
Find User	MaxUsers	Determine if the database can handle another user
	SaveUser	Used to be saved as new user's username and password
	FindUser	Check the user login details for correctness
database	User Info	Used to store user information

Table 11: Attributes for Domain Model (UC-1)

UC-3: New-User Domain Model

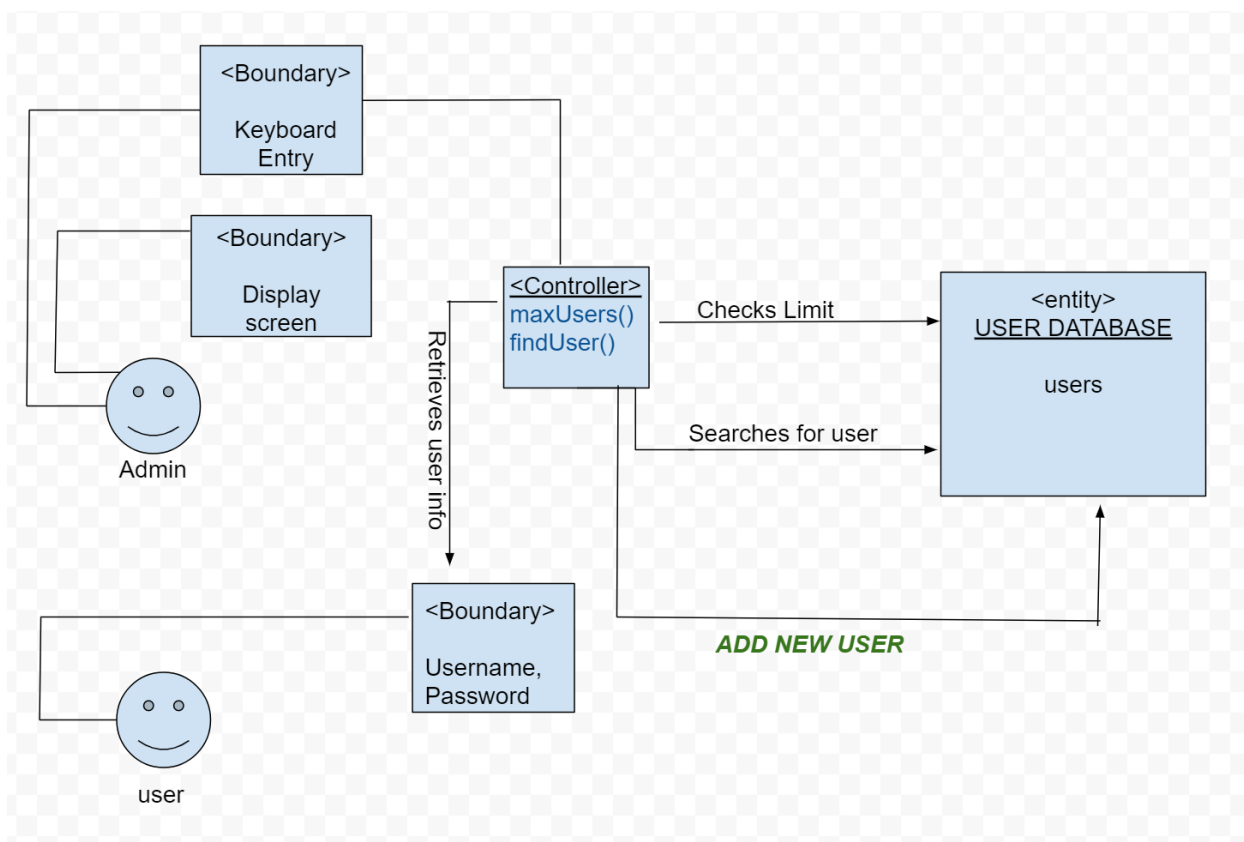


Figure 18: Domain Model for UC-3: New User

Concept Pair	Association description	Association name
InterfacePage <---> Device Controller	User sends request via the interface to the controller to create new user	Conveys request
deviceController <---> database Connection	Controller searches database to see if space available for new user, then continues	Searches
InterfacePage <---> Device Controller	User enters username and password	Provides data
deviceController <---> database Connection	Controller searches database for pre-existing user with same username, continues if no match found	search

deviceController <---> database Connection	Saves user information to database	Requests save
---	------------------------------------	------------------

Table 12: Associations for Domain Model (UC-3)

Concept	Attributes	Attribute Description
Search Request	Max amount of users	Used to determine if spot is available for new user to be entered
	New user's ID	Used to be saved as new user's username and password
database	User Info	Used to store user information

Table 13: Attributes for Domain Model (UC-3)

UC-8: Add-Device Domain Model

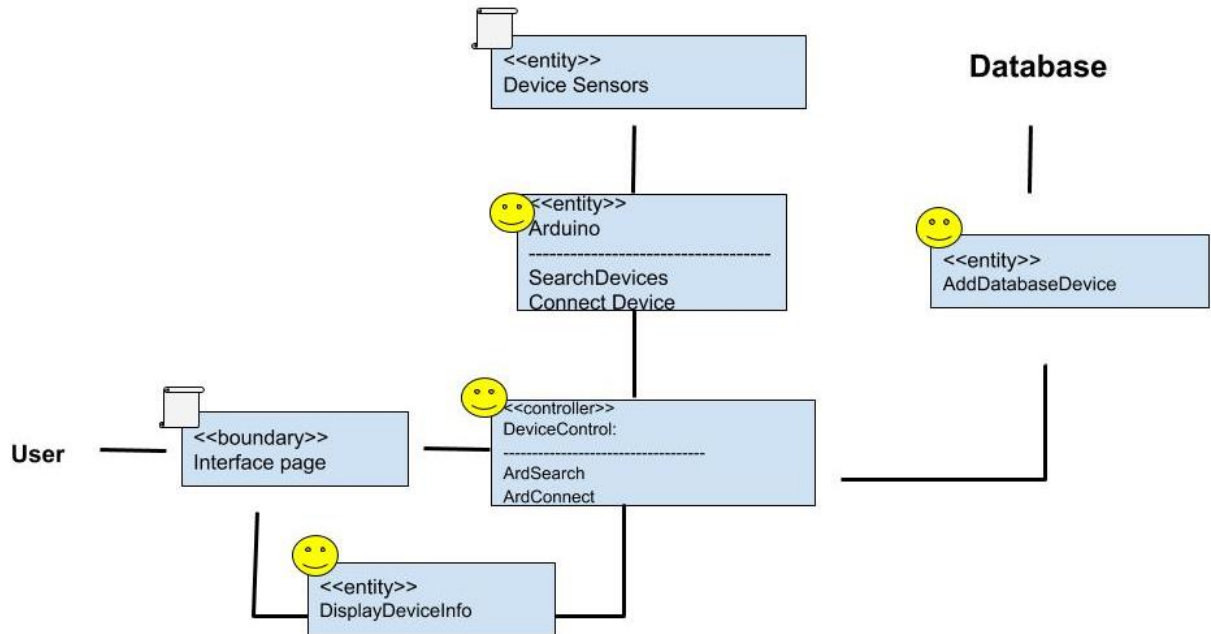


Figure 19: Domain Model for UC-8

Concept pair	Association description:	Association Name:
Interface page <-> DeviceControl	Sends request to add-device to DeviceControl. Then, if device is found, prompts user to connect device.	UserMachineInteraction
DeviceControl <-> Arduino	DeviceControl prompts Arduino to search for devices and connect to them.	DeviceManagement
Arduino <-> Device Sensors	Arduino searches for active devices near it.	Arduino Search
Arduino <-> DeviceControl	Arduino gives device data to DeviceControl after search	Show devices
DeviceControl<->AddDatabaseDevice	DeviceControl sends device data to AddDatabaseDevice; prepares it to be uploaded into Database	Convey device data

AddDatabaseDevice <-> Database	Inputs device data into Database	DatabaseManagement
DeviceControl <-> DisplayDeviceInfo	DeviceControl conveys what type of page should be sent to user (success, confirmation, error, etc.)	Write page
DisplayDeviceInfo <-> Interface page	Shows page to user	UpdateStatus

Table 14: Associations for Domain Model (UC-8)

Concept:	Attributes:	Attribute Description
Arduino:	SearchDevices: Connect Devices:	-Searches for connectable devices -Connects device in range
DeviceControl:	ArdSearch: ArdConnect:	-Commands Arduino to search for connectable devices. -Commands Arduino to connect to device in range.

Table 15: Attributes for Domain Model (UC-8)

UC-9: Device-Status Domain Model

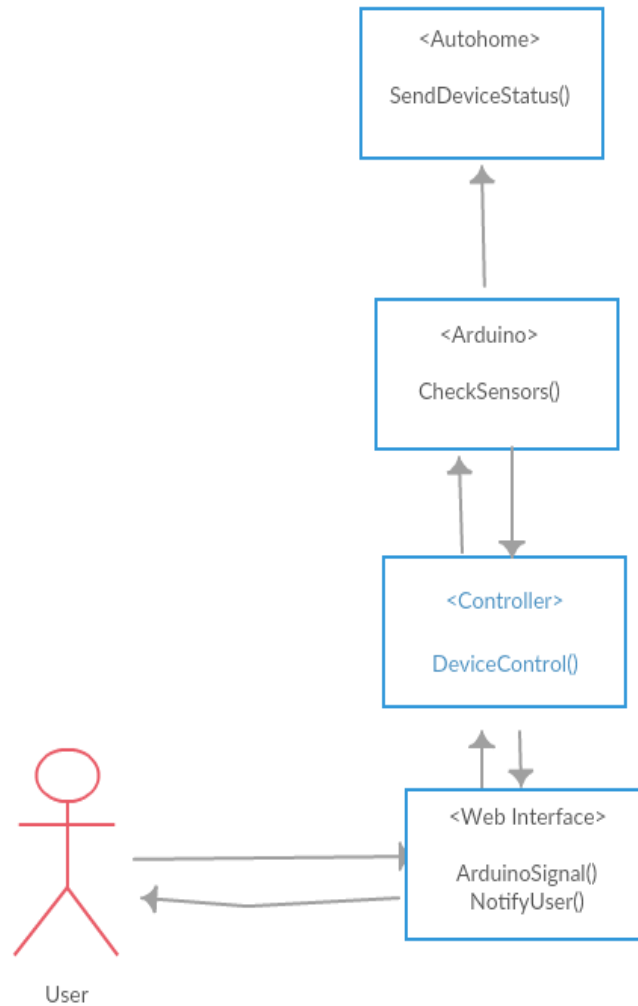


Figure 20: Domain Model for UC-9

Concept name:	Attributes	Responsibility Description:
DeviceControl	Controller	Application on user's device that connects to Arduino and sends device entry data to Database.
NotifyUser()	Web Interface	User Web interface that receives data from user and displays data from DeviceControl.

Interface page	Web Interface	Shows possible actions user can take on the User Web Interface. Shows the user which devices are currently connected and their current status
Arduino	CheckSensors() Controller	Sends data between DeviceControl to Device sensors.
Checksensors	Arduino	Arduino checks the signals sent from all the device sensors to notify the user of the device status
SendDeviceStatus	Autohome	Sends Device Sensor data to Autohome. Autohome device sensors send device status back to arduino so that the user eventually sees the current status of the devices
ArduinoSignal	Controller Arduino Web Interface	Web Interface acknowledges user's attempt to check device status and sends signal to the controller

Table 16: Attributes for Domain Model (UC-9)

Concept pair	Association description:	Association Name:
Interface page <-> DeviceControl	Sends request to check signal to DeviceControl. Then, if device is found,shows user the current status of the device	UserMachineInteraction
DeviceControl <-> Arduino	DeviceControl prompts Arduino to search for devices and connect to them.	DeviceManagement
Arduino <-> Autohome	Arduino searches for active devices near it.	Arduino Search
Arduino <-> DeviceControl	Arduino gives device data to DeviceControl after search	Show devices
DeviceControl <-> NotifyUser	DeviceControl conveys the status of every device and sends any possible error messages that are encountered.	Write page
NotifyUser <-> Interface page	Shows page to user	UpdateStatus

Table 17: Associations for Domain Model (UC-9)

System Operation Contracts

Name: Lock

Responsibilities: Log out and lock access to user's personal account

Use Case(s): UC-2

Exceptions: None

Preconditions:

- User is logged into the system

Post conditions:

- User is logged out of the system
 - Log out is logged in the system
-

Name: Unlock

Responsibilities: Log the user into their personal account

Use Case(s): UC-1

Exceptions: None

Preconditions:

- User has account in the system

Post conditions:

- User is logged into the system
 - Log in is logged in the system
-

Name: Add user

Responsibilities: Add a new user profile into the system

Use Case(s): UC-3

Exceptions: Only users with administrator privileges can do this

Preconditions:

- Username to be added is not already in the system
- Person adding user has administrator privileges

Post conditions:

- New user is added
 - Addition is logged in the system
-

Name: Remove user

Responsibilities: Remove a user profile from the system

Use Case(s): UC-4

Exceptions: Only users with administrator privileges can do this

Preconditions:

- User to be removed is not logged in
- Person adding user has administrator privileges

Post conditions:

- User is removed
 - Removal is logged in the system
-

Name: Access history

Responsibilities: Allow administrator is see status/history of devices connected to system and user history

Use Case(s): UC-5

Exceptions: Only users with administrator privileges can do this

Preconditions:

- User trying to access history has administrator privileges

Post conditions:

- History access is logged into the system
-

Name: Lock Monitor

Responsibilities: Check the status of door lock

Use Case(s): UC-6

Exceptions: None

Preconditions:

- None

Post conditions:

- Access history is updated whenever the status of the sensor is changed
-

Name: Monitor Device

Responsibilities: Monitor the status of all devices connected to the system

Use Case(s): UC-7

Exceptions: None

Preconditions:

- User is logged into the system

Post conditions:

- None
-

Name: Add device

Responsibilities: Add additional devices to monitor through the system

Use Case(s): UC-8/UC-16

Exceptions: None

Preconditions:

- None

Post conditions:

- System is updated with the new device
-

Name: Device Status

Responsibilities: Periodically update the database with its status

Use Case(s): UC-9

Exceptions: None

Preconditions:

- Device has connection with the server

Post conditions:

- Device status is logged
-

Name: Read device status

Responsibilities: Take database interface and display it to the user

Use Case(s): UC-10

Exceptions: None

Preconditions:

- Device has connection with the server

Post conditions:

- Device status is logged
-

Name: Light off

Responsibilities: Turn the lightbulb off

Use Case(s): UC-11

Exceptions: None

Preconditions:

- Controller has connection with server

Post conditions:

- Device status is logged
-

Name: Light on

Responsibilities: Turn the lightbulb on

Use Case(s): UC-12

Exceptions: None

Preconditions:

- Controller has connection with server

Post conditions:

- Device status is logged
-

Name: Music play

Responsibilities: Turn the music player on

Use Case(s): UC-13

Exceptions: None

Preconditions:

- Controller has connection with server

Post conditions:

- Device status is logged
-

Name: Music stop

Responsibilities: Turn the music player off

Use Case(s): UC-14

Exceptions: None

Preconditions:

- Controller has connection with server

Post conditions:

- Device status is logged
-

Name: Music control

Responsibilities: Control the user's music, skipping tracks, changing sources, etc.

Use Case(s): UC-13

Exceptions: None

Preconditions:

- Controller has connection with server

Post conditions:

- Device status is logged
-

Traceability Matrix: Domain Concepts

REQ	P	U	U	UC	UC	UC	UC	UC	UC	UC	UC	UC	UC	UC	UC	UC	UC	UC
	W	C1	C2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
AccountAccess	5	X	X	X	X													
AccountManagement	5			X	X				X								X	
DeviceManagement	4			X			X	X	X	X	X	X	X	X	X	X	X	X
DeviceControl	2						X	X	X			X	X	X	X	X	X	X
UserMachineInteraction	1	X	X														X	
Database Management	1			X	X	X			X		X						X	
Max	P W	4	4	4	4	1	5	4	2	4	4	5	5	5	5	2	2	2
Total	P W	6	6	15	11	1	6	6	12	4	5	6	6	6	6	6	13	6

The Gantt chart displays project milestones and tasks from February 2018 to May 2018. The chart is divided into two main sections: February to April 2018 and April to May 2018. Tasks are represented by horizontal bars of various colors (black, red, green, blue) with labels indicating the task name and the group member responsible. Key milestones include 'Report #2', 'Preparation for full project demo', 'Test project demo', and 'Full report #3'.

Task	Group Member	Start Date	End Date
Report #2	[all group members]	12/15/18	3/5/19
Report #2 part 1	[all group members]	12/15/18	3/5/19
Report #2 part 2	[all group members]	12/15/18	3/5/19
Report #2 part 3	[all group members]	12/15/18	3/5/19
Project Assembly	[all group members]	12/15/18	3/5/19
Write Arduino Code	[all group members]	12/15/18	3/5/19
Search Tokumaru, Aaron Chomko	[all group members]	12/15/18	3/5/19
Write User Interface Code	[all group members]	12/15/18	3/5/19
Search Tokumaru, Aaron Chomko	[all group members]	12/15/18	3/5/19
Align and integrate speech recognition software	[all group members]	12/15/18	3/5/19
Complete data processing, capture and storage functionalities	[all group members]	12/15/18	3/5/19
Develop and design website	[all group members]	12/15/18	3/5/19
Develop and design website	[all group members]	12/15/18	3/5/19
Establish database	[all group members]	12/15/18	3/5/19
Database Backup, Jimmy Yeh	[all group members]	12/15/18	3/5/19
Preparation for full project demo	[all group members]	3/5/19	3/5/19
Test project demo	[all group members]	3/5/19	3/5/19
Full report #3	[all group members]	3/5/19	3/5/19
Report #3	[all group members]	3/5/19	3/5/19
Report #3 part 1	[all group members]	3/5/19	3/5/19
Report #3 part 2	[all group members]	3/5/19	3/5/19
Report #3 part 3	[all group members]	3/5/19	3/5/19
Project Assembly	[all group members]	3/5/19	3/5/19
Write Arduino Code	[all group members]	3/5/19	3/5/19
Search Tokumaru, Aaron Chomko	[all group members]	3/5/19	3/5/19
Write User Interface Code	[all group members]	3/5/19	3/5/19
Search Tokumaru, Aaron Chomko	[all group members]	3/5/19	3/5/19
Align and integrate speech recognition software	[all group members]	3/5/19	3/5/19
Complete data processing, capture and storage functionalities	[all group members]	3/5/19	3/5/19
Develop and design website	[all group members]	3/5/19	3/5/19
Develop and design website	[all group members]	3/5/19	3/5/19
Establish database	[all group members]	3/5/19	3/5/19
Database Backup, Jimmy Yeh	[all group members]	3/5/19	3/5/19
Preparation for full project demo	[all group members]	3/5/19	3/5/19
Test project demo	[all group members]	3/5/19	3/5/19
Full report #3	[all group members]	3/5/19	3/5/19
Report #3	[all group members]	3/5/19	3/5/19
Report #3 part 1	[all group members]	3/5/19	3/5/19
Report #3 part 2	[all group members]	3/5/19	3/5/19
Report #3 part 3	[all group members]	3/5/19	3/5/19
Project Assembly	[all group members]	3/5/19	3/5/19
Write Arduino Code	[all group members]	3/5/19	3/5/19
Search Tokumaru, Aaron Chomko	[all group members]	3/5/19	3/5/19
Write User Interface Code	[all group members]	3/5/19	3/5/19
Search Tokumaru, Aaron Chomko	[all group members]	3/5/19	3/5/19
Align and integrate speech recognition software	[all group members]	3/5/19	3/5/19
Complete data processing, capture and storage functionalities	[all group members]	3/5/19	3/5/19
Develop and design website	[all group members]	3/5/19	3/5/19
Develop and design website	[all group members]	3/5/19	3/5/19
Establish database	[all group members]	3/5/19	3/5/19
Database Backup, Jimmy Yeh	[all group members]	3/5/19	3/5/19
Preparation for full project demo	[all group members]	3/5/19	3/5/19
Test project demo	[all group members]	3/5/19	3/5/19
Full report #3	[all group members]	3/5/19	3/5/19
Report #3	[all group members]	3/5/19	3/5/19
Report #3 part 1	[all group members]	3/5/19	3/5/19
Report #3 part 2	[all group members]	3/5/19	3/5/19
Report #3 part 3	[all group members]	3/5/19	3/5/19
Project Assembly	[all group members]	3/5/19	3/5/19
Write Arduino Code	[all group members]	3/5/19	3/5/19
Search Tokumaru, Aaron Chomko	[all group members]	3/5/19	3/5/19
Write User Interface Code	[all group members]	3/5/19	3/5/19
Search Tokumaru, Aaron Chomko	[all group members]	3/5/19	3/5/19
Align and integrate speech recognition software	[all group members]	3/5/19	3/5/19
Complete data processing, capture and storage functionalities	[all group members]	3/5/19	3/5/19
Develop and design website	[all group members]	3/5/19	3/5/19
Develop and design website	[all group members]	3/5/19	3/5/19
Establish database	[all group members]	3/5/19	3/5/19
Database Backup, Jimmy Yeh	[all group members]	3/5/19	3/5/19
Preparation for full project demo	[all group members]	3/5/19	3/5/19
Test project demo	[all group members]	3/5/19	3/5/19
Full report #3	[all group members]	3/5/19	3/5/19
Report #3	[all group members]	3/5/19	3/5/19
Report #3 part 1	[all group members]	3/5/19	3/5/19
Report			

In the next few weeks, we hope to have most functions written for the Arduino successfully, have the website up and running, have the mobile application built, setup the database properly so that users can create and manage account. We hope to have most devices

supported at that time utilizing both Bluetooth and WiFi. If possible, we'd like to implement a speech recognition program that executes commands, however the main priority is to get the application to work and properly manage (automate) home devices first.

References

- [1] Home automation - <https://www.vectorsecurity.com/UserFiles/File/PDF/blog/Smart-Home-Interactive.pdf>
- [2] Arduino microcontroller - <https://www.arduino.cc/>, <https://www.elegoo.com/download/>, <https://www.linuxjournal.com/content/learning-program-arduino>
- [3] Version control system - <https://github.com/>
- [4] Software Engineering book - http://eceweb1.rutgers.edu/~marsic/books/SE/book-SE_marsic.pdf
- [5] Android development - <https://medium.com/google-developers/developing-for-android-introduction-5345b451567c>, http://www.ebookfrenzy.com/pdf_previews/AndroidStudioEssentialsPreview.pdf
- [6] Smart home sensors - <https://www.smarthomeusa.com/plan-your-own-smart-home-system/>, <https://www.networkworld.com/article/2925722/security0/home-security-demystified-how-to-build-a-smart-diy-system.html>