Software Requirements Specification
Mechtronics Enigeering

Team 32, Wingman, SmartVault
Edward He
Erping Zhang
Guangwei Tang
Peng Cui
Peihua Jin

Table 1: Revision History

Date	**Developer(s)**	**Change**
2022-10-05	Edward He, Erping Zhang Guangwei Tang, Peng Cui Peihua Jin	Revision 0
2022-11-20	Guangwei Tang, Erping Zhang	Update functional and non-functional Requirements
2023-1-16	Peng Cui, Edward He	Change FSM and update context diagram
2023-3-26	Peihua Jin	Change some incorrect Concepts
2023-4-3	Peihua Jin, Peng Cui	Update control and monitored variables, check for grammar errors

This document is a modified Volere template and has been adopted according to project needs. Since this is a Mechatronics project, some subsections are different comparing to the original Volere template. The document follows the general organization of the Volere template with some additional sections to better suit for Mechatronics.

Neglected subsections: the hands-on users of the product, persona, priorities assigned to users, users participation, maintenance users and service technicians,implementation environment of the current system, partner or collaborative applications, off-the-shelf software, anticipated workplace environment, enterprise constrains, relevant facts, business rule.

Contents

List of Tables

List of Figures

1 Project Drivers

1.1 The Purpose of the Project

The purpose of the project is to build a Mechatronics system called "Smart-Vault" that is able to assist user in finding their belongings in a given area.

1.2 The Purpose of the Document

This document is intended to provide detailed set of requirements project SmartVault. The documentation will cover the functionality of the system and the requirements that the system is expected to fulfill. In addition to the function requirements of software system, non-functional requirements will also be included in this document. Any additional useful information for building the system is also covered and written in details. This document is used as a reference and guideline for the development of the system and is to ensure that the built system is fulfilling the necessary requirements and meeting the desired goals.

1.3 The Stakeholders

1.3.1 The Client

- Dr. Spencer Smith, professor from McMaster University, Computing and Software department.

1.3.2 The Customers

- people who often can't find their belongings due to poor organization
- people who has bad memory

1.3.3 Other Stakeholders

N/A

2 Project Constraints

2.1 Mandated Constraints

MC1	The cost of purchasing components and parts should not exceed $750
Description	Cost are restricted as a hard requirement in project designing stage
MC2	The submission date of the complete project design is by the end of academic year
Description	Based on course requirement, project has a solid deadline
MC3	The design should be an integration of hardware and software
Description	As part of the project hard requirement for Mechatronics group, the design can not be all software based

2.2 Naming Conventions and Terminology

2.2.1 Definitions of All Terms

Object	The physical objects need to be tracked and searched in the room
Camera Mount	The motorized mount that hold the camera and adjust the angle of the camera view
Servo	An electromagnetic device that converts electricity into precise controlled motion by use of negative feedback mechanisms
Field Oriented Control(FOC)	A variable-frequency drive control method in which the stator currents of a three-phase AC or brushless DC electric motor are identified as two orthogonal components that can be visualized with a vector
Relocation	The change of the position or location of an object in the room
Controller Board	A hardware chip set with General Purpose Input and Output ports, which send data from camera to laptop and control the rotation of camera mount
Serial Communication	A communication method that uses one or two transmission lines to send and receive data, and that data is continuously sent and received one bit at a time
User Interface	A embedded software program that allow user to interact with the searching system
Camera	The device that collect video data and send the data to rest of the system
Human Detection	The action that system can recognize the existance of human shown in the room
File Storage	The physical space in system which record information about each object

2.3 Relevant Facts and Assumptions

2.3.1 User Characteristics

The users of SmartVault are people who have trouble finding their belongings in their daily lives. It is preferred that the user to have general knowledge of what is presented in the designated area. Users should be familiar and comfortable with technology. At minimum, the users should be able to read, type and navigate on a web-page.

2.3.2 Reader Characteristics

The intended readers for this document are the developers responsible for developing SmartVault. Dr.Smith, a professor at McMaster University and his teaching team will also be the intended reader of this document. The document is written in a technical manner and requires the reader to possess both software and hardware knowledge. The reader should be familiar with image processing and the ability to understand the basic logic of object detection. The reader should also understand the software design cycle and the steps taken to develop a complete software program. In order to fully understand the document, it is suggested that the reader has the ability to read and comprehend the communication and control system between a software program and hardware device.

2.3.3 Assumptions and Dependencies

AD1: The user can perform simple computer operations.
Rationale: The user can type, moving mouse and clicking on the window.

AD2: The user can understand English.

AD3: The operation of the project should take place indoors with adequate lighting.

AD4: All objects can be seen by the product without hiding.

AD5: The general motion of the user can be seen by the device.

3 Functional Requirements

3.1 The Scope of the Work

The following picture shows the design of the context diagram of the project. In this diagram, the user can interact with the SmartVault by logging in successfully. The user can also provide timing information about the object to get feedback from SmartVault. A camera will be mounted inside the room and it will keep sending images to Smatvault for image processing. The motor is used to changes the angular position of the camera so that the desired location is achieved. After SmartVault has done image processing, it will update the data in the file storage and will extract information when the user needs it.

3.1.1 The Context of the Work

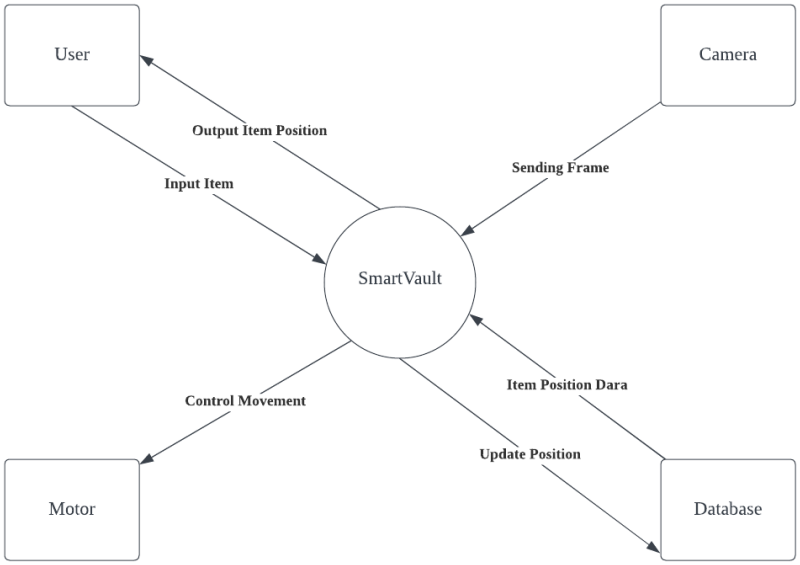

Figure 1: Context of Work

3.1.2 Work Partitioning

The following table shows the work partitioning of the project. all possible types of those events are included in the table. A functional Decomposition Diagram is also shown to show the whole working condition of the project.

Table 2: Event List

Event Name	**Input and Output**	**Types of Input or Output**
User input	Item description(in)	Username and password, timing information about the object want to find, operational actions
UI output	Information(out)	Item pictures, picture showing the position of the selected item.
Provide item position	Item position (out)	Item pictures, picture about the position of the object
Live feed from camera	Live frames(in)	pictures taken by the camera.
Motor Control	Degree and direction of rotation (out)	rotating angle and speed.
Update file	New item position(out)	Replace or add position data for an item in the database.
Acquire position	Item timing information	time interval information related to the item.

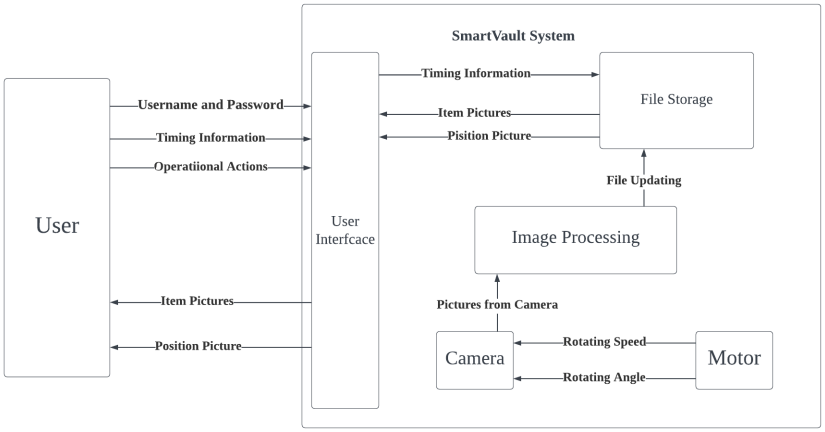

Figure 2: Functional Decomposition Diagram

3.2 The Scope of Product

3.2.1 Finite State Machine Description

To make the behaviour of the product to achieve the target task, the Finite State Machine is created to describe the behaviour with detailed description provided after the picture.

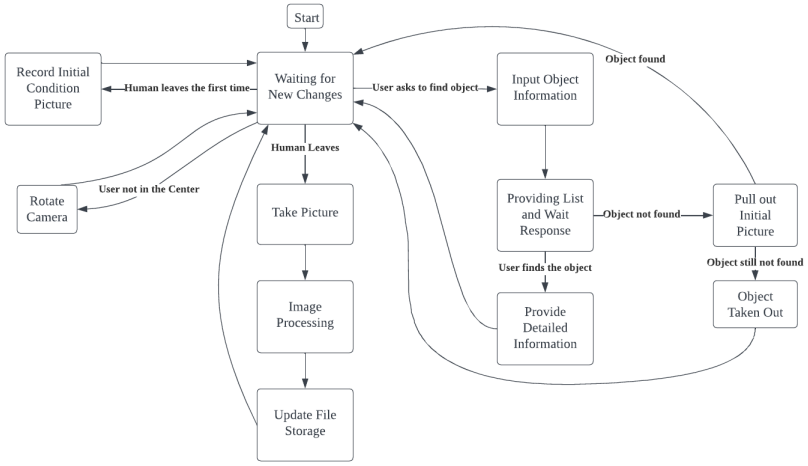

Figure 3: The Picture of Finite State Machine

Record Initial Condition: When the user leaves the room for the first time, the device will take a picture of the room as the initial position information for each object and save it to the file storage module.

Waiting for New Changes: The device will wait for future changes to the system.

Take Picture: If the user leave the room after taking the initial picture, the camera will take another picture and send both pictures to the image processing module.

Image Processing: The system will take the difference of the two pictures, identify new existence or position change of the objects shown in the image.

Update File Storage: The device will record the updated information (including time, Location, picture of object, and so on) of the object into the file storage module.

Input Object Information When the user want to find one object in the room, the user interface will ask the user to input information of the object (like size, color, last time saw it, and so on).

Providing List and Wait Response: After the user has input the information, the system will provide a list of objects with pictures that satisfy the input information.

Provide Detailed Information: When the user find the target object, the device will display the detailed information about the object like showing the current location on the screen.

Pull Out Initial Picture: When the object is not found in the list, the device will display the picture taken just after the program starts to let the user find the object.

Object Taken Out: If the object is still not found, the system will think that the object is taken out of the room or is hidden behind another object.

Rotate Camera: If the device found the human body detected is not in the center of the screen or the camera is covered by something else, it will send signals to the motor to rotate the camera until problem solved.

3.2.2 Individual Product Use Cases

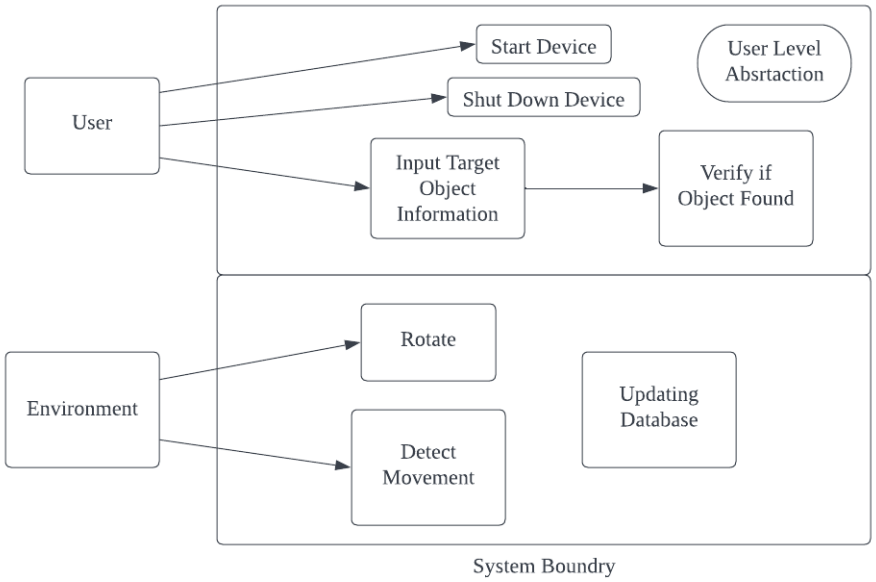

Figure 4: The Picture of Use Case Diagram

The Use Case Diagram shown above describes some actions that the user can interact with the system. The line in the middle shows the boundaries between user and the internal program. The out-most large box shows the boundary for the whole system. All user can do is just to start and shut down the program. The user can also find the target object by typing the information about the object and verify if the object is found. The system will interact with the environment by rotating its camera and detecting the movement of the object in the room. It will also be able to update its database when the position of the object is changed.

3.3 Monitor and Control Variables

3.3.1 Monitor Variable

Table 3: Table of Monitored Variables

Variable Name	**Unit**	**Range**	**Comment**
Object position	pixel sizes for an image	size of the whole picture	a rectangle showing marking the object
Human position	pixel sizes of an image	size of the whole picture	a structure shows human body
Object characteristic	based on software libraries	not applicable	Characteristics about the object to identify it from one another
Object last seen time	timing values from seconds to days	1 day to 365 days	the timing information used to search for an object

3.3.2 Control Variable

Table 4: Table of Control Variables

Variable Name	**Unit**	**Range**	**Comment**
Degree of movement of the motor	degrees	2 degrees to 180 degrees	the angular movement can be made by the motor.
Direction of movement of the motor	not applicable	either positive or negative	the direction that the motor can move
Speed of turning if the motor	degrees per second	10 to 90 degrees per second	the speed of the motor

3.4 Functional Requirements

The following are the functional requirements of the project. They are separated into 2 main parts: Image Processing and Storage, and User Interface Menu.

3.4.1 Image Processing and Storage Requirements

IPR1: The system should be able to identify human's body.
Rationale: When human enters the camera frame, the system needs to be able to identify it and presents a strcture on the image.

IPR2: The system should be able to identify the movement or new existence of different objects.
Rationale: By applying Image processing, the system is able to identify those changes.

IPR3: The system should be able to take a photo once the user leave the room.

IPR4: The system should be able to differentiate one item from another

items which are identified by the system through 3 main parameters, item_shape, item_color and item_size.

IPR5: The system must be able to store all the photos into a file, and indicate the time when it was taken.

IPR6: The system must be able to update the file storage module.

IPR7: The system must be able to name each item with a unique ID.

IPR8: The system should be able to arrange the photos stored in the file in ascending or descending order according to the time it was taken.

IPR9: The system should be able to arrange the photos stored in the file in ascending or descending order according to their IDs.

3.4.2 User Interface Menu Functional Requirements

UIR1: The UI should be able to let user to choose whether to highlight a certain item or not.

UIR2: The UI should be able to let user to switch the ordering method.

UIR3: The UI should be able to let the user login by inputting correct usernme and password.

UIR4: The UI must be able to allow the user to view the system's status at any given point in time.

UIR5: The UI must be able to provide Technical to the user.
Rationale: To ensure that the system will remain operational.

4 Nonfunctional Requirements

The next paragraphs will talk about about non-functional requirements in the designing of the SmartVault, which will be discussed in several different parts.

4.1 Look and Feel Requirements

4.1.1 Appearance Requirements

APR1: The device should not have exposed internal electronic wiring.

APR2: The sharp corners should not be exposed to the users and should be covered by some soft materials.

4.1.2 Style

Not Applicable.

4.2 Usability and Humanity Requirements

4.2.1 Easy of Use Requirements

EUR1: The device should be easy to installed in the room.

EUR2: Fonts, blanks and graphics shown on the screen should be big and visible enough for the user to use without inserting wrong information.

4.2.2 Personalization and Internationalization Requirements

Not applicable.

4.2.3 Learning Requirements

LER1: The software part of the product should be easy to install.

LER2: The product should be easy to be set up in its working environment.
Rationale: The device can identify objects in the room in a short time after powering up.

4.2.4 Understandability and Politeness Requirements

UPR1: The graphics used should be readable and visible to the user.
Rationale: Easy for clients to find objects that wish to find in the room.

4.3 Accessibility Requirements

ACR1: The display window should be concise enough for user to understand

4.4 Performance Requirements

4.4.1 Speed and Latency Requirements

SLR1: The response time of the product to show the location of the object should be less than 5 seconds.

4.4.2 Safety-Critical Requirements

SCR1: The base of the device should be strong enough without falling from the floor.

SCR2: The device should not infringe on user's privacy.

SCR3: The normal rotating speed of the motor should be slow enough without hurting people.
Rationale: The average rotating speed should be controlled to about 30 degrees per seconds.

4.4.3 Precision and Accuracy Requirements

PAR1: The time based value used in the device should be precise to seconds.

PAR2: The location value used in the device should be precise to whole number.

PAR3: Other numerical values used in the device should be rounded to one decimal place.

4.4.4 Reliability and Availability Requirements

RAR1: The device should not over-rotate to an unexpected angle.
Rationale: The range of the angle is set to be -180 degrees to 180 degrees.

RAR2: The device should be available to work for the whole time of except for maintenance or updating time.

4.4.5 Robustness or Fault-Tolerance Requirements

RFR1: The data stored in the file should not be deleted or changed even after the program shuts down.

RFR2: The device should be able to notify user for an error occurs in the program.
Rationale: The user should be notified if he or she has a wrong input or having inappropriate actions.

4.4.6 Capacity Requirements

CAR1: The device should be able to store the any information about the location of each object detected from the camera.

4.4.7 Scalability or extensibility Requirements

Not applicable.

4.4.8 Longevity Requirements

LOR1: The device should store the information about the objects until it is changed into a different environment.

4.5 Operational and Environmental Requirements

4.5.1 Expected Physical Environment

EPE1: The device is supposed to work in any indoor space.

4.5.2 Requirements for Interfacing with Adjacent Systems

Not applicable.

4.5.3 Productization Requirements

Not applicable.

4.5.4 Release Requirements

No applicable.

4.6 Maintainability and Support Requirements

4.6.1 Maintenance Requirements

MAR1: The maintenance for the device should be done by the developers.

4.6.2 Supportability Requirements

SUR1: The device is supported by any computers supports both C and python programming languages.

4.6.3 Adaptability Requirements

Not Applicable.

4.7 Security Requirements

4.7.1 Access Requirements

AER1: Any one except for the users is not allowed to access the any file that stores the information about the objects in the room.

4.7.2 Integrity Requirements

INR1: Data in the files should not be changed unnecessarily.

INR2: The files are locked when the device shuts down.

4.7.3 Privacy Requirements

PRR1: Other users are not allowed to access any file in the computer.

PRR2: The camera should only work for a specific user.

4.7.4 Audit Requirements

Not applicable.

4.7.5 Immunity Requirements

Not applicable.

4.8 Cultural and Political Requirements

4.8.1 Cultural Requirements

Not applicable.

4.8.2 Political Requirements

Not applicable.

4.9 Legal Requirements

4.9.1 Compliance Requirements

CPR1: The performance of the product should not violate the laws that protect the privacy of thee user.

4.9.2 Standards Requirements

Not applicable.

5 Traceability and Priority

For both functional and non-functional requirements, the dependency matrix is made and priorities are assigned to each requirement. The Traceability matrix and priority table are shown below. In addition, the likelihood change of the requirements and the future plan of the requirements are also shown below.

5.1 Traceability Matrix

Table 1: The Table of Requirement Traceability Matrix

Functional Requirement ID	Nonfunctional Requirement ID
IPR2, IPR5, IPR6	LER2
IPR8, IPR9, UIR3	UPR1, PAR2
UIR4	RFR2
IPR6, IPR7, IPR8	CAR1

5.2 Priority Table

Table 2: The Priority Table for each Requirement		
Functional Requirement ID	Nonfunctional Requirement ID	Priority
IPR1, IPR2, IPR3, IPR3, IPR4, IPR5, IPR6, IPR7, UIR1, UIR3	LER2, UPR1, SCR2, PAR1, PAR2, PAR3, RFR1, RFR2, CAR1, EPE1, INR1, CPR1	HIGH
IPR8, IPR9, UIR2, UIR4	APR1, APR2, EUR2, ACR1, SCR3, RAR1, RAR2, SUR1, AER1, INR2, PRR1, PRR2	MID
UIR5	EUR1, LER1, SCR1, MAR1	LOW

5.3 Likelihood of Changes

Table 3: Requirements that are Likely to Change		
Requirement ID	Likelihood	Ways to Change
IPR2(func)	Likely	The image processing method may be changed to achieve a more accurate result.
IPR9(func)	likely	Since time is the easiest way to track an object, the assistance information may be changed so the ordering method may also be changed.
UIR1(func)	likely	The device will remember each object that is moved so this function may be useless.
UIR2(func)	likely	The sorting method is determined by the program not the user, so the ording method may not depend on the user.
EUR1(nonfunc)	likely	The physical installation of the device may be changed after the model is built.
SCR1(nonfunc)	likely	The base may be changed when the installation method of the device changed/
CAR1(nonfunc)	very likely	The device may only choose information that make the sorting algorithm easier.

5.4 Requirement Timeline

Table 4: The Timeline for each Requirement	
Requirement ID	Finish Date
IPR1, IPR3, IPR6, ACR1, PAR1, PAR2, PAR3, SUR1	2022.10.31
RFR1, CAR1, AER1, INR1, PRR1, PRR2	2022.11.15
IPR2, IPR4, IPR5, UIR1, UIR3, UIR4, LER2, RFR2	2022.11.28
EUR2, SLR1, INR2	2022.12.15
IPR7, IPR8, IPR9, UIR2, UPR1, SCR2, SCR3, RAR1	2022.12.30
EUR1, LER1, SCR1, RAR2, EPE1, MAR1, CPR1	2023.1.15

6 Project Issues

6.1 Open Issues

- Limit of 180 rotation degree of servo motor
- Accuracy of object detection
- How to distinguish two objects with very limited resources
- How to recognize the same object in different angles
- How to guarantee the stability of the serial communication

6.2 Off-the-Shelf Solutions

- Huskylens - is a AI camera which can learn new objects and recognize them. It has the machine learning technology enables projects to interact with people and environments which allows many kinds of system control.
- NVIDIA Jetson TX2 - is an embedded AI computer device. It has 8GB memory and 59.7GB/s of memory bandwidth which provides a high

quality AI performance to build efficient AI models including computer vision.

6.3 New Problems

6.3.1 Effects on the Current Environment

Any changes to the exist database may cause the data related to each object missing

6.3.2 Effect on the Installed Systems

Changes to the motorized camera mount will affect the algorithm or logic of the controller board

6.3.3 Potential User Problems

Changes to the user interface may change the way that user used to search the object

6.3.4 Limitations in the Anticipated Implementation Environment

NA

6.3.5 Follow-Up Problems

The changes in computer vision algorithm may cause the whole system malfunction

6.4 Tasks

6.4.1 Project Planning

NA

6.4.2 Planning of the Development Phases

NA

6.5 Migration to the New Product

6.5.1 Requirements for Migration to the New Product

- All the objects data should be stored
- Motorized camera mount should be calibrated before using
- The specification of camera should be kept in same as possible

6.5.2 Data that Has to be Modified or translated for the new system

- Objects data need to be transferred into the new system

6.6 Risks

- Connection lost between the board and camera during the motor movement.
- Inappropriate distance measure and control between the object and the motor which furthermore cause damage or stuck.
- Physical damage from collision with objects .
- Physical damage from wire twisting during rotation.
- Unexpected movement caused by the delay of data transfer

6.7 Costs

Product	Price
USB Camera	$30
PTZ Mount	$25
Arduino	$30
Motors	$15
Total	$100

6.8 User Documentation and Training

6.8.1 User Documentation Requirements

- User manuals
- Installation manuals
- Technical specifications to accompany the product

6.8.2 Training Requirements

NA

6.9 Waiting Room

- Expanding chassis's activity area including rotation angle and planar movement.
- Developing new algorithm regarding data transfer to enable faster real-time reaction.
- Adding alarm in case that object not found in the assigned area as an application of storage security.

6.10 Ideas for Solutions

- Use DC motor with decoder to implement the unlimited rotation degree of camera mount
- Use Field Oriented Control algorithm to implement the unlimited rotation degree of camera mount
- Use frame-to-frame compassion to detect the relocation of objects
- Predict the possible location of objects by tracking the users path in the room
- Beeper alert when the camera view is not cleared

References

7 Appendix

7.1 Symbolic Parameters

N/A

7.2 Reflection

Our team consists of 5 Mechatronics student and is developing a Mechatronics system where both hardware and software components are essential for success. There are may different skills and knowledge that ties into the success of the project. Many of these required skills might new and unfamiliar to some of the group members. However, as this is a project over a span of 6 months, there are plenty of time and opportunities to pick up and master some new skills. The learning of such skills will be delegated to different team members and mastered by them before delivering and teaching others member about it.

The project is heavily relying on a functional, reliable, and efficient software system that involve live time object detection and image processing. The coding will be done in Python, so each member needs to have adequate skill of programming in Python and able to develop and test a python program. Knowledge of image processing and object detection also should be acquired. In order to master the skills for programming in python and image processing, team members should practice and conduct research on different object detection method. It is required for the assigned team members to compare and learn from different method and making sure that our chosen method would yield the best results given our requirements. This part of the learning is assigned to Edward, Peng and Jinhua. Edward has gained relevant knowledge through courses he took and projects he did. Peng and Jinhua have great enthusiasm for this topic and would like to conduct research on it.

Another software component is the user interface. Members of the team should be able to build and maintain a working web-page using JavaScript. This skill is obtained by Edward through his co-op and other experiences. Erping and Jinhua also are assigned to perfect their abilities in such area. The path to success for this is to read relevant documents and tutorials, as well as practising with small projects. In order to communicate between the software and hardware, related programming skills are needed. Erping as the

lead developer for this topic has multiple experiences and with the help of Guangwei who has exceptional skills in this field, they will be able to excel the required skills and guide other members of the team.

On the hardware side, Guangwei will be the lead and responsible for the learning and teaching of the related skills. 3D modeling is needed for the building and 3D printing the camera mount. This skill is assigned to Guangwei and can be mastered through practising and watching tutorials. Arduino board is the chosen board for handling the control of the motors and repositioning of the camera. Sufficient knowledge and skills to program Arduino is needed. This task is assigned to Erping and Guangwei. They both have experiences in this field and will be able to teach the rest of the team. Knowledge associated with handling motor movement is also needed and can be mastered by reviewing course notes and refreshing past knowledge.

Aside from the hardware and software system, there are also skills associated with team management and team communication. In order to maintain a desired work plan and team coordination, it is important for all members of the team to possess the skills to work under a team environment. Members need to be comfortable and familiar with Github and in order to excel, one must practice different Github actions and commends. One also can improve on this by watching tutorial videos. Working in a team environment, it is important to have good communication skills and making sure people are on the same page. This can be mastered by actively engage in team meetings and ask for help when needed.

To conclude, for a professional project like the capstone, there are many aspects to it and many skills are required. It is important to delegate different learning tasks to individual and be able to teach the new skill-set to the rest of the team. The following are the skills mentioned in previous paragraphs and is the overview of the assignment of the learning process:

1. communication and team management (All)
2. Github (All)
3. Python, image processing and object detection (Edward, Jinhua, Peng)
4. Arduino and control (Erping and Guangwei)
5. Motor control (Erping and Guangwei)
6. Javascript (Edward, Peng, Jinhua)

7. 3D modeling and 3D printing (Erping and Guangwei)
8. Writing and presenting reports (All)