Surveillance Camera System

Table of Content

I. Project Background

1. Topic

Security has long been a concern for households and organizations. A secured site is the first step to ensure good life quality and productive work environment. However, traditional camera surveillance systems on the market is limited by their dissociation from users as well as their lack of remote accessibility. Facing such real-world problems, our HQFC project team is incentivized to create a more interactive and innovative variant for those surveillance systems. As Internet of Things emerges, the project team brings forth the idea of creating a camera surveillance system that allows users to supervise their facilities ubiquitously through a web application while also being able to control the camera angles. Furthermore, the skills and knowledge that the project team obtain from working on this project will certainly become useful for our future careers.

2. People

- **Name: Nguyen Hoang Chuong**
  - Student ID: s3651318
  - Email: s3651318@rmit.edu.vn
  - Phone number: 0934820601
  - Interest: IoT, Machine Learning, Cloud Computing
  - Relevant skills: Python, C, Arduino, Web Programming, Project Management
  - Primary role: Project Leader, Front-end Developer, Raspberry Pi Developer, Tester

- **Name: Tseng Chia Fu**
  - Student ID: s3634996
  - Email: s3634996@rmit.edu.vn
  - Phone number: 0909844458
  - Interest: UX/UI, Working With Hardware, Sketching/Making Hardware Prototype
  - Relevant skills: Python, Web Programming, C, Arduino
  - Primary role: Chassis Designer, Proofreader, Raspberry Pi Developer, Tester

- **Name: Pham Ngoc Minh Hang**
  - Student ID: s3672206
  - Email: s3672206@rmit.edu.vn
  - Phone number: 0901358246
  - Interest: IoT, Web Programming.
  - Relevant skills: Python, Web Programming, Node JS
  - Primary role: Arduino Developer, Android Developer, Trello Manager, Tester

- **Name: Pham Minh Quang**
  - Student ID: s3740924

- Email: s3740924@rmit.edu.vn
- Phone number: 0707945961
- Interest: Web Programming, UI/UX, Project Management
- Relevant skills: React/Redux, NodeJs, Python, C++, Project Management
- Primary role: Back-end Developer, Android Developer, Arduino Developer, Tester

3. Aim and Goals

The primary aim for this project is to explore and experiment the applications of the Internet of Things technologies in security surveillance. The system to be built must support live remote surveillance and direct control of the camera via the Internet connection. In order to achieve this set aim, the project team has to accomplish the following sub-goals:

- **Learning goal:** Although most of the team members have had previous experience in working with hardware like Arduino, the team will still have to do more researches and revision to understand more in depth. Specifically, we will have to figure out how to configure Arduino so that it is able to receive commands sent from a web application and execute them. Next, the project team needs to learn the method of sending live video feed from a camera to a web server with the use of Raspberry Pi. The team will also have to look into managing and coordinating multiple platforms and hardware together.
- **Building goal**: The project team's building goal is to produce a system that allows remote surveillance of a physical location and remote control of the camera; both through interactions made on a web application with secured access.
- **Testing goal**: Throughout the process of building the intended system, the project team plan on frequent testing to identify and resolve bugs and other problems. Ideally, by the end of the project, the team will have addressed and fixed all the critical bugs of the system. The system's main functions should work smoothly.
- **Team communication and coordination goal**: For the team to produce the best work in this project, we must ensure good communication and coordination between team members. Specifically, the team schedules weekly physical meetings while still keeping up with the project off campus by using tools like Facebook Messenger, Google Drive, and Trello. All the member are expected to respond to messages in a timely manner as well as staying informed about the current and upcoming tasks. The team will also be flexible in holding additional physical meetings if necessary.

4. Scope

- **Within scope:**
  - Users can watch the livestream video captured by cameras on a web interface through a Wifi connection

- Users can control the movements of the camera remotely through the web interface
- Utilize Raspberry Pi to control Arduino
- Utilize Arduino to control camera movements
- The camera motions include turning left and turning right
- The web application will be hosted using Amazon EC2 service
- The livestream will be implemented using Wowza Cloud service
- Secured access to the web interface which requires users to login to their account
- Playback from a livestream is saved and stored using cloud storage **(extra function)**
- User can download a playback video after a livestream ends **(extra function)**
- Camera motions can be controlled by an Android application **(extra function)**

- **Out of scope**
  - Camera cannot change angles vertically
  - Capture the image from the video and save it to the local storage
  - Capture screenshots from livestreams
  - The camera does not support motion detection.
  - Camera cannot turn more than 90 degree for each sides

II. Project Progress

1. Description

- **Group A (Raspberry Pi team, including Chuong and Fu):**

Initially, group A had faced many difficulties because no members had experienced with the core technologies, equipments used in this project before, which are Raspberry Pi and Wowza live stream service. However, by working hard, we had gradually surpassed all issues, finishing the live steam part on schedule.

The most challenging part group A has experienced is finding the library that has an ability to send stream from Raspberry Pi to the cloud. In week 5, we followed many tutorials on the Internet and successfully made a livestream using a webcam in our laptop and a software named OBS Studio. This made us think the process of streaming using Raspberry Pi could done easily by applying the same principle as streaming using laptop. Reversely, it was neither easy nor the same principle could be applied, because Raspberry Pi is not strong enough to handle a streaming software such as OBS Studio. After that, group A struggled a lot finding how to send video stream from RPi to cloud service. The team started thinking of changing our plan to use other livestream services. As a result, 2 back-up were made which are using Dataplicity and Youtube, however, these back-up plans have their own drawback. Moreover, if these plans were used, we could not achieve our learning goal which is able to create our own livestream using Raspberry Pi and cloud service. Although the backup plans have some drawbacks, they

helped group A to focus on our initial plan because at least, the project could be finished (but not with high quality). Consequently, in week 6, the team could totally focus on working with our initial plan and finally we are able to send the stream to Wowza cloud service using a library called Ffmpeg.

- **Group B (Arduino team, including Hang and Quang)**

At the beginning, the team planned to build a web application that can communicate with and control an Arduino remotely via the Internet. The team was inexperienced in using web technologies and hardware interchangeably. Merging multiple programming languages was a new and challenging concept. The task appeared to be daunting.

Later on, in week 5, the team discovered Johnny-Five, which is a JavaScript library that allows remote control of an Arduino. Traditionally, commands for Arduino are written with C. Using JavaScript is a step up for the team as it is also used for the web application, which eliminates the challenging task of integrating multiple programming languages on multiple platforms. We also found out about PubNub. PubNub is a third party Application Programming Interface (API) that helps establishing a connection between an IoT (Internet of Thing) device and a remote web server. After a period of reading the online documentation for Johnny-Five and Pubnub, the team started to take the first steps and made a web application to remotely control a LED attached to an Arduino board. Comprehending the concepts and working with these new technologies were initially difficult. However, the team was able to successfully create the prototype with more intensive research and referring to online documentation.

With the knowledge obtained from the first prototype, in week 6, the team moved on to work with the actual stepper motor instead of the LED. The calibration process for the motor was made simpler with our experience from the previous prototype. Nevertheless, we encountered an issue that prevented the motor from rotating counter-clockwise. After much research, reading and experimenting, the team figured out the problem which originated from the Arduino's firmware. The firmware was incompatible with our stepper model. Having the issue pinned down, the team modified the firmware code according to the stepper's datasheet and fixed the issue. We successfully made a small system that can remotely control a stepper motor through a web application.

- **The project team (including group A and group B)**

In the initial project timetable, integrating 2 parts together was the task in week 9 and week 7 was reserved for testing 2 part separately. However, because the team would have to show a demo to the lecturer in week 7, the team decided to to make a minor change in the project plan. As a result, all team members agreed that in week 7, the project team integrated 2 parts to have a first prototype to show to the lecturer. Thankfully, both 2 sub teams finished their assigned tasks on schedule (week 6) with high-quality and after 2 meetings, the first prototype of the full system was created.

Due to the reason that the team had no experience of building a website with high security, we decided to use Amazon Cognito to develop a Login function in week 8. As a result our web security have been more increased because the user account is managed and protected by Amazon Cognito.

In week 9, we conducted the first test on our product to make sure the core features of our camera works properly. At the same time, Fu started learning how to use Autocad to design a chassis but then we realized that we did not have enough time to learn a new software. Therefore we stopped learning and decided to pay for a third party service to do that. As the result, in week 10, the team had have a new chassis *(showed in image 1*) which could contain every components of our product. Additionally, since the project is ahead of schedule, we decided to add more features to our product. Consequently, 2 new features were added which are the Android Application to control the camera and video playback capture. We also tried to build a function, which is to help capturing pictures from the livestream video but we failed.

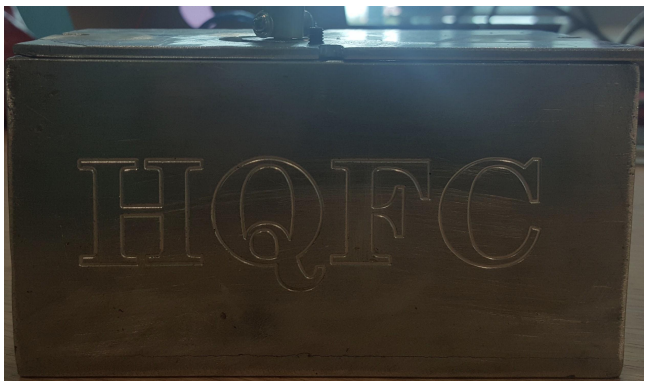


Image 1: Final chassis

In week 11 and week 12, the team decided to stop developing and we totally focused on testing the our camera to make sure it works without any problem in the presentation. Specifically, we requested assistance from fellow classmates to test our project in week 11. Moving to the final week, the project team had one meeting which only served for the purpose of testing and finalizing the product. More detail of testing phase (week 9, 11 and 12) is described in ***Test*** section.

2. Scope

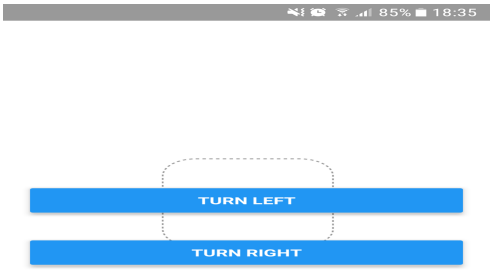
The initial scope of the project has changed from the beginning of the semester. We decided to expand the project by integrating more features into the security system. Consequently, we successfully developed 2 new functions which are Android application allowing user to control the camera with their phone, and the function to record livestream as a playback video . The team also tried to develop a function to capture image from the livestream video and save it to the cloud storage is also expected as an extra function, however, we failed to develop it. Therefore, capturing screenshot had been eliminated from the scope of this project.

3. Outcomes to date

Through the course of the project, the team has achieved the following outcomes:

- The final system: we have successfully built the originally intended surveillance system. Our final system consists of two components which are software and hardware.
  - Hardware: a metal chassis that contains a Raspberry Pi, an Arduino, a motor, and a camera. The camera is connected to the Raspberry Pi and is configured to broadcast its feed to our web application. The livestream has good image quality. The Raspberry Pi is also responsible of relaying the rotation commands to the Arduino, which then controls the motor accordingly.
  - Software: a secured and polished web application with users authentication and SSL certificate. Through this web application, users can view the livestream from the camera. Users can also modify the camera angles by interacting with the "Turn left" and "Turn right" buttons. The web application can be found at https://d2c62ivfvfrdl3.cloudfront.net

  ⇒ The final system is fully functional which can operate in a stable state. Overall, the project team have accomplished all the goals we previously visioned at the start of the project.
- Additional features: we have also extended some new features for our system to increase its usability
  - Android application: a mobile application running on Android that allows users to control camera angles without interacting with the web application.
  - Video playback capture: this feature allows every livestream to be recorded and stored in a cloud storage. After the livestream ends, user can download and watch playback videos.



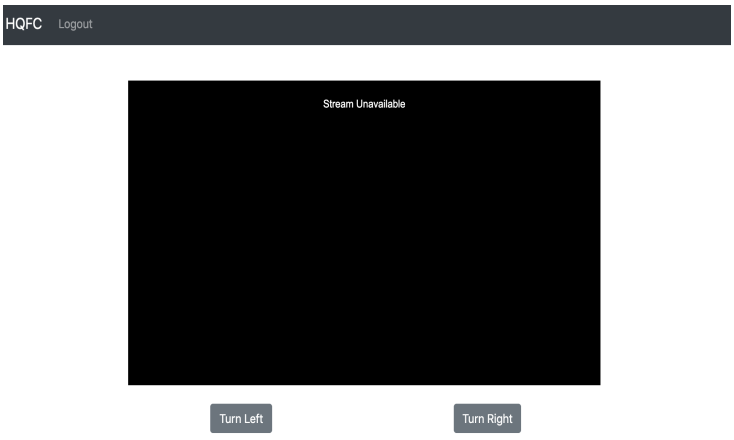


Image 2: Mobile Application UI *Image 3: Web Application UI*

Image 4: Final surveillance camera

4. Progress

From week 3 to week 6, everything had been proceed following the initial timetable. The team started slowly by spending week 3 making plan, collecting all equipments and week 4 was used for learning how to use new equipments (Raspberry Pi, Arduino) as well as new technologies (Wowza live stream service). After having gained enough basic knowledge for our project, in week 5 the team began the first steps to build a surveillance camera. Since the team had worked very hard in week 5 and week 6, 2 sub groups finished their task before Thursday week 6. At that time, the team's progress was slightly faster than the project plan.

However, first change was made in week 7 due to the reason that the would have a demo with the lecturer in this week. Consequently, the task of week 7 and week 9 were exchanged, resulting in the integration 2 parts, which was previously a task in week 9, became a new task in week 7. Likewise, the task of week 9 became conducting the first testing. By doing this, the team could deliver the first prototype to the lecturer and did not negatively affect the project schedule.

Except for the task exchange of week 7 and week 9, the schedule for the other weeks have been kept the same as the initial plan. In detail, the project team used week 8 for building log in function, creating a chassis, week 10 for developing extra functions. Finally we spent week 11 and 12 to test the camera many times. Since there is no big change happened to the schedule and the project team had sticked with the initial plan, the project was completed on schedule with high quality.

5. Testing

Test	Expected Outcome	Test Result	Solution

Livestream on web server using Raspberry Pi	Raspberry Pi successfully send stream to the Wowza cloud and the livestream could be viewed via the Web application	Problem with the free license: the live stream is delayed for approximately 30 seconds and the maximum length of the livestream is 15 minutes	Purchase a better licence for higher frequency cloud service
Arduino engine - motor rotation	The motor could be control to rotate left and right via the Web application	The motor worked properly without any problem	
First integration of Arduino and Raspberry Pi	The motor rotate the attached camera successfully. Allowing a smooth 180 degrees turn each sides. And livestream happens without any issues	- Have not had a chassis to contain all components. - The camera footage was upside down. - USB cable of Arduino sometimes did not work	- Attempt to build a chassis for the product - Buy longer cable to correct the camera footage - Buy 1 cables to use and 1 backup cable
Usability Test	Gaining thoughts of the product from fellow classmates as well as discovering bugs and issue	- No bugs and issues were found - Test user's comment: Lower latency livestream, UI aesthetic improvement	- Attempt to apply recommendation by designing a front-end
Final Test	- Raspberry Pi Camera able to stream live footage - Motor can be controlled via web app and mobile app - Everything will work while being contained inside the chassis.	- Motor was fried due to heavy usage and testing, resulting unable to turn or support Pi Camera correctly every time.	- A replacement motor for the fried motor was made

All test plan went accordingly to what was expected, by finishing two seperate parts and ready them for merging. After merging the project, few testing sessions was made for the merged prototype. Since there were no considerable issues with the prototype, therefore it did not affect the project too much, thus there were no changes that was made on the final product.

While there are no changes regarding the final product, but as mentioned above that the test plan went well, but the scheduled test plan session had been adjusted from Week 9 to Week 7. This change allows the team utilize the time better, and manage the rest of the project reasonably.

Week 11 and week 12 testing plans also went smoothly according to plan, week 11 testing focused on usability of the product, we asked our fellow classmates for a usability test and documented the process. The test allowed us to gain more insights regarding the whole of the project with minor bugs or issues as well as gathering the thoughts of the test users themselves. In week 12 a final testing was performed to check the integrity of the project for a smooth presentation that follows in the same week.

6. Tools and Technologies

Tools	
Visual Studio Code v1.29.0	This tool allowed the team to merge Arduino camera movement function and raspberry pi livestreaming together into one web application, displaying the livestream footage while having camera control button on screen.
Google Drive	The team uses Google drive to share data and important files containing codes together, also connecting to google doc which serves as a platform where the team is allowed to edit one document together, cutting down on integration time as if using traditional document application.
Trello/ Trello iOS v4.6.1/ Trello Android v5.3.0.11552-production	Trello is a task management app that gives you a visual overview of what is being worked on and who is working on it, this improves the team task management significantly compared to assigning tasks on other chat platform, reducing the time and effort to memorize each task.
Facebook Messenger iOS v192.0/ Android v192.0.0.31.101	Messenger acts as an important aspect for communication within the team. Discussion of group meetings, project planning or issue occurring, etc.
Arduino IDE v1.8.7	This tool uploads the code regarding to control the motor with Arduino to Arduino board. Thus

	writing new code or overwriting old codes onto the Arduino board memory.
Autocad 2019 23.0 (**Removed**)	AutoCAD is a 2-D and 3-D computer-aided drafting software application used in architecture, construction, and manufacturing to assist in the preparation of blueprints and other engineering plans.
Expo.io (**New**)	Expo is a free and open source toolchain built around React Native to help build native iOS and Android projects using JavaScript and React.

Technologies	
Wowza Engine v4.7.6	Wowza Engine is used to receive a stream from Raspberry Pi on local network, after it will send Raspberry Pi stream to Wowza Cloud
Wowza Streaming Cloud (**New**)	Wowza Streaming Cloud is used to make a livestream on Web server. It helps to receive a stream from Wowza Engine, and after that provide an embedded code to embed to the website
Ffmpeg library (**New**)	FFmpeg is designed for the processing/transcoding of video and audio, what makes FFmpeg different from others is that it is a command-line-based program. Therefore, it is easier and possible to use on Raspberry Pi.
Johnny five library (**New**)	Johnny-Five (J5) enables JavaScript programmers to develop around embedded development boards such as an Arduino, Raspberry Pi or Tessel 2. Johnny-Five supports more than X different development boards for use in robotics, IoT and other hardware interactions.

Pubnub (**New**)	PubNub is a global Data Stream Network (DSN), it helps software and hardware developers to build realtime web, mobile and Internet of things applications.
Raspbian v4.14	The OS of Raspberry Pi, must be used in order to access other tools using Raspberry Pi.
EC2 from AWS	AWS (Amazon Web Server) is used to host the web application that is responsible for showing camera footage and controlling the camera movement.
Amazon Cognito from AWS	Used to build Login function. The team decided to used it instead of building our own user management because it helps to increase the security of our website since user account is managed and protected by Amazon Cognito
Javascript v1.8.5	A technology that is used with the tool Visual Studio Code to make functions for the web application.
Python v3.70	A technology that is used with Arduino IDE to control the movement of the camera
HTML5	HTML5 is used to display the web application, in this case, the button placement and livestream footage placement.
CSS3	Using this for web application to customize the aesthetic of the website UI.
Windows 10 Version 1809/ Mac OS Mojave	Main technologies utilized for research and building purposes.
React Native (**New**)	React Native is a JavaScript framework for writing real, natively rendering mobile applications for iOS and Android.

Advanced Firmata (**New**)	Allows Arduino to work with Johnny Five Library.

Equipments	
Arduino Uno R3/ 5 LEDs/ 5V DC Step motor	These are used for motor controlling section as well as data transfer testing.
Raspberry Pi 3/ HDMI Cable/ SanDisk 16GB Micro SD Card	These components are necessary to operate Raspberry Pi.
Raspberry Pi Camera Module 2 (Replacement of USB camera)	USB camera is replaced with Camera Module 2, which is a camera specifically designed for Raspberry Pi. The reason for changing of equipment is due to the Camera Module 2 is lightweight and easy to maneuver around the chassis. Moreover, Camera Module 2 support ffmpeg library used for sending stream, whereas USB camera does not.
Ethernet Shield for Arduino **(Not used)**	In the process of making the project, the team found out that Raspberry Pi is able to connect to Arduino. Furthermore, Raspberry Pi is able to host an internet connection onto the Arduino as well. This allows the usage of Pubnub to connect Arduino to the web application.

IV. Elevator Pitch

Our surveillance system is equipped with a high-quality livestream from a camera that can be viewed and controlled on a secured web application or a smartphone, surpassing traditional systems in term of interactiveness, accessibility and availability. Built with the highest working standards and constantly checked for errors, our product is well-built and well-tested. Compared to traditional surveillance cameras, our system has many strengths of an IoT devices and it has large potentials to completely change how people monitor their properties. If our system could get

invested, we will improve the livestream latency and make our product more economical.