VIDEO DOORBELL

By:
Trey Brown
Michael Cheng
Jesse Jento
Brittany Marietta

California University of Pennsylvania
CSC 490: Senior Project I Design Document

Instructor Comments/Evaluation

Table of Contents

Abstract

The purpose of this document is to outline, in detail, the design of our final product. We will specify how our hardware will be set up and how it will communicate with the software. In addition to the hardware, we will be outlining all the details of our software and how we will bring it to fruition. This document will be used as a guide for the rest of the development process. This document will need to be referenced many times in the future, so it must be as specific and final as possible.

Description of Document

Purpose and Use

The purpose of this document is to provide the team with a detailed outline of the product that they can follow for the rest of development. The document must specify all of the classes that will be used and how they will interact. It must also define what will be used to develop the software (Language, Development software, etc.).

Ties to the Specification Document

This document seeks to correct mistakes made with the Specification Document. We will then build upon those improvements and make them more detailed. We will decide upon our programming language and the structure of the code.

Intended Audience

The intended audience of this document is mostly developers and instructors. The document’s main purpose is to aid the developers along the way throughout development. Therefore, not many people other than the team and the instructor will get much use out of it.

Project Blocking Diagram with Description

Our product will require 3 parts. These 3 parts include: a front-end application portion, a back-end server portion, and the physical Video Doorbell hardware portion. The hardware portion will send an initial signal (button pressed) to the server requesting to send a video-feed to the front-end portion. The front-end application will communicate with the server in several ways. The first way is that it will receive requests from the back-end portion to receive the video-feed and voice-feed. The application will then transmit a response back to the server either accepting or denying the video-feed and voice-feed request. Assuming the application accepts the video-feed the back-end portion then transmits the video-feed to the front-end portion. The back-end server gets the video-feed from the Hardware portion and stores it once the video-feed is completed. The front-end portion will then use the devices microphone to transmit a voice-feed to the hardware portion. This allows for an open line of communication between the front-end portion and the hardware portion.

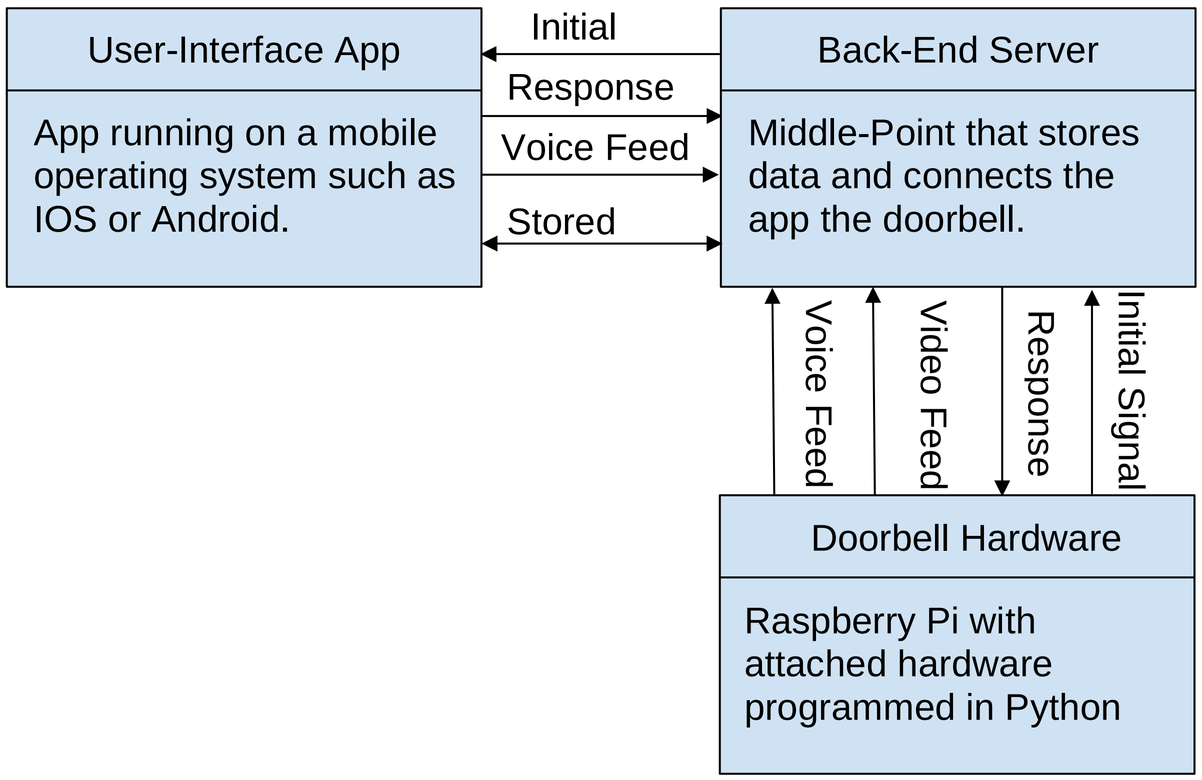
User-Interface App
App running on a mobile operating system such as IOS or Android.
Initial
Response
Voice Feed
Stored
Back-End Server
Middle-Point that stores data and connects the app the doorbell.
Voice Feed
Video Feed
Response
Initial Signal
Doorbell Hardware
Raspberry Pi with attached hardware programmed in Python

Figure 1: Block Diagram

Design Details

System Modules and Responsibilities

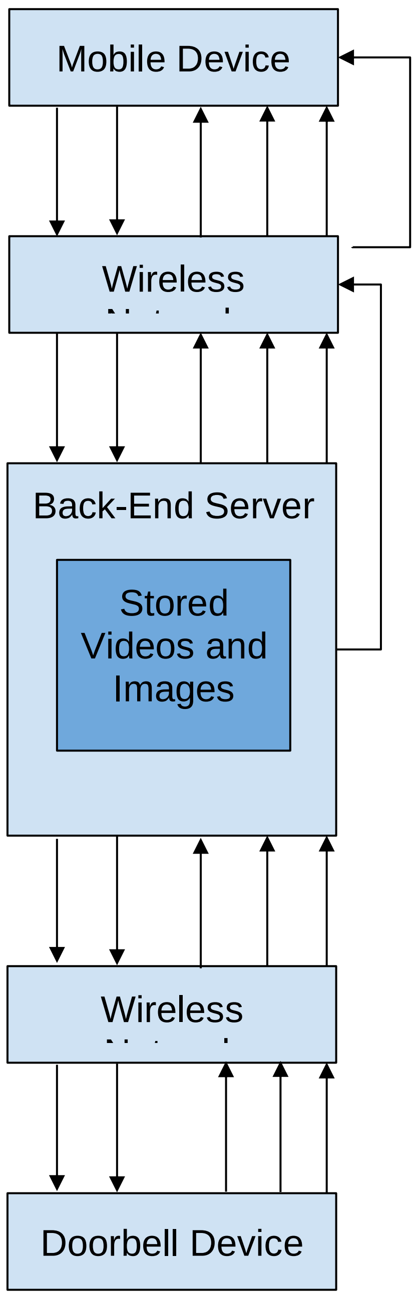

Figure 2: Architecture Diagram

Module Cohesion

Our project shows module cohesion through the Doorbell class. The Doorbell class will contain all the necessary functions to control all the hardware and communicate with the server. There will be subclasses for each piece of hardware that will have their own associated functions that control them. The parts will demonstrate sequential cohesion. The buttons output will become the requests input and so on.

Module Coupling

The application will use module coupling through the different things the user can view. The user will have a separate screen and functions for communicating with the user(visitor) at the door. There will then be a separate view option where the user(homeowner) can view saved content. This will give them access to the saved data on the server which they can then choose to delete or download. This represents coupling because they both are functioning within the app but they have separate classes and functions. Viewing the data on the server doesn’t require any communication with the Doorbell.

Design Analysis

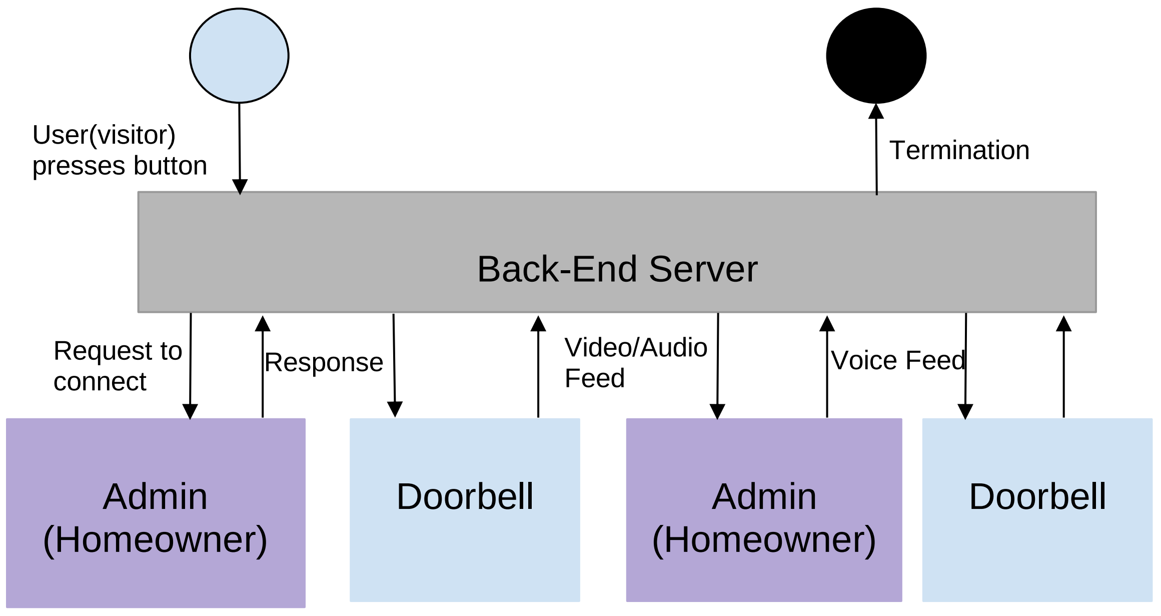

Figure 3.a: Data Flow Diagram Scenario 1

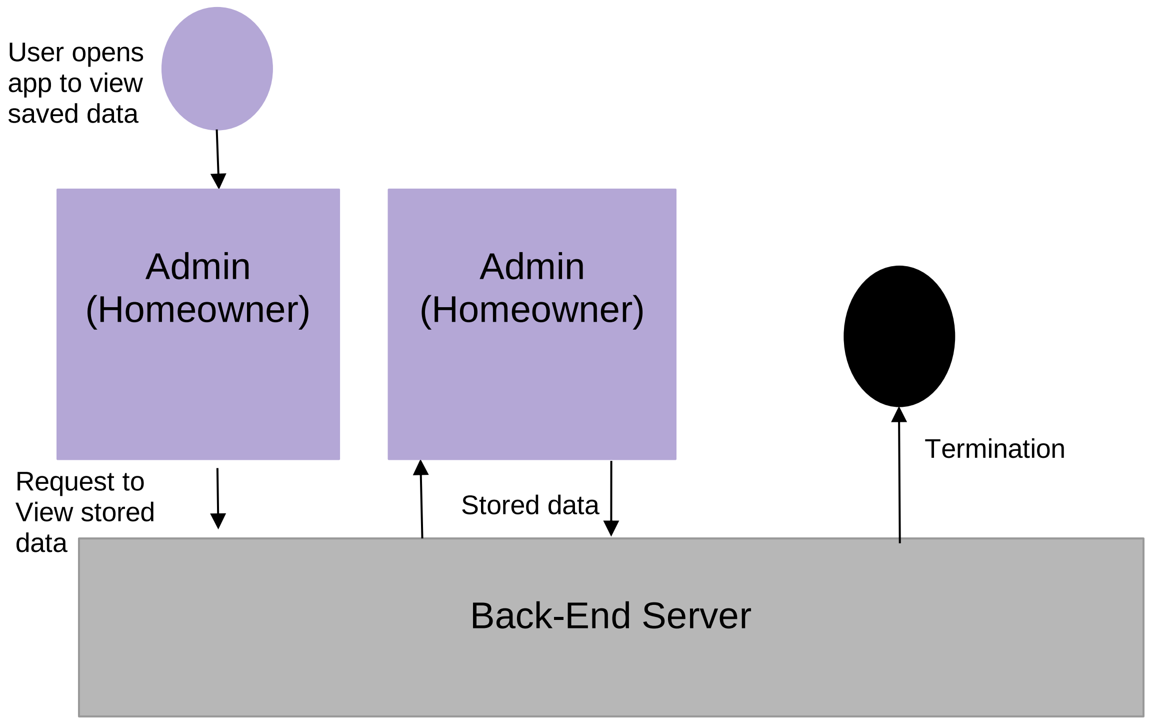

Figure 3.b: Data Flow Diagram Scenario 2

Design Organization (Object Oriented Design)

Detailed Tabular Description of Classes/Objects

Class name: DoorBell

Class description: The DoorBell class is the main class of the program. It calls on all the rest of the classes. This class handles the main interaction between the Admin and the program. It calls on all other classes when needed to perform tasks.

Class data members: Camera Class, Viewer Class

View Object Class member functions: pingAdmin(), pingCamera()

Class name: Camera Class

Class description: The Camera is attached to the doorbell and is pinged when the button is pressed. When the button is pressed the camera takes a picture of what it can see. When the user answers the ping from the doorbell the camera starts broadcasting live video/audio to admin's device.

Class data members: Camera Object

View Object Class member functions: takePicture(), deletePicture()

Class name: ImagePayload Class

Class description: Contains the picture taken by the camera when the buttons pressed. Also contains a timestamp of when the picture was taken.

Class data members: Image Object, String Object

View Object Class member functions: timeImage(), disImage()

<u>**Class name:**</u> **Stream Class**

<u>Class description:</u> This is the class where we connect the video from the camera on the doorbell to the admin’s device. We also send audio back from the admins device to be broadcast over the speaker on the doorbell.

<u>Class data members:</u> Video Object, Audio Object

<u>View Object Class member functions:</u> connectDoorbell(), disconnect()

<u>**Class name:**</u> **Speaker Class**

<u>Class description:</u> This is where we send the audio from the admin device to be played out of the speaker on the doorbell.

<u>Class data members:</u> Audio Object

<u>View Object Class member functions:</u> mute(), broadcastAudio()

<u>**Class name:**</u> **Viewer Class**

<u>Class description:</u> This is what we use to display the snapshots taken by the camera. When the admin is pinged, the snapshot will be displayed on the device. Admin can go back at anytime to view/delete images.

<u>Class data members:</u> Image Object

<u>View Object Class member functions:</u> deleteImage(), saveImage()

Functional Description of Classes/Objects

Doorbell Class

pingAdmin()

Input:

The pingAdmin() function will generate a true flag when the doorbell button is pressed.

Output:

The pingAdmin() function will output a notification on the admin's smartphone which indicates that the doorbell button is being pressed.

Return Parameters:

The pingAdmin() function will return true or false values or exceptions.

Types:

The pingAdmin() function will use data types based on the language used to code the project.

pingCamera()

Input:

The pingCamera() function will generate a true flag when the doorbell button is pressed.

Output:

The pingCamera() function will not return any output.

Return Parameters:

The pingCamera() function will return exceptions.

Types:

The pingCamera() function will use data types based on the language used to code the project.

Camera Class

takePicture()

Input:

The takePicture() function will take a picture and record live footage for a brief period.

Output:

The takePicture() function will return the picture and live footage that was taken and store them in the admin's account.

Return Parameters:

The takePicture() function will return strings or exceptions.

Types:

The takePicture() function will use data types based on the language used to code the project.

deletePicture()

Input:

The deletePicture() function will generate a true flag when the admin accepts deletion of the selected picture.

Output:

The deletePicture() function will output a dialog box which requests the admin to confirm deletion of the selected picture.

Return Parameters:

The deletePicture() function will return true or false values or exceptions.

Types:

The deletePicture() function will use data types based on the language used to code the project.

ImagePayload Class

timeImage()

Input:

The timeImage() function will generate a true flag when the picture is taken.

Output:

The timeImage() function will return the time when the picture was taken and store it in the admin's account.

Return Parameters:

The timeImage() function will return strings or exceptions.

Types:

The timeImage() function will use data types based on the language used to code the project.

disImage()

Input:

The disImage() function will generate a true flag when the picture is taken.

Output:

The disImage() function will display the selected picture.

Return Parameters:

The disImage() function will return strings or exceptions.

Types:

The disImage() function will use data types based on the language used to code the project.

Stream Class

connectDoorbell()

Input:

The connectDoorbell() function will generate a true flag when the doorbell button is pressed.

Output:

The connectDoorbell() function will return the true flag.

Return Parameters:

The connectDoorbell() function will return true or false values or exceptions.

Types:

The connectDoorbell() function will use data types based on the language used to code the project.

disconnect()

Input:

The disconnect() function will generate a true flag when the doorbell button is not pressed.

Output:

The disconnect() function will return a true flag.

Return Parameters:

The disconnect() function will return true or false values or exceptions.

Types:

The disconnect() function will use data types based on the language used to code this project.

Speaker Class

mute()

Input:

The mute() function will generate a true flag when the doorbell button is not pressed.

Output:

The mute() function will return a true flag.

Return Parameters:

The mute() function will return true or false values or exceptions.

Types:

The mute() function will use data types based on the language used to code the project.

broadcastAudio()

Input:

The broadcastAudio() function will generate a true flag and record audio when the doorbell button is pressed.

Output:

The broadcastAudio() function will return the recorded footage and store it into the admin’s account.

Return Parameters:

The broadcastAudio() function will return true or false values, strings, or exceptions.

Types:

The broadcastAudio() function will use data types based on the language used to code the project.

Viewer Class

deleteImage()

Input:

The deleteImage() function will generate a true flag when the admin deletes a picture and confirms the deletion.

Output:

The deleteImage() function will display a dialog box that confirms successful deletion of the picture.

Return Parameters:

The deleteImage() function will return exceptions.

Types:

The deleteImage() function will use data types based on the language used to code the project.

saveImage()

Input:

The saveImage() function will generate a true flag when the admin saves a picture and confirms the save.

Output:

The saveImage() function will display a dialog box that confirms successful save of the picture.

Return Parameters:

The saveImage() function will return exceptions

Types:

The saveImage() function will use data types based on the language used to code the project.

Files Accessed

The only files accessed on the admin's device would be located inside of the application. All of the pictures taken through the doorbell app will be stored in the app. These files would stay on the app for up to 60 days or until the admin deletes the images.

Real-time Requirements

The application will need to work in a short amount of time to be practical. We are aiming to have less than a five second delay between the visitor pressing the button and the admin receiving the ping on their phone. Once the admin presses accept it should take no longer than five seconds to start the live feed. The camera quality that needs to be achieved is 720p which won't be hard with a reliable internet connection.

Messages

Message Type	**Source/Destination**	**Data**
sendMessage()	Backend <-> Phone	Send any error messages (string)
receiveMessage()	Backend <-> Phone	Receive any error messages (string)
sendVideo()	Phone <-> Doorbell	Phone and doorbell send video stream back and forth
sendPicture()	Doorbell -> Backend	Image sent to backend from doorbell
recognizePress()	Doorbell -> Backend	Physical button press signal sent to backend for ping

Narrative/PDL

Application View

Main Menu:

1. View Video
2. View Pictures

View Video:

1. Button Pressed?
    a. Satisfied?
        i. Yes? View video feed
        ii. No? Empty screen/feed

View Pictures:

1. Recently Taken
    a. Save photos taken within last 24 hours
    b. Clear all recently taken photos
2. Saved Photos
    a. Load Selected Saved Photos
    b. Delete Photos

Doorbell

1. Button Pressed
    a. Satisfied?
        i. Yes?
            1. Take Photo
                a. Save

```
            i. Satisfied?
                1. Yes? Send to server
                2. No? Throw Error
    2. Ping homeowner
        a. Homeowner Responds?
            i. Yes? 2-3 Max Video Feed
            ii. No? Wait 1min and press again
ii. No?
    1. Nothing done
```

Decision: Programming Language/Reuse/Portability

The language that will be used for the app development portion is C# and Xamarin will be used to develop the iOS and Android versions of the application. For the hardware side, Python will be used because it works very seamlessly into the Raspberry Pi architecture.

Implementation Timeline

This timeline includes deadlines for specific implementation steps to keep the project organized and on time.

Month	Jan	Feb	Mar	April	May	May+
Component Design	🟥	🟥				
Software Design		🟥	🟥			
Validation			🟥	🟥		
Testing	🟩	🟩	🟩	🟩	🟩	
Present					🟩	🟩

Design Testing

Design testing will be done regularly throughout the entirety of the project at every level. When each main function of the project is completed like video feed and the application, testing will be done to ensure proper implementation of the project. With the project’s hardware aspect, multiple versions will be configured during the design portion of the project and tested for its ability to meet all requirements of the product. Once all functions for the hardware operation are complete, they will be tested to eliminate any possible bugs before working on the application. With the testing of hardware operation, we will be able to ensure that under any circumstances like pressing the button many times in a row or holding it in that a bunch of pictures and pings to the admin don’t occur. The testing will all be done within the development team, not only because of the project requirements but because we don’t want to outsource the testing to a pool of people. Making sure the testing is done in-house ensures that all requirements are met.

Appendix A: Schematic & Bill of Materials (BOM)

Hardware Costs:

- Raspberry Pi 4 Model B has an MSRP of $39.99.
- Camera Module has an MSRP of $9.99 - $14.99.
- Speaker has an MSRP of $7.99 - $9.99.
- Miscellaneous Electrical Components have an estimated cost of $14.99.
- Breadboard has an estimated MSRP of $4.99 - $8.99.

Misc. Fees:

- The Google Play store has a $25 registration fee.
- The Apple App Store has a $99 registration fee.

Appendix B: Technical Glossary

Admin - Short for administrator and analogous with the homeowner.

Android - a Mobile operating system developed by Google.

Apple App Store – The marketplace for uploading and download applications on iOS devices

Breadboard - A construction base for wiring up systems of electronics.

C# - A programming language used for the main operation of the product.

Google Play – Androids store for uploading and downloading applications

iOS - The iPhone Operating System.

Push Button - Physical button that the user will press to initiate device operation.

Python – A programming language used for programming the Raspberry Pi unit

Raspberry Pi - A small computer that has input and output functionality.

Raspberry Pi Camera Module - A camera module sold to seamlessly work with Raspberry Pi.

Miscellaneous Electrical Components - Hardware components used to connect, amplify, and control electrical signals.

User - Described as the person interacting with the physical product, analogous with a visitor.

Video Doorbell - The title of our project to provide a solution to missing visitors at a person's home.

Xamarin - .NET is a developer platform made up of tools, programming languages, and libraries for building many different types of applications (Microsoft).

Appendix C: Team Details

This document was created, revised, and finalized by the following individuals:

Trey Brown –

Trey was responsible for all the formatting of the document, assuring everything is neat and orderly. He was also responsible for the following sections of the document:

- Messages
- Narrative/PDL
- Decision: Programming Language/Reuse/Portability
- Implementation Timeline
- Design Testing
- Appendices

Michael Cheng –

Michael was responsible for the proofreading and editing of the document as well as the following sections:

- Functional Description
- Input/Output/Return Parameters/Types
- Files Accessed

Brittany Marietta –

Brittany was the leader for the design document, setting up the google doc and diving tasks up based upon strengths. Brittany was also responsible for the following sections:

- Abstract
- Description of the Document
- Purpose and Use
- Ties to the Specification Document
- System Modules
- Design Analysis

Jesse Jento –

Jesse was responsible for the proofreading and editing of the document as well as the following sections:

- Design Organization
- Detailed Tabular Description of Classes/Object
- Real-time Requirements

Appendix D: Workflow Authentication

I, Trey Brown, hereby attest that I have performed the work as documented herein.

Printed name	Signature	Date

I, Michael Cheng, hereby attest that I have performed the work as documented herein.

Printed name	Signature	Date

I, Brittany Marietta, hereby attest that I have performed the work as documented herein.

Printed name	Signature	Date

I, Jesse Jento, hereby attest that I have performed the work as documented herein.

Printed name	Signature	Date

Appendix E: Report from the Writing Center

Cal U Vulcan Learning Commons Report

Client: Trey Brown

Staff or Resource: Brittany Kach

Date: December 05, 2019, 12:30pm - 1:00pm

What course was serviced by this visit?: CSC 490

Did the student request that the instructor receive a visit report?: No

Please provide any additional comments relevant to this session.:

How did the process of this consulting session address the established goals?: The client brought in his design document for a senior project. Reviewed paper for organization and format, slight advisements on the abstract, few typos/grammatical changes.