**Smart-glass based**

**Remote Guidance System**

**Research Report**

Group 21

Lyndon Prado - 9740783

Tingcong Jimmy Li - 100017000

Keagan Foster - 101609822

Ayub Khan - 100667654

Dineth Gunawardena - 100862158

Kosala Edirisinghe - 101265981

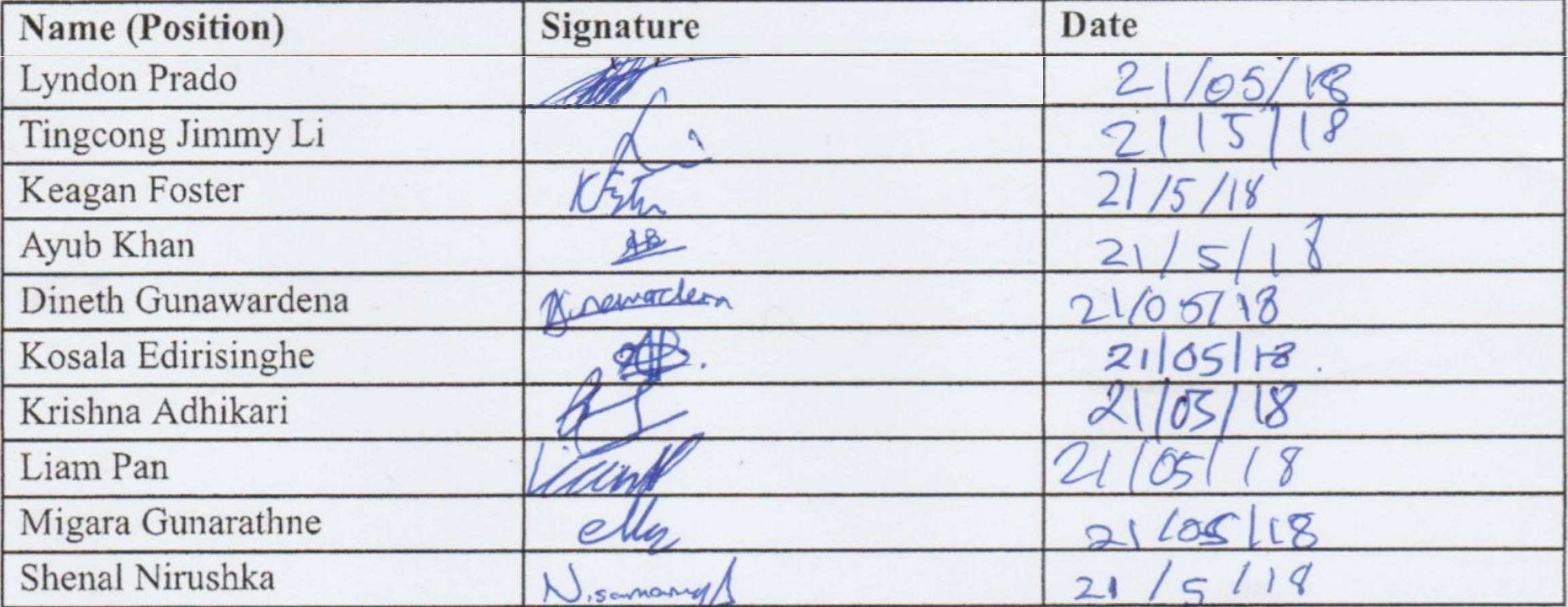
Krishna Adhikari - 4953193

Liam Pan -101106174

Migara Gunarathne - 101002269

Shenal Nirushka – 101054998

**Table 2. Document Sign Off**



# Table of Contents

[Table of Contents 2](#_Toc514963409)

[1. Abstract 5](#_Toc514963410)

[2. Introduction 6](#_Toc514963411)

[2.1. Objectives and/or aims 6](#_Toc514963412)

[2.2. Research questions and/or hypotheses 6](#_Toc514963413)

[2.2.1. How do we best design the system to allow instructors to train operators remotely? 6](#_Toc514963414)

[2.2.2. How do you effectively manage a team of 10 team members while maintaining a high standard of quality? 6](#_Toc514963415)

[2.2.3. What practices need to be utilised to ensure the project meets the clients expectations? 6](#_Toc514963416)

[3. Literature Review 7](#_Toc514963417)

[3.1. How do we best design the system to allow instructors to train operators remotely? 7](#_Toc514963418)

[3.1.1. Remote communications and guidance 7](#_Toc514963419)

[3.1.1.1. Ways and methodologies of human learning 7](#_Toc514963420)

[3.1.1.2. Overview of solutions 7](#_Toc514963421)

[3.1.2. System design 8](#_Toc514963422)

[3.1.2.1. Software Architecture 8](#_Toc514963423)

[3.1.2.1.1. Architectural Patterns 8](#_Toc514963424)

[3.1.2.1.1.1. Layered architecture 9](#_Toc514963425)

[3.1.2.1.1.2. Event-Driven Architecture 9](#_Toc514963426)

[3.1.2.1.1.3. Microkernel architecture 9](#_Toc514963427)

[3.1.2.1.1.4. Microservices architecture pattern 9](#_Toc514963428)

[3.1.2.1.1.5. Space-based architecture 9](#_Toc514963429)

[3.1.2.1.2. What makes a “good” Architecture 9](#_Toc514963430)

[3.1.2.2. Platforms 11](#_Toc514963431)

[3.1.2.2.1. Vuzix smart glass 11](#_Toc514963432)

[3.1.2.2.2. Alternate android development languages/ IDEs 11](#_Toc514963433)

[3.1.2.3. Programming languages 12](#_Toc514963434)

[3.1.2.4. Machine learning 13](#_Toc514963435)

[3.1.2.4.1. Application of Deep learning in object detection 13](#_Toc514963436)

[3.1.2.4.1.1. Neural network 13](#_Toc514963437)

[3.1.2.4.1.2. Dataset 13](#_Toc514963438)

[3.1.2.5. Object recognition 13](#_Toc514963439)

[3.1.2.6. Computer vision 14](#_Toc514963440)

[3.1.2.7. Data transmission 15](#_Toc514963441)

[3.1.2.8. Multi-threading 15](#_Toc514963442)

[3.1.2.9. User interface 16](#_Toc514963443)

[3.2. How do you effectively manage a team of 10 team members while maintaining a high standard of quality? 17](#_Toc514963444)

[3.2.1. Project management 17](#_Toc514963445)

[3.2.1.1. Software engineering practices 17](#_Toc514963446)

[3.2.1.2. Development methodologies 20](#_Toc514963447)

[3.2.1.2.1) Test Driven Development 20](#_Toc514963448)

[3.2.1.2.2) Behaviour Driven Development 21](#_Toc514963449)

[3.2.1.3. Trello vs jira 23](#_Toc514963450)

[3.2.2. SDLC 23](#_Toc514963451)

[3.2.2.1. Agile 23](#_Toc514963452)

[3.2.2.1.1. Scrum 24](#_Toc514963453)

[3.2.2.1.2. Extreme Programming (XP) 25](#_Toc514963454)

[3.2.2.2. Waterfall 25](#_Toc514963455)

[3.2.3. Tools 26](#_Toc514963456)

[3.2.3.1. Documentation standards, tools and applications 26](#_Toc514963457)

[3.2.3.2. Coding standards 26](#_Toc514963458)

[3.3. What practises need to be utilised to ensure the project meets the clients expectations? 27](#_Toc514963459)

[3.3.1. Requirements gathering 27](#_Toc514963460)

[3.3.2. Implementation 28](#_Toc514963461)

[3.3.2.1. Iterative and incremental Implementation 28](#_Toc514963462)

[3.3.3. Testing 28](#_Toc514963463)

[3.3.3.1. Acceptance testing 28](#_Toc514963464)

[3.3.3.2. System testing 28](#_Toc514963465)

[3.3.3.3. Integration testing 29](#_Toc514963466)

[3.8.4. Unit testing 29](#_Toc514963467)

[4. Methods 30](#_Toc514963468)

[4.1. Research design 30](#_Toc514963469)

[4.1.1) Purpose 30](#_Toc514963470)

[4.1.2) Research questions 30](#_Toc514963471)

[4.1.3) Techniques 30](#_Toc514963472)

[4.2) Data or other information collection methods 31](#_Toc514963473)

[4.3) Procedure 31](#_Toc514963474)

[Execution of Research 31](#_Toc514963475)

[Instructions to Participants 32](#_Toc514963476)

[Experiment Manipulations 32](#_Toc514963477)

[5. Results/Conclusion 33](#_Toc514963478)

[6. Discussion 34](#_Toc514963479)

[6.1. Answering our research questions 34](#_Toc514963480)

[6.1.1. How do we best design the system to allow instructors to train operators remotely? 34](#_Toc514963481)

[6.1.2. How do you effectively manage a team of 10 team members while maintaining a high standard of quality? 34](#_Toc514963482)

[6.1.3. What practises need to be utilised to ensure the project meets the clients expectations? 35](#_Toc514963483)

[6.2) Conclusion 35](#_Toc514963484)

[7) References 36](#_Toc514963485)

# 1. Abstract

There are multiple ways to ensure that training is supplied to the intended parties. There are some instances where it is not possible to train an individual, because the expert that is needed to train the subordinate is not available in close proximity. In these cases, remote communication is needed. In this report, we focus on smart glasses technology to implement a method in which to help communication between an instructor and an operator, so that the issue of remote training can be mitigated. We investigate the technologies that can be used to implement hand gestures for intuitive interactions with the instructor and the operator, and we also do research on the project management techniques to run this development project.

# 2. Introduction

## 2.1. Objectives and/or aims

This report presents the reader a new approach in how instructors can interact with operators remotely whilst improving productivity. This approach involves the use of smart glass technologies to help increase the productivity of the operators and the instructors, as they can communicate in real time to each other. As discussed in section 2.2.1 several considerations have already been considered for the methods currently used for remote communication and guidance. Section 2.2 goes over the current research that was done in terms of the new platform that the project will be implement and the software practices and tools used within the project. Section 2.3 are the questions that we must ask ourselves to direct the path of research and to ensure that the content being researched is within the required scope. Section 3 goes through the methods that we use to undertake our research of the project and how we create the prototypes to test our hypothesis. Section 4 goes through the results obtained so far, from the prototypes that are developed and our findings from testing the prototypes. Section 5 is where we discuss about these findings from out prototypes and tests, and we explain what next step are ahead of us are.

## 2.2. Research questions and/or hypotheses

### 2.2.1. How do we best design the system to allow instructors to train operators remotely?

### 2.2.2. How do you effectively manage a team of 10 team members while maintaining a high standard of quality?

### 2.2.3. What practices need to be utilised to ensure the project meets the clients expectations?

# 3. Literature Review

## 3.1. How do we best design the system to allow instructors to train operators remotely?

### 3.1.1. Remote communications and guidance

To ensure that the most effective methodology of remote guidance is used, we must first research the most effective ways, that human beings learn new concepts. Below we discuss the different methods in which humans learn, and from that, it will help us determine how to design a platform that uses the best methods of learning.

#### 3.1.1.1. Ways and methodologies of human learning

Human beings have 5 senses as inputs to the environment that surrounds us. Every day as we are learning new concepts, we use these senses to help us get an understanding about the world around us. Those senses are Visual, Audio, Taste, Smell and Touch [1]. According to a study done by Oklahoma state university, we learn 83 percent of the time by seeing, 11 percent of the time by hearing, 3.5 percent of the time by touching, 1.5 percent of the time by smelling and 1 percent of the time by tasting [2]. Even though visual learning is the most effective way to learn new concepts, different types of visual learning may not be equivalent in effectiveness.

In terms of the conventional methods to learn new aspects and ideas are visual, auditory and kinesthetics [3]. According to nwlink website, learners use all three modalities, though depending on the learning style, one of these modalities may be a higher preference to the specific person.

According to a study done in university of California Santa Barbara, it was shown that having a first person’s perspective of completing a task has improved results in terms of learning, in comparison to third person perspective [4].

Therefore, the solution to the proposed problem of remote communication must be one that uses a visual aspect to it and is also in first perspective.

The following section will discuss more about solutions taking visual and first-person aspects into account.

#### 3.1.1.2. Overview of solutions

There have been multiple solutions that have been implemented for multiple situations regarding remote communications. Some examples are remote guidance for the blind, using a teleassistance system and navigation, although it is noted that these methods are often “costly, poor reliability and (have a) lack of comfort” for the users of these proposed devices [5]. Technologies are being developed for remote surgical procedures such that the surgical/medical practitioner can be in another part of the world but still be able to conduct surgery on the patient remotely, which is referred to as telemedical. The best type of communication that Humans respond to is visual, so having visual communication to have rapid responses from recipients, when giving instructions to them remotely, is a must. The downside to having visual communications is that it creates an overhead in data transmission. There is current research to create efficient algorithms for “energy-efficient on-device preprocessing of data” when sending Atrial fibrillation detection signals remotely [6]. Many industries are looking for efficient ways to transmit communications remotely, whilst maintaining the quality of the communication.

Remote conference meetings happen using social media, such as video chat (skype). This method for remote collaboration is good for briefings between two parties, but in real time applications, where the recipient must execute a task whilst in communication with the remote party, there has yet to be a system in place where these tasks can happen seamlessly.

### 3.1.2. System design

#### 3.1.2.1. Software Architecture

The term software architecture refers to the high-level structures of a software application, the guidelines to creating such structures and the documentation of these structures. Each structure encompasses a significant set of decisions about the organization of software elements, relations among them and their properties. These software components usually comprise of various classes and functions whose interactions are organized through the creation of UML diagrams. These structural decisions help decrease costs by minimizing the risk of making changes after the structure has been implemented and optimize performance, quality and manageability. Well documented software architectures help in facilitating a clear understanding between the developers and the stakeholders, finalize decisions about the high-level design and allow reuse of design components in further projects [7] [8].

##### 3.1.2.1.1. Architectural Patterns

Architectural Patterns are a set of principles that provide an abstract framework for a system. They provide a general reusable solution to frequently recurring problems and improves partitioning. Architectural Patterns are like software design patterns but have a broader scope in comparison and can be used to solve various software issues such as hardware performance limitations, high availability and minimizing the risk factor in software development. It can be thought of as a set of principles that shape an application. Different architectures may share similar characteristics and implement similar patterns [9] [10].

###### 3.1.2.1.1.1. Layered architecture

Layered architecture is where components are organized into horizontal layers, each layer performing a specific role within the application (e.g., presentation logic or business logic). For a conventional web application or android application, there is the presentation layer, business layer, persistence layer and the database layer [11].

###### 3.1.2.1.1.2. Event-Driven Architecture

Most common distributed asynchronous architecture used to develop a highly scalable system. There are usually two main types of event driven architecture. One being a mediator topology, and the other one is a broker topology. Mediator topology has a single event queue and a mediator is used to direct each of the event to relevant event processors. On the other hand, the broker topology does not involve no event queue. Event processors are responsible for obtaining events, processing and publishing another event indicating the end [12].

###### 3.1.2.1.1.3. Microkernel architecture

Microkernel architecture pattern is used for implementing product-based applications. The microkernel architecture pattern consists of two types of architecture components: a core system and plug-in modules [13].

###### 3.1.2.1.1.4. Microservices architecture pattern

Microservices architecture pattern is where multiple service components are used to ensure tasks are completed on the request of the clients. Each service component is made up of smaller modules and each service component serves one purpose. There is an interface layer that interfaces with multiple service components, and Client requests are directed to the indented service component [14].

###### 3.1.2.1.1.5. Space-based architecture

The space-based architecture pattern is specifically designed to address and solve scalability and concurrency issues. The space-based pater is also referred to as the cloud architecture pattern. The architecture is built by having processing units that split the application functionality. The processing units usually have the application modules, with an in-memory data grid and an

optional asynchronous persistent store for failover [15].

##### 3.1.2.1.2. What makes a “good” Architecture

According to addison wesley, a list of following conditions needs to be met for the system being developed, to be of good quality [16]:

* The architecture should be the product of a single architect or a small group of architects with an identified leader
* The architect (or architecture team) should have the functional requirements for the system and an articulated, prioritized list of quality attributes (such as security or modifiability) that the architecture is expected to satisfy.
* The architecture should be well documented, with at least one static view and one dynamic view using an agreed-on notation that all stakeholders can understand with minimum effort.
* The architecture should be circulated to the system’s stakeholders, who should be actively involved in its review.
* The architecture should be analyzed for applicable quantitative measures (such as maximum throughput) and formally evaluated for quality attributes before it is too late to make changes to it.
* The architecture should result in specific (and small) set of resources contention areas, the resolution of which is clearly specified, circulated, and maintained. For example, if network utilization is an area of concern, the architect should produce (and enforce) for each development each development team guidelines that will result in a minimum of network traffic. If performance is a concern, the architects should produce (and enforce) time budgets for the major threads.

**The structural rules of thumb are as follows:**

* The architecture should feature well-defined modules whose functional responsibilities are allocated on the principles of information hiding and separation of concerns. The information-hiding modules should include those that encapsulates idiosyncrasies of the computing infrastructure, this insulating the bulk of the software from change should the infrastructure should change.
* Each module should have a well-defined interface that encapsulates of “hides” changeable aspects (such as implementation strategies and data structure choices) from other software that uses its facilities. These interfaces should allow their respective development team to work largely independently of each other
* Quality attributes should be achieved using well-known architectural tactics specific to attribute
* The architecture should never depend on a version of a commercial product or tool. If it depends upon a commercial product. It should be structured such that changing to a different product is straightforward and inexpensive.
* Modules that produce data should be separate from modules that consume data. This tends to increase modifiability because changes are often confined to either the production of the consumption side of data. If new data is added, both sides will have to change, but the separation allows for a staged (incremental) upgrade.
* For parallel-processing systems, the architecture should feature well-defined processes or tasks that do not necessarily mirror the module decomposition structure. That is, processes may thread through more than one module; a module may include procedures that are invoked as part of more than one process
* Every task or process should be written so that its assignment to a specific processor can be easily changed, perhaps even at runtime.
* The architecture should feature a small number of simple interaction patterns. That is, the system should do the same things in the same way throughout. This will aid in understandability, reduce development time, increase reliability, and enhance modifiability. It will also show conceptual integrity in the architecture, which, while not measurable, leads to smooth development.

#### 

#### 3.1.2.2. Platforms

When discussing IDEs 3 of the options that were brought up were Android Studio, Xamarin Studio and Visual Studio. Xamarin’s main advantage is supports cross platform development very well, it provides native level functionality and it is open source technology with strong corporate support [17]. Some of its cons are that it has it has limited access to open source libraries has a larger app size and has compatibility issues with third party libraries and tools [17].

An important advantage of visual studio is that it supports NDK, which we will be using for this application, something which android studio does not do well. It does however lack some comfort features that android studio has, like android studios drag and drop UI editor [18].

##### 3.1.2.2.1. Vuzix smart glass

The purpose of this project is to develop a system capable of running on the Vuzix Smart M100 Smart Glass. [19] The device comes with an array of sensors as well as camera and a near-eye display. While the not all sensors will be utilised in this project, the device is lacking the latest version of Android (currently running Android 4.0.4). One of the primary goals of the project is to stream videos across a network. [20] Without the latest version of Android(>=4.3), the function of video capturing and streaming must be programmed on a lower level using older Android APIs, in doing so the performance of the system is significantly reduced. Furthermore, lower level programming will also result in high chance of system failure due to its complexity.

From observation, battery life of the device lasts for an estimated duration of 20 minutes while in idle. While this is not important during development, it could be problematic during demonstration.

##### 3.1.2.2.2. Alternate android development languages/ IDEs

Most like other smart glasses Vuzix m100 runs on an android operating system. It consists of fabulous features like other high performance smart glasses. In addition, it has 3D gesture controlling feature which will be used. To develop with this smart glass the registration must be done as a developer. This can be done with the Vuzix official website. The SDK consists of the SDK documentation and a virtual device for the android development environment. To develop apps for the glass, the android SDK and JDK must be installed. IN this project we are going to use Android studio IDE for developing. The best alternatives for the Android IDE are the Eclipse IDE, IntelliJ and NetBeans. Since android is based on Java as the primary language, there are few languages that can be used to support the java environment such as Python, Corona, C++, Kotlin etc.

#### 3.1.2.3. Programming languages

Out of the various programming languages available for android development 3 languages that were nominated by the team are Java, C#, and Kotlin. When it comes to Java and C# there is not much of a difference between the two languages. The main advantage C# has over Java is that is supported on multiple platforms. Since we will be only developing the application for use in android neither Java nor C# has an advantage over each other.

When comparing Java to Kotlin there are several advantages and disadvantages that Kotlin has compared to Java, but these advantages are more useful only if u already know to program in Java. [21]

|  |  |
| --- | --- |
| **Kotlin** | |
| Pros | Cons |
| Interchangeable with java | Increased runtime size |
| Easy learning curve (if already familiar with java) | less readable initially |
| First class android studio support | It has a smaller community therefore less community support |
| Code is much more compact | less resources compared to Java (Books, tutorials and courses) |

Due to the complexity of implementing video capture and streaming with native APIs, an alternative solution was to use the HLS for streaming. HLS was introduced to Android since 3.0. The solution was to use WebView to capture video from the device’s camera and stream using HLS. The initial assumption was that the WebView will not be able to communicate with Android NDK(C++), which is essential for image processing, object recognition and other resource heavy features. It was later discovered [22] that, the communication could be facilitated via Java JNI, which is situated in the frontend. The WebView will be initiated by Java, Java methods that could call C++ will be called by the WebView. This solution can overcome the complexity of coding in low Android APIs while retaining the ability to perform resource heavy processing.

#### 3.1.2.4. Machine learning

##### 3.1.2.4.1. Application of Deep learning in object detection

###### 3.1.2.4.1.1. Neural network

The most basic definition of a neural network is as follows: “a computer system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs.” in "Neural Network Primer: Part I" by Maureen Caudill, AI Expert, Feb. 1989. [23]

Neural networks are used for and work very well with: capturing associations or discovering regularities within a set of patterns; where the volume, number of variables or diversity of the data is very great; the relationships between variables are vaguely understood; or, the relationships are difficult to describe adequately with conventional approaches.

Neural networks have limitations such as: the user only has the role to feed in input and watch it train and await the output of the neural network. Also, that with backpropagation, “you almost don’t know what you’re doing”.

###### 3.1.2.4.1.2. Dataset

Datasets are used to train the neural network model that is developed. They are used so that all the possible conditions that are possible for a specific input and be considered, and that the model is improved to consider the nuances of the different conditions. This is like how humans learn, where we base our understanding of the world and the objects in it, to our internal database of examples of what that object is [24].

Datasets are obtained through collection of data in the real world or the relevant environment. Conventional statistical collection is used to collect data by using metrics on the objects or things that are within the environment. This is done manually or by systems implemented (data entry). Once a significant amount of data is collected, then is it possible to model a neural network model for the dataset, to help with future predictions. In terms of image recognition, there will be variations within the images and objects that need to be recognise. The learning model takes these variations into account by being exposed to the numerous amounts of different variants of the particular object being studied. With the model being trained by the datasets, the image recognition algorithm can determine, for instance, a person’s hand, after being exposed to multiple images of what a hand looks like.

#### 3.1.2.5. Object recognition

Object recognition is a system’s means of using algorithms implementing different techniques to identify an object present in an image or video. Object recognition is relatively important in this project since the smart glasses are expected to recognize one particular object the user is working with and bring up information regarding the object. If we are taking the example of the user operating with the smart glasses in a meat factory, the user would need to know about the dimensions and quality of the meat. Algorithms can help by calculating the height, area and the meat quality by comparing it with a color scale [25].

The gist of how object recognition works, is by running certain regions of an image several times over a classifier until patterns match with a known object in the system’s database. However, this process can be considered relatively slow if the results are needed quickly, especially if the scenario this process is used in, is a self-driving car, where results are expected at the speed the real time environment changes, else a collision could occur. The process can be sped up by using a different technique of the classifier being run only a single time over an image, so that all objects can be recognized in one go. Algorithms can use different techniques like pattern recognition or feature extraction to recognize an object [25].

Pattern recognition is an appearance-based method where templates of objects are used while feature based methods try to compare object and image features until a successful match has been found. Both methods have its own disadvantages. Pattern recognition can be unreliable due to the image of an object appearing differently under changes in lighting, viewing direction and shape, whereas, feature based recognition has the issue of relying only on a single point on an object for all the matches [25].

#### 3.1.2.6. Computer vision

In computer science, there are many branches and one of them is computer vision. Computer vision managers to identify and process images to give similar results same as human beings. Essentially it is giving the ability of human vision to a computer. It makes the utilization of artificial intelligence to distinguish and interpret what the image is about. For example, for this project computer vision is required when processing the images of meat, it should determine the quality of it and should provide the accurate dimensions to give out actual measurements

Computer vision is one of the integral part of internet. This technology helps us in many ways such as identifying fault products, avoiding obstacles and following signals through artificial intelligence. It’s clear that there is a rapid advancement of computer vision market growth. If we go into depth it helps us to identify the objects easily and, we could increase the accuracy of it. Therefore, for this project we need an application to meet up with the stated requirements and computer vision makes a great impact on its success.

The project will be utilising the OpenCV [26] library for video processing. The library supports an array of languages such as C++, C, Java. Video processing methods include hand tracking, gesture tracking and video segmentation, which will be implemented by the server to increase the performance of the system, since the library will be processing captured video data by the devices in real-time.

Hand gesture tracking will be utilising the color differences in the capture video to extract the model of a hand [27] [28]. Once the shape of the hand has been identified, the finger locations will also be identified. By tracking the location of the fingers, most importantly the index finger, its movement paths will be used to sketch an image. Overlaying the movement paths over an image will effectively create the effect of drawing on an image. Tracking the movement of the hand will create the effect of swiping left or swiping right, which can be used to call other methods, such as send sketch and dismiss sketch.

#### 3.1.2.7. Data transmission

Data transmission as the name suggests is the transfer of data from one device to another or from one device to multiple devices. Communication can be over copper wires, fiber cables or wireless. In our case the Communication channel is wireless over Wi-Fi. Used by various electronic devices, a Wi-Fi setup router serves as a communication hub which is also responsible for exchange of data packets between devices. These networks are extremely low in range as it requires the user to be near a router to connect [29]. There are different types of Data communications channels such as simplex, half duplex, full duplex.

Out of which the channel most efficient for a live 2-way audio and video transmission is a full duplex. Full Duplex channels allow communication in both directions always. A full duplex can best be described as a 2-way highway where one lane carries data to the destination and the other lane carries data to the source from the destination both taking place simultaneously. On the other hand, simplex channels only allow a one-way communication to take place [30]. An example of this can be found in our home where data is transmitted to our T.V sets but no data is sent back. As for half duplex it can support a two-way transmission of data but not at the same time. Data transmission takes place using IP packets to help route and address data packets to their destination. Some examples of these are UDP, TCP, RAW and ICMP. Out of the four the most reliable transport system to transport data between devices is TCP. TCP is a stream protocol that guarantees that data reaches the remote host [31].

Data transmission may take place in one of two ways either by serial transmission or by parallel transmission. The more often used type is serial communication where messages are broken down into smaller packages and transmitted over a single wire to its rightful destination. The message is then put together when it reaches the destination [32]. Each package is attached with a parity bit which alerts you in case of an error. Parallel communication work in a similar manner but the smaller packages are transmitted over several wires simultaneously instead of one single wire. Though Parallel transmission is a faster means of transmitting data, it is not an efficient choice over long distances.

#### 3.1.2.8. Multi-threading

A program is run by executing instructions in the processor. These instructions are loaded to the memory from the drive then executed by the processor. In early stages of computing, the processor could execute one program in a given time. One program needed to be finished to start the next program. Then the processes were introduced. In a process, the program is capsuled, and each process has its own memory allocations. If process A wants to do some I/O interactions or it was interrupted by another process which has a higher priority, then the process A goes to the temporary blocked state then another process can be executed. As opposed to the early stage computing, multiple processes can be run at the same time in a processor and the computation became more and more efficient.

Since the rapid growth of the computation speed, running multiple processes wasn’t enough to satisfy the requirements of a computer. Then the Multithreading was invented. Multithreading is same as processes. However, in multithreading, the process is divided into subtasks and each subtask is a thread. Each thread shares the process memory. Most of the modern processors support the multithreading technology. To implement multithreading in a program, the program needs to be written by using multithreading functions. If not, the processor considers the program as a single thread process. Android supports programming with multithreading. When an android program is launched the main thread is created. This thread handles all the tasks if there are not any other threads were used when the app was being programmed. There are some built-in threads in android which it makes easier to program. The AsyncTask is a framework which handles the threads and handlers. HandlerThread is useful for callbacks. ThreadPool is useful when the app is divided into small task then given into ThreadPool to execute. IntentService is ideal for background tasks. These all classes are made to make multithreading easier in android.

#### 3.1.2.9. User interface

When It comes to smart glass interface design less means more. the best approach to take is a minimalistic approach to the design rather than making an interface that has too much information on the display. The Interface should be nearly invisible only showing up when commanded by the user [33]. This is especially important in this project as the smart glasses we are working with have a small screen, it will be too difficult to fit int a lot of detail into the interface always. We should provide some form of visual cues to let the user know how to use the interface.

We should keep in mind that we are designing the interface for daily use perhaps for extended periods of time. We should choose the colors and animation carefully taking into consideration human factors like fatigue and error perception [33]. Another major factor to take into consideration designing the interface is the environment in with the user will be using the device. [33]

In order not to clutter the screen with too many buttons and menus it is better to replace most of the basic functionality with hand gestures. For example, we can have the user make a specific hand movement to indicate that the user wants to go back. We can also use hand gestures for the user to navigate any menus that are present in addition to physical controls on the device itself.

When designing any menus that will be present in the interface it best that the android material design guideline is followed since it a is tried and trusted set of design principle that a proven to work. The three basic principles of material design are realistic design cues, Bold, graphic and intentional, motion provides meaning. Even though we will be using these design principles mainly for the designs of the menus most of the principles can be applied to other areas of the application. [33]

## 3.2. How do you effectively manage a team of 10 team members while maintaining a high standard of quality?

### 3.2.1. Project management

To be able to successfully run this project, management techniques must be implemented, so that as the team will be able to effectively and efficiently use the time, budget and resources available. There are various ways to run a project depending on different variables and objectives of the project being executed.

There are 6 styles of management: **Autocratic**, where managers make decisions unilaterally, without much (or any) input of subordinates. **Consultative**, which is like Autocratic, but it allows more discussion than the autocratic method, but is essentially dictatorial, the leader consults the team members but makes the final decisions. **Persuasive**, which is also like the autocratic management style, but makes the decision based on the persuasion of the subordinates. **Democratic**, which is where the involvement of the team is used when engaging in decision making. **Laissez-faire**, which is a complete opposite of autocracy, where team members can make most of the decisions. **Management By Walking Around (MBWA),** this management style involves listening, where managers gather information by listening to the thoughts of team members that can stop problems at their source [34]. As a team we try to maintain a combination between Consultative, Persuasive and Democratic. In some situations, being Democratic is good because it is fair for the whole group, but issues arise when there are equal numbers for and against a particular decision. Therefore, in those situations, the leader has to make the decisions so that the team can progress.

#### 3.2.1.1. Software engineering practices

Software engineering is all about looking at a particular problem and coming up with a solution to that problem. However, finding a solution does not guarantee that it is the best one. Software engineering practices relate to day-to-day principles, patterns, design, standards, and practices that software engineers adhere to ensure that the final product behaves reliably and efficiently, are affordable to develop and maintain while addressing all requirements defined by the client. Having these sets of principles helps development team to accelerate the development efficiently and managers to organise and manage the projects better. Below are a few recommended software engineering practices:

**Understand the requirement**

Although it might sound like an obvious thing to do. However, surprisingly, a lot of projects fails simply because they fail to understand the requirement. It is very crucial to understand the use case and users for the end-product before you start developing the software. Remember that you won’t be using the final product. So, there should not be any assumptions made while developing the product. Developers also need to understand that requirement might change during the development phase. If the requirements are created based on the use cases and user scenarios and discussed early on with the client, it helps both parties to be on the same page.

**Have a clear vision of the project**

Once the requirement is agreed upon between client and development team, everyone on the team needs to have a clear vision of the project. If the whole team is not clear about what is being built, chances are very high they end of developing something that the client does not want, and the users do not need. Hence, it is important that there is a team meeting talking about the crucial features, what a minimal viable product looks like and how should the team move forward.

**Divide and Conquer**

Once the bigger picture is clear and the whole team can see why the product is required and what problem domain is it going to address, the team should sit down and see how these features can be broken down into several tasks and subtasks. Once broken down, the problem can be dealt with in smaller manageable chunks. This helps both while developing and testing the product as you are looking at isolated modules rather than a large feature set.

**Don’t Repeat Yourself**

This principle extends to the previous principle making the code blocks not only smallest independent unit but making them reusable too. The teams should aim to write reusable blocks as much as possible. This enforces that each of the class/function must have a single, unambiguous, authoritative representation within an entire system. It not only helps limit the code size and write reusable blocks but helps narrow down the bug while debugging or write unit tests to confirm it is doing what it is intended for.

**Don’t reinvent the wheel**

Quite often while working on big projects, there will be several functionalities or processes which are generic with the chances of them being implemented already. It is therefore important to research about different libraries and packages that might prove to be helpful. It not only saves the development time but since it has already been tried and tested by the wider community, the chances of finding bugs is reduced too. Learn to reuse open sourced libraries and packages instead of reinventing the wheel.

**Consistent coding standards**

It is important to agree upon the coding standards before you start coding, especially in larger teams with many developers. Even though the software works as intended, it becomes hard to maintain the codebase especially if there are new features that need to be added in future. It can be quite frustrating to debug in future if coding standard is not consistent across the project which ultimately consumes more time. Hence, it is important that the standards like commas, indentation, use of variables, comments are agreed upon by entire team before you start coding.

**Unit testing**

Unit testing is an integral part of development. The unit test helps identify bugs in the system in initial stages which can save huge resource and time if the bug was to be found towards the end of the project. Creating unit tests helps reinforce creating modularised codes as the class/function should be able to be tested on their own. They should not rely on external classes and functions. It also helps make sure each class have a unique, independent functionality and do not contain a lot of codes. Generally, a test that takes more than 0.1 seconds to run isn’t a unit test. (Powell-Morse, 2018)[35] Finally, it helps to make sure that every block of code is bug-free and good to be integrated with rest of the system.

**Version Control and Repositories**

Version control is an incredibly important part of software development. It helps to track changes across the project and allows developers to collaborate their works. It acts as a single source of truth as it tracks all the activities of all the users. Having multiple versions mean you can roll back to the last stable version if there is a bug in the system without much downtime which might mean saving a lot of cost in the production environments. It accelerates the development of the software allowing multiple developers to work on multiple branches simultaneously. However, best practices should be followed by the entire development.

**Code Reviews**

Code reviews serve multiple purposes in today's development. Firstly, it makes sure the whole team is aware of what each of the classes are responsible. When you review someone else’s code, you get a better understanding of what a class or function is responsible for. Secondly, it also helps maintain the coding standard as if you notice that any developer is deviating from the standards they can be notified immediately. Finally, by reviewing each other’s code, developers can make sure that the changes do not have any bugs and will not break the existing codebase. It could be costly for the team if codes are merged without being reviewed only to find out that they had a bug and broke the system.

**Continuous integration**

Continuous integration plays an important role as we move towards modular development. As the features are broken down into several features to let several developers work on it simultaneously, it is equally important to merge and link these code blocks once they are completed. In a large team, there are several automated builds are carried out when a particular task is completed. It makes sure that even though the teams are working simultaneously and separately, the code can still work as one. By having multiple builds, the team can be sure of which build breaks the software and act on it. It also provides the big picture of the project and what has been achieved.

**Integration testing**

Integration test should be carried out after each feature has been completed. Although unit test and code reviews help to detect bugs early on, it is still important to test the system end-to-end to see if the integrated product works as intended. These test help to confirm the minimum viable product is bug-free and help locate any bugs that might have occurred during the integration process. Doing this regularly helps to make sure the new builds have not introduced new bugs and the system still works as intended.

##### 3.2.1.2. Development methodologies

A software development methodology us a framework that is used to structure, plan and control the process of developing a software. However, it is important to choose the right methodologies for the project. For example, if a heavily plan-driven model is chosen for a project with high level of uncertainty, it can get cumbersome very quickly. So, it is important to research and understand how these methodologies. Test Driven development and Behavior driven development have been briefly introduced in the following section.

###### 3.2.1.2.1) Test Driven Development

Test driven development (TDD) is a programming practise which encourages the developers to write tests before even writing any codes. This is completely opposite to the traditional development where the software is first designed, implemented and then only tested. Since, there are no codes at the start, the tests are bound to fail. The developers then refactor or write codes to pass the unit tests. This approach follows the following steps throughout the development process.

* Write a test
* Execute the tests
* If the test passes, add new tests.
* If the test fails, refactor the codebase until the test passes.
* Once the test is passing, add another test.

Before writing a piece of code, the developers first write a test describing how that code is going to behave. Then they refactor the code to pass the test. There were case studies conducted with three development teams at Microsoft and one at IBM that adopted TDD. The results of the case studies indicated that pre-release defect density of the four products decreased between 40% and 90% relative to similar projects that did not use the TDD practice. However, the team did experience a 15%-35% increase in initial development after adopting TDD. (Nachiappan, 2008).[36]

There are several advantages to this approach and a few have been mentioned below.

**TDD helps prevent bugs**

As you create comprehensive test coverage of the codebase, it reduces errors significantly. The test serves the specification of the code and will not pass until the code is behaving in that way. This ensures the codes are working as intended and reduces the possibility of bugs significantly. As the whole test-suite is run every time, it ensures no new bugs were introduced with the new feature implemented.

**Self-Explanatory code:**

Codes following TDD tend to have less duplication, fewer bugs, quality codebase and a better overall architecture. Since a unit test only focus on one aspect of the feature, the codebase is refactored and a lot cleaner.

**Save Money:**Although at the start, it might be time consuming to write tests before you start. However, in the long run, it can save a lot of time of debugging, running test manually, new bugs being introduced. The cost of change is also a lot less towards the later stages of the project.

TDD may, however, not be suitable and even fail in a few scenario. A few of them being:

* Since TDD depends on unit tests in some ways, not having knowledge about the unit test framework can affect the development.
* It could be hard to create an maintain tests for the system that have rapidly changing requirement.
* If the developers do not have a lot of idea about TDD, they may write too many test at once or forget to run the tests frequently
* Writing tests for trivial code or writing large test cases may be cumbersome

###### 3.2.1.2.2) Behaviour Driven Development

Behaviour Driven Development (BDD) is a software development methodology that combines practices from test-driven development (TDD) and domain-driven design. (Powell-Morse, 2018) [37]. BDD aims to explain the behaviour. The main aim of BDD was to improve communication amongst all the stakeholders so that everyone is in the same page so as to what is needed. This helps remove ambiguity about the requirements as both the technical and business people can understand and communicate with each other effectively. BDD allows you to describe a system’s behaviour without explaining how the behaviour should be implemented.

In BDD, there are four major keywords used to describe scenarios namely **Given**, **When**, **Then** and **And**. **Given** is used to provide the context of the scenario, **When** describes the event and **Then** highlights the outcome. **And** keywords are used to provide more information in a scenario.

Taking an example of a scenario for creating a user account on the system, it would look something like this:

**Scenario 01: Creating a user account**

**Given** the user is creating an account  
**When** they enter the following required details:

**Required details:**

* First Name
* Last Name
* Email
* Confirm Email
* Password
* Gender
* Date of Birth

**And** submits his request

**Then** the account is created

**Account** a confirmation email is sent to user’s email

Looking at above scenario, no matter if it is the business analyst, client, tester or developer, everyone can easily understand what is going on, what is required and what the outcome looks like. So, it really helps to communicate the requirement across all stakeholders. Furthermore, BDD also reduces the time spent in regression test suites significantly as the tests are being written and executed throughout the development process. It is reducing the chances of unexpected bugs as codebase is constantly being monitored and tested.  
  
However, as BDD requires clear specification before development, the team is required to sit down and write in-depth user stories for each particular scenario and feature. It might be a problem for certain team or organistaions. Likewise, BDD relies on content outside feedback. It involves higher level of input form users, customers and domain experts. This may pose a problem for some organisations.

All in all, while all the different methodologies have their own set of advantages and disadvantages, it is important to be able to select the right methodology for the project. It is essential to understand the nature and strengths as well as pitfalls of different methodologies so that you can make an informed decision about which methodology is best suited for the project. If chosen correctly, the development methodologies can help plan, structure, and control the process to create a bug-free, efficient and right product for clients.

#### 3.2.1.3. Trello vs jira

Trello and Jira are both tools that can be used for project management. Our group decided to use Jira over Trello for the following reasons [38]:

* Jira is considered the more qualified of the two tools being widely used in many professional projects. Therefore, experience using Jira would be more valuable and useful in the future.
* Jira offers various dashboards for scrum techniques which our team could profit from as we are adhering to the scrum methodology.
* Jira offers a lot more third-party integrations than Trello does.
* Trello works with Kanban boards while Jira provides proper issue tracking. Jira has a lot more advanced features which Trello board doesn’t have like time tracking, reporting and issue management.
* When approaching the topic of data storage, Jira offers the option of on-premise in addition to cloud storage.

### 3.2.2. SDLC

Presented in this chapter are multiple methodologies of running and managing project that were researched in the start of the year. Choosing the correct methodology for this specific project involves looking at the upsides and downfalls of some of the most commonly used methods in the industry and identifying how these could specifically benefit this project.

There is a total of three methods researched in this chapter, Scrum and XP which fall under the agile ideology but have distinct differences and Waterfall which is an older method of project management.

#### 3.2.2.1. Agile

Agile is a term used to describe approaches to software development emphasizing incremental delivery, team collaboration, continual planning, and continual learning. Agile was coined in 2001 in the Agile Manifesto [39]. The manifesto set out to establish principles to guide a better approach to software development and at its core, the manifesto declares 4 value statements representing the foundation of the agile movement. As written, the manifesto states

Its 4 mains values are:

* Individuals and interactions over processes and tools
* Working software over comprehensive documentation
* Customer collaboration over contract negotiation
* Responding to change over following a plan

The manifesto also states 12 key principles:

* Customer is everything and increases his satisfaction by rapid delivery of software
* Welcome and inclusion of late changing requirements.
* Continuous delivery of working software.
* Enhancement of technical excellence and good design by keeping continuous attention.
* Simplicity is essential.
* Progress measurement through working software.
* Face to face communication is the best communication.
* Develop projects in healthy environment with trustworthy.
* Have self-organizing teams.
* Self-judgment at regular intervals to become more effective.
* Sustainable development, ability to maintain a constant.
* Co-operation between developers and business persons.

All agile methods are built around these core values and principles which seek to promote the efficiency and effectiveness of projects. [18] A study conducted in 2008 with 337 respondents illustrates the perceived benefit that agile methods provide to development teams, including many respondents agreeing with:

* Higher requirements elicitation over time
* Higher initial requirements modelling accuracy
* Greater attention to technical excellence and good design
* Better architecture modelling

However, in the same report, it is clear that many respondents felt that the agile approach also made the project more complex and many were not satisfied with the balance of documentation and implementation.

##### 3.2.2.1.1. Scrum

Scrum is a lightweight Agile method, used to help complete complex projects iteratively, by using techniques to increase flexibility and productivity. Scrum breaks delivery into timeboxed iterations, called Sprints, and tracks progress daily through Stand-up meetings.

Scrum has quickly become the most prominent agile method within the IT industry and many others. In 2017, Scrum Alliance had conducted their yearly survey on the current state of Scrum in the world, in which over 2000 participants from 91 countries took part [40]. This report has shown that Scrum is the most used agile method, with 94% of their participants using Scrum in their Agile practises. Comparing this to the 2013 survey that Scrum Alliance conducted with only 499 participants, in 2013 only 80% of participates used Scrum in their Agile practices [41].

According to Rising and Janoff, A team-based approach for controlling the chaos of conflicting interests needs to be done iteratively by incrementally developing systems and products when requirements are rapidly changing. Scrum can improve communications and maximize co-operation. This method is scalable from small single projects to entire organizations [42].

The creator of Scrum Jeff Sutherland stated in a blog in 2003 that scrum team sizes should be less than 7 [20], however over the following years in his official scrum guide he stated 9 was the maximum team size for Scrum to be efficient. While our team size is 1 larger than the recommended team size, it can still be utilized properly and effectively, with the 2017 Scrum Alliance survey showing that 13% of Scrums teams are over 10 members [43]. Our project also has the weakness of lacking experience with management and operating with any project, and Scrum is better utilized with a more experienced team. Despite this, we believe that the many advantages Scrum brings to a project will be greater than these risks found.

##### 3.2.2.1.2. Extreme Programming (XP)

XP focuses on tactical best-practices for building software rather than the best ways to get the overall project to the release on time and on budget. It prescribes a very specific set of software development practices, like pair programming, test-driven development, and continuous integration.

XP is a flexible agile methodology emphasising interconnection of the proposal and implementation stage [44].

Basic activities are:

1. Planning and Managing

2. Designing

3. Coding

4. Testing

Extreme Programming also focuses on following some of the following characteristics to maintain high standards in development [44]:

* Continuous revision of the program code
* Testing
* Short iterations of development

Extreme Programming does not emphasise documentation of the development and its strict control. The basic element is a high level of communication among all team members and customers as well as frequent iterations. An advantage of this methodology was its possibility to react fast on customer’s changes in requirements and possibility to adapt the program to users with specific disorders, even at the cost of removal of a great part of already written code. This methodology prefers fast reaction to a change before the plan completion, which can prove to be important in the case of the developing software. However, as a result of a lack of focus on documentation and design, agile software projects often use Scrum for project management while drawing tactical practices from XP, with few organisation using purely XP [45].

#### 3.2.2.2. Waterfall

Waterfall project management is a relatively old project management methodology that works best for projects completed in a linear fashion. Unlike the flexible nature of the agile methodologies, the waterfall model is defined by its strict and linear principles. Projects start at the first phase and only progress to the next when everything in the previous phase has been completed.

Dr. Winston Royce, the man who is often called the “father of waterfall” and the author of the seminal 1970 paper Managing the Development of Large Software Systems [46], apparently never intended for the waterfall caricature of his model to be anything but part of his paper’s academic discussion leading to another, more iterative version. In his paper Royce argued for a more-iterative version of his “waterfall” model and went so far as to say that even his more-iterative model was “risky and invites failure.” He further goes on to discuss the additional steps he felt were needed even to allow even the more-iterative models to be successful.

There are a couple of key traits that define the waterfall method:

* Systems are developed in a sequential process
* All requirements are known upfront
* Analysis is done once, and precedes design
* Design is done once, and precedes coding
* All coding is done once, and precedes testing and integration

Waterfall has advantages of being a clear and well-defined management process of running a project, in that requirements are clearly specified early on and don't change, giving clarity in the scope of the project and timeline. However, for a project like the Capstone Software Engineering Project, this method will not be optimal.

### 3.2.3. Tools

#### 3.2.3.1. Documentation standards, tools and applications

Documentation standards are a list of requirements that all members of the team will be expected to follow. By following these, all documents that are created by the team can be standardised and more easily understood by all parties. Documentation standards are necessary because if they are not used, our documentation may not be easily navigable.

After making considerations as to the relative experience of the members, a set of standards was created that was relatively simple compared to a standards document that is generally used in the industry. This is because having a set of standards that is too stringent may be too harsh on the team members who are still new to this kind of project, especially as every member of the team is a student. As such, having standards that are easy to follow will reduce the workload for them.

The standards were created based on simplifying standards found on both the Institute of Electrical and Electronics Engineers (IEEE) [47] and a more informal document found on Toolbox [48].

To assist us in formatting our references, we will be using Endnote X8, as it can store our references, and given the information, can simplify the process of formatting each reference in the style of the intended referencing standard. The referencing standard we will be using is the IEEE standard.

#### 3.2.3.2. Coding standards

When it comes to writing code, various people use assorted styles of coding, this works fine when working alone it creates issues when collaborations take place and to prevent any issues that may come up teams follow a specific set of coding standards. When creating the coding standards one of the thing that need to be taken into consideration is the readability of the code, any one that reads the code should be able to do it easily. Another thing to consider is making the code maintainable, we should be able to go back and read the code after a long time and still understand what it is doing, this can be done, for example by making the code tidier and giving variable names that describe the data it contains. The coding standards differ according to the language or organization, there is no correct coding standard, but it is better to use one that most of the team members agree with. [49]

## 3.3. What practises need to be utilised to ensure the project meets the clients expectations?

### 3.3.1. Requirements gathering

The requirement gathering begins with understanding the problems presented by the client and the objective of the project to produce a product vision. The product vision describes the product in its entirety and what the product is trying to achieve. A series of requirement elicitations are conducted with the stakeholders to define and refine the project scopes. Good requirements cannot be gathered just from multiple meetings with the stakeholder or client. You can never fully be sure you have gathered all the specific requirements by just asking the client what he wants the system to do or not do. Requirements gathering practices include a wide range of interviews, questionnaires, user observation, brainstorming and prototyping. Requirements gathering process begins with an elicitation process followed by an analysis and specification. Though the requirements gathering process may sound simple, there are issues that may arise during the process [50].

These may include:

* **Problem of scope**:

Where the scope of the system is not well defined,

* **problem of understanding:**

Where the client is not sure what he wants the system to do

* **Problems of volatility:**

Where requirements may change over a period.

Some steps can be taken to improve the quality of the requirements for example:

* **Visualization:**

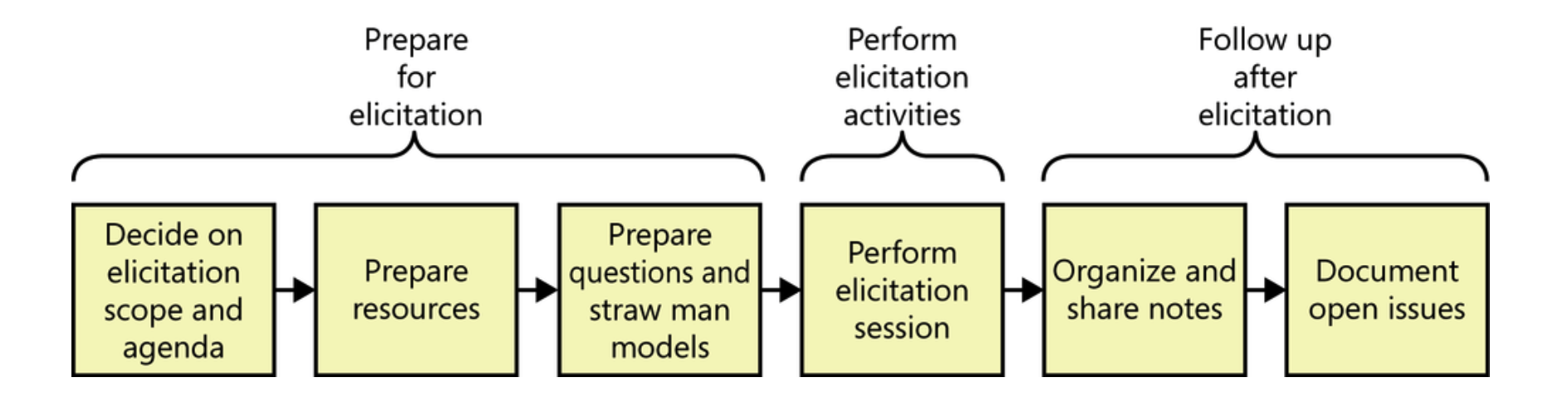
Using tools to better visualize end product

* **Consistent Language**:

Using simple and consistent definitions for describing requirements

* **Documenting Dependencies**:

Document dependencies among requirements [50]



Requirement elicitation is a repeated process that plays an important role in the requirement gathering phase.

### 3.3.2. Implementation

#### 3.3.2.1. Iterative and incremental Implementation

Iterative is where a functionality is added in a repeated cyclic manner/process. Incremental is adding new functionality in small additions. Iterative and Incremental are used side by side to ensure that each feature/functionality that is added to the system is properly tested and integrated with the system, and that it does not break the system when added [51].

### 3.3.3. Testing

This Chapter covers the various types and methodologies of software testing such as Integration testing, Unit Testing, Acceptance Testing and System Testing. Research was carried out in different types of testing in order to finalize the appropriate testing strategy for the Smart Glass Project. After extensive research in highly recommended Testing strategies used in industries a 4-level testing strategy was finalized for the project. This strategy will make sure the system is tested at all major stages of development to ensure quality and accuracy. Testing at multiple stages also keeps the team on track and not deviate from the actual task at hand and help realize which aspects of the system needs special attention. Each stage of the testing strategy is explained further in detail below.

#### 3.3.3.1. Acceptance testing

Considered the last stage of testing in the 4-level testing strategy. Acceptance Testing is the final and arguably the most important stage of testing. This consist of the final system being tested in front of the client who shall decide if the system meets the specified requirements and is efficient enough to be deemed complete. In case of failure the System must go back into development [52] [53].

#### 3.3.3.2. System testing

System Testing takes place toward the end of the system development when all major features of the system have been implemented and can work together as a whole. System testing is carried out to ensure the system meets all major requirements given by the client.

This will ensure whether the system is ready to be shown to the client. System testing generally consist of a usability test carried out using users who have not seen or worked on the system before [52] [53].

#### 3.3.3.3. Integration testing

Integration testing is performed when a low-level design of the system has been established. Where members of the team have worked on different modules of the system individually/separately. An integration test helps determine whether the modules work efficiently when integrated. This helps identify defects in interaction between different components before moving on to the development of a high level complex system [52] [53].

#### 3.8.4. Unit testing

Carried out at the beginning of stage of coding when there are multiple pieces of code being developed by various members of the software development team. Unit tests are developed to test various pieces of code to detect bugs and errors early on to avoid compromising the system later. This helps reduce risk and save time. Unit Tests are generally carried out by the developers themselves [52] [53].

# 4. Methods

## 4.1. Research design

Research design is defined as the strategy which was used to obtain the information and the knowledge about the particular research. There are 3 main steps of every research design. The research purpose explains the reasons for doing the research. Research questions describes the decisions which should be made during the project. Techniques is defined as the methods, strategies and the methodology which we used for researching.

### 4.1.1) Purpose

This group consists of 10 group members who study different types of courses. Most group members don’t have the knowledge in most areas which will be used in this project. To deliver a successful product to the client, the members must have the knowledge and the skills in required areas.

### 4.1.2) Research questions

Deciding which topics should be researched were decided in the beginning of the project. Then those topics were assigned to the group members. There were some topics that every group member had to research. In addition, members were able to research some extra information in various areas.

### 4.1.3) Techniques

* Machine Learning
* Object Recognition
* Computer Vision
* Data Transmission
* Multi-Threading
* Alternate Android development Languages/IDEs
* Android Development with Vuzix smart glass
* SDLC
* User Interface
* Project management
* software engineering practices
* Trello vs jira
* Tools
* Documentation standards, Tools and applications
* Testing

These topics were chosen to be researched. Each member received 2 more tasks to be researched. Some topics were divided into subtopics. Software engineering practices was researched by every member in the group. Members preferred to choose the research areas which they are not familiar with. By doing that the members are able to improve their knowledge in a new field. When the researching is done members have gained the knowledge and the information which will be used in this project.

## 4.2) Data or other information collection methods

The data searching strategy included online electronic data searches and manual searches of various national and international resources that relate to our topic of interest. In this context, following online journal databases were searched:

-ASQ Digital Library

-IEEE Explore

-Springer Link

-Academia.edu

-Research Gate

-Agile Alliance

-Science Direct

In addition, many online blogs and information websites were utilized to the same effect to gather knowledge about less documented areas of interest.

## 4.3) Procedure

Developing a functional prototype requires researching with a proper research plan. The steps followed for the execution of research are shown below:

### Execution of Research

1. **Identifying the problem:** As a team, we must find out what the application being created is required to do. The first step would be to make a requirements document which must be validated with the client, to ensure there isn’t any misunderstanding.
2. **Classifying the requirements:** The requirements must be analyzed, with solutions for each requirement, helping us come up with features for the design of the prototype.
3. **Skill Training:** Study research must be conducted by the members of the team to achieve the skills needed to develop the solution. It is important to create a research plan with different sections assigned to specific team members. Team members can share their skills by sharing material which might be useful to other members.
4. **Developing the prototype:** Team members should use the skills their acquired to develop a functional prototype which meets the requirements of the user.
5. **Usability Testing:** While the prototype might be functional, verifying the requirements, it might not validate the expectations of the end users. Therefore, usability tests must be done with the target users acting as participants ensuring that the software would satisfy the customer.
6. **Iterative Design:** The prototype undergoes constant testing and analyzing, with refinements being implemented to making the best possible prototype [54].

### Instructions to Participants

Participants would follow the procedure below when the usability evaluation is being carried out.

1. The participant must read thoroughly through consent agreement form.
2. The participant must answer demographic questionnaire before the experiment.
3. The participant must talk about their thought process while doing the tasks.
4. The participant must fill out a satisfaction questionnaire after completing the evaluation test.

### Experiment Manipulations

Fixing issues of the system is heavily reliant on the results of the evaluation tests. Therefore, it is important that the results are accurate. The following methods can be used to remove possible errors from the tests:

* **Counterbalancing-**  As participants progress from task to task they might find their way through the interface easier finishing the other tasks faster, despite having flaws in the interface. This is known as a confound, which is a factor which might affect the experiment by being an undesirable explanation to the test results. If counterbalancing was followed in the procedure, half of the participants would follow one task order while the other half follows a different task order removing the confound. [55]
* **Task randomization-** As participants progress from task to task they might find their way through the interface easier finishing the other tasks faster, despite having flaws in the interface. This is known as a confound, which is a factor which might affect the experiment by being an undesirable explanation to the test results. If counterbalancing was followed in the procedure, half of the participants would follow one task order while the other half follows a different task order removing the confound. [56]

### 

# 5. Results/Conclusion

At this stage of the project, the team has tested the connection between 2 smart glasses, with the streaming program written in android IDE. This development was a spike and not a part of the actual implementation of the product, to determine if the method of development is going in the correct direction. This development spike showed that a connection between the 2 glasses is possible, but the programs was not stable, and the connection only lasted for around 30 seconds. We determined that the version of the android run on the smart glasses Vuzix m100, was very limited in terms functionality and compatibility; and, the hardware performance capability of the glasses, may not be powerful enough to undertake the image processing.

# 6. Discussion

## 6.1. Answering our research questions

### 6.1.1. How do we best design the system to allow instructors to train operators remotely?

We found that through our research, that having a first-person perspective is an effective way that human beings can be taught to execute a instruction. We found that this was not possible previously, in a remote setting, because of the technological limitations. Technological advancements in smart glass technology has allowed for this possibility of having a perspective of an instructor point of view (first person). We have yet to gather data on the effectiveness of the glasses, and this will be undertaken when the usability tests are executed.

Using the research about the system design and architecture, we have designed the system in what we believe would be most practical and effective. We gained knowledge on how we are going to approach implementing video processing, object recognition and machine learning to help assist the operator and instructor in the effectiveness of remote training using our system. We also discovered the limitations on the hardware and software we have access to, which define the restrict of features we can implement and therefore altering our design. Also knowing what hardware, we are designing our system towards, let us design the user interface to best suited the smart glasses which should improve the effectiveness of the system.

### 6.1.2. How do you effectively manage a team of 10 team members while maintaining a high standard of quality?

Using the founding of our research about managements styles, we determined early on that using a Scrum based style of management would best suite our team due to a number of advantages discussed within the literature review. As well, we choose tools such as Jira and chose standards for our coding and documentation that would best support our team layout. However, after utilising Scrum throughout the semester we have found that Democratic management is too slow in terms of making decisions, when the whole teams motives are not unified. This could be due to several factors such as inexperienced leadership and use of Scrum, therefore leading us to defer to a more waterfall-based style of project management that was easier to implement.

### 6.1.3. What practises need to be utilised to ensure the project meets the clients expectations?

We found that to ensure good design and development of the system requires close interaction with the client. The first step is to get the requirement gathering correct, so that the client and our team have a mutual understanding of what is going to be delivered at the end of the project, however we also must be flexible to change requirements if necessary. We are using a iterative implementation of the system which will allow us to demonstrate to the client our progress, and get regular feedback to ensure we are meeting his expectations. And throughout the development stages, we need to continually test the system using a variety of tests to ensure that it meets both our functional and non-functional requirements.

## 6.2) Conclusion

As the project has not concluded, the research will continue unto the second semester. As we gain results from our developments, we will update our findings and determine if we have successfully answered our research questions with significant and solid evidence and research. At this moment, it is indefinite that we proved, that remote guidance in first person, using hand gestures, is an effective way to communicate, train and liaise with other people in remote locations. We may not be able to 100% prove this as everyone has different learning style, but we may be able to show the effectiveness of this platform to a high percentage of most cases.

# 7) References

1. (Lyndon)  
   <https://www.howtolearn.com/2012/12/learning-is-multi-sensory-how-to-engage-all-the-senses-so-children-really-benefit/>
2. (Lyndon)  
   <http://4h.okstate.edu/literature-links/lit-online/others/volunteer/4H.VOL.115%20Learning%20Styles_08.pdf/>
3. (Lyndon)  
   <http://www.nwlink.com/~donclark/hrd/styles/vakt.html>
4. (Lyndon)  
   <https://www.researchgate.net/publication/309891761_It's_All_a_Matter_of_Perspective_Viewing_First-Person_Video_Modeling_Examples_Promotes_Learning_of_an_Assembly_Task>
5. (Lyndon) <https://ieeexplore-ieee-org.ezproxy.lib.swin.edu.au/abstract/document/4581561/>
6. (Lyndon) <https://ieeexplore-ieee-org.ezproxy.lib.swin.edu.au/document/7985896/>
7. (Ayub) <https://msdn.microsoft.com/en-us/library/ee658098.aspx>
8. (Ayub)<https://en.wikipedia.org/wiki/Software_architecture>
9. (Ayub)<https://msdn.microsoft.com/en-us/library/ee658117.aspx>
10. (Migara) <https://ieeexplore.ieee.org/document/5661769/>
11. (Lyndon)  
    <https://www.safaribooksonline.com/library/view/software-architecture-patterns/9781491971437/ch01.html>
12. (Lyndon)  
    <https://towardsdatascience.com/event-driven-architecture-pattern-b54fc50276cd>
13. (Lyndon)  
    <https://www.oreilly.com/library/view/software-architecture-patterns/9781491971437/ch03.html>
14. (Lyndon)  
    <https://www.safaribooksonline.com/library/view/software-architecture-patterns/9781491971437/ch04.html>
15. (Lyndon)  
    <https://www.oreilly.com/ideas/software-architecture-patterns/page/6/space-based-architecture>
16. (Lyndon)  
    software architecture in practice 2nd edition - Addison wesley (book)
17. <https://dzone.com/articles/java-vs-kotlin-which-language-android-developer-sh> (Shenal)
18. <https://www.altexsoft.com/blog/mobile/pros-and-cons-of-xamarin-vs-native/>(Shenal)
19. (Tingcong Jimmy Li) <https://www.vuzix.com/Products/m100-smart-glasses>
20. (Tingcong Jimmy Li) <https://github.com/fyhertz/libstreaming/blob/master/README.md>
21. <https://code.tutsplus.com/articles/java-vs-kotlin-should-you-be-using-kotlin-for-android-development--cms-27846> (Shenal)
22. (Tingcong Jimmy Li) <https://stackoverflow.com/questions/9193522/webview-from-ndk-in-android?utm_medium=organic&utm_source=google_rich_qa&utm_campaign=google_rich_qa>
23. (Lyndon)  
    <http://pages.cs.wisc.edu/~bolo/shipyard/neural/local.html>
24. (Lyndon)  
    <https://towardsdatascience.com/how-do-we-train-neural-networks-edd985562b73>
25. (Dineth) <https://www.mathworks.com/discovery/object-recognition.html>
26. (Jimmy)

<https://docs.opencv.org/3.4.1/index.html>

1. (Jimmy) <https://medium.com/@muehler.v/simple-hand-gesture-recognition-using-opencv-and-javascript-eb3d6ced28a0>
2. (jimmy)  
   <https://github.com/bengal/opencv-hand-detection/blob/master/hand.c>
3. (Ayub)<https://www.quantil.com/content-delivery-insights/content-acceleration/data-transmission/>
4. (Ayub) <http://www.on-time.com/rtos-32-docs/rtip-32/programming-manual/tcp-ip-networking/ip-packet-types.htm>
5. (Ayub) <http://softwaretestingfundamentals.com/software-testing-levels/>
6. (Ayub) <http://theteacher.info/index.php/fundamentals-of-computer-science-unit-1/3-data-transmission/topics/2939-serial-parallel-simplex-half-duplex-and-full-duplex-data-transmission>
7. <https://blog.mvp-space.com/why-should-i-use-material-design-for-my-app-5a90edb3bba> (Shenal)
8. (Lyndon) <https://online.grace.edu/news/business/types-of-management-styles/>
9. (Krishna) Powell-Morse, A. (2018). Behavior-Driven Development: What is it and how do you use it?. [online] Airbrake Blog. Available at: <https://airbrake.io/blog/software-design/behavior-driven-development>
10. (Krishna) Nagappan N, Williams L, Bhat T & Maximilien M, 2008, Realizing quality improvements through test driven development: results and experiences of four industrial teams
11. (Krishna) Powell-Morse, A. (2018). Behavior-Driven Development: What is it and how do you use it?. [online] Airbrake Blog. Available at: <https://airbrake.io/blog/software-design/behavior-driven-development>
12. (Dineth) <https://usersnap.com/blog/jira-vs-trello/>
13. (Keagan )<http://agilemanifesto.org/>
14. (Keagan) Rising, L., & Janoff, N, The Scrum Software Development Process for Small Teams. IEEE Software, 17(4), 26–32.
15. (Liam) <https://it.toolbox.com/blogs/craigborysowich/sample-standards-amp-procedures-manual-for-document-structure-amp-style-033109>
16. (Keagan) <http://www.ijetmas.com/admin/resources/project/paper/f201411181416300558.pdf>
17. (Keagan) Rising, L., & Janoff, N, The Scrum Software Development Process for Small Teams. IEEE Software, 17(4), 26–32.
18. (Keagan) <https://ac.els-cdn.com/S1877050911000330/1-s2.0-S1877050911000330-main.pdf?_tid=b95ea4a1-d045-40db-afdb-ccc06fae4663&acdnat=1525911045_f5c8db2edd82c0d99ec966141324f0df>
19. (Keagan) <http://www.ambysoft.com/surveys/practicesPrinciples2008.html>
20. (Keagan) <http://delivery.acm.org/10.1145/50000/41801/p328-royce.pdf?ip=136.186.248.243&id=41801&acc=ACTIVE%20SERVICE&key=65D80644F295BC0D%2E8CC58AB4E41E133F%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35&__acm__=1525912292_8ca27b62648d06479559fcfc5678f82a>
21. (Keagan)  <https://www.scrumalliance.org/ScrumRedesignDEVSite/media/ScrumAllianceMedia/Files%20and%20PDFs/State%20of%20Scrum/2017-SoSR-Final-Version-(Pages).pdf>
22. (Liam) <https://development.standards.ieee.org/myproject/Public/mytools/draft/styleman.pdf>
23. (Shenal) <https://swinbrain.ict.swin.edu.au/wiki/C_Plus_Plus_coding_recommendations>
24. (Ayub)<https://en.wikipedia.org/wiki/Requirements_elicitation>
25. (Lyndon)  
    <https://www.testingexcellence.com/iterative-incremental-development-agile/>
26. (Ayub) <http://softwaretestingfundamentals.com/software-testing-levels/>
27. (Ayub) <https://www.seguetech.com/the-four-levels-of-software-testing/>
28. (Dineth) <https://www.interaction-design.org/literature/article/design-iteration-brings-powerful-results-so-do-it-again-designer>
29. (Dineth)

<http://www.usabilityfirst.com/glossary/counterbalancing/>

1. (Dineth) <https://measuringu.com/random/>