Smart Home System

By: Vaishnavi, Isha, Vivien, Vy, Erica, Asfandiyar

Smart Home System

Since technology has become more and more ubiquitous, homes contain devices that homeowners can use that allow for more functionality within the system leveraging both security and convenience.

Welcome to the future of living - the Smart Home System. Why smart homes?

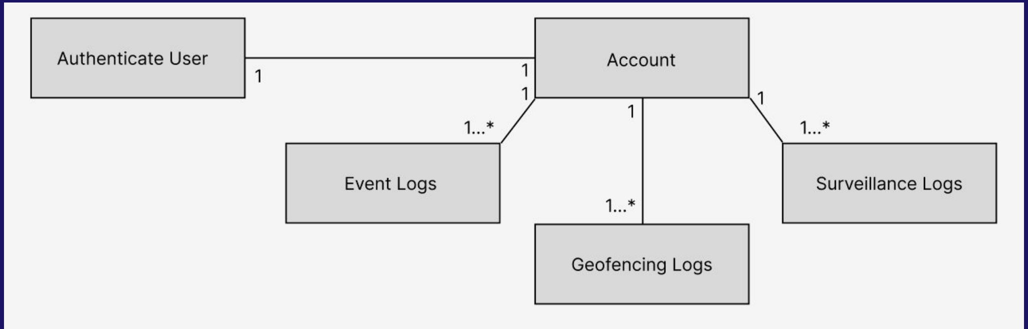
- Security: With rising concerns about home security, a Smart Home System provides advanced intrusion detection and access control, using cutting-edge technologies such as cameras, sensors, RFID tags, and more.
- Convenience: Imagine a home that adapts to your preferences, turning on lights, adjusting the thermostat, and even brewing your morning coffee as you wake up - all automatically.
- Energy Efficiency: Smart homes optimize energy consumption, reducing your carbon footprint and saving money on utility bills.

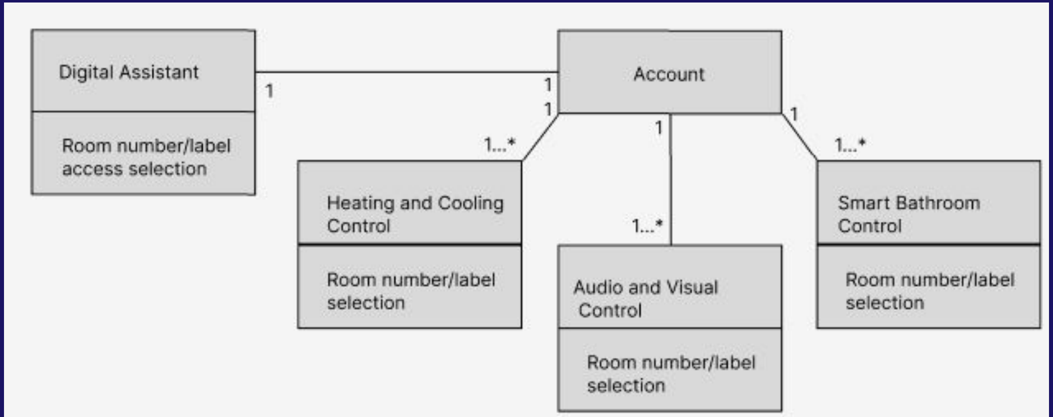
The Domain – Smart Home Technology

- Smart home technology encompasses the base ideas of innovation, convenience, and security.
- Ubiquitous Nature - multiple networks of devices and systems with the core ideal of making homeowners lives safer, more convenient, and overall easier.
- With rising concerns about home security, Smart Home System provides advanced intrusion detection and access control.
- This technology is built on the basis of providing control and monitoring abilities for homes remotely allowing for customization and personalization.

Conceptual Class Diagrams

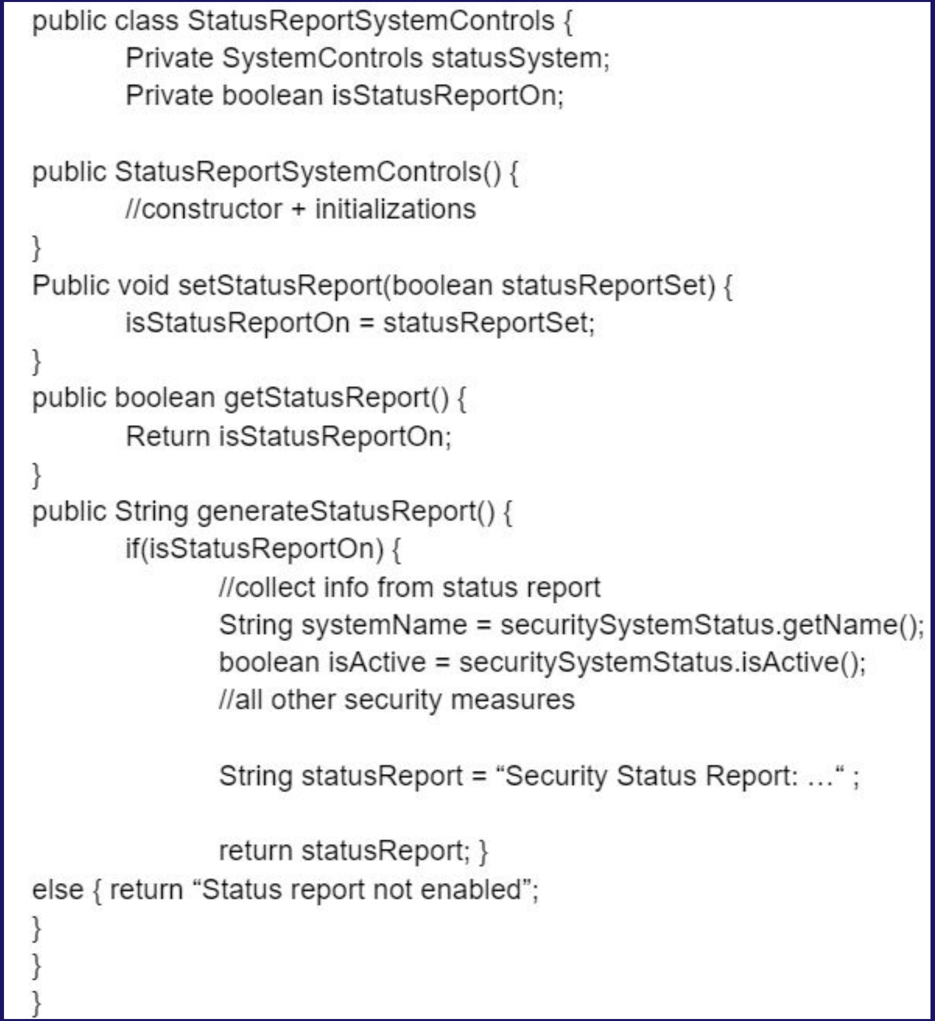
- 1st diagram is for user authentication & account logs
- 2nd diagram is of 4 key features




Digital Assistant
Room number/label access selection
1
1
Account
1
1
1...*
1
1
1...*
Heating and Cooling Control
Room number/label selection
1...*
Smart Bathroom Control
Room number/label selection
Audio and Visual Control
Room number/label selection

Pseudocode

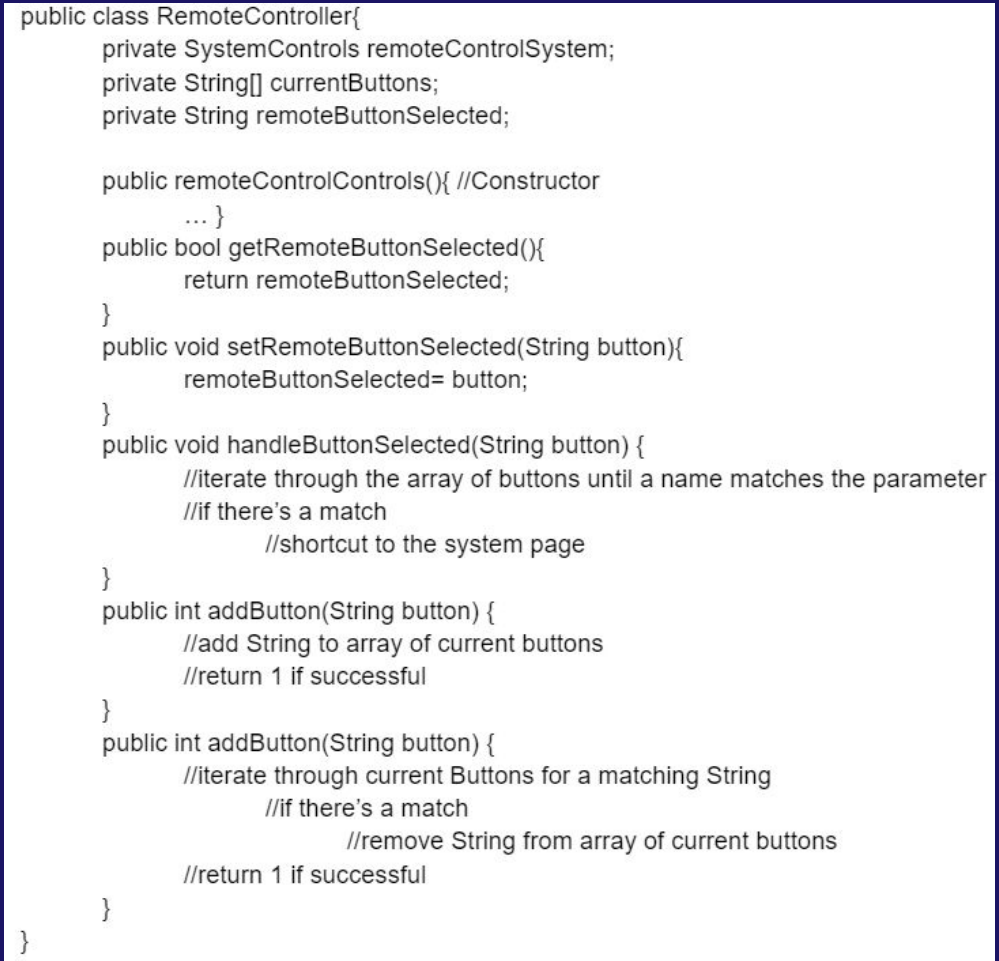
- Status report system controls class



```
public class StatusReportSystemControls {
        Private SystemControls statusSystem;
        Private boolean isStatusReportOn;

public StatusReportSystemControls() {
        //constructor + initializations
}
Public void setStatusReport(boolean statusReportSet) {
        isStatusReportOn = statusReportSet;
}
public boolean getStatusReport() {
        Return isStatusReportOn;
}
public String generateStatusReport() {
        if(isStatusReportOn) {
                //collect info from status report
                String systemName = securitySystemStatus.getName();
                boolean isActive = securitySystemStatus.isActive();
                //all other security measures

                String statusReport = “Security Status Report: …“ ;

                return statusReport; }
else { return “Status report not enabled”;
}
}
}
```

Pseudocode

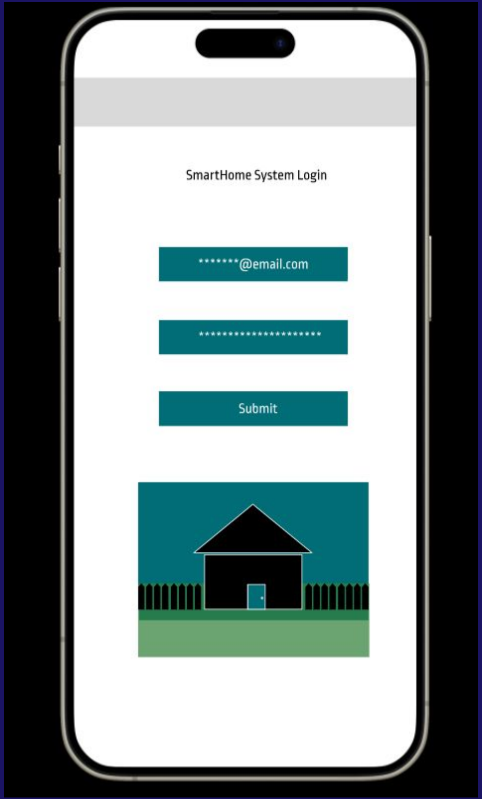
- Remote controller class



```
public class RemoteController{
    private SystemControls remoteControlSystem;
    private String[] currentButtons;
    private String remoteButtonSelected;

    public remoteControlControls(){ //Constructor
        … }
    public bool getRemoteButtonSelected(){
        return remoteButtonSelected;
    }
    public void setRemoteButtonSelected(String button){
        remoteButtonSelected= button;
    }
    public void handleButtonSelected(String button) {
        //iterate through the array of buttons until a name matches the parameter
        //if there's a match
            //shortcut to the system page
    }
    public int addButton(String button) {
        //add String to array of current buttons
        //return 1 if successful
    }
    public int addButton(String button) {
        //iterate through current Buttons for a matching String
            //if there's a match
                //remove String from array of current buttons
        //return 1 if successful
    }
}
```

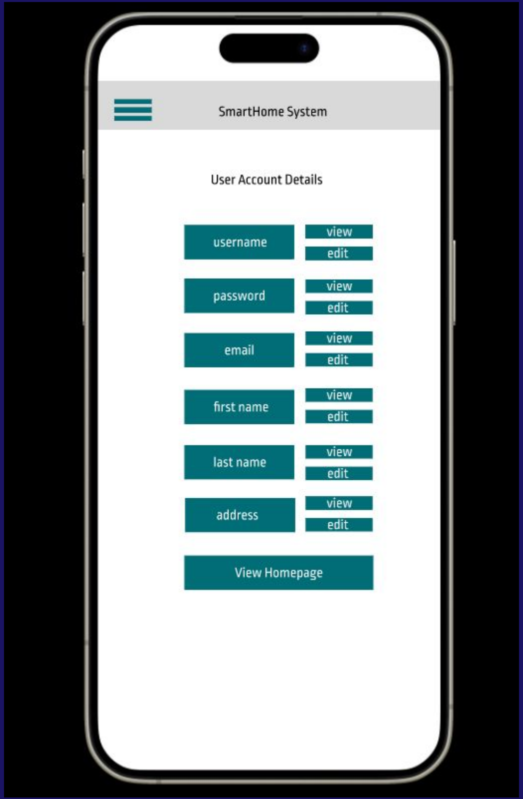
UI Design

Logging in and user account details


SmartHome System Login

*******@email.com

****

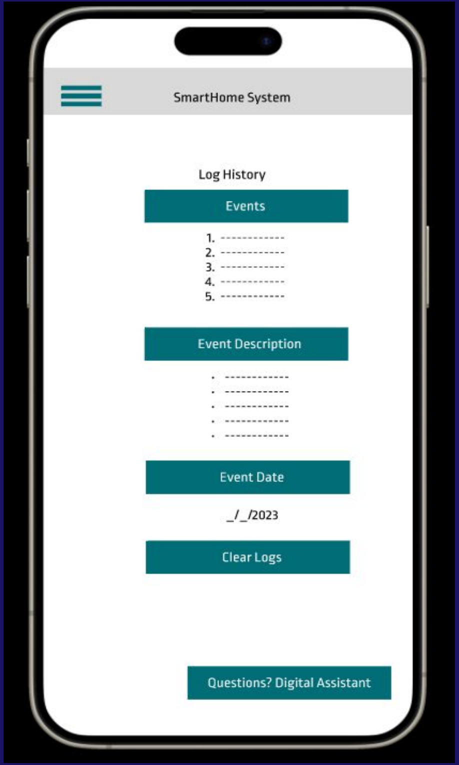


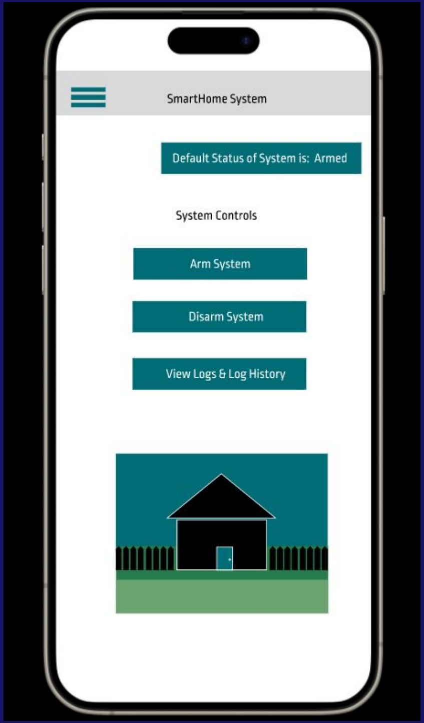
Important Key Features

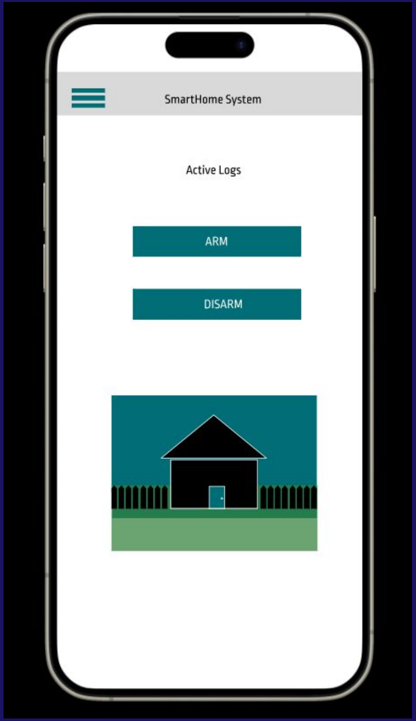
Key Features

Event Logging

Users are able to navigate through the system to view past alerts, logs of specific security events such as system status (arming and disarming of system).



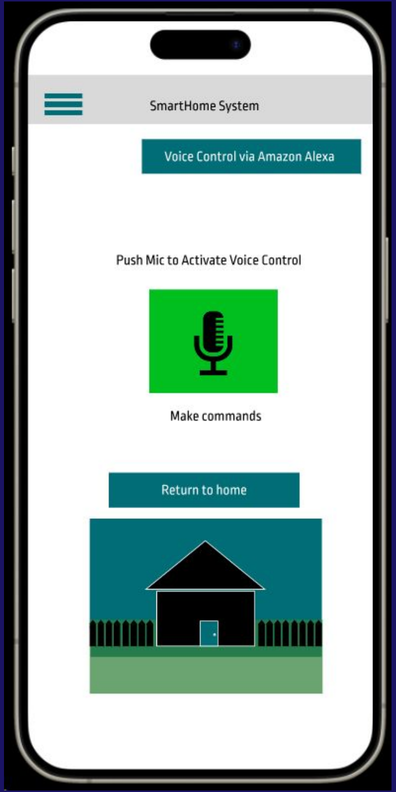




Key Features

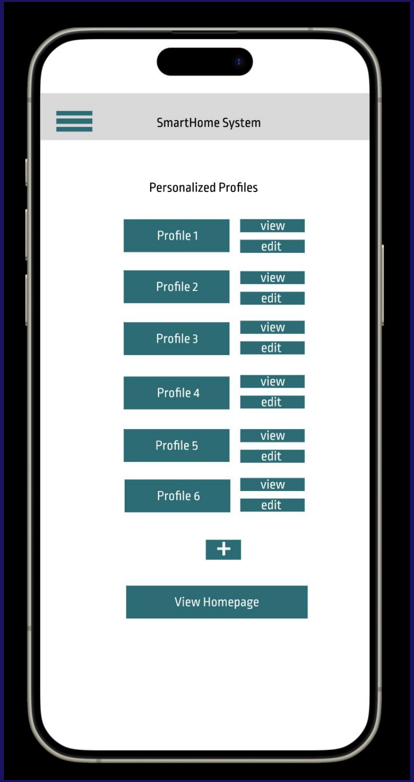
Voice Control

Users can utilize voice commands to control various aspects of their smart home system like lighting, temperature, and security.


SmartHome System
Voice Control via Amazon Alexa
Push Mic to Activate Voice Control
Make commands
Return to home

Personalized Profiles

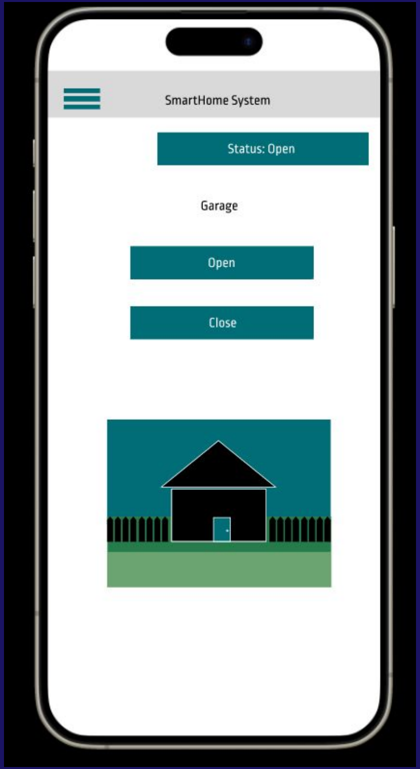
Users have the ability to create multiple profiles that fit their needs and preferences when it comes to managing their Smart Home System.



Key Features

Garage Control

Users are able to view the status of their garage: open, closed, opening, closing and are able to control these features in the app.



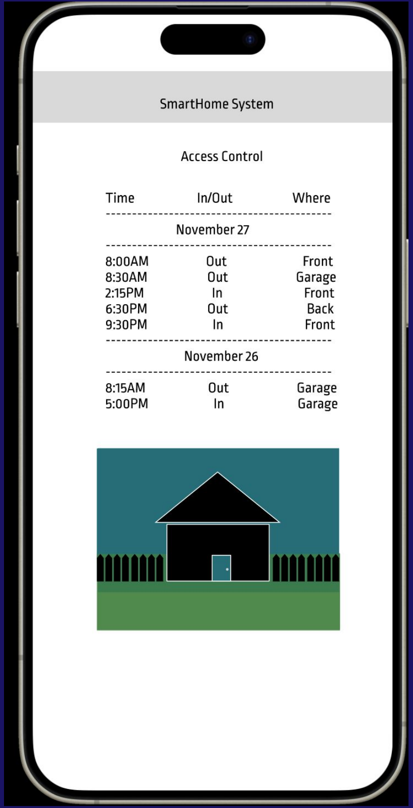
SmartHome System

Status: Open

Garage

Access Control

Users can identify who has entered or exited the property based on installed sensors and surveillance of the system.



SmartHome System

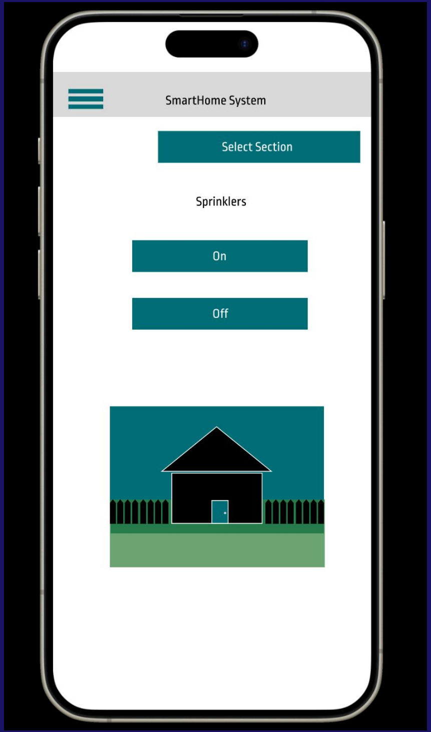
Access Control

Time	In/Out	Where
November 27		
8:00AM	Out	Front
8:30AM	Out	Garage
2:15PM	In	Front
6:30PM	Out	Back
9:30PM	In	Front
November 26		
8:15AM	Out	Garage
5:00PM	In	Garage

Key Features

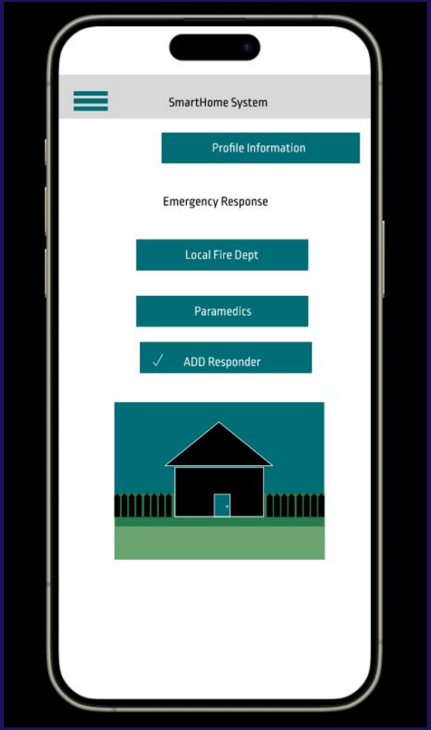
Smart Gardening

Users are able to turn off/on sprinklers to control their lawn care needs.



Emergency Response

Users can add multiple emergency contacts and have the option to share their location data with first responders.



Major Challenges

- Figuring out what application/platform to use to build models such as our ER Diagrams, Class Diagrams, and etc.
- Overload of the components that made up the System Controls Class
- Difficulty producing the cardinality and relationships between the tables representative of these components (System, Logs, SystemControl)
- Determining the complexity and scalability of our Smart Home System
- Authorization & Access of both Real-time and Past Event Log Data
- Understanding how might the connection of the Smart Home System look like for different homes

Solutions

- Choose Figma for the creation of our Smart Home System diagrams ultimately because of its user-friendly interface.
- We decided to separate the components of our System Controls class based on the control of the component and section that would fall under the home.
- We fixed the cardinality relationships between the tables by considering many to many relationships and keys that could be added to connect these relationships.
    - Company, System, and SystemControl
- To determine the scalability of our system, we had to consider how we wanted to setup the core functionality our system and consider what components users would utilize the most.
- To ensure proper authorization and access, we had to consider only users that were authenticated are able to view real-time and historical data.
- The Smart Home System can be setup up differently for different homes, due to user preferences regarding the functionalities that matter to them.

Lessons Learned For Success

- Learned the importance of fully dressed use cases, use case diagrams, class diagrams, ER diagrams, tables, general UI design, and algorithm designs
- Use cases and diagrams helped us understand the many reliant interconnections of our system specifications and also allowed us to build our system incrementally
- Progressively adding to GitHub helped us succeed by staying on track
- Team communication helped us divide work and successfully meet project deadlines
- Ultimately, we created a reliable application to manage a home efficiently using software development practices learned in class and applied in the real world

Thank you!

Questions?

github link: https://github.com/AsfandiyarK/CS3704Project-SmartHomeSystem.git