CA326 Functional Specification - Home Surveillance System

Team Members:

- Ross Franey
- Kieran Turgoose

Table Of Contents

1. Introduction

1.1 Overview of Project

This project is a cost effective and easy to use home surveillance / security system which will enable users to keep track of several cameras positioned in various locations of their choosing (either physically connected via USB, or using IP cameras to connect over Wi-Fi) via their smartphone or personal computer. The system will be based around a Raspberry pi, and will include motion detection software which will activate image capturing on each individual camera if and only if there is movement in that camera's frame. These images will be then taken in a pre-set recurring time frame and stored in a drop-box folder. This will also mean that there will be images available if there is a breach of security in the owner's home, which can be handed over to authorities to strengthen any resulting legal case that may arise.

The user will also have the option of setting up a live link with their video camera through streaming software such as Skype, using a Virtual Networking Computing (which from now on will be referred to simply as VNC) Server, in order to view, in real time, the video feed of that camera, which would facilitate not only a security function, but also communication with whoever may be in the scope of the camera's view, allowing the system to be used for the purpose of monitoring young children and / or elderly people that may be under the care of the user.

As well as the motion detection camera software, the system can be used with a motion sensor placed on doors/windows etc. which also activates the alarm and subsequently initializes the image capturing process of the camera.

The main functions of the system can be summarized as follows:

- Remote notifications of detected motion to one's android device, along with images.
- Ability to fetch a live stream of cameras over Wi-Fi.
- Motion detection of varying sensitivity depending on the use of the camera (EG monitoring a sleeping baby vs making use of the camera for security reasons.
- Option to sound an alarm once motion is captured.
- GPS system that will make use of the android devices GPS location to determine whether notifications are needed based on whether you are in your home or not. This allows the system to be configured to automatically arm the system once the device is identified as being away from home.
- A mobile application that provides the user with the view of current enabled camera(s).

1.2 Project Scope

This system was developed with the home-user in mind, and as such will provide a convenient, cost effective method of monitoring a home, without unnecessary complexity of use, or cost. A large distinguishing factor of this system is that it will aim not just to satisfy the security needs of its users, but also be useful in a more general, everyday life sense. The versatility of this product means that the set of target users spans a wide range of demographics. New / busy parents, child minders, people with elderly relatives, small businesses in need of till surveillance, etc. More specific and detailed use cases and scenarios can be seen in section 2.3 / 2.4 in this document.

1.3 Document Outline

Section 1 of this document has been an Introduction to the project, including an overview of the idea, scope, our motivations as well as external references from which the team drew knowledge to aid in the planning of the system.

Section 2 of this document has been designed to provide a more in-depth non-technical description of the main functions of this surveillance system. This includes the different interfaces the user will interact with, the different options available to the user in terms of configuration and preferences, and example use cases to display the practicality and simplicity of the system, for a multitude of diverse user demographics.

Section 3 of this document will give a general overview of the system requirements, i.e. the various technical frameworks it must support to function as planned. This will include software/hardware requirements, security requirements, functionality, usability, and accessibility requirements that this system must adhere to, so that it can benefit the widest spanning user-base possible.

Section 4 will focus on the individual components/modules of the system, the design decisions in terms of how each will interact with the other, Including a component Diagram, and an explanation of the interaction in the diagram.

Section 5 is a high-level overview of the design of the system. A context diagram will represent this design as well as an accompanying explanation of the interactions in the diagram.

Section 6 outlines the Preliminary schedule for the completion of individual tasks involved in the overall completion of the project. Included in this section is a GANNT chart, as well as a task view chart. Section 6 also outlines the hardware and software requirements needed on the *teams* end to carry out the tasks mentioned.

Section 7, labelled "Appendices", will explain the anticipated problems, proposed solutions, and learning challenges associated with the development of this system.

1.4 Motivations

Our main motivation for the development of this system was the fact that currently, home security systems are not something that most people consider a viable option, as they are perceived as being expensive, almost “luxury” systems, that the average person could not afford. Unfortunately, this is largely true. However, we believe considering how much progress society has made in terms of technological advancement, home security should be established as a concept that all home owners consider adding to their home, a new “societal norm” if you will. This trend can be seen in several areas of the Tech-World, that is, the idea of new technology being inaccessible and high-end, until cheaper and cheaper methods are developed, eventually normalizing the use of the technology and making it a staple in society. A good example of this is the development process of smart phones. Once a luxury product, the smartphone is now considered a necessity for an ever-growing majority of people.

Our focus is to achieve this goal of introducing the idea that security should be a fundamental concern for all home owners, and should not be overlooked due to fears of complexity/price, while also adding useful features that can help people in other aspects of their life aside from security, as mentioned above. To do this, we needed to choose a medium that is already a cornerstone in the everyday life of the public.

Enter the above-mentioned smartphone. Nowadays many people use smartphones to manage a variety of different aspects of life be it nutrition, time keeping, socializing, fitness, music, media/entertainment, etc. As such, this is a comfortable and well-known platform for a huge number of people. The system should not make users feel like they must learn something new, which can be an off-putting idea for many, particularly for people who may not necessarily be tech-literate. The aim of this project is to deliver a system which has all the functionality of a high-priced industrial security system and more, but packaged in the familiar, non-intimidating format of a mobile application.

1.5 Reference Links

Image Storage:

https://www.dropbox.com/

https://www.raspberrypi.org/

Security checks:

http://searchsecurity.techtarget.com/definition/Secure-Sockets-Layer-SSL

http://info.ssl.com/

https://www.dropbox.com/help/27

Learning Python:

https://www.codecademy.com/learn/python

https://www.python.org/

http://www.learnpython.org/

Glossary References:

http://searchnetworking.techtarget.com/definition/virtual-network-computing

http://www.dictionary.com/browse/gps

https://www.extremetech.com/computing/124317-what-is-raspberry-pi-2

1.6 Glossary

- **Raspberry Pi**: "*The Raspberry Pi is a small, barebones computer developed by The Raspberry Pi Foundation, a UK charity, with the intention of providing low-cost computers and free software to students. Their ultimate goal is to foster computer science education and they hope that this small, affordable computer will be a tool that enables that.*"" - ExtremeTech.com
- **VNC Server**: "*Virtual network computing (VNC) is a type of remote-control software that makes it possible to control another computer over a network connection. Keystrokes and mouse clicks are transmitted from one computer to another, allowing technical support staff to manage a desktop, server, or other networked device without being in the same physical location.*" - searchnetworking.techtarget

- **GPS**: "*Global Positioning System: a global system of U.S. navigational satellites developed to provide precise positional and velocity data and global time synchronization for air, sea, and land travel.*" - Dictionary.com
- **SSH**: "*SSH can refer both to the cryptographic network protocol and to the suite of utilities that implement that protocol. SSH uses the client-server model, connecting a secure shell client application, the end at which the session is displayed, with an SSH server, the end at which the session runs.*" - searchnetworking.techtarget

2. Description

2.1 Features

The main functions in this system involve the user being able to interact with a variety of web cameras that have been set up around the home, place of work, etc. This interaction can be from a remote location using a mobile device, to monitor a certain area using motion detection and image capture, and to take actions based on the images being captured. Although the following features will be explained in more detail in section 3, below is a general outline of the main features the system should provide:

- An Android Application
- Motion Detection System
- Image Storing
- Triggerable Alarm (Through multiple pathways)
- Automatic Arming (based on GPS tracking)
- Ability to respond to security notification with multiple “one-click” options
- Multiple Alarm Sound options
- User Preferences/Settings (camera sensitivity, enable/disable GPS, change alarm sound etc.)

2.2 Target User Characteristics & Environments

In regards to the expected profile of prospective users, the main categories can be summarized as one, or any combination of the following:

- New Parents who would benefit from the baby monitoring capabilities of this product.
- Minders/Sitters/Watchers be it for a property, young children, or for the elderly who could utilize such a system to give the cared subjects independence while still monitoring their activities, and even having the opportunity to communicate with them at the tap of a button.
- General Home owners who are living alone, or who’s house is left unoccupied for large portions of the day.
- Anybody interested in adding extra security to an asset or number of assets. For example, one may want to add security to the garage/driveway in which they keep their car. Another example may be a business owner

interested in monitoring a safe or till.

2.3 Use Cases

Use Case 1	Configuring a Profile
Goal in Context	Set up the application to work with the camera(s)
Preconditions	Possession of Security Pi system, Skype & Dropbox account, Android Device
Success End Condition	Application configured & synchronized with Drobox & Skype
Failed End Condition	Application not synchronized with accounts correctly
Actors	User
Step	Action
1	Download the application from the Google Play Store
2	Upon prompt, enter Dropbox Username & Password
3	Upon prompt, enter Skype Username & Password
4	Enter the name & Dropbox Folder of First Camera
5	Create Application Username & Password
6	Add New Camera
Branches	
6B	Skip

Use Case 2	Investigate Triggered Alarm
Goal in Context	Respond to an alarm by viewing images captured
Preconditions	Security System armed, App configured and Motion detected
Success End Condition	View images of motion captured from android device

Failed End Condition	Not capable of viewing Images
Actors	User
Step	Action
1	Android Device receives a buzz notification
2	User opens application and sees the highlighted camera in question
3	User opens Camera and sees the recently captured images
4	User decides there is nothing concerning and stops alarm & image capturing
Branches	
4B	User opens Skype connection with camera
4C	User presses "Call emergency services" / "Call Neighbor" button
4D	User saves one or more of the captured images to phone

Use Case 3	Automatic Arming
Goal in Context	Automatically arm the Surveillance system once the user leaves the home
Preconditions	Raspberry Pi Powered on, Application granted GPS location access
Success End Condition	System is armed once user leaves home
Failed End Condition	System fails to arm once user leaves home
Actors	User
Step	Action
1	User Leaves Home, forgetting to arm the system
2	User's device receives notification offering to arm the system
3	User Chooses "Yes"

4	System is now armed and waiting to detect motion
Branches	
3B	User Chooses "No"
4B	System remains unarmed

Use Case 4	Open Skype Connection
Goal in Context	Open skype on the Raspberry pi and connect with it using a VNC Server
Preconditions	App profile Authenticated/w Skype account, System is armed
Success End Condition	User can see live video from the Camera
Failed End Condition	Skype call fails to initialize
Actors	User
Step	Action
1	User opens a camera inside the application
2	User wants to communicate with other side, chooses "Live Feed" option
3	User is prompted to enter VNC Server Password
4	Password accepted and user can see Skype call on Raspberry Pi screen
Branches	
4B	Password is declined and user is brought back to App's Dropbox folder

2.4 Example Scenarios

Outlined below are examples of situations in which this system will prove a useful tool for users in several scenarios already listed, to display the usefulness and functionality of the system in everyday life.

Scenario 1: Home Security /w Burglary

In this scenario, a user has set up the Raspberry Pi and camera by their living room window. Once they leave the home, a notification will be received on their phone and upon accepting, the system will be armed. During the time in which the user is away from home, while there is no motion being detected, the camera simply takes a photo, compares against the last, and discards it. When there is motion detected however, the speed of the image capturing picks up substantially and the user receives a notification on their mobile device. Once they open the application, they will see which camera the notification is coming from, and will open the Dropbox folder for that camera to investigate. Here, they will see live images from in real time being uploaded of an intruder in their home, as he enters through the window. From this position, the user will have the option to save the images, make a phone call (contact the police, for example), or open a live video feed connection with the camera. The user touches the button to call the police and are immediately connected so that they can explain their situation, and can provide all the footage necessary to use as evidence against the intruder.

Scenario 2: Monitor a Sleeping Baby

In this example, a parent has set up a camera in the room of their young baby to give them time to do necessary chores around the house while the baby sleeps, and be notified if the baby wakes up. Upon receiving a notification on their mobile phone, they open the app and move into the camera in question. From here, they can see that the baby has moved but it appears to be still asleep. To investigate further, the parent opens a live video stream with the camera and finds that the baby is in fact awake and unsettled. From here, the parent triggers the alarm (which in this case is a soothing lullaby melody), and can now drop what they were doing to tend to the baby's needs.

Scenario 3: Monitoring Front Door

Here, the camera has been set up to give a view of the outside of the home, namely the front door entrance. The user's phone receives a motion notification (and the door knocks). The user opens the camera's folder to investigate. If in this case, the camera shows somebody who is not welcome and/or not expected, the user is now aware without having to expose their presence. In this case however, the user sees a delivery man that had been expected. The user is busy however and thus, opens a skype connection and notifies the delivery man that he/she is on the way and will answer the door shortly.

Scenario 4: Monitoring Elderly Relative/Care Subject

In this scenario, the user is taking care of their elderly grandmother who is currently sleeping in the living room (where the surveillance camera has been set-up). The user takes a trip to the shop to buy groceries for the evening, however on the way a notification is sent to the user's phone and upon investigation, the user finds that their grandparent has fallen and is in need of assistance. The user can now open a skype connection with the camera, comfort and direct the elderly subject, and begin to make their way home immediately to assist.

2.5 Constraints

One unfortunate constraint is related to the type of camera that can be used in conjunction with the system. The

Raspberry Pi official camera module is the most obvious choice and can be disguised very well. Although many other cameras can be used, they are, naturally enough, limited to the cameras that are compatible with the Raspberry Pi. There is a list of such cameras on the official Raspberry Pi website. Although the set of all compatible cameras is not limited to this list, these are the only cameras guaranteed to be supported by the Pi, and as such are a safe bet.

Another constraint to be considered is the number of cameras. The number of cameras on this system will be limited to a maximum of 4, in order to avoid slowing down the system or risking IO problems related to the uploading and comparing of images.

The alarm on this system will be limited to the one output device, rather than being individual to each camera. The reason for this is that multiple cameras will be connected to the one Raspberry Pi, which will also house the audio output system. The same is true for the ability for communicating with the other side via skype as communication would be achieved via this same audio device.

3. System Requirements

3.1 External Requirements

Hardware: For the user to set up this surveillance system at home, a Raspberry Pi, and at least one camera is required (ideally the Raspberry Pi official camera module). As well as this, an android device is required to make use of the mobile application.

Software: This system required the Raspberry Pi to be running a version of Raspbian OS. As well as this, it is required that the user's phone is running android (preferably 6.0(+) Marshmallow).

Environment: For the system to send real-time notifications to the user's mobile phone, it is necessary that the user has some sort of internet connection outside of the home, in which they are connected to through their android device. A home internet connection (or wherever the system is stationed), naturally, is also required. A drop box account is also required as this is where the system will store images, and therefore is the medium through which users will be capable of viewing images from the camera in question once the system has detected motion.

3.2 Product Functions

The Application

The system will provide the application downloadable from the google play store, which will feature a main page of all available cameras that the user has set up in a tile format. These cameras will only be activated once there has been motion detected at which point, a notification would be sent to the user's phone, the app would be opened, and the user will find the corresponding tile highlighted. From here the user would have a multitude of options such as viewing the live feed from the camera, opening the folder where images are being stored in order to view what the motion was, deactivate the alarm triggered by the motion, and also send a text message to authorities to alert them of the disturbance if the user feels such action is necessary based on the images they see have been captured. The application would also be connected to the user's drop box account, so that images being uploaded in real time by the system can be viewed regardless of where the user is.

Motion Detection

This system is built around the concept of capturing images only when there has been motion detected. In order to achieve this, photographs will be taken every X number of seconds, where X is a predetermined variable. These images will be stored and compared against the previous image. If the previous image differs in any way, the camera will begin to take photographs more rapidly and store them, as well as notifying the above-mentioned application. For all the images that are no different to the previous, the new image will simply be discarded. This means the system only needs to store new images when there is a change in scenery.

Image Storing

As well as showing real time images to the user via the mobile application, the system will also store each image which differs from the previous. These images will be stored on drop box for the user to be able to access and download/save the images.

Multiple Alarm Triggers

The system will provide two mediums through which the alarm sound may be triggered: a magnetic motion sensor which can be put on doors or windows, and the motion detection software of the camera(s). The alarm can be triggered through both methods. The magnetic sensor is limited to being in the same room as the Raspberry Pi unlike a camera, as it is connected directly to the device. Although not the main security feature of this system, it could be a useful extra precaution for the user, depending on their needs.

GPS Tracking & Automatic Arming

The system will also make use of the GPS feature on one's android device which will allow it to detect whether the user is at home at any given time. The user can configure the system to automatically arm itself when the GPS on their mobile phone knows that they are out of range of the house. The application will automatically send the users phone a notification with the option to arm the system (if it is turned on at home). The user can then choose to accept or deny the offer. If the former is chosen, the user will be brought to the configuration page of the application to choose from the preferences/settings mentioned earlier, as well as which cameras they want to arm.

One-Click responses

If the user finds that the images being captured by one of the cameras are suspicious, and they are not near home to investigate themselves, the app will also have the option to call numbers that the user will define as neighbors, with the simple click of a button. If, for example the images lead the user to believe there is a burglary underway, the user can also tap a single button to call emergency services. The system also features an option to tell the camera to stop waiting for motion, or to stop an alarm that has been triggered.

Multiple Alarm Sounds

One important feature of the system is the ability to change the sound clip being played when the alarm is triggered. This is useful for when the alarm is triggered on a camera that is not set up for a security function, but rather to keep watch on a sleeping child for example. In this situation, the alarm could be changed to something akin to a lullaby melody while the user makes his/her way to attend to the child's needs. As mentioned in the constraints section, this is limited to the cameras that are physically connected to the Raspberry Pi itself, rather than any IP cameras that may be connected over Wi-Fi.

User Preferences

The user would also be given the option to change the settings and preferences of certain cameras. For example, the user can change a camera from a “security” styled setup to a monitoring setup, in which the user would decrease the sensitivity of motion captured by the camera to monitor say, a sleeping baby. The user would not want to capture small movements in sleep, but only larger more obvious movements that would be indicative of a child waking.

3.3 Usability Goals

Scalability

Using multiple cameras on this system is a very important feature, so the user can monitor more than a single area at any one time. The obvious concern here is the trade-off between more cameras and the additional overhead in terms of bandwidth and storage. Thus, while currently the system is planned to cap it at four cameras at any one time, in the future an obvious area of expansion to consider is expanding this limit.

Performance

As this system relies heavily on the user’s internet connection, this will likely be the first limiting factor in terms of the performance of this system. Another factor could be the number of cameras attached to the system which is why, as explained above, the camera limit has been set to four at any one time. Performance will have a strong correlation with the scalability mentioned above. Naturally, if all four cameras have detected motion at the same time, and are all updating their respective drop box folders performance will suffer. However, again, with the cap of four cameras it shouldn’t be a concern.

However, assuming an adequate internet connection, and with the restriction on the number of cameras, the performance of this system should be suitable. This will be achieved by the fact that the camera is neither constantly streaming video, or even constantly storing images, so memory should never be a concern. At any one time, while there is no motion detected, the system should only ever have at most two images stored, one of which will be deleted in absence of motion.

Portability

A big advantage of this system over similar security/surveillance systems is how easy it is to move around. If a user wishes to set up the system in a different location, all that is required is to move a very small Raspberry Pi and any cameras necessary for their particular goals.

Setup

It is important this system is easy for any prospective user to set up, and not just those familiar with the Raspberry Pi computer, or technology in general. To achieve this, a simple instruction section will be included in the application for the user to reference if there are ever any confusions in regards to how to use the system. This will include features such as a FAQ section, and step by step guides using images/screen shots of the set-up process. However, there will not be much involved in this process as the Raspberry Pi will be set up in advance to run the security system on start-up, so powering on the system should be all that is required.

Maintenance

This system will require almost no maintenance from the user, other than managing drop box folders, removing no longer required images, etc. On behalf of the team, as each segment is broken down into separate distinct classes in

code, this should make it easier for any problems that do occur in the future to be identified and fixed immediately.

Security

The primary security concerns for this system are related to Dropbox and the integrity of one's drop box folders where images will be uploaded. This should not be a problem considering drop box files are encrypted using 256-bit Advanced Encryption Standard (AES), Secure Sockets Layer(SSL) and Transport Layer Security (TLS) to, per their website, "protect data in transit between Dropbox apps and our servers; it's designed to create a secure tunnel protected by 128-bit or higher Advanced Encryption Standard (AES) encryption."

This, in combination with the fact that regular tests are carried out by drop box for security vulnerabilities to protect against attacks, will ensure that user's images are suitably safe while they are stored online.

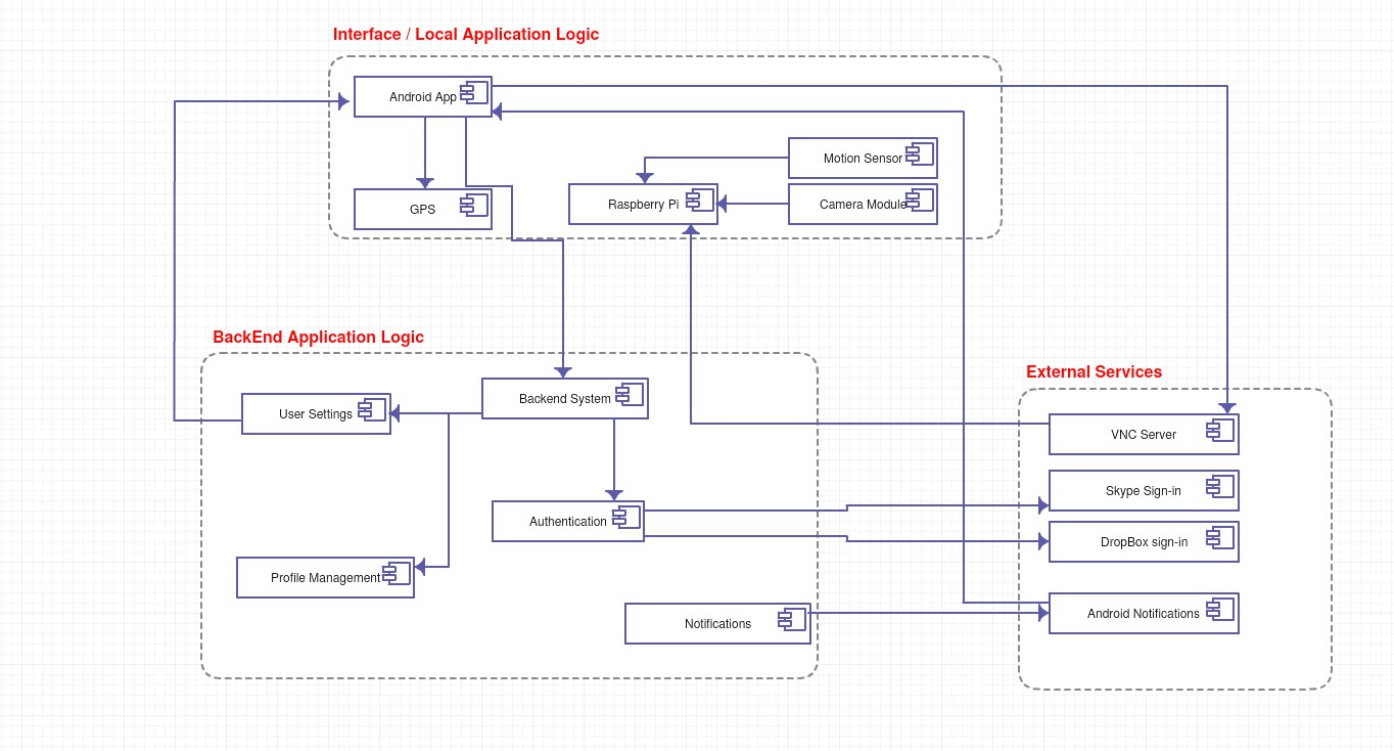
Finally, the application will require the user to grant Dropbox access via the Authorize button in the user's Dropbox account, adding an extra layer of security. Instructions on exactly how to set this up will be included in the application as mentioned above.

User Interface

The UI for this system should be sleek, minimalistic, and clean while still providing the user with all the functionality mentioned above. It is important that the application is not overly cluttered or complex looking as an integral aspect of the advantage of this system over others in the field is the fact that it is accessible both in price and usability, to all home owners, as specified above.

4. System Architecture

4.1 System Module/Component Diagram



4.2 Interaction Diagram Explanation

- Here, the **Local application logic** section covers all the high-level interactions between each physical component and the application itself. The Android application will interact with the GPS in order to satisfy the Automatic arming function. The application will also interact with the Backend system module to access the components in that respective section. Also in the local application, the Raspberry pi will receive data from the camera module and motion sensor to determine whether images are to be stored. As well as this the VNC server will be set up to view the Raspberry Pi's live feed.
- The **Backend application logic** section represents the different relationships between the lower level components of the system. Here the backend system component acts as a link between the Local and Backend sections. The user settings module will allow the user to change preferences in the application itself (as per the functional requirement described above), in the profile management option. As well as this, the Authentication module will require the user to sign into their Skype and Dropbox accounts in order to gain full access to the functionality of the system. The notifications to the user's phone will be handled by the Android notification system, which will allow the user's phone to be notified at any time, and not just when the application is open.
- Finally, the **External services** section refers to the 3rd party components used in this system.

5. High-Level Design

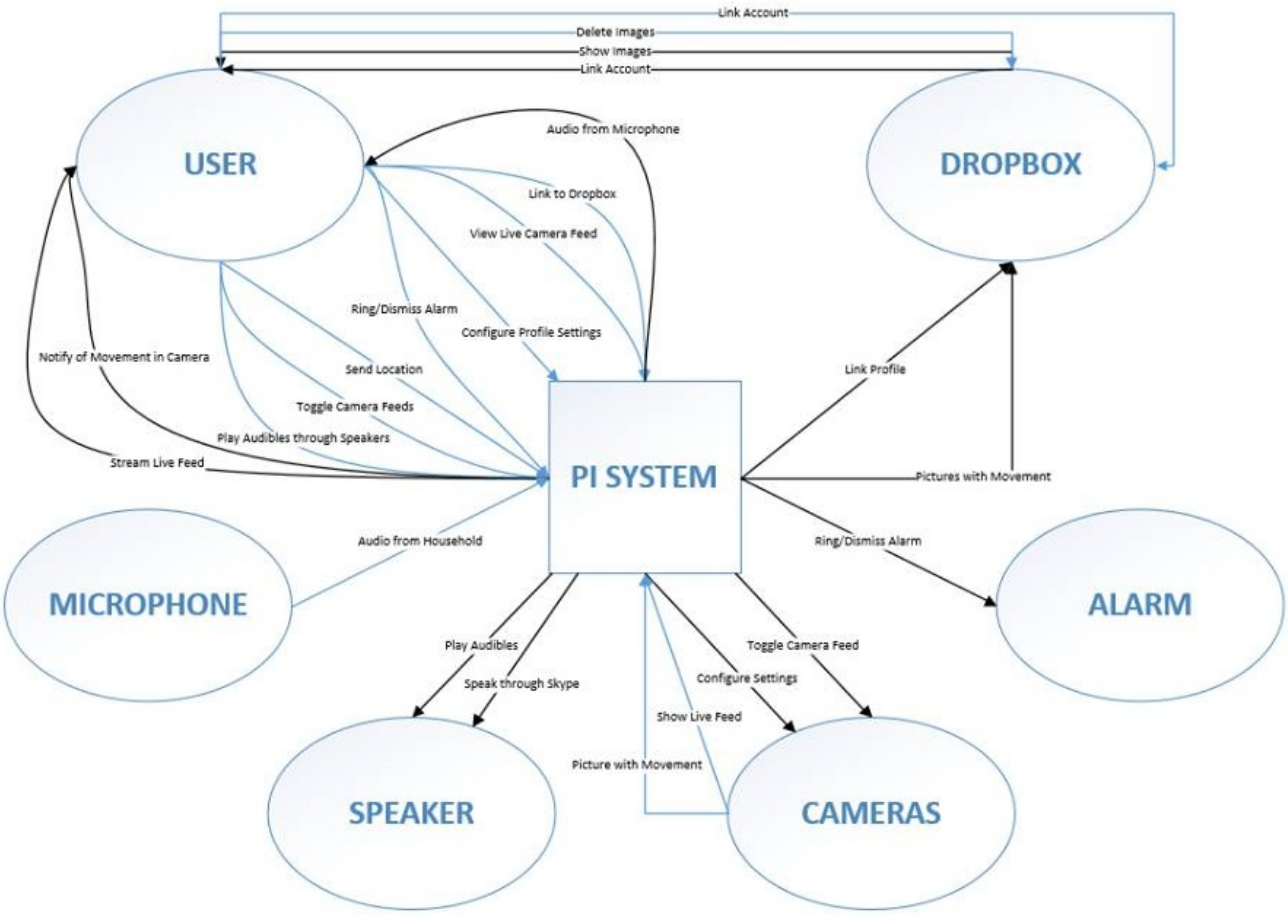
5.1 Design Overview

Here we provide a Context Diagram illustrating how our system will be connected and what data will be transferred

between each component. We chose this diagram as it is the easiest way to visualize which parts of the project must be linked to each other. We feel that this diagram accurately conveys a high-level representation of how the components of this system will be connected and the information needed to be passed between components.

This diagram shows how the Raspberry Pi interacts with the external hardware and software that will be used as part of the system development. The Pi will need to be linked to the Android application, which is represented as the "USER", the DROPBOX account, the external SPEAKER, the ALARM system and the CAMERAS. The only real interconnection between the external entities will be between the user's android application and their corresponding Dropbox account which will allow the use of Dropbox functionality to view and delete images stored from the motion detecting camera.

5.2 Context Diagram



6. Preliminary Schedule

6.1 Overview of Preliminary Schedule:

The schedule below (6.3) was designed using smartsheet.com and shows a breakdown of what we have initially

suspected to be the main tasks as part of this project. It is divided into columns of the Task Name, Start Date, End Date, Duration in days, and status of each task. These start and end dates are of course tentative and subject to change. Also included are our schedules for the tasks that were already completed as part of this project for completeness sake (i.e. Creating the GitLab Account, the initial Proposal Form, and this Functional Specification.

Below, a GANTT chart can be seen which shows this same information in a more visual representation of the time allocated to each task. The current completed tasks have been represented as a blue bar, while the incomplete tasks are represented by a green bar. Over the course of the project we will be updating this GANTT chart as tasks are completed or as the timeline is adjusted for individual tasks.

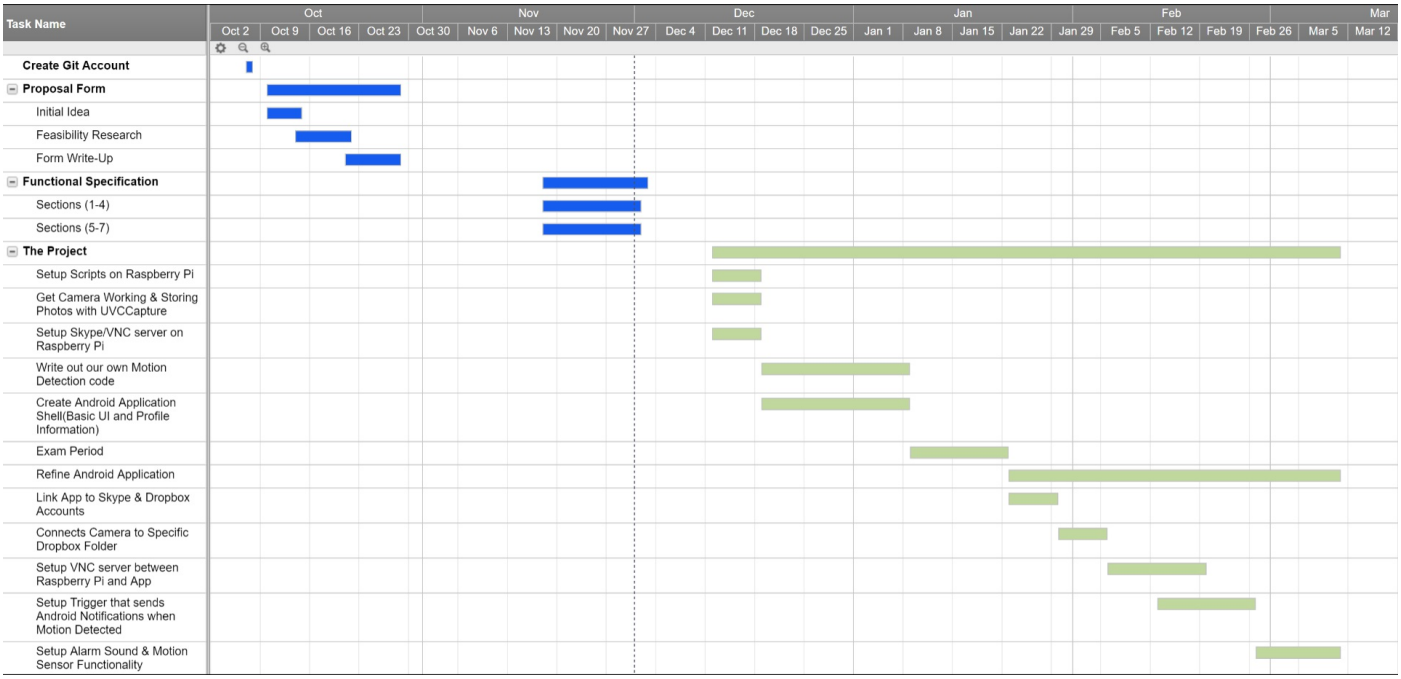
The **Hardware** requirements for this project are as follows:

- Raspberry PI
- Maximum of 4 Cameras (1 official Raspberry pi, 3 USB/IP)
- One external speaker
- Motion sensor
- Laptop/Monitor

The **Software** requirements for this project are as follows:

- Raspbian OS
- Python
- Skype
- Dropbox
- TightVNC
- UVCCapture
- Java & Android App Development Software

6.2 GANTT Chart



6.3 Task View

Task Name	Start Date	End Date	Duration	Status
Create Git Account	**07/10/16**	**07/10/16**	**1d**	**Completed**
Proposal Form	**10/10/16**	**28/10/16**	**18d**	**Completed**
Initial Idea	10/10/16	14/10/16	4d	Completed
Feasibility Research	14/10/16	21/10/16	7d	Completed
Form Write-Up	21/10/16	28/10/16	7d	Completed
Functional Specification	**18/11/16**	**02/12/16**	**14d**	Completed
Sections (1-4)	18/11/16	01/12/16	13d	Completed
Sections (5-7)	18/11/16	01/12/16	13d	Completed
The Project	**12/12/16**	**10/03/17**	**89d**	In Progress
Setup Scripts on Raspberry Pi	12/12/16	18/12/16	7d	Not Started
Get Camera Working & Storing Photos with UVCCapture	12/12/16	18/12/16	7d	Not Started
Setup Skype/VNC server on Raspberry Pi	12/12/16	18/12/16	7d	Not Started
Write out our own Motion Detection code	19/12/16	08/01/17	7d	Not Started
Create Android Application Shell(Basic UI and Profile Information)	19/12/16	08/01/17	21d	Not Started
Exam Period	09/01/17	22/01/17	14d	Not Started
Refine Android Application	23/01/17	10/03/17	47d	Not Started
Link App to Skype & Dropbox Accounts	23/01/17	29/01/17	7d	Not Started
Connects Camera to Specific Dropbox Folder	30/01/17	05/02/17	7d	Not Started
Setup VNC server between Raspberry Pi and App	06/02/17	19/02/17	14d	Not Started

Setup Trigger that sends Android Notifications when Motion Detected	13/02/17	26/02/17	14d	Not Started
Setup Alarm Sound & Motion Sensor Functionality	27/02/17	10/03/17	12d	Not Started

7. Appendices

7.1 Possible Requirements Problems

Requirement: The Application

Possible Problem: Implementing App folder that has access to Dropbox file system.

Proposed Solution: If there is an issue with implementing Dropbox files we can simply sync the phones photo album with the drop box account and link there instead.

Possible Problem: Integrating VNC Server with App.

Proposed Solution: Use Remote Ripple app which is developed by the same company as TightVNC server.

Possible Problem: Sending notifications from the Pi to the Application.

Proposed Solution: As a means of notifying the application (and subsequently the android device) as soon as motion is detected, rather than worrying about pushing notifications from the software we have written on the Raspberry Pi itself, the device could simply receive a notification each time the size of the camera's respective Dropbox folder grows.

Requirement: Motion Detection

Possible Problem: Images that are stored in Dropbox would trigger a notification of the phone regardless of there being a change or not, resulting in a "false alarm" (or in this case, successive false alarms).

Proposed Solution: In order to fix the above problem, images for comparison could be stored locally on the Raspberry Pi itself rather than going to the Dropbox server. This means the only time the size of the Dropbox folders will change is when that particular camera has captured motion, meaning the notifications will only be sent to the android device at appropriate times.

Requirement: Multiple Alarm Triggers

Possible Problem: There is no way to distinguish between the sensor-triggered alarm (from the Raspberry Pi magnetic motion sensor) and the camera's motion detection alarm notification.

Proposed Solution: In the python Code stored on the raspberry Pi, it may be possible to set different notifications for

each type of sensor, and possibly even each camera, which would be a welcome edition. The only uncertainty around this is whether it is possible in android notifications there Android programming is a completely unfamiliar area to the team.

Requirement: GPS Tracking

Possible Problem: It may not be possible or viable to write our own python code in order Utilize the GPs capabilities of the android device in a way that is useful to this requirement. Again, the ambiguity here derives from the lack of experience programming android applications.

Proposed Solution: 3rd party software may be a more viable option such as google maps or other similar services that track GPS location. Generally, this is considered an unsatisfactory solution by the team, however it is preferred over the alternative of simply not implementing the GPS feature.

Requirement: One Click Response

Possible Problem: If the user hits the emergency services option accidentally, it could result in unnecessarily wasting the time of the emergency services which is a very serious issue.

Proposed Solution: In order to prevent the above from happening, a solution may be to slightly bend the simplistic design goal of the "one click" aspect of that particular response, and add an extra "are you sure" prompt, to safeguard against an accidental click. As the other options are not necessarily largely consequential, the team feels no need to add this restraint to all options.

Requirement: Multiple Alarm Sounds

Possible Problem: It may not be feasible to allow the user to use their own custom sounds as alarm noises, due to compatibility issues, file type discrepancies, sound length, etc. As well as the need to store these files somewhere on the application.

Proposed Solution: Build all alarm sounds that may be useful (such as the previously mentioned lullaby melody) into the code itself, meaning they can be stored locally on the Pi.

Requirement: User Preferences

Possible Problem: Dynamically changing the sensitivity of the motion capture code as the user enters new preferences.

Proposed Solution: Connect to the VNC server once the user opens the preferences page, and use it to edit the line in code that determines the sensitivity of that specific camera, then shut down the VNC server once the user saves/confirms these preferences. This is an acceptable solution however it would mean the user would have to enter the password of the VNC server once they choose to save/confirm their preferences. Of course, the trade off to this solution would be the overhead required to do something relatively simple like change camera sensitivity, but it may just be the best (feasible) solution.

7.2 Challenges and Learning Requirements

Using the Raspberry Pi

The task of facilitating much of this system using a Raspberry Pi is certainly going to be a potential challenge and a massive learning experience for both team members, as neither have had any appreciable experience interacting with a Raspberry Pi in the past. We will need to quickly discover the basics of setup and configuration and how to use the Pi in conjunction with our external software (for example Skype, Emacs editor, Motion Detection Packages, VNC server, and of course programming on the device, either through SSH or directly). This will certainly prove a valuable learning experience for both of us, but will also be a significant challenge both in terms research and implementation considering our relatively tight time schedule, and it's intertwining with examinations.

Another challenge associated with the Raspberry Pi will be the development of Configuration Scripts which will allow the Pi to run our program correctly upon startup. This is an essential aspect of the final product when considering the user may not be tech literate, and even if this is not the case, could not be expected to SSH into the Raspberry Pi and run a terminal command each time the system is needed. This aspect of the project will facilitate the improvement in the basic bash scripting knowledge we have so far gained through this course and allow us to implement those skills in a real-life scenario for the first time.

No Python Experience

This is considered one of the largest challenges of the project and as such, a suitable chunk of our time schedule has been dedicated to completing this task. As well as this it has been placed very early on the schedule, which should allow any unforeseen problems to be dealt with as soon as possible. Both members are going into this project completely new to the language of Python. The general concepts surrounding Object Oriented Programming should, hopefully, carry over from our time spent learning Java and C++. However, developing our largest and most challenging program in an untouched language is certainly going to be one of the main challenges that this project will produce, yet when completed, will surely have provided us with an invaluable improvement to our programming capacities.

Creating an Android Application

For our project, we must build an android application, which again is something that neither team member has prior experience in. The basic design and construction of the application is expected (perhaps naively) to be relatively straightforward, due to the extensive wealth of help online and also our experience coding in OO languages. However, the main challenges will likely come with linking this application to external / 3rd party software such as Skype and Dropbox, and also the inclusion of features such as GPS using our own code. The research material on these features may be more limited and will certainly require more skill and thought when implementing them correctly.

Designing the application is also an opportunity to expand our skills in UI design and usability. As multiple modules have been focused on this topic over the course of the past three years, this is a valuable opportunity to implement what has been learned in a real-world scenario.

GPS Inclusion in Code

Including GPS functionality into the code whereby the Pi will be able to detect the user's location based on their android device with the application installed is another feature that we plan to include in this project and is also untouched territory for us. We have no Application development practice under our belts, as mentioned above, and to join with this we have also never implemented any sort of location based programming in any aspects so far in our past coding experiences. Although a proposed solution of using third party software has been mentioned, this is not an ideal solution, and we are confident that with enough research, we can overcome this problem using our own solution. It is certainly one of the largest areas of concern for the team.

No Dropbox Experience

Prior to the development of this project, neither member had any experience in using the Dropbox file storage system. This is an aspect of learning that is expected to be straightforward and relatively pain-free. However, the main challenge will of course come with correctly applying this to link correctly with both the Raspberry Pi, and our android application. Integrating third party software in programming is something that has never been done to this degree, and this in combination with assuring the Dropbx account is also synchronized with an android application (another unfamiliar hurdle) is certainly a challenge.