California State University, Long Beach
College of Engineering
CECS 490B

SafeSense

A Room Occupancy Detection System

Final Group Report

Team SafeSense

Jianni Cariaga | Reed Ellison | Khang Tran

May 8, 2020

Table of Contents

Section 1: Introduction

1.1 Mission Statement

Inspired by the invaluable services that firefighters and other first responders perform to ensure the safety of individuals in danger, our mission is to develop an effective tool for first responders that will help reduce the overall time spent during indoor emergency protocols. Whether there is a fire evacuation or a search and rescue, **nobody gets left behind**.

1.2 Team SafeSense

Jianni Cariaga

I am a fifth-year undergraduate computer engineering major with a minor in web technologies and applications. I currently work part-time as an experience office assistant at the CSULB College of Engineering. My technical interests include IoT and UX/UI design.

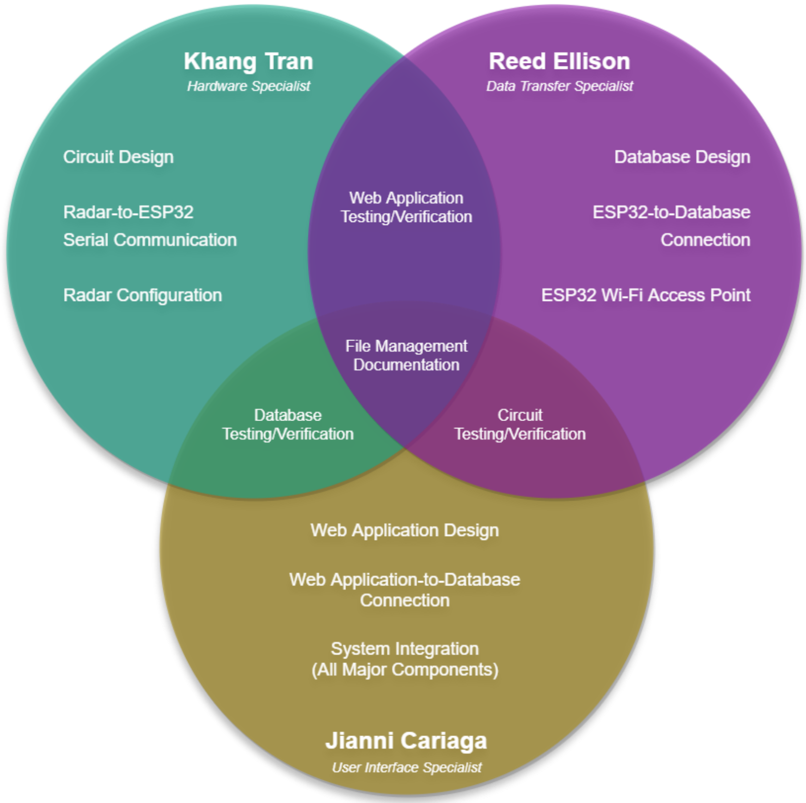
Reed Ellison

I am a fourth-year undergraduate computer engineering major with a minor in computer science. I currently work part-time as a software engineering intern at Siemens. My technical interests include software engineering an artificial intelligence.

Khang Tran

I am a senior-level undergraduate international transfer student majoring in computer engineering with a minor in computer science. My technical interests include embedded programming and field programmable gate arrays.

1.3 Team Roles and Contributions



Khang Tran performed the role of *Hardware Specialist* where his contributions focused primarily on integrating the hardware components of SafeSense. He is responsible for building the SafeSense device's hardware circuit and programming the ESP32 microcontroller to communicate serially with the XENSIV™ radar via UART.

Reed Ellison performed the role of *Data Transfer Specialist* where his contributions focused primarily on exchanging data between the ESP32 microcontroller and database. He is responsible for setting up the ESP32 microcontroller's built-in Wi-Fi access point capabilities, posting sensor data to the database, and getting user input data from the database.

Jianni Cariaga performed the role of *User Interface Specialist* where his contributions focused primarily on designing and building the SafeSense web application. He is responsible for establishing SafeSense's visual identity and integrating the major components of SafeSense (hardware, database, and web application) at a system level.

All team members shared the responsibility of testing and verifying the functionality of each other's unique tasks. In addition, all team members routinely organized and maintained their shared file repository on Google Drive and contributed to all SafeSense documentation including weekly reports, group reports, and demo presentations.

Section 2: SafeSense Overview

2.1 Technical Summary

SafeSense is a wireless network of user-configurable devices that utilizes cutting-edge radar technology to detect whether a room is occupied. Its primary applications are a first responder tool and a discreet surveillance system. The major components of SafeSense include a XENSIV™ radar, ESP32 microcontroller, MySQL database, and web application.

Users can monitor room occupancy via web application and configure individual SafeSense devices to accommodate a variety of indoor layouts. Device configurations include Maximum Range, Sensitivity, and Show Activity. Devices can be mounted on walls or ceilings to cover entire spaces or points of interest due to its compact size and low power consumption.

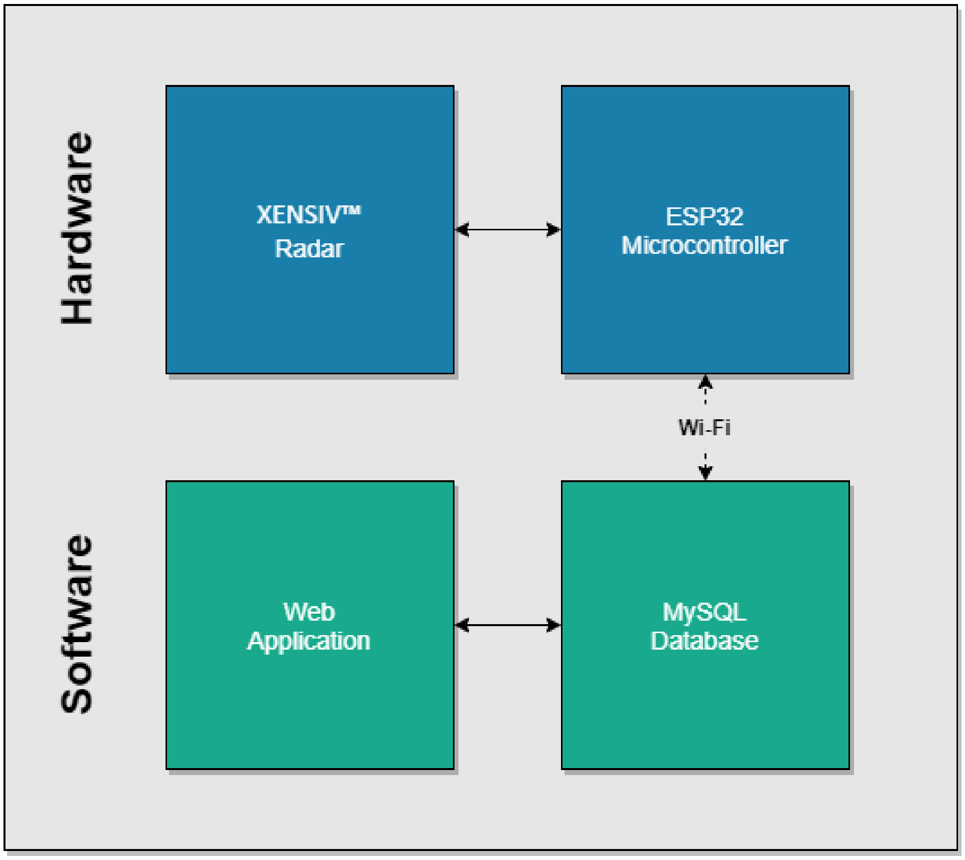
2.2 Motivation

SafeSense was inspired by the invaluable services that firefighters and other first responders perform to ensure the safety of individuals in immediate danger. However, their jobs also put them at risk.

Team SafeSense aspires to develop an effective tool for first responders to reduce the overall time spent during indoor emergency protocols.

When there is no indoor emergency taking place, SafeSense can be repurposed as a discreet surveillance system for home or business owners who are concerned about potential intruders or suspicious activity. At Team SafeSense, individual safety is our number one priority –first responders included.

2.3 System-Level Block Diagram



The block diagram above illustrates the major components of SafeSense at a system level. The major components include a XENSIV™ radar, ESP32 microcontroller, MySQL database, and web application. The system is divided into two sections: hardware and software.

The hardware section of the block diagram consists of a XENSIV™ radar and an ESP32 microcontroller. Both components are directly connected to exchange sensor data and the user's settings. The ESP32 microcontroller is connected to the MySQL database via Wi-Fi.

The software section of the block diagram consists of a MySQL database and web application. Both components communicate via HTTP GET and POST methods. The MySQL database sends, retrieves, and stores information between the hardware and web application.

Section 3: Hardware

3.1 XENSIV™ Radar

The XENSIV™ radar by Infineon Technologies utilizes their cutting-edge 60 GHz radar chip technology to provide innovative sensing capabilities to many applications. Along with its small form factor and low power consumption, the radar can detect human presence via RF signals ranging from 0.2 m to 5 m at a 90° horizontal and 80° vertical FOV.

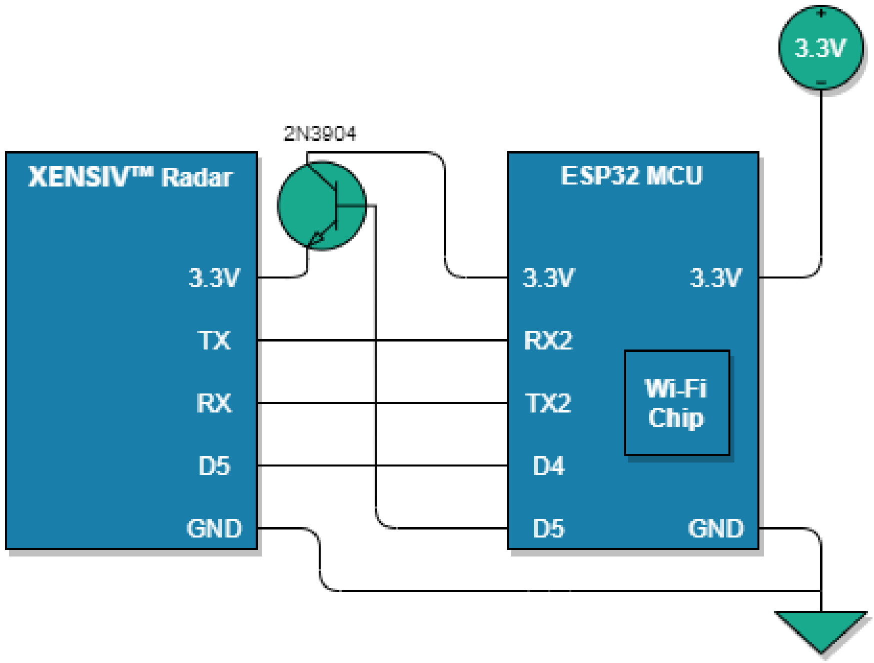
For more information about the XENSIV™ Radar, click here.

3.2 ESP32 Microcontroller

The ESP32 microcontroller by Espressif Systems utilizes a low-power, low-cost SOPC designed for IoT applications. With a reliable operating temperature ranging from –40°C to 125°C, the ESP32 microcontroller is ideal for most environments. In addition, the ESP32 microcontroller has an integrated Wi-Fi and Bluetooth chip allowing for wireless interfacing.

For more information about the ESP32 microcontroller, click here.

3.4 Hardware Schematic



The schematic above illustrates the detailed interconnections between the major hardware components of SafeSense. The ESP32 microcontroller is powered by a 3 AA battery holder, and the XENSIV™ radar is powered by the microcontroller's 3.3V output pin. The 2N3904 NPN transistor bridges the radar's power supply to compensate for voltage drop.

The XENSIV™ radar transmits sensor data directly from its D5 pin to the ESP32 microcontroller's D4 pin when human presence is detected. The ESP32 microcontroller then binds the sensor data to a MySQL INSERT statement and executes the statement in the database via Wi-Fi using its built in Wi-Fi chip.

The ESP32 microcontroller also receives device settings from the database and transmits them to the XENSIV™ radar via UART. The XENSIV™ radar receives these device settings and applies them immediately. After, the XENSIV™ radar sends an "OK" message back to the ESP32 microcontroller when new settings are applied (seen only on a serial monitor).

Both the XENSIV™ radar and ESP32 microcontroller are connected to a common ground. The ESP32 microcontroller checks for sensor data from the XENSIV™ radar at 1 second intervals. Additionally, the ESP32 microcontroller retrieves device settings from the database at 1 second intervals. This timing is fixed to prevent the ESP32 microcontroller from overheating.

To see the SafeSense device's source code, click here.

3.3 Bill of Materials

Item Name	Company	Vendor	Quantity	Cost
XENSIV™ Radar	Infineon Technologies	digikey.com	1	$286.06 *
ESP32 Microcontroller	Espressif Systems	mouser.com	1	$10.00
2N3904 NPN Transistor	ST Micro	sparkfun.com	1	$0.50
3 AA Battery Holder	Eagle Plastic Devices	mouser.com	1	$1.51
			Estimated Total:	$297.07

* Position2Go Radar (Infineon Technologies)

The table above illustrates the cost of each major component in the hardware schematic and the estimated total cost of SafeSense per unit. Note, the XENSIV™ radar has not yet been released to the public, and therefore does not have a determined price. Infineon Technologies' Position2Go radar was used as a placeholder based on functional similarities.

Section 4: Software

4.1 MySQL Database

MySQL is a freely available open source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL). SQL is the most popular language for adding, accessing and managing content in a database. It is most noted for its quick processing, proven reliability, ease and flexibility of use.

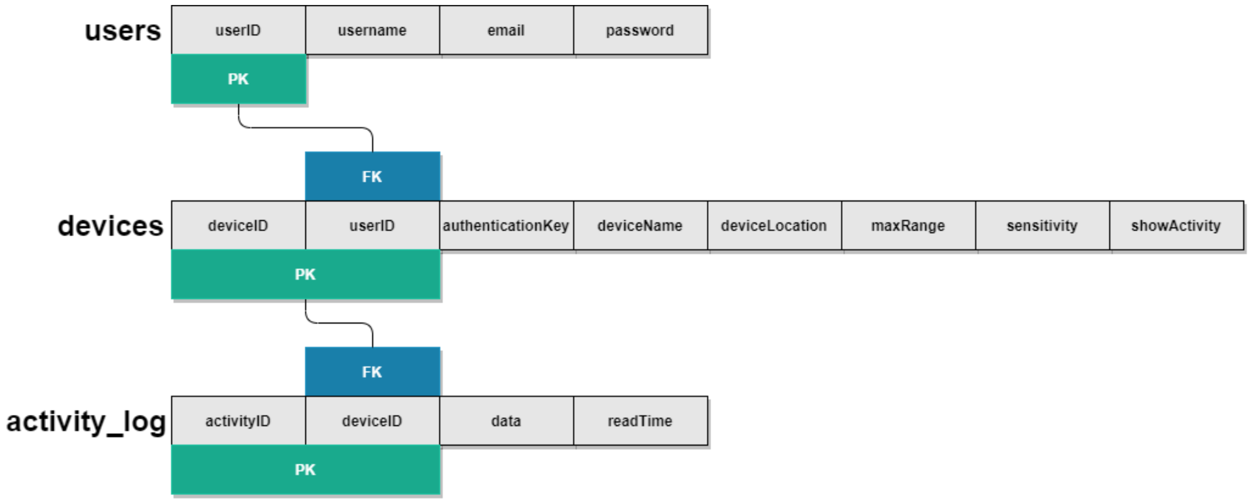
For more information about MySQL databases, click here.

4.2 Web Application

The SafeSense web application is designed to allow the user to create an account, register a SafeSense device, view sensor data, configure device settings, and configure account settings. The web application was written in the Brackets text editor and tested on the XAMPP local server. The web application is currently deployed on SiteGround.

To see the SafeSense web application, click here.

4.3 Database Relational Schema



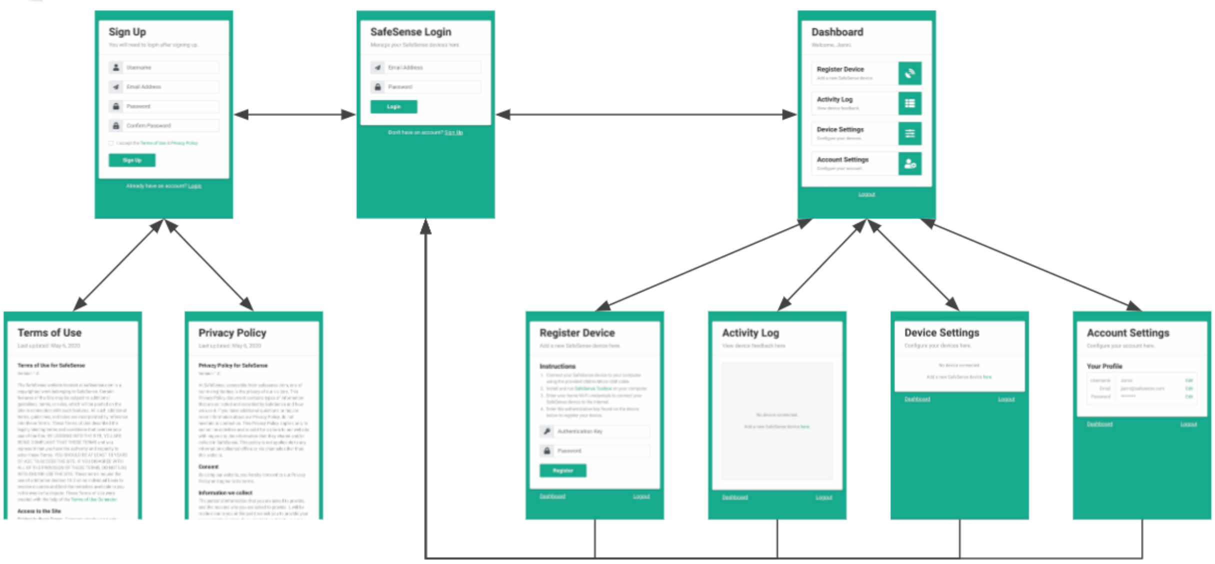
The relational schema above illustrates the SafeSense database's tables, columns, and their relationships. There are three tables: users, devices, and activity_log. Each user can have 0 to many device entries, and each device can have 0 and many activity_log entries. All columns in each table are hashed to protect the user's sensitive information.

In the users table, all users are uniquely identified by their userID which is auto incremented by 1 per user registered. The table stores each user's username, email, and password. The email column is indexed as unique, so duplicate entries of the same email address cannot exist in the table. However, duplicate usernames and passwords may exist.

In the devices table, all devices are uniquely identified by their deviceID which is auto incremented by 1 per device connected. The table stores the device's owner, unique authentication key, and default or current device name, location, and settings. The userID column is set to NULL until a user registers a device by entering its authentication key.

In the activity_log table, all sensor data is uniquely identified by their activityID which is auto incremented by 1 per sensor data posted. The table stores each device's sensor data information (activity) and timestamp (readTime). The deviceID column is automatically indexed to its corresponding registered device's ID from the devices table.

4.4 UI Flowchart



The flowchart above illustrates the SafeSense web application's user interface and navigation. The web application consists of 29 files and roughly 3,600 lines of code written in a combination of HTML, CSS, JavaScript, and PHP. The web application is designed to be compatible with all resolution sizes including those of smartphones and tablets.

The front-end aspect of the SafeSense web application utilizes HTML and CSS to structure and style all content layout. Each web application page has its own unique HTML, and all pages point to a single CSS file. JavaScript was used for dynamic input validation by manipulating CSS for specific HTML elements such as input requirements and menu popups.

The back-end aspect of the web application utilizes PHP to get and display data from the database or post and store data to the database. Most pages have one to several PHP scripts that run when a form is submitted. These scripts bind the user's input data to a MySQL statement which is sent and executed in the database.

All code was written on the Brackets text editor and tested on a XAMPP local web server. The final version of the SafeSense web application was uploaded to the SafeSense's SiteGround web hosting account's file directory for integration and testing with the database and SafeSense devices.

To see the SafeSense web application source code, click here.

Section 5: Project Management

5.1 Fall 2019 Project Timeline

	Task	Complete	Category	Team Member(s)	Start Date	Deadline
1	Contact Infineon Technologies	✔	Hardware	Jianni Cariaga	10/28/2019	11/5/2019
2	Order Recommended Radar	✔	Hardware	Jianni Cariaga	10/31/2019	11/7/2019
3	Study Radar Documentation	✔	Hardware	Khang Tran	11/6/2019	11/12/2019
4	Research Microcontrollers	✔	Hardware	Reed Ellison	10/28/2019	11/12/2019
5	Order Microcontrollers	✔	Hardware	Reed Ellison	11/12/2019	11/19/2019
6	Study Microcontroller Documentation	✔	Hardware	Reed Ellison, Khang Tran	11/14/2019	11/21/2019
7	Research UI Options	✔	Software	Jianni Cariaga	10/18/2019	11/21/2019
8	Research Database Options	✔	Software	Jianni Cariaga, Reed Ellison	11/14/2019	11/26/2019
9	Study UI/Database Documentation	✔	Software	Reed Ellison	11/26/2019	11/30/2019
10	Organize All Documentation	✔	Administrative	Jianni Cariaga	11/30/2019	12/5/2019
11	Build Early Prototype		Hardware	Reed Ellison, Khang Tran	11/30/2019	12/10/2019
12	Test Early Prototype		Hardware	Reed Ellison, Khang Tran	12/10/2019	12/15/2019
13	Complete Final Documentation	✔	Administrative	Jianni Cariaga	11/30/2019	12/15/2019

The list above illustrates the SafeSense project timeline for the fall 2019 semester. The fall 2019 semester was the planning phase of SafeSense where team roles and contributions focused primarily on research. Reed Ellison and Khang Tran researched hardware components while Jianni Cariaga researched software components while completing administrative tasks.

Team SafeSense reached out to Infineon Technologies by Professor Cregg's referral. The company provided the team several XENSIV™ radars to test to see if they suit SafeSense's needs. Once the team decided to integrate the radars into SafeSense's overall design, Reed Ellison began research on microcontrollers that can interface with the radars.

After two weeks of research, Reed Ellison proposed that the team should utilize the ESP32 microcontroller due to its serial communication features and Wi-Fi capabilities. During this time, Jianni Cariaga began research on how to implement a user interface for SafeSense. Due to time constraints, the team decided to implement a web application interface.

One all major components were decided, Reed Ellison and Khang Tran attempted to build an early prototype of the hardware system, but they were unsuccessful due to time constraints. Jianni Cariaga organized the team's file management system and completed their final documentation for the planning phase of SafeSense.

5.2 Spring 2020 Project Timeline

	Task	Complete	Category	Team Member(s)	Start Date	Deadline
1	Build Circuit	✔	Hardware	Khang Tran	2/3/2020	2/7/2020
2	Test Circuit	✔	Hardware	Jianni Cariaga, Reed Ellison	2/10/2020	2/14/2020
3	Program ESP32 (Serial Communication)	✔	Hardware	Khang Tran	2/10/2020	2/14/2020
4	Test Serial Communication	✔	Hardware	Jianni Cariaga, Khang Tran	2/17/2020	2/21/2020
5	Program ESP32 (Wi-Fi Access Point)	✔	Hardware	Reed Ellison	2/10/2020	2/14/2020
6	Test Wi-Fi Access Point	✔	Hardware	Jianni Cariaga, Khang Tran	2/24/2020	2/28/2020
7	Build Database	✔	Software	Reed Ellison	2/24/2020	2/28/2020
8	Test Database	✔	Software	Jianni Cariaga, Khang Tran	3/2/2020	3/6/2020
9	Connect Hardware and Database	✔	HW + SW	Reed Ellison	3/9/2020	3/20/2020
10	Test Hardware and Database Connection	✔	HW + SW	Jianni Cariaga, Khang Tran	3/23/2020	3/27/2020
11	Build Web App	✔	Software	Jianni Cariaga	2/3/2020	3/27/2020
12	Test Web App	✔	Software	Reed Ellison, Khang Tran	3/30/2020	4/3/2020
13	Connect Database and Web App	✔	Software	Jianni Cariaga	4/6/2020	4/10/2020
14	Test Database and Web App Connection	✔	Software	Reed Ellison, Khang Tran	4/13/2020	4/17/2020
15	Full System Test	✔	HW + SW	All Team Members	4/20/2020	5/1/2020
16	Organize All Documentation	✔	Administrative	Jianni Cariaga	4/27/2020	5/1/2020
17	Complete Final Documentation	✔	Administrative	Jianni Cariaga	4/27/2020	5/8/2020

The list above illustrates the SafeSense project timeline for the spring 2020 semester. The spring 2020 semester was the building phase of SafeSense where team roles and contributions focused primarily on designing and building SafeSense's major components (hardware, database, and web application) in parallel before full-system integration.

Khang Tran's project tasks involved SafeSense's hardware where he designed and built the final circuit. He also wrote the embedded code to implement serial communication between the XENSIV™ radar and ESP32 microcontroller. Jianni Cariaga and Reed Ellison were tasked to test and verify the functionality of the SafeSense device's serial communication.

Reed Ellison's project tasks involved wireless communication and databases where he wrote the embedded code to utilize the ESP32 microcontroller's built-in Wi-Fi chip. He also designed and built the SafeSense database. Jianni Cariaga and Khang Tran were tasked to test and verify communication between the SafeSense device and database.

Jianni Cariaga's project tasks involved designing and building the SafeSense web application. He also worked on Team SafeSense's administrative tasks including all documentation and email exchanges with Infineon Technologies. Reed Ellison and Khang Tran were tasked to verify communication between the database and SafeSense web application.

5.3 Project Tools

The various tools involved during the planning and building phases of SafeSense include administrative tools and technical tools. The administrative tools below helped Team SafeSense organize shared files, create documentation, and communicate. The technical tools below helped Team SafeSense design, build, and test the major components of SafeSense.

<table>
<tr><th>Administrative Tools</th><th>Technical Tools</th></tr>
<tr><td>Task Management
Wrike (Fall 2019)
Airtable (Spring 2020)</td><td>Hardware
Infineon Toolbox (Radar SDK)</td></tr>
<tr><td>File Management
Google Drive</td><td>Software
Arduino IDE
Brackets Text Editor
XAMPP (phpMyAdmin)</td></tr>
<tr><td>Documentation
Draw.io
Google Docs
Google Slides
Microsoft Word</td><td>Web Hosting
SiteGround (phpMyAdmin)</td></tr>
<tr><td>Communication
Discord
Skype
Zoom</td><td></td></tr>
</table>

The Infineon Toolbox was recommended by Infineon Technologies to help Team SafeSense become more familiar with their XENSIV™ radar technology. The Radar SDK provided live sensor data and a variety of device settings to use in the SafeSense web application. Team SafeSense ultimately chose to implement the maximum range, sensitivity, and status settings.

The Arduino IDE was utilized to write the ESP32 microcontroller code. Additionally, the Brackets Text Editor and XAMPP were utilized to design and test the SafeSense application locally before deploying on SiteGround. The database was also created on SiteGround using the phpMyAdmin tool.

5.4 Technical Challenges

There were a few technical challenges that have presented themselves during the development of SafeSense. These challenges include implementing the serial communication between the XENSIV™ radar and ESP32 microcontroller and writing the SafeSense web application's back-end code.

Implementing serial communication between the XENSIV™ radar and ESP32 microcontroller was a challenge because we had limited documentation for the XENSIV™ radar. The radars we were provided by Infineon Technologies have not been released to the public, so we did not have access to its datasheets and other documentation.

However, we were able to resolve this challenge by contacting Infineon Technologies. Team SafeSense scheduled Skype meetings and exchanged emails with contacts from Infineon Technologies discussing what they needed to meet their project requirements. The contacts sent a custom-configured XENSIV™ radar to implement serial communication.

Developing the SafeSense application has undergone a transition in programming languages and tools throughout the spring 2020 semester. Initially, Jianni Cariaga was going to program an iOS app in Swift using the Firebase database, but then realized that there was not enough time to learn these new skills and complete the SafeSense web application.

To resolve this issue, Jianni Cariaga decided to use what he already knew: front-end web development in HTML, CSS, and JavaScript. He spent his free time learning PHP so he can write the back-end code of SafeSense which was more manageable to learn than iOS development. In the end, the SafeSense application became a web application instead of an iOS application.

Appendix

A.1 References

Arduino Language Reference
https://www.arduino.cc/reference/en/

BootstrapCDN Reference
https://getbootstrap.com/docs/4.4/getting-started/introduction/

ESP32 Microcontroller Overview
https://www.espressif.com/en/products/hardware/esp32/overview

Flaticon (SafeSense Logo)
https://www.flaticon.com/free-icon/radar_439941?term=radar

Font Awesome (Web App Icons)
https://fontawesome.com/how-to-use/on-the-web/referencing-icons/

How to Create a Login System in PHP
https://www.youtube.com/watch?v=LC9GaXkdxF8

How to Create an ESP32 Wi-Fi Access Point
https://randomnerdtutorials.com/esp32-access-point-ap-web-server/

How to Insert ESP32 Data into a Database
https://randomnerdtutorials.com/esp32-esp8266-mysql-database-php/

JavaScript Language Reference
https://www.w3schools.com/jsref/

jQuery AJAX Methods
https://www.w3schools.com/jquery/jquery_ref_ajax.asp

MySQL Connector for Arduino Projects
https://github.com/ChuckBell/MySQL_Connector_Arduino

PHP and MySQL Language Reference
https://www.w3schools.com/php/default.asp

SiteGround (Web Hosting Service)
https://www.siteground.com/

XENSIV™ 60 GHz Radar Overview
https://www.infineon.com/cms/en/product/promopages/60GHz/

A.2 SafeSense Source Code

To see the SafeSense GitHub repository, click here.