FIT3161 Project Proposal for Group 5: Remote Video Surveillance System Project

TABLE OF CONTENTS

1. Introduction

Our project, proposed by Clayton Campus tutor Daniel Jitnah, involves the act of constructing a scalable multi-client video monitoring surveillance system. The project involves employing multiple computers as nodes in the system, with a central computer acting as the main surveillance network, and utilizing USB webcams hooked to computers, to perform motion-detection of its area. The aim of this proposed system is to address resource concerns commonly seen in other public surveillance models whereby now, video streams will be sent in short clips to the server only when valid motion has been detected by the camera's surroundings. A motion detection methodology will be implemented in order to achieve our project aim. Our project will be implemented using two development languages, C++ and JavaScript. C++ will handle the video alteration side of the system, utilizing the popular open source computer vision library OpenCV. JavaScript is used for implementing the web application and server/client-side interactions in the system, with the application deployed using the MEAN Stack. A hybrid development cycle will be utilized during project initiation, with the waterfall model and scrum methodology being combined together. Project deliverables are developed and updated along the way, with progress reports and sprint backlogs being key documents for pushing project development. A detailed testing plan for software quality control will be developed via analysis of high-level requirements of the project before final deployment.

Word Count: 6000 Words

2. Literature Review

2.1 Introduction

The aim of our literature review will be to explore the previous research that has been conducted in regards to our project, with the aim of supplying and contextualizing the background for study, as well as to discuss related research, provide synthesis of previously published works and their implications, and finally, to critically evaluate contributions & shortcomings of past research in order to supplement our rationale for pursuing our project of study.

We have conducted our literature review based on 10 related papers regarding Motion Surveillance, with the aim of resolving questions surrounding our understanding of our project's design & implementation. The following key topics will need to be researched extensively during the Literature Review:

1. Surveillance System Designs
2. Motion Detection Algorithms
3. Real-Time Transmission Protocols
4. Software Libraries & Stacks

At the conclusion of our Literature Review, we hope to have gained a much better understanding of our project's context, and will be able to effectively answer the following questions/enquiries:

1. How has similar projects conducted in the past approached the problem, and how will our project differ from past projects?
2. Analyze online resources integrating current technologies that we will require (such as transmission protocols, web protocols, database design, Virtual Machine Design etc.) in our project in order to provide for us a broader contextual understanding of how we will integrate technologies into the project.
3. Based on current research regarding motion detection, which technique best suits our needs for the project?

2.2 Source Selection Criteria

The usefulness of our Literature Review depends on our ability to scout literature that possess huge use to us. We developed some selection criteria to allow us to assess a paper's usefulness:

1. What is the resolution and hypothesis presented in the literature paper? Are the questions and answers presented useful, and holds value to our project's context?
2. What is the main form of research conducted in the literature paper? Is the data presented in the paper quantifiable and measurable for testing?
3. What is the currency of the literature paper? Are the methods and data presented still accurate and updated to today?
4. How credible is the literature paper? Are the author's paper's widely read in the industry? Have they been cited a number of times in external papers? Is the paper endorsed or published under a credible organization?

2.3 Review of Key Topics

Our literature review will be conducted based on the premise of the four key topics proposed. The sources we have chosen are analyzed extensively regarding their credibility and use, with the Review Table shown in Appendix A. From here, we will mention our sources based on their respective Paper ID's, shown in Appendix A.

2.3.1 Surveillance System Designs

Four papers have been reviewed, with all of them proposing unique solutions for a surveillance system. All four papers show high traits of credibility, with 3 of them being published under ResearchGate or IEEE, and another being published under Princeton University, a highly recognized educational institution. The currency of all four papers are positive, with the exception of Paper 1, which is published during 2011, However, the technologies proposed and used in Paper 1, are industry-leading, and is still considered the industry standards of today (2019), with specifics being aimed towards the use of the H.264 video compression codec, and the very relevant transmission protocols, RTP, and UDP. The most-cited paper, which was Paper 4, had 97 external papers citing the paper, demonstrating its immense credibility and reliability. We have discovered that of the papers presented, video data transmission are performed either using MJPEG (Motion JPEG) as seen on Paper 4, and the H.264 Codec, seen on Paper 1 and 3. We have decided to encode our videos using the H.264 codec after analysis of the papers, this is because MJPEG video compression compresses videos frame by frame, whereas H.264 compresses across frames. This results in differing priorities between the use of the two compression techniques, MJPEG produces higher quality images, however, will use a larger bandwidth compared to H.264. As our project aims to address resource overhead concerns during wireless transmissions, and also during storage, we have decided upon the H.264 codec. A common trait that we have seen in the 4 papers, are that the motion detection algorithm utilized in all the papers, are always either performed via background subtraction, or frame differencing techniques. In the context of our project, efficiency and computation times are of utmost importance, alongside detection accuracy, and the choice of our motion detection algorithm will be elaborated on in the next section. Another common concept seen in the four papers is the term 'Edge Computing Node', which is the process of performing computation near the area where computation is required, instead of at the central location. This perfectly describes our project's design of having local computers performing the bulk of the computation (motion detection, video transmissions, etc.) at the site. All in all, from analysing the surveillance system design papers we have researched, we have been able to gain a much better understanding of the context of our project, with key design components and their choices, being easier to make after analysing the four papers. These key design choices researched are, the video compression algorithm, the motion detection algorithm, and the system design breakdown.

2.3.2 Motion Detection Algorithms

The importance of selecting a motion detection methodology, and implementing the motion detection methodology as a custom-made algorithm for our project, cannot be understated. The key backbone of our project relies in delivering an efficient, computationally fast, and accurate motion detection algorithm. Hence, alongside the first four motion detection algorithms proposed by the surveillance system design papers, we have dedicated an additional four papers focusing specifically on motion detection methodologies, which are Paper's 5-8.

<u>Summary of Motion Detection Methodologies Research</u>

Paper 1: Analyzes motion Detection using three motion detection algorithms. Background Subtraction method produces too much noise, not suitable.

Paper 2: Provides a good overview of the types of motion detection algorithms such as Background Subtraction, Frame Differencing, Template matching, Foreground detection.

Paper 3: Utilizes a state-of-the-art human detection algorithm called "Open-pose". This is an open source algorithm for detecting human body joints in videos. We are not able to directly incorporate this algorithm into our project as we plan to construct our own, however, we can utilize methodologies and ideas used in OpenPose, for our Human Motion Detection algorithm.

Paper 4: Utilizes several motion detection algorithms for different purposes. a "Harr Cascade classifier" algorithm is used for face detection, whereas "CareSafe", and "WalkSafe", are algorithms used to detect cars and lanes on the road. The surveillance system proposed is able to determine the algorithm to use depending on the situation. Since our project mainly focuses on human motion detection, we may utilize features presented from the "Harr Cascade classifier" algorithm.

Paper 5: Proposes a new background subtraction algorithm called "ViBE" which is hugely popular across the industry. The paper analyzes many trends of other background subtraction modelling methods, and talks about how it is difficult to evaluate different background subtraction algorithms, as there lacks a standardized evaluation framework. The paper also clearly states how "ViBE" is different, and a justified necessity in the motion detection algorithm academia.

Paper 6: Proposes a combination of Frame Differencing and template matching algorithm for performing object tracking. The algorithm does not require color information to track objects, and is shown to be accurate for detecting a single moving object under bad lighting conditions.

Paper 7: Evaluates the performance of three different visual motion detection algorithms: Frame Differencing, Background subtraction, and a modified frame subtraction algorithm. The paper discusses the advantages and disadvantages of the algorithms. Background subtraction offers the best motion detection accuracy, and motion tracking, but however, is the least computationally efficient, as it took the longest to iterate, out of the other algorithms. Frame subtraction on the other hand, is the fastest algorithm, but however produced too many false positives. This was because the simple frame subtraction method had no logic for dismissing repetitive movements. The modified frame subtraction algorithm modifies the simple version such that it could ignore repetitive movements. This paper proposes interesting implications, as the purpose of the surveillance system will impact on the decision of the motion detection algorithm. Our system's purpose is to detect motion occurring, but not to track the object's movement over time, hence, we only want to mark one instance of movement, instead of marking and tracking the detected object. Hence, the modified frame subtraction algorithm will be particularly useful for our project, and we can utilize the logic presented in the algorithm.

Paper 8: Paper which has a similar purpose to Paper 7, comparing background subtraction algorithm against two other frame differencing algorithms with regards to tracking a moving object. The same conclusions arose from this paper, that is, frame differencing technique is more computationally efficient than background subtraction technique.

<u>Evaluation of Motion Detection Methodologies Research</u>

We have analyzed a great deal about motion detection, as shown in the eight papers which proposes their motion detection methodologies. We have come to understand various techniques of achieving motion detection such as background subtraction, Frame differencing, and template matching, alongside their implications when it comes to computational efficiency, detection accuracy, real-time reliability, and obviously, the difficulty factor when we start implementing them (as our motion detection algorithm). As a summary, we have decided to implement our motion detection algorithm using a frame differencing technique, with a key factor, which is to not register the same object more than once, after initial detection. We believe this methodology is the most suitable for our system as we do not want to track objects, rather only detect when objects become present in the frame. Another reason for our choice is that, based on research of the papers shown, frame differencing technique is far more computationally efficient when compared against other techniques, and is also far easier to implement.

2.3.3 Software Libraries & Stacks

Software Libraries & Stacks are an important part of how we are going to build up our system. As our project members have had sufficient experience working with the MEAN Stack (a software stack for developing dynamic web applications), we have decided to focus more on video frame modification libraries. OpenCV (Open source computer vision) caught our eye, it was the most popular library for performing real-time computer vision modification, and has been in continual development since 2000, showing how mature the library has become. We analyzed Paper 9 in order to gain some understanding of how we can utilize OpenCV when performing video frame analysis of our USB webcams. Paper 9 gave us insights into how we will read, write, and display videos from the web cameras, into our computers for analysis, with our development language of choice, C++.

2.3.4 Real-Time Transmission Protocols

The selection of an effective transmission protocol is important in regard to video file transmission across our local computers to the central server. We need to choose a good transmission protocol which will ensure the best transmission efficiency (bandwidth saved) while also ensuring fast transmission times to handle the real-time aspects of our system. Paper 10 gave us insights on the various internet protocols used for real-time multimedia communication, discussing RTP (Real-Time Transport Protocol), RTCP (Real-Time Transport Control Protocol), and RTSP (Real-Time Streaming Protocol). Paper 1 also provided some insight, stating that the TCP (Transmission Control Protocol), “is unable to meet the requirements of real-time streaming media” (Ge & Shan & Kou, 2011). However so, TCP is useful in establishing the congestion control and quality assurance of the communication channel, which may be useful for ensuring the reliability and security of our system’s communication channels. After some analysis, we have decided to use RTSP as our video streaming protocol, as it is an extension of the RTP and RTCP, and is supported by the web application technologies that we will use, specifically, the MEAN Stack. Furthermore, RTSP is flexible, and can utilize other protocols such as UDP and TCP to provide additional functionality such as authentication, security, and reliable connections. Finally, RTSP also offers lower latency and network overheads in multimedia streaming as compared with other protocols such as HTTP, which is important for the real-time aspects of our system.

2.4 Review Conclusion

We have successfully conducted an effective Literature Review based around 10 papers surrounding key topics of our Motion Surveillance Project. Four Papers address the first key topic, Surveillance System Designs, Another Four Papers address the second key topic, Motion Detection Algorithms, and the last two papers address the final key topics, which are Real-Time Transmission Protocols, and Software Libraries, respectively. After the literature review, we have gained a much better understanding of our project’s context, and are now much more capable in our project’s system design and implementation stages. We are now able to answer our enquiries proposed in the beginning with much ease. Our project differs hugely in the aspect where we maximize efficiency during motion detection and video transmission. We are now also able to confidently discuss the technologies we plan to integrate, and how they structure together (such as H.264 codec, RTSP, MEAN Stack, Frame Differencing algorithm etc.).

3. Project Management Plan

To achieve a successful development of the project, the project management plan is presented here in terms of project scope, project organization, management process, and schedule and resource requirements.

3.1 Project Overview

The objective of the project is to construct a scalable multi-client video monitoring surveillance system. The system aims to address resource concerns commonly seen in other public surveillance system, such as a large amount of useless video data and a large bandwidth transmission. Our surveillance model sends a short clip of video to the server only when a valid motion is detected in the coverage area. To measure the progress of the project, the following key major milestones are identified:

Milestone 1: Develop the Motion Detection Algorithm

Milestone 2: Develop Web Application

Milestone 3: Set up Database Schema

Milestone 4: Set up Video Connections & Database Connections

Milestone 5: Link Wireless Video Cameras to Central Network

Milestone 6: Project Testing

Milestone 7: Deliver Final Product

3.2 Project Scope

To facilitate project scope, direction, and progress, a set of project deliverables have been documented, alongside important high level functional/non-functional project requirements. Key qualities have been documented, which will act as the product user acceptance criteria.

3.2.1 Project Deliverables

The components diagram we have constructed of our system (Figure 2 of Appendix), allows us to clearly articulate the deliverables for this project. We are able to see that our system will be constructed based on large modules of the following:

1. Motion Detection Module: A module that is able to detect motion from input videos
2. Video Capture Module: A module that enables the local computer to read and write videos that are captured from its USB webcam.

3. Database Design Module: Module that integrates database schema and integration into video store of the central
4. Web Application Interface Module: The user interface design that refers to the layout of elements in the web application to be constructed.
5. Web application Module: A fully functional web application that integrates the server side, client side and database.

Alongside product-related deliverables, some project management-related deliverables will include:

1. Work Breakdown Structure – Document illustrating how the project's deliverables will be broken down across team members
2. Project Schedule – Document illustrating the estimated completion times for each project milestone
3. Project Requirements Traceability Matrix – Document illustrating all product requirements, alongside test validations to ensure all requirements are met
4. Project Risk Register – Document illustrating all potential risks, alongside tactics to mitigate or accept the risk
5. Product backlog – Document showing the progress that is occurring so far with the project
6. Project Progress Report: A written document that is used to monitor and control the project process
7. Code Review document: A written document that documents the review session of any bugs/fixes/improvements to be made to the current system code.
8. System Manual: A manual that explains how the system works and gives instructions about how to use the system.

3.2.2 Product Characteristics & Requirements

1. Real-Time Video Surveillance – Central Server is able to tap into any local camera feed at any time
2. Multi-Client Capability – System is scalable to as many local camera's possible, with the central server monitoring all
3. Motion-Detection Capability – Local camera's implement a motion detection methodology in order to detect movement in videos
4. User-Friendly & Accessible – Web application developed is easy to use and accessible

5. Efficient Bandwidth consumption – System as a whole does not consume large amounts of bandwidth
6. Organized Data Storage – Data gathered in the system is systematically organized using a database schema
7. Web Application Server – Implementation of the web application server will allow for monitoring and management capabilities

We have documented all our project requirements into a requirements traceability matrix, seen in Figure 3 of the Appendix, listing all functional and non-functional requirements stated above. This document will allow us to test that all requirements are met during the project timeline.

3.2.3 Product User Acceptance Criteria

We have derived a set of user acceptance criteria based on the critical specifications require for the project, and defined them based on 3 qualities: Performance, Usability, and Reliability/Availability.

<u>Performance Qualities</u>

1. Video frames processed and stored in the database must be of the same quality as the camera's live stream.
2. Processing time for writing a specific timestamp of the live stream into the central server upon motion detection must be fast (0-5 seconds).
3. Web application developed to perform live streaming and management operations must be fast when performing web and database operations
4. The system must be able to connect at least four cameras at the same time
5. The system is able to view multiple live streams at once from the central computer
6. The latency of living streaming must be less than 1 second
7. The accuracy of motion detection must be more than 70 percent

<u>Reliability/Availability Qualities</u>

1. Web application developed must be robust and free from fatal crashes.
2. If a local camera happens to be unavailable, the central monitoring computer is able to immediately notice.
3. Past stream recordings must be stored up to 7 days
4. Every local camera must be able to consistently write video frame data towards its respective local data store with accurate timestamps without delays.

5. Central monitoring computer is able to tap into any local camera at any time without fail.

Usability Qualities

1. The supervisor operating the central monitoring computer should be notified of motion detection of an unspecified site in a way that is clear.
2. Usability of the web application developed to inspect live streaming and perform management operations should be easy to use and intuitive.

3.3 Project Organization

3.3.1 Process Model

We will adopt a hybrid development model for our project, combining Waterfall and Scrum methodology. Since our project is constructed from multiple large modules, with project requirements basically fixed, a waterfall methodology is useful here. Scrum methodology comes in during the system development stage of the module. Rationale for this is so we can quickly develop the module, and ensure that it will be functional fast.

3.3.2 Project Responsibilities

In our project, high-level functional requirements stated previously already highlights the major technical system functions and activities. Some major project management functions and activities are listed below:

1. Executing regular Scrum Meetings – Important for kicking off the project development phase, coordinating group tasks, addressing all confusions
2. Project Deliverables logging – Important step, where an individual has to manage all the current deliverables to update as the project progresses such as risk register, project schedule, meeting minutes, progress report, testing logs etc.
3. Code Review Sessions: Session which discusses current code produced so far, with written document produced that documents the review session of any bugs/fixes/improvements to be made to the current system code.
4. System Manual: A manual that explains how the system works and gives instructions about how to use the system.
5. Hosting conference call sessions – Important for group meetings when members are away from each other
6. Final System Testing – Important testing phase during end of project development
7. Module Testing – Important testing phase during end of each module development cycle

A Responsibility Assignment Matrix is created, showing the person in charge for each function or activity.

Responsibility Assignment Matrix

Project Activity/Function	Individuals Responsibility
1	Cheng Zeng
2	Terence Ng
3	Peisong Yang
4	ALL
5	Cheng Zeng
6	ALL
7	ALL

3.4 Project Management Process

3.4.1 Risk Management

In order to effectively manage potential risks in our project, our group underwent a 'risk identification' brainstorming session, where everyone undergoes a creative session of coming up with a list of everything that could go wrong with the project. We then proceeded to undergo an evaluation process, documenting the risk and impact of a potential risk, as well as identifying the root cause of a risk. After identifying and evaluating the risk, we came up with a risk strategy for each risk, assigning priority scores. All risk strategies are considered such as risk avoidance, risk sharing, risk reduction, and risk transfer. The risk register we have developed is shown in Figure 4 of the Appendix.

3.4.2 Monitoring & Controlling Mechanisms

To achieve the objective of monitoring and controlling the project process, we will compare and evaluate the actual results against the planned project plan. In our project, a project review meeting is held every week, in which we review the status of the project. A progress report is produced every week after the project review meeting, summarizing what tasks we completed, what objectives were met, and how well we performed in the current week. The progress report which will be documented every week, will consist of five main sections:

1. A brief outline of project progress up to present point
2. Measurements of deliverable produced to current point
3. Comparison of the deliverable produced, against the expected deliverable produced in the project schedule, then identifying and evaluating the differences and their causes.
4. Identification of potentially new risks and obstacles that have occurred or will occur
5. A brainstormed plan which will outline corrective actions to allow the group to keep on track with the project schedule.

3.4.3 Communication & Reporting Plan

The following communication mechanisms are brainstormed to ensure effective communication & reporting during the project:

1. Facebook messenger or email is for informing minor issues
2. Phone call or virtual meeting is for discussing major issue and notifying emergencies
3. Face to face meeting is most important and crucial, held for various purposes such as scrum meetings, and for milestone meetings.

These three mechanisms can increase collaboration and ensure every team member is on the same page through the whole development process.

The progress status report format has been constructed in a standardized format shown in Figure 5. A meeting minutes report format has also been created, shown in Figure 7. The sprint backlog template is in Figure 6.

The communication matrix below details the communication & reporting requirements for the project.

Communication Matrix

Communication Type	**Objective of Communication**	**Medium**	**Frequency**	**Deliverables**
Project review meetings	Review status of the project	Online	Weekly	Progress Report
Scrum Sprint Planning Meeting	Come up with the plan of a new sprint	Face to face	Once every two weeks	Sprint Backlog
Daily Scrum meeting	Discuss what every member is up to	Face to face	Twice a week	Meeting Minutes
Sprint review meeting	Retrospect the recent sprint	Face to face	Once every two weeks	Retrospective Report & Audit Report
Interface design meeting	Discuss and produce the design solution	Face to face	Once	Web Application Interface Design Documentation

3.4.4 Review & Audit Mechanisms

Since we apply hybrid project methodology, the whole project is split into sprints. The audit mechanism will be conducted in a sprint review meeting at the end of the sprint. During the meeting, the products of a sprint undergo quality assurance testing according to the sprint backlog and the product user acceptance criteria. If the current deliverable does not cover all sprint backlog or fail to meet the acceptance criteria, a follow-up session, including adjustments of the sprint backlog and refinement of the deliverable, is scheduled for the next sprint.
All project documents such as the progress or audit report, will be kept in a document repository (i.e. Google Drive), ensuring that all members are able to review the project process at any time. Software deliverables are stored in GitHub to achieve version control.

3.5 Project Schedule & Resource Requirements

3.5.1 Project Schedule

By dividing the project into parts and estimating the duration of completing each section, we made the project schedule form which is presented in Figure 8 of the appendix. In addition to the project schedule, we also constructed a work breakdown structure (Figure 9) to highlight key areas each member is responsible for leading. Something to note is, since the project life cycle runs under a Hybrid model, there is no need to specify a dependencies diagram, as modules are developed independently one by one, with system integration happening near the end of the project.

3.5.2 Project Resource Requirements

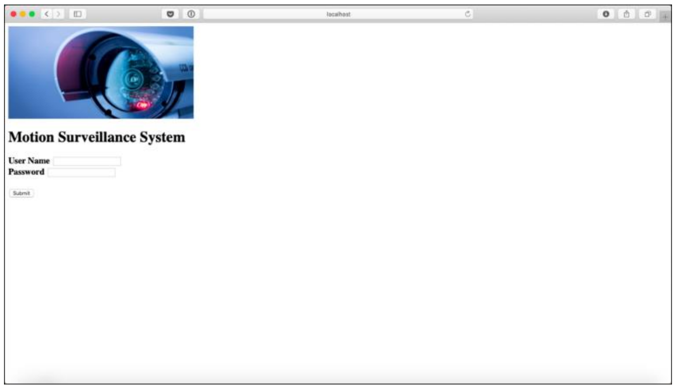
The human resources needed for the project are the group members. We will need everyone to be responsible for the motion detection algorithm section, database section and web application section. Members will take part in each development cycle, with the expert for each major section taking control of the Scrum meetings and spring backlog. Furthermore, the software we need (Visual Studio for C++, MongoDB or other database language and Visual Studio Code for Javascript) are all open-source free software, which we do not need to maintain ourselves. The situation for hardware differs, we will need local USB webcams for the sake of video-surveillance, and a laptop for each webcam. This means that we may require a lot of hardware when testing out our system, due to the scalable nature of our project. On another note, the central server we use, will need to be able to store large amounts of video data.

4. External Design

4.1 User Interface

The user interface for our system will be defined under the web application. The user can perform all system operations here (e.g. view all live streams, check out past streams, perform CRUD operations on videos etc.). Some prototype images for the web application can be seen below.

Login Page – Prompts the supervisor to input a username and password to start using the web application.

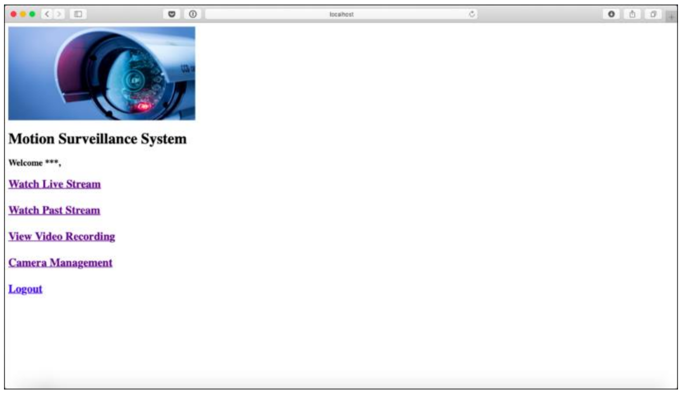

localhost

Motion Surveillance System

User Name
Password

Submit

Home Page – Displays all functionalities of the system.


localhost

Motion Surveillance System

Welcome ***,

Watch Live Stream

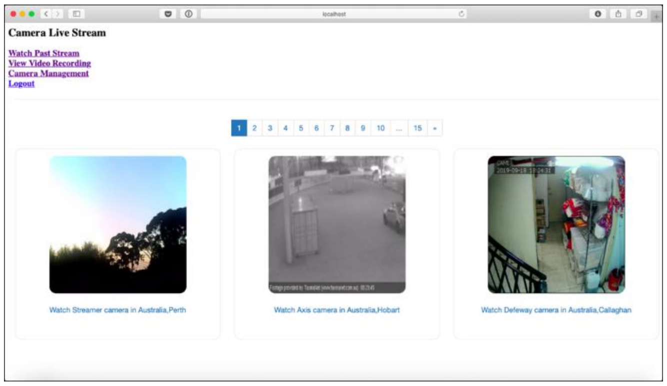
Watch Past Stream

View Video Recording

Camera Management

Logout

Live Stream Page – Shows all available camera's live stream. The supervisor can click on any specific camera to direct to that cameras page, displaying a better video resolution.


localhost

Camera Live Stream

Watch Past Stream
View Video Recording
Camera Management
Logout

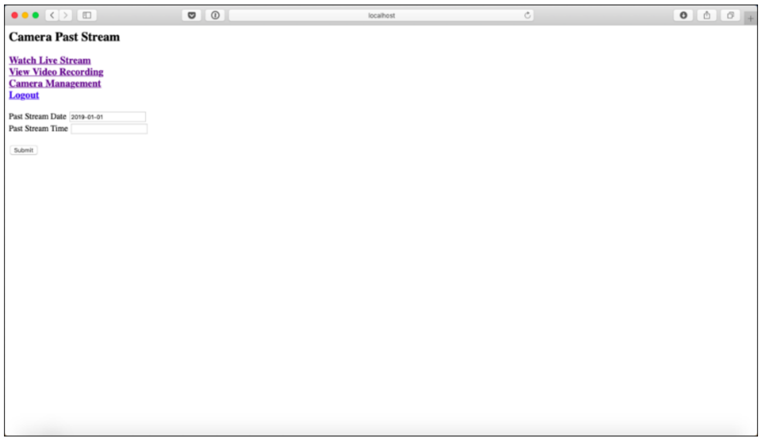
1 2 3 4 5 6 7 8 9 10 ... 15 »

Watch Streamer camera in Australia,Perth

Watch Axis camera in Australia,Hobart

Watch Defeway camera in Australia,Callaghan

Past Stream Page – Prompts supervisor to input the date and time of the past stream they intend to watch. After submitting, the web application plays a past stream.

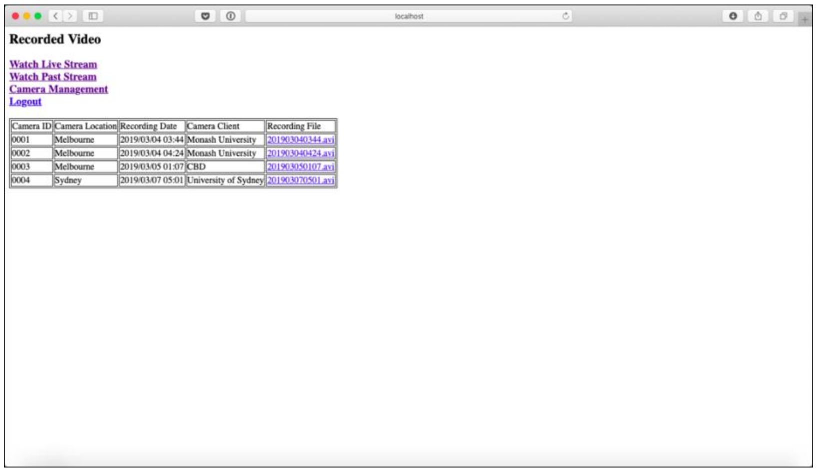


Camera Past Stream

Watch Live Stream
View Video Recording
Camera Management
Logout

Past Stream Date 2019-01-01
Past Stream Time

Motion-Detection Recordings Page – Displays a table of video snapshots of camera streams that have experienced motion detection. The supervisor will be able to filter the table based in the camera ID, location, client, or the time of recording. A recording file is stored which by default, will be a 20 second clip of the motion detected video.

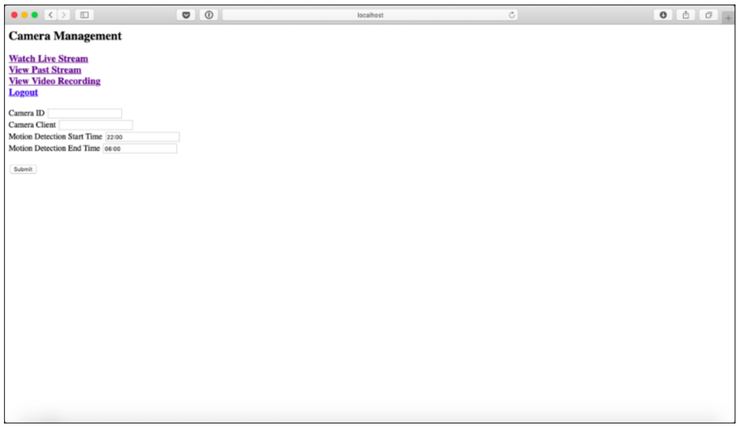


Recorded Video

Watch Live Stream
Watch Past Stream
Camera Management
Logout

Camera ID	Camera Location	Recording Date	Camera Client	Recording File
0001	Melbourne	2019/03/04 03:44	Monash University	201903040344.avi
0002	Melbourne	2019/03/04 04:24	Monash University	201903040424.avi
0003	Melbourne	2019/03/05 01:07	CBD	201903050107.avi
0004	Sydney	2019/03/07 05:01	University of Sydney	201903070501.avi

Camera Management Page – Enables the supervisor to specify and alter behaviors of its local cameras. Functionalities are subject to change and get extended.



Camera Management

Watch Live Stream
View Past Stream
View Video Recording
Logout

Camera ID
Camera Client
Motion Detection Start Time 22:00
Motion Detection End Time 06:00

4.2 Packages & APIs

Not many packages and APIs will be used as we plan to implement an original system, however, we will use OpenCV in order to perform video alteration, and use the MEAN Stack as the fundamental technology to build our web application.

4.3 Time & Space Characteristics of Software

The reaction time of our web application will be similar to normal websites online. The motion detection algorithm will be able to process the video stream in real-time and should be able to process detection fast (within seconds). The quality of the live streaming function of the cameras to the central server will largely depend on the implementation based on the real-time streaming protocol employed. The space performance of our system will need to be higher since we have to store large amount of data(videos) on the server. To improve space performance of our program, we will use the H.264 video compression codec to store files before we upload them to our server. An ideal storage required for a 24-hours-video should be less than 1 Gigabyte.

5. Methodology

5.1 Toolchains & Approaches

Two main programming languages will be utilised to develop our project, they are C++ with OpenCV and JavaScript with MEAN Stack.

5.1.1 C++ with OpenCV

C++ will be used to implement the functional requirements for video stream processing. During any video camera stream, the video will consist of multiple frames per second which make up the stream. To process this, we utilize a crucial library called OpenCV for read/write operations. We plan to utilize the C++ compiler for performing video read/write operations which offers fast computation. The IDE used for development will potentially be Visual Studio due to its support for C++. C++ will also be used to develop our motion detection algorithm, which will be embedded into all the IP cameras of the system.

5.1.2 JavaScript with MEAN Stack

We will use Javascript with the MEAN stack to build a web application which can service all the central monitoring functionality alongside perform management operations on our system.

Web Application UI and Design Functionality

The web application's User Interface will be built using Angular.JS, a client-side framework offered from the MEAN Stack. It allows us to create a dynamic and responsive web interface. The application will offer the live streaming of all local cameras in the system, and the ability to perform management operations such as altering daytime/night-time monitoring specs, performing CRUD operations on the database etc.

Web Application Database

The central database will store the client's information, motion capture recordings of all edge nodes alongside relevant information, such as occurrence location and time of motion detection. Database is established via MongoDB, another MEAN stack tool. The user will be able to query this database via the web application in order to play a certain motion-detected video.

Web Application Server-Side Routing

The Web Application will be connected to a server using Node.js & Express.js server-side framework offered by the MEAN Stack. This allows the user to perform routing operations such as tapping into the IP camera of any local camera.

5.2 Version Control

GitHub will be our version control tool for establishing a collaborative software development environment. Version control will allow us to manage our code progress and revert back changes if anything happens to go wrong throughout our development timeline.
To mitigate the risk of hardware/software malfunction, Google Drive is used to back up software deliverables. Even if it is not the ideal tool for software development version control, but it can enable us to avoid loss of deliverables.

5.3 Data Management

- ➢ Every local computer will store video streams into their local disk, which will be accessible by the central server via an FTP server.
- ➢ Motion-Detected videos will be sent from the local computer, to the central server, and stored in the MongoDB database. These videos are then accessible by the central server by querying the web application, powered by the MEAN Stack.

The FTP server prototype concept will be powered by Google’s Virtual Machine.

5.4 Algorithm Explanation

We are going to use the Frame Differencing technique to implement motion detection function. Frame Differencing is one of the most widely used methods in motion detection and is performed by segmenting foreground objects from the background in a sequence of video frames. This is done by taking a frame at time t as the background, then we determine whether motion has been detected by comparing a new frame acquired at time t+1 with background. These two frames are converted to greyscale. Each frame is stored as a matrix of pixel, so the subtraction of two frames results in a matrix of differences of pixels. A threshold is introduced to categorize pixels. If the differences of pixel are greater than the threshold, the pixel is replaced by a white one, otherwise replaced with a black one. Then the white contours are seen as a movement.

5.5 System Integration

The components diagram shown in Figure 2 of the Appendix provides a larger context on how the system is integrated as a whole.

6. Test Planning

All modules of the project will be tested in order to ensure that all deliverables meet the requirements of the project, and adhering to the product user acceptance criteria. This will be done in stages. Testing criteria will strictly follow requirements stated in our requirements traceability matrix (shown in Figure 3). At each stage of testing, we will produce test cases that cover all corresponding requirements and standards. The test cases will be performed by team members and the test result will be recorded in a test report.

6.1 Test Coverage

There are four stages of testing in our project. The first stage is during the implementation of video capture module and motion detection module using C++.

The second stage occurs during web application and database development. We test functional requirements that relate to the modules here.

During the third stage the client side of the web application is tested.

The last stage tests that all edge nodes (local computers) are connected in the system, and files can be transferred through the wireless network.

6.2 Test Methods

At each testing stage, different methodologies will be applied to test functional and non-functional requirements.

First stage consists of conducting unit tests on the video-capture component and the motion detection component individually. Afterwards we apply integration testing, testing how the two components integrate together.

At the second and third stage, functionality testing is applied to ensure the web application functions correctly. We first check the database connection, performing CRUD operations.

Furthermore, other non-functional requirements tests will be conducted, such as usability testing, interface testing, compatibility testing, performance testing and security testing.

Finally, the completely integrated system will be tested as a whole by using system testing. This is the last testing conducted before product delivery.

6.3 Sample Test Case

In each testing stage, a requirements traceability matrix test is produced to record the relationships between software requirements and cases. It includes all identified requirements in this stage and its corresponding test case.

Test case shown in the requirements traceability matrix is identified in a test suite. A test suite is a collection of test cases, with each test case consisting of a:

1. Test Case ID - identifies test cases
2. Test Title - brief description of the test
3. Test Procedure - procedures of performing the test
4. Expected results - description of the expected results of the test

The data collected from each case will be recorded in a test report. The test report presents three main data about a test case:

1. Who conducted the test case? What was the test environment, and when was it conducted?
2. Was there any deviation from the test procedure?
3. Was there any deviation from the expected rest results?

The three documentations below illustrate the documents discussed above.

Requirements Traceability Matrix

Requirement ID	**Requirement Description**	**Test Case ID**

Test Suite

Test case ID	**Test Title**	**Test Procedure**	**Expected Results**

Test Report

Test case ID	Who conducted the test?	When was the test performed?	What was the test environment?	Were there any deviations from procedure?	Were there any deviations from the expected result?

7. References

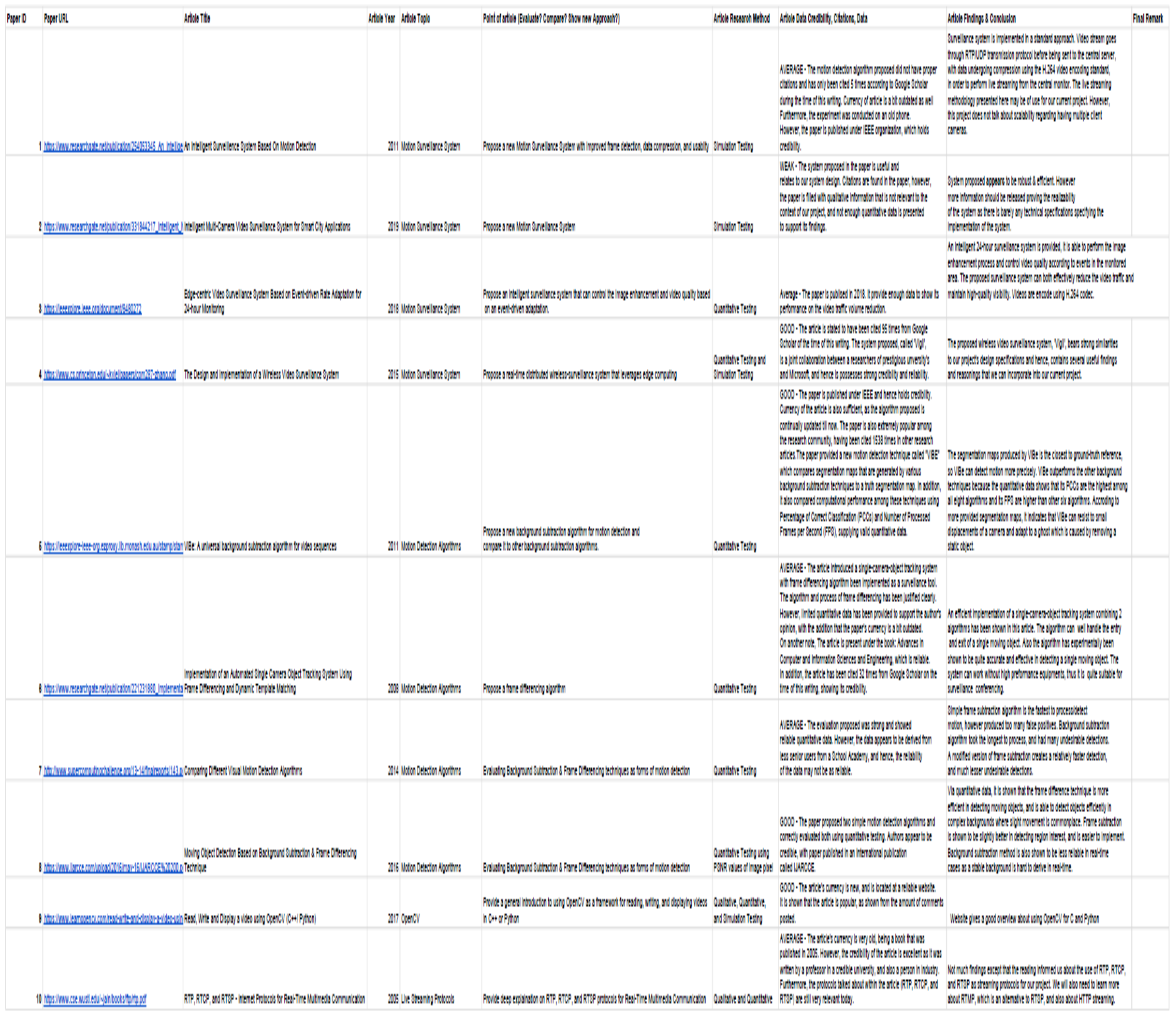
1. Alshammari, A., & B Rawat, D. (2019). Intelligent Multi-Camera Video Surveillance System for Smart City Applications. Retrieved from https://www.researchgate.net/publication/331844217_Intelligent_Multi-Camera_Video_Surveillance_System_for_Smart_City_Applications
2. Barnich, O., & Van Droogenbroeck, M. (2011). ViBe: A Universal Background Subtraction Algorithm for Video Sequences. IEEE Transactions On Image Processing, 20(6), 1709-1724. doi: 10.1109/tip.2010.2101613
3. Durresi, A., & Jain, R. (2005). RTP, RTCP, and RTSP - Internet Protocols for Real-Time Multimedia Communication. Retrieved 24 October 2019, from https://www.cse.wustl.edu/~jain/books/ftp/rtp.pdf
4. Gupta, K., & Kulkarni, A. (2008). "Implementation of an Automated Single Camera Object Tracking System Using Frame Differencing and Dynamic Template Matching." Retrieved from https://www.researchgate.net/publication/221231880_Implementation_of_an_Automated_Single_Camera_Object_Tracking_System_Using_Frame_Differencing_and_Dynamic_Template_Matching
5. Hussain, Z., Naaz, A., & Uddin, N. (2016). "Moving Object Detection Based on Background Subtraction & Frame Differencing Technique". IJARCCE, 5(5). Retrieved from https://www.ijarcce.com/upload/2016/may-16/IJARCCE%20200.pdf

6. Ng, T., Yang, P., & Zeng, C. (2019). FIT3161 Project Design & Prototype Specification for Group 5: Remote Video Surveillance System Project [Ebook].
7. Reed, A., Daniels, Z., & Grubb, C. (2014). Comparing Different Visual Motion Detection Algorithms. Retrieved from http://www.supercomputingchallenge.org/13-14/finalreports/143.pdf
8. Ruiquan, G. (2011). https://www.researchgate.net/publication/254053345_An_intelligent_surveillance_system_based_on_motion_detection. Retrieved from http://254053345_An_intelligent_surveillance_system_based_on_motion_detection
9. Saha, A. (2017). Read, Write and Display a video using OpenCV (C++/ Python) | Learn OpenCV. Retrieved 24 October 2019, from https://www.learnopencv.com/read-write-and-display-a-video-using-opencv-cpp-python/
10. Sakaushi, A., Kanai, K., Katto, J., & Tsuda, T. (2018). Edge-centric Video Surveillance System Based on Event-driven Rate Adaptation for 24-hour Monitoring. Retrieved from https://ieeexplore.ieee.org/document/8480272
11. What is the difference between MJPEG and H.264? – The Angelcam blog – company & industry news. (2015). Retrieved 24 October 2019, from https://blog.angelcam.com/what-is-the-difference-between-mjpeg-and-h-264/
12. Zhang, T., Chowdhery, A., Bahl, P., Jamieson, K., & Banerjee, S. (2015). The Design and Implementation of a Wireless Video Surveillance System. Retrieved from https://www.cs.princeton.edu/~kylej/papers/com287-zhang.pdf

8. Appendix

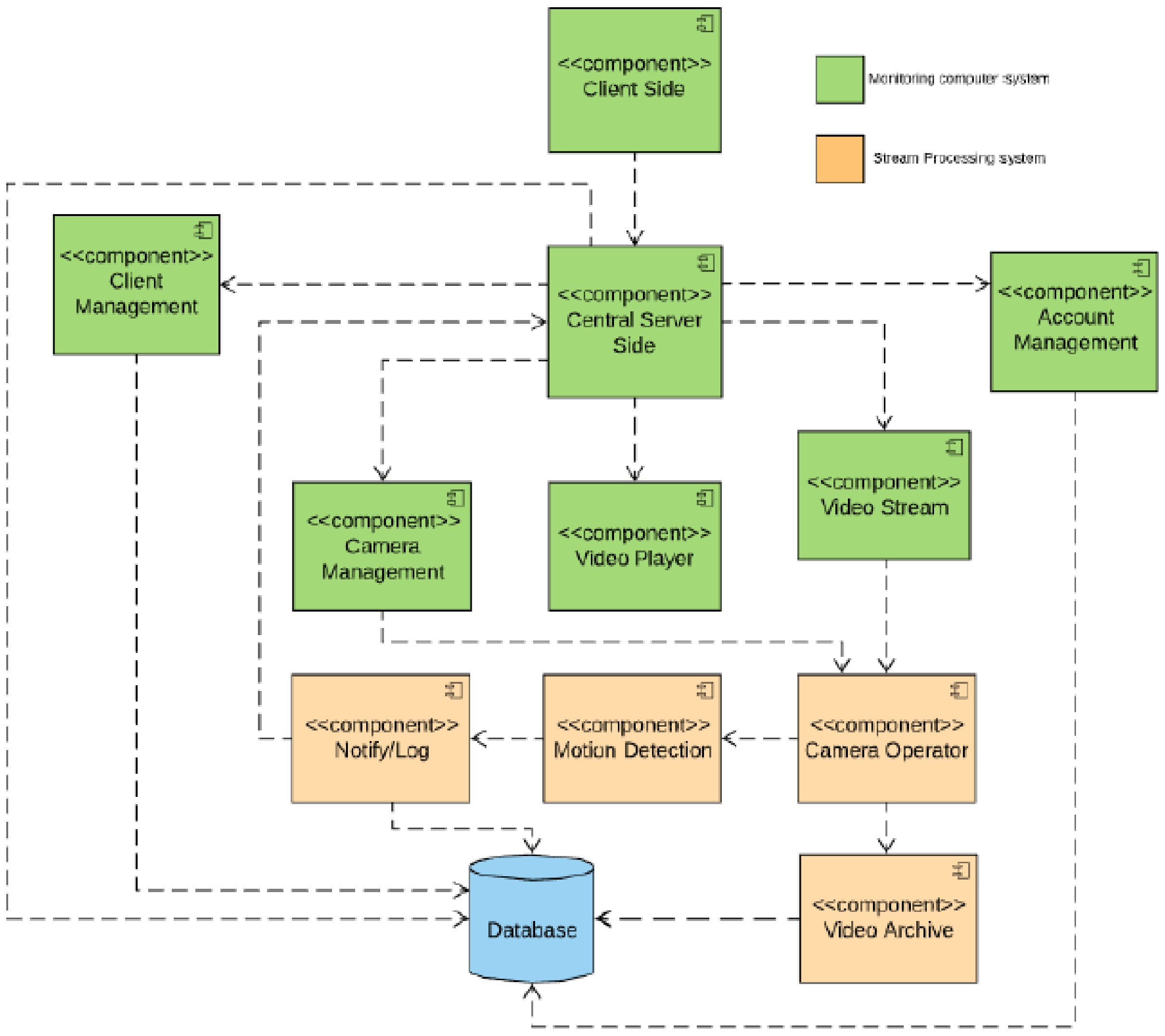
Figure 1: Literature Review Table

Apologies for the terrible picture quality as we cannot squeeze all the info into one picture. For a clearer view, please check our Google Drive Folder, FIT3161_2019S2_Team5 → Project Proposal → Literature Review Table



Paper ID	Paper URL	Article Title	Article Year	Article Topic	Point of article (Evaluate? Compare? Show new Approach?)	Article Research Method	Article Data Credibility, Citations, Data	Article Findings & Conclusion	Final Remark
1	https://www.researchgate.net/publication/254053345_An_intellige	An Intelligent Surveillence System Based On Motion Detection	2011	Motion Surveillance System	Propose a new Motion Surveillance System with improved frame detection, data compression, and usabilty	Simulation Testing	AVERAGE - The motion detection algorithm proposed did not have proper citations and has only been cited 5 times according to Google Scholar during the time of this writing. Currency of article is a bit outdated as well Furthermore, the experiment was conducted on an old phone. However, the paper is published under IEEE organization, which holds credibility.	Surveillance system is implemented in a standard approach. Video stream goes through RTP/UDP transmission protocol before being sent to the central server, with data undergoing compression using the H.264 video encoding standard, in order to perform live streaming from the central monitor. The live streaming methodology presented here may be of use for our current project. However, this project does not talk about scalability regarding having multiple client cameras.	
2	https://www.researchgate.net/publication/331844217_Intelligent_M	Intelligent Multi-Camera Video Surveillance System for Smart City Applications	2019	Motion Surveillance System	Propose a new Motion Surveillance System	Simulation Testing	WEAK - The system proposed in the paper is useful and relates to our system design. Citations are found in the paper, however, the paper is filled with qualitative information that is not relevant to the context of our project, and not enough quantitative data is presented to support its findings.	System proposed **appears** to be robust & efficient. However more information should be released proving the realizability of the system as there is barely any technical specifications specifying the implementation of the system.	
3	https://ieeexplore.ieee.org/document/8480272	Edge-centric Video Surveillance System Based on Event-driven Rate Adaptation for 24-hour Monitoring	2018	Motion Surveillance System	Propose an intelligent surveillance system that can control the image enhancement and video quality based on an event-driven adaptation.	Quantitative Testing	Average - The paper is publised in 2018. It provide enough data to show its performance on the video traffic volume reduction.	An intelligent 24-hour surveillance system is provided, it is able to perform the image enhancement process and control video quality according to events in the monitored area. The proposed surveillance system can both effectively reduce the video traffic and maintain high-quality visibility. Videos are encode using H.264 codec.	
4	https://www.cs.princeton.edu/~kylej/papers/com287-zhang.pdf	The Design and Implementation of a Wireless Video Surveillance System	2015	Motion Surveillance System	Propose a real-time distributed wireless-surveillance system that leverages edge computing	Quantitative Testing and Simulation Testing	GOOD - The article is stated to have been cited 95 times from Google Scholar of the time of this writing. The system proposed, called 'Vigil', is a joint collaboration between a researchers of prestigious unversity's and Microsoft, and hence is possesses strong credibility and reliability.	The proposed wireless video surveillance system, 'Vigil', bears strong similarities to our project's design specifications and hence, contains several useful findings and reasonings that we can incorporate into our current project.	
5	https://ieeexplore-ieee-org.ezproxy.lib.monash.edu.au/stamp/stam	ViBe: A universal background subtraction algorithm for video sequences	2011	Motion Detection Algorithms	Propose a new background subtraction algorithm for motion detection and compare it to other background subtraction algorithms.	Quantitative Testing	GOOD - The paper is published under IEEE and hence holds credibility. Currency of the article is also sufficient, as the algorithm proposed is continually updated till now. The paper is also extremely popular among the research community, having been cited 1538 times in other research articles.The paper provided a new motion detection technique called "ViBE" which compares segmentation maps that are generated by various background subtraction techniques to a truth segmentation map. In addition, it also compared computational perfomance among these techniques using Percentage of Correct Classification (PCCs) and Number of Processed Frames per Second (FPS), supplying valid quantitative data.	The segmentation maps produced by ViBe is the closest to ground-truth reference, so ViBe can detect motion more precisely. ViBe outperforms the other background techniques because the quantitative data shows that its PCCs are the highest among all eight algorithms and its FPS are higher than other six algorithms. Accroding to more provided segmentation maps, it indicates that ViBe can resist to small displacements of a camera and adapt to a ghost which is caused by removing a static object.	
6	https://www.researchgate.net/publication/221231880_Implementa	Implementation of an Automated Single Camera Object Tracking System Using Frame Differencing and Dynamic Template Matching	2008	Motion Detection Algorithms	Propose a frame differencing algorithm	Quantitative Testing	AVERAGE - The article introduced a single-camera-object tracking system with frame differencing algorithm been implemented as a surveillance tool. The algorithm and process of frame differencing has been justified clearly. However, limited quantitative data has been provided to support the author's opinion, with the addition that the paper's currency is a bit outdated. On another note, The article is present under the book: Advances in Computer and Information Sciences and Engineering, which is reliable. In addition, the article has been cited 32 times from Google Scholar on the time of this writing, showing its credibility.	An efficient implementation of a single-camera-object tracking system combining 2 algorithms has been shown in this article. The algorithm can well handle the entry and exit of a single moving object. Also the algorithm has experimentally been shown to be quite accurate and effective in detecting a single moving object. The system can work without high preformance equipments, thus it is quite suitable for surveillance conferencing.	
7	http://www.supercomputingchallenge.org/13-14/finalreports/143.p	Comparing Different Visual Motion Detection Algorithms	2014	Motion Detection Algorithms	Evaluating Background Subtraction & Frame Differencing techniques as forms of motion detection	Quantitative Testing	AVERAGE - The evaluation proposed was strong and showed reliable quantitative data. However, the data appears to be derived from less senior users from a School Academy, and hence, the reliability of the data may not be as reliable.	Simple frame subtraction algorithm is the fastest to process/detect motion, however produced too many false positives. Background subtraction algorithm took the longest to process, and had many undesirable detections. A modified version of frame subtraction creates a relatively faster detection, and much lesser undesirable detections.	
8	https://www.ijarcce.com/upload/2016/may-16/IJARCCE%2020.p	Moving Object Detection Based on Background Subtraction & Frame Differencing Technique	2016	Motion Detection Algorithms	Evaluating Background Subtraction & Frame Differencing techniques as forms of motion detection	Quantitative Testing using PSNR values of image pixel	GOOD - The paper proposed two simple motion detection algorithms and correctly evaluated both using quantitative testing. Authors appear to be credible, with paper published in an international publication called IJARCCE.	Via quantitative data, it is shown that the frame difference technique is more efficient in detecting moving objects, and is able to detect objects efficiently in complex backgrounds where slight movement is commonplace. Frame subtraction is shown to be slightly better in detecting region interest, and is easier to implement. Background subtraction method is also shown to be less reliable in real-time cases as a stable background is hard to derive in real-time.	
9	https://www.learnopencv.com/read-write-and-display-a-video-usin	Read, Write and Display a video using OpenCV (C++/ Python)	2017	OpenCV	Provide a general introduction to using OpenCV as a framework for reading, writing, and displaying videos in C++ or Python	Qualitative, Quantitative, and Simulation Testing	GOOD - The article's currency is new, and is located at a reliable website. It is shown that the article is popular, as shown from the amount of comments posted.	Website gives a good overview about using OpenCV for C and Python	
10	https://www.cse.wustl.edu/~jain/books/ftp/rtp.pdf	RTP, RTCP, and RTSP - Internet Protocols for Real-Time Multimedia Communication	2005	Live Streaming Protocols	Provide deep explaination on RTP, RTCP, and RTSP protocols for Real-Time Multimedia Communication	Qualitative and Quantitative	AVERAGE - The article's currency is very old, being a book that was published in 2005. However, the credibility of the article is excellent as it was written by a professor in a credible university, and also a person in industry. Furthermore, the protocols talked about within the article (RTP, RTCP, and RTSP) are still very relevant today.	Not much findings except that the reading informed us about the use of RTP, RTCP, and RTSP as streaming protocols for our project. We will also need to learn more about RTMP, which is an alternative to RTSP, and also about HTTP streaming.	

Figure 2: System Components Diagram



Monitoring computer system
Stream Processing system
<<component>> Client Side
<<component>> Client Management
<<component>> Central Server Side
<<component>> Account Management
<<component>> Video Stream
<<component>> Camera Management
<<component>> Video Player
<<component>> Notify/Log
<<component>> Motion Detection
<<component>> Camera Operator
<<component>> Video Archive
Database

Figure 3: Requirements Traceability Matrix

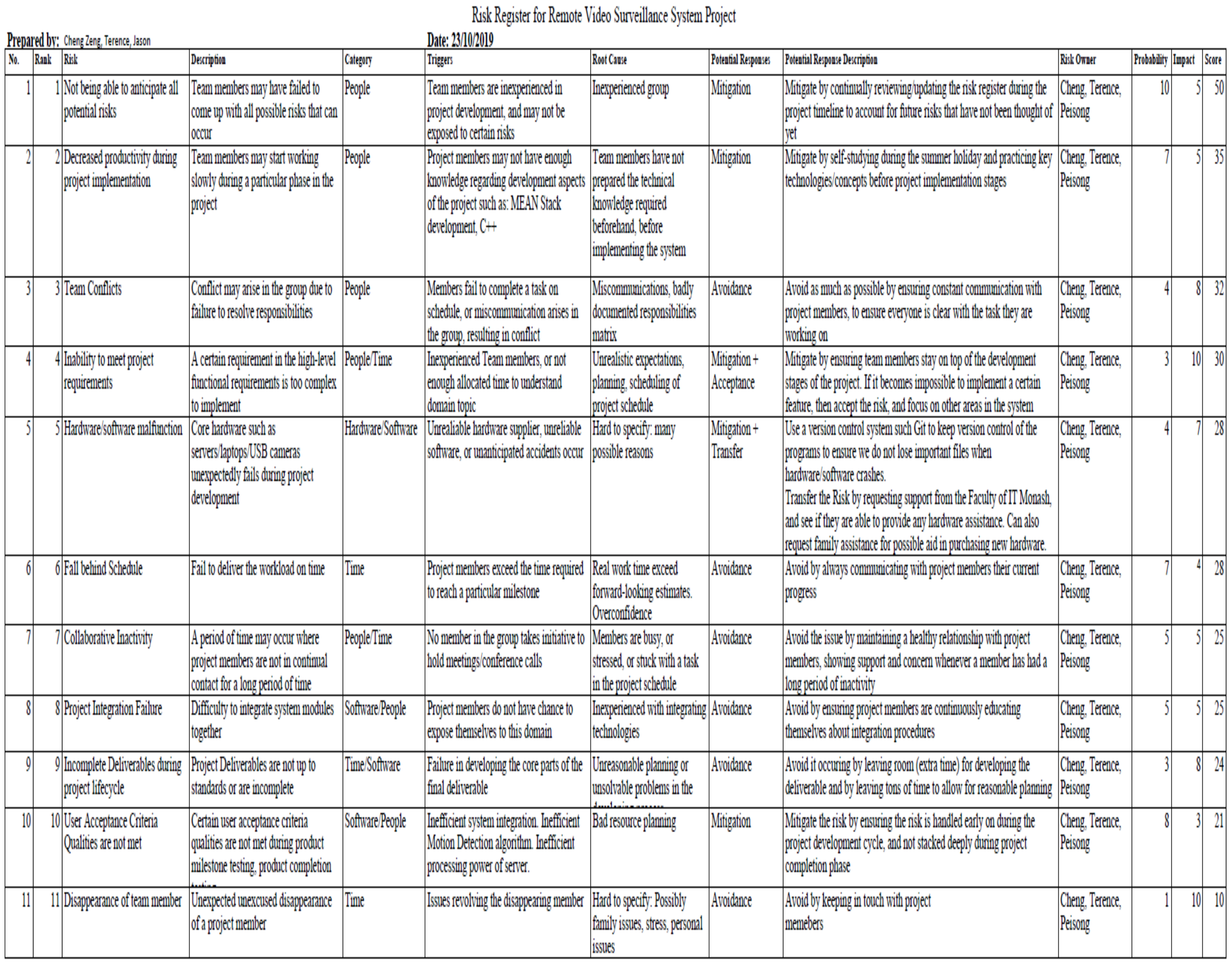
REQUIREMENTS TRACEABILITY MATRIX					
Project Name:	Remote Video Surveillance System Project				
Project Manager Name:	Terence, Jason, Shawn				
ID	***Requirements (Functional or Non-Functional)***	***Assumption(s) and/or Customer Need(s)***	***Source***	***Source***	**Status**
001	Ability to record video frames from a camera and store it into a local database	Functional requirement for storage purposes	Functional Requirement	Daniel	In Progress
002	Laptop with Internet Access for all team members	Non-Functional requirement which ensures communication and collaboration from team members across areas	Hardware Requirement	Daniel	Completed
003	Ability to process video frames into a valid video format for viewing	Functional requirement for video transmission purposes	Functional Requirement	Daniel	In Progress
004	Ability to detect motion from a camera and issue an alert message to the central monitoring computer	Functional requirement for surveillance purposes	Functional Requirement	Daniel	In Progress
005	Ability by a local computer to store snapshot of its video upon motion detection to the central database, hence the snapshot can be played anytime.	Functional requirement for storage purposes	Functional Requirement	Daniel	In Progress
006	Ability for the central monitoring computer to view any camera stream connected to it on demand.	Functional requirement for surveillance purposes	Functional Requirement	Daniel	In Progress
007	Ability for the central monitoring computer to store all motion detection clips alongside a valid schema for storing event logs, client information etc.	Functional requirement for organization purposes	Functional Requirement	Daniel	In Progress
008	Ability for the central monitoring computer to view past streams from any local camera via accessing the local computer's video database.	Functional requirement for storage purposes	Functional Requirement	Daniel	In Progress
009	Ability from the central monitoring computer to associate different cameras to different clients and manage this association (via database management).	Functional requirement for security purposes	Functional Requirement	Daniel	In Progress
010	Ability to carry out management tasks on the monitoring computer via interaction with a web application.	Functional requirement for usability purposes	Functional Requirement	Daniel	In Progress
011	Ability to perform all desired activities from the central monitoring computer onto each local computer via a robust web application.	Functional requirement for usability purposes	Functional Requirement	Daniel	In Progress
012	Ability to design a valid and responsive motion detection algorithm embedded into all local USB webcams by employing a motion detection methodology.	Functional requirement for surveillance purposes	Functional Requirement	Daniel	In Progress
013	If a local camera goes down, the central monitoring computer is able to immediately notice	Functional requirement for surveillance purposes	Functional Requirement	Daniel	In Progress
014	Web Application developed is intuitive and easy to use	Non-Functional requirement for usability purposes	Non-Functional Requirement	Daniel	In Progress
015	The supervisor of the central monitoring computer should be notified of motion detection of a coverage site in a way that is clear	Non-Functional requirement for usability purposes	Non-Functional Requirement	Daniel	In Progress

Figure 3 Extension: Requirements Traceability Matrix Test Coverage Summary

Software Module(s)	*System Component(s)*	Tested In	Implemented In	Validation
Read/Write video algorithm, Video Transmission Protocols	Database, USB Webcam			
Video Transmission Protocols	Web Application, Video Transmission			
Motion Detection Algorithm	USB Webcam, Central Server			
Motion Detection Algorithm	Database, USB Webcam			
MEAN Stack	Database, Web Application			
Database Operations	Database			
FTP Server	Central Server, Local Computer			
Database Schema	Central Server, Database			
Database Operations	Central Server, Web Application, Database			
MEAN Stack	Database, Central Server, Web Application			
Motion Detection Algorithm	USB Webcam			
MEAN Stack	Central Server, Web Application, USB Camera			
MEAN Stack	Web Application, Video Transmission			
MEAN Stack	Web Application, Video Transmission			

Figure 4: Risk Register

Once again, apologies for the terrible picture quality as we cannot squeeze all the info into one picture. For a clearer view, please check our Google Drive Folder, FIT3161_2019S2_Team5 → Project Proposal → PM Deliverables → Risk Register



Risk Register for Remote Video Surveillance System Project

Prepared by: Cheng Zeng, Terence, Jason

Date: 23/10/2019

No.	Rank	Risk	Description	Category	Triggers	Root Cause	Potential Responses	Potential Response Description	Risk Owner	Probability	Impact	Score
1	1	Not being able to anticipate all potential risks	Team members may have failed to come up with all possible risks that can occur	People	Team members are inexperienced in project development, and may not be exposed to certain risks	Inexperienced group	Mitigation	Mitigate by continually reviewing/updating the risk register during the project timeline to account for future risks that have not been thought of yet	Cheng, Terence, Peisong	10	5	50
2	2	Decreased productivity during project implementation	Team members may start working slowly during a particular phase in the project	People	Project members may not have enough knowledge regarding development aspects of the project such as: MEAN Stack development, C++	Team members have not prepared the technical knowledge required beforehand, before implementing the system	Mitigation	Mitigate by self-studying during the summer holiday and practicing key technologies/concepts before project implementation stages	Cheng, Terence, Peisong	7	5	35
3	3	Team Conflicts	Conflict may arise in the group due to failure to resolve responsibilities	People	Members fail to complete a task on schedule, or miscommunication arises in the group, resulting in conflict	Miscommunications, badly documented responsibilities matrix	Avoidance	Avoid as much as possible by ensuring constant communication with project members, to ensure everyone is clear with the task they are working on	Cheng, Terence, Peisong	4	8	32
4	4	Inability to meet project requirements	A certain requirement in the high-level functional requirements is too complex to implement	People/Time	Inexperienced Team members, or not enough allocated time to understand domain topic	Unrealistic expectations, planning, scheduling of project schedule	Mitigation + Acceptance	Mitigate by ensuring team members stay on top of the development stages of the project. If it becomes impossible to implement a certain feature, then accept the risk, and focus on other areas in the system	Cheng, Terence, Peisong	3	10	30
5	5	Hardware/software malfunction	Core hardware such as servers/laptops/USB cameras unexpectedly fails during project development	Hardware/Software	Unrealiable hardware supplier, unreliable software, or unanticipated accidents occur	Hard to specify: many possible reasons	Mitigation + Transfer	Use a version control system such Git to keep version control of the programs to ensure we do not lose important files when hardware/software crashes. Transfer the Risk by requesting support from the Faculty of IT Monash, and see if they are able to provide any hardware assistance. Can also request family assistance for possible aid in purchasing new hardware.	Cheng, Terence, Peisong	4	7	28
6	6	Fall behind Schedule	Fail to deliver the workload on time	Time	Project members exceed the time required to reach a particular milestone	Real work time exceed forward-looking estimates. Overconfidence	Avoidance	Avoid by always communicating with project members their current progress	Cheng, Terence, Peisong	7	4	28
7	7	Collaborative Inactivity	A period of time may occur where project members are not in continual contact for a long period of time	People/Time	No member in the group takes initiative to hold meetings/conference calls	Members are busy, or stressed, or stuck with a task in the project schedule	Avoidance	Avoid the issue by maintaining a healthy relationship with project members, showing support and concern whenever a member has had a long period of inactivity	Cheng, Terence, Peisong	5	5	25
8	8	Project Integration Failure	Difficulty to integrate system modules together	Software/People	Project members do not have chance to expose themselves to this domain	Inexperienced with integrating technologies	Avoidance	Avoid by ensuring project members are continuously educating themselves about integration procedures	Cheng, Terence, Peisong	5	5	25
9	9	Incomplete Deliverables during project lifecycle	Project Deliverables are not up to standards or are incomplete	Time/Software	Failure in developing the core parts of the final deliverable	Unreasonable planning or unsolvable problems in the	Avoidance	Avoid it occuring by leaving room (extra time) for developing the deliverable and by leaving tons of time to allow for reasonable planning	Cheng, Terence, Peisong	3	8	24
10	10	User Acceptance Criteria Qualities are not met	Certain user acceptance criteria qualities are not met during product milestone testing, product completion	Software/People	Inefficient system integration. Inefficient Motion Detection algorithm. Inefficient processing power of server.	Bad resource planning	Mitigation	Mitigate the risk by ensuring the risk is handled early on during the project development cycle, and not stacked deeply during project completion phase	Cheng, Terence, Peisong	8	3	21
11	11	Disappearance of team member	Unexpected unexcused disappearance of a project member	Time	Issues revolving the disappearing member	Hard to specify: Possibly family issues, stress, personal issues	Avoidance	Avoid by keeping in touch with project memebers	Cheng, Terence, Peisong	1	10	10

Figure 5: Project Progress Report Format

Progress Report on Milestone X			
1. Key Update 1 2. Key Update 2 3. Key Update 3			
Milestone X Development			
Development Stage	Expected Completion	Actual Completion	Status
Requirement Analysis	XX/XX/XXXX	XX/XX/XXXX	On Track
System Design			Delayed
Implementation			Concern
Testing			
Deployment			

Figure 6: Sprint Backlog Spreadsheet Format

A User Story = a key piece of high-level functional requirement extracted from the project specification, told in the perspective on a user.

We then derive X tasks from the user story, with team members committing to the tasks they want to complete, and noting down the estimate hours of completion on the sprint backlog.

User Story	Tasks Derived from the User Story	Day 1	Day 2	Day 3	Day X
Story X	Task 1	4	4	2	0
	Task 2	2	4	1	0
	Task 3	8	2	5	0
	Task 4	4	4	4	0
	Task 5	2	2	1	0
	Task 6	1	1	3	0
	Task 7	9	4	2	0

Figure 7: Meeting Minutes Report Format

Location:	[Address or room number]
Date:	[Date]
Time:	[Time]
Attendees:	[List attendees]

Previous Meeting Minutes Summary

1. Group concluded to pursue X instead of Y
2. Time schedule altered to enable more focus on completing X
3. Summary XXX
4. Summary XXX

Agenda items

1. Agenda 1
2. Agenda 2
3. Agenda 3
4. Agenda 4
5. Agenda 5

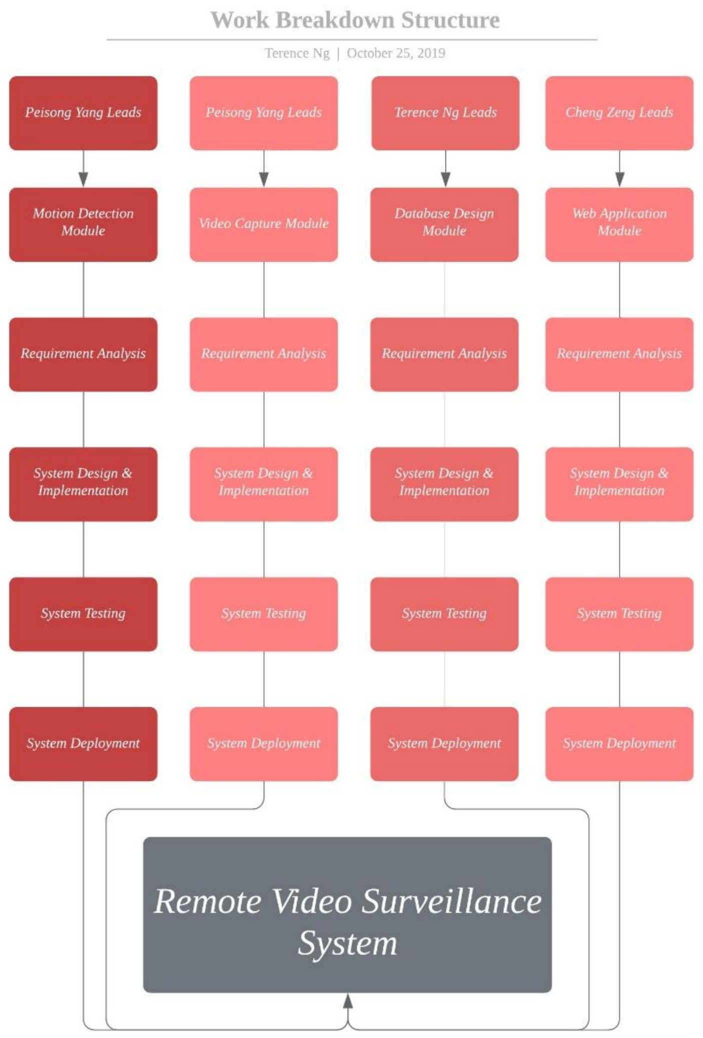
Action items	Owner	Deadline	Status
Item 1	Name X	Date X	In Progress
Item 2	Name X	Date X	Compete
Item 3	Name X	Date X	X
Item 4	Name X	Date X	X
Item 5	Name X	Date X	X

Figure 8: Project Schedule

Task ID	Description	Duration	Start date	End date
1	Initialize the development environment of OpenCV in C++ (set up in Visual Studio), including establish the git resources	2 days	1.3.20	2.3.20
2	Design and implement the Motion Detection algorithm	14 days	3.3.20	16.3.20
3	Milestone 1: Algorithm for motion detection completed	0 days		16.3.20
4	Set up the JavaScript with MEAN stack for the website application development.	2 days	16.3.20	17.3.20
5	Write the user interactive functions of the website (log in and management interface)	12 days	18.3.20	31.3.20
6	Milestone 2: Develop Web Application	0 days		31.3.20
7	Prepare the setting-up for the server database	2 days	1.4.20	2.4.20
8	Implement the database functions	7 days	3.4.20	9.4.20
9	Milestone 3: Set up Database Schema	0 days		9.4.20
10	Testing the motion detection algorithm output videos and the database separately	2 days	10.4.20	11.4.20
11	Linking the data(stored videos) to the established database	3 days	12.4.20	14.4.20
12	Evaluate the algorithm produced and the efficiency of the linked database system	5 days	15.4.20	19.4.20
13	Milestone 4: Set up Video Connections & Database Connections	0 days		19.4.20
14	Link the cameras (including their process computers) with the database and web application based on the FTP	7 days	20.4.20	27.4.20
15	Design and complete the interface of the web application	5 days	28.4.30	2.5.20
16	Test and improve the system on the web application	7 days	3.5.20	9.5.20
17	Milestone 5: Link Wireless Video Cameras to Central Network	0 days		9.5.20
18	Final test the software(web application, motion detection C++ program, database)	7 days	10.5.20	17.5.20
19	Milestone 6: Project Testing	0 days		17.5.20
20	Evaluate the surveillance system and collect data about the storage using estimation, the efficiency of the algorithm and other data	5 days	18.5.20	22.5.20
21	Review the implementation process and summary the experience	2 days	23.5.20	24.5.20
22	Produce the instruction of the web application and analysis about the estimating usage and performance	3 days	25.5.20	27.5.20
23	Write the progress reports of producing the project	10 days	28.5.20	6.6.20
24	Milestone 7: Deliver Final Product	0 days		6.6.20

Figure 9: Work Breakdown Structure

The Work breakdown structure is half redundant because as discussed previously, our project development cycle will be broken down into system modules, with each module undergoing a waterfall methodology. During the development step in the waterfall methodology, the Scrum methodology is employed. And hence, work is essentially broken down into system modules, which can already be seen in Figure 2 of the system components diagram in the Appendix, and also in Section 3.2.1 of the project deliverables. An important note to emphasize however is that, the work breakdown structure by personnel, will be split based on their experience in a system module. Hence, a "lead" member will be responsible for executing Scrum meetings, and documenting the Sprint backlogs for the development of a specific module, and, as stated already, all members will be responsible for the development of all system modules. The work breakdown structure, alongside the lead member responsible for a module, is shown below.


Work Breakdown Structure
Terence Ng | October 25, 2019
Peisong Yang Leads
Peisong Yang Leads
Terence Ng Leads
Cheng Zeng Leads
Motion Detection Module
Video Capture Module
Database Design Module
Web Application Module
Requirement Analysis
System Design & Implementation
System Testing
System Deployment
Remote Video Surveillance System