Software Requirements Specification

Cyclops Ride Assist: Real-time bicycle crash detection and blindspot monitoring.

Team 9
Aaron Li (lia79)
Amos Cheung (cheuny2)
Amos Yu (yua25)
Brian Le (leb7)
Manny Lemos (lemosm1)

Table of Contents

List of Tables

List of Figures

1. Revision History

Table 1.1: Revision History

Date	Developer(s)	Change
2022-10-03	Aaron Li, Amos Cheung, Amos Yu, Brian Le, Manny Lemos	Document created
2022-10-20	Amos Yu	Improved formatting

2. Project Drivers

2.1. Project Purpose

The purpose of this project is to create a system that helps cyclists to have a more secured experience especially on roads without bike lanes.

Cyclops Ride Assist (CRA) is going to be an all-in-one, easily mountable, and quick to setup system that adds modern car safety features onto a bike, such as blind spot detection and crash detection. The system will also have a built in headlamp that will illuminate during night time, not just for the cyclist to see but also for cars around to realize the bike.

2.2. Project Scope

CRA is going to be a bike assist system with convenient mounting, accurate crash detection, video buffering and saving, reliable blindspot monitoring and a user controlled headlamp that helps cyclist to have a peace of mind while riding on the road. Although CRA is primarily targeted towards road cyclists, it will able be useful for cyclists who ride on mountains or trails.

2.3. Behaviour Overview

The user can press the power button to turn on the CRA. Once it turns on, it will start to record the forward point of view of the bike. If a bike crash is detected, the system will store the past BUFFER_TIME_MINUTES of footage so the user can look back at the events leading up to the crash. Also on the back side of the system, CRA will watch out for cars approaching the bike at blind spots and alert the cyclist with an indicator. The system also has a headlamp that is easily switched on and off for dark environment.

2.4. Project Stakeholders

The project stakeholders are as follows:

- The project proposers (Aaron Li, Amos Cheung, Amos Yu, Brian Le, Manny Lemos)
- The project supervisor (Dr Spencer Smith)
- The teaching assistants (Nicholas Annable)
- The user (cyclist)

2.5. Product Users

The user will be all cyclists.

3. Project Constraints

3.1. Mandated Constraints

A list of constraints which will adhered to during the design and development of this system.

Mandated Constraint 1	**This capstone project must be completed prior to the final demonstration.**
Rationale 1	Project deadlines provided in the course outline dictate project milestone which must be met. One such milestone is the final demonstration which occurs between March 20-31 2023.

Mandated Constraint 2	**The total cost of the components used in this design must not exceed $750.**
Rationale 2	The final deliverable must be a competitor in the open market. Further, using funds to purchase an off-the-shelf product is not allowed.

Mandated Constraint 3	**The system must be able to analyze inputs to produce desired results in real time.**
Rationale 3	Real time analysis and response is an integral component of the cyclops ride assist system. More precisely, desired results are only of value if they can be delivered on time every time.

3.2. Naming Conventions and Definitions

3.2.1. Naming Conventions

Name	Explanation
Client	See user.
CRA	Abbreviation of Cyclops Ride Assist.
Cyclist	A person who operates a bicycle as a means of transportation.
User	A person who will operate the final product. See Cyclist.

3.2.2. Constants

- Gravity = 9.81 m/s^2

3.2.3. Monitored Variables

Monitor Name	Monitor Description	Monitor Type	Units
αx	Measures acceleration parallel to the path of the bicycle.	acceleration	m/s2
αy	Measures acceleration perpendicular to the path of the bicycle along the plane of the ground.	acceleration	m/s2
αz	Measures acceleration in the vertical direction.	acceleration	m/s2

Monitor Name	Monitor Description	Monitor Type	Units
tilt	Measures the vertical tilt of the system relative to a calibrated absolute level.	rotation	rad
vfront	Video feed from the front facing camera.	Video	N/A
vrear	Video feed from the rear facing camera.	Video	N/A
sw_flashlight	Switch which controls the flashlight.	Boolean	N/A

3.2.4. Controlled Variables

Controlled Name	Controlled Description	Controlled Type	Units
led_blind_spot	Indicates a vehicle is in the bicycles blind spot.	Boolean	N/A
flashlight	Indicates the state of the flashlight.	Boolean	N/A

3.3. Relevant Facts and Assumptions

3.3.1. Relevant Facts

- A Raspberry Pi can consume up to 6.4 watts of power under maximum CPU load.

3.3.2. Assumptions

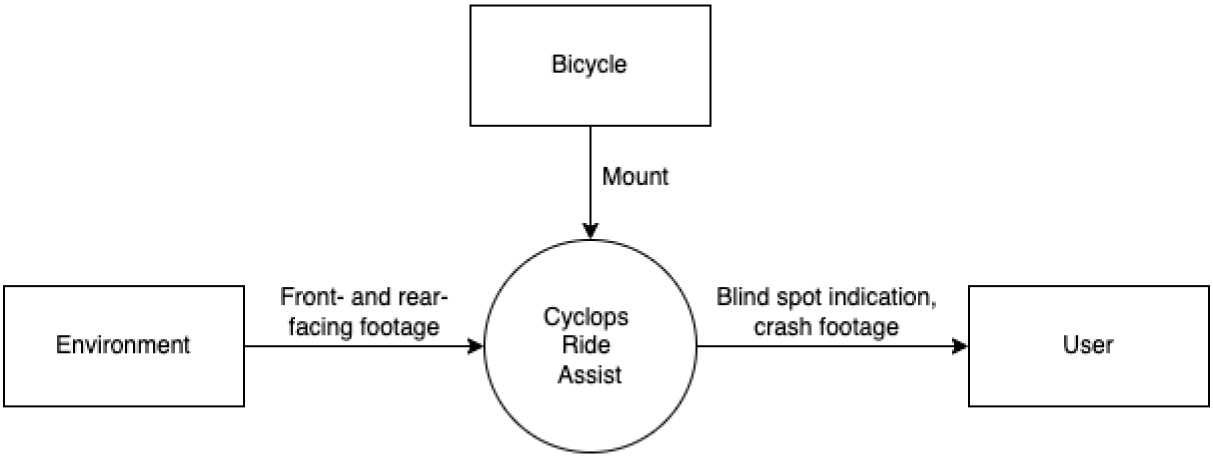
Assumptions will enable developers to cull the scope of the problem(s) being undertaken. As such, assumptions will detail limitations of the system.

Assumption 1	**Cyclists will mount and dismount their bikes with care.**
Rationale 1	Violent mounting and dismounting of one's bicycle may result in unintended triggering of crash detection and subsequent video logging. The system will not be able to distinguish between violent (dis)mounting and true crashes.

Assumption 2	**While on the road, cyclists will abide by traffic laws. This means travelling in marked bike lanes where available.**
Rationale 2	The system will not be able to distinguish between parked vehicles, which may appear momentarily in a cyclists blind-spot, and moving vehicles. As a result, if a cyclist is not travelling in designated bike lanes they may be subject to increased instances of false blind-spot detection triggers.

4. Context Diagrams

Figure 4.1: CRA System Context Diagram



5. Functional Decomposition Diagrams

Figure 5.1: CRA Functional Decomposition Diagram

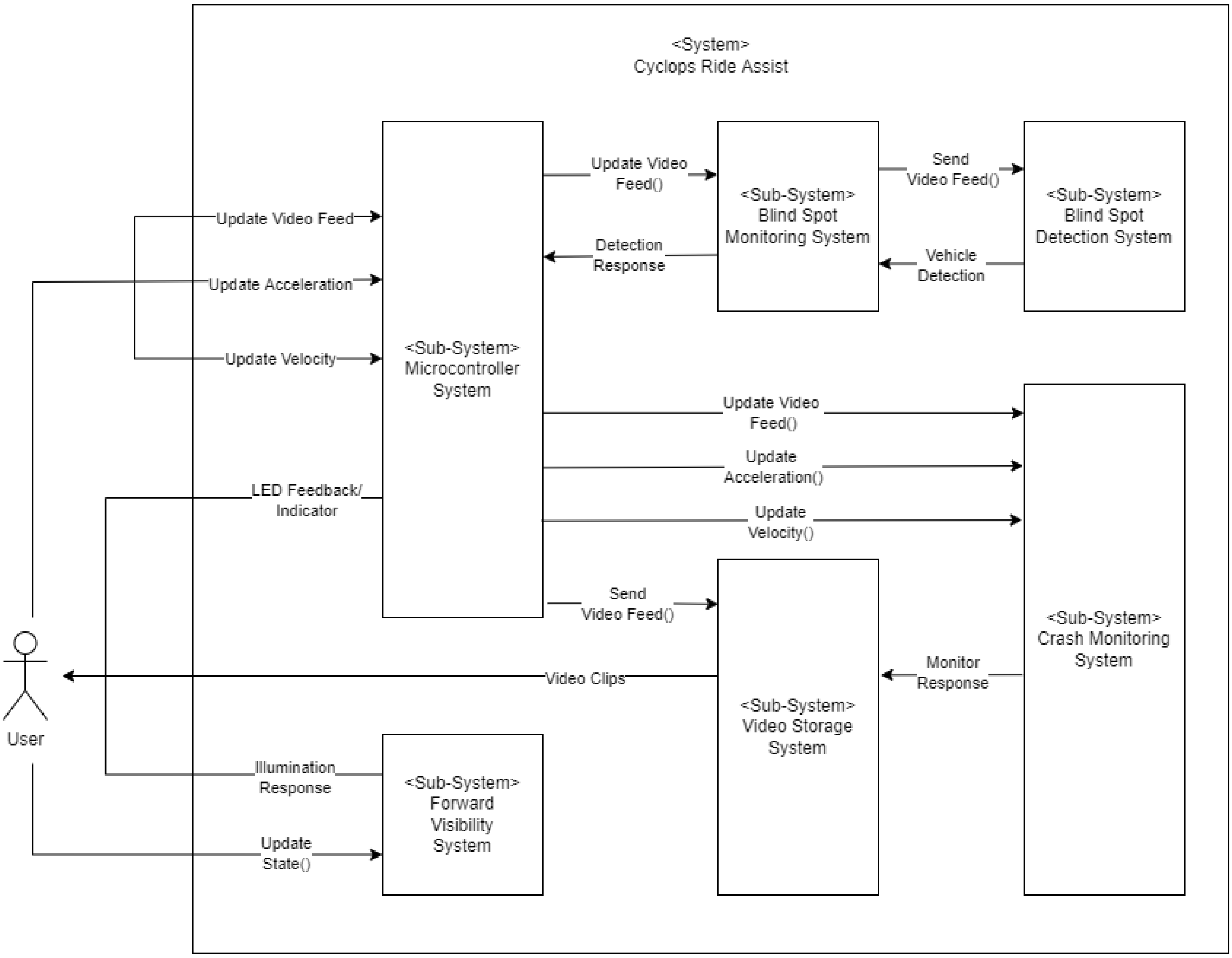


Figure 5.2: CRA Data Flow Diagram

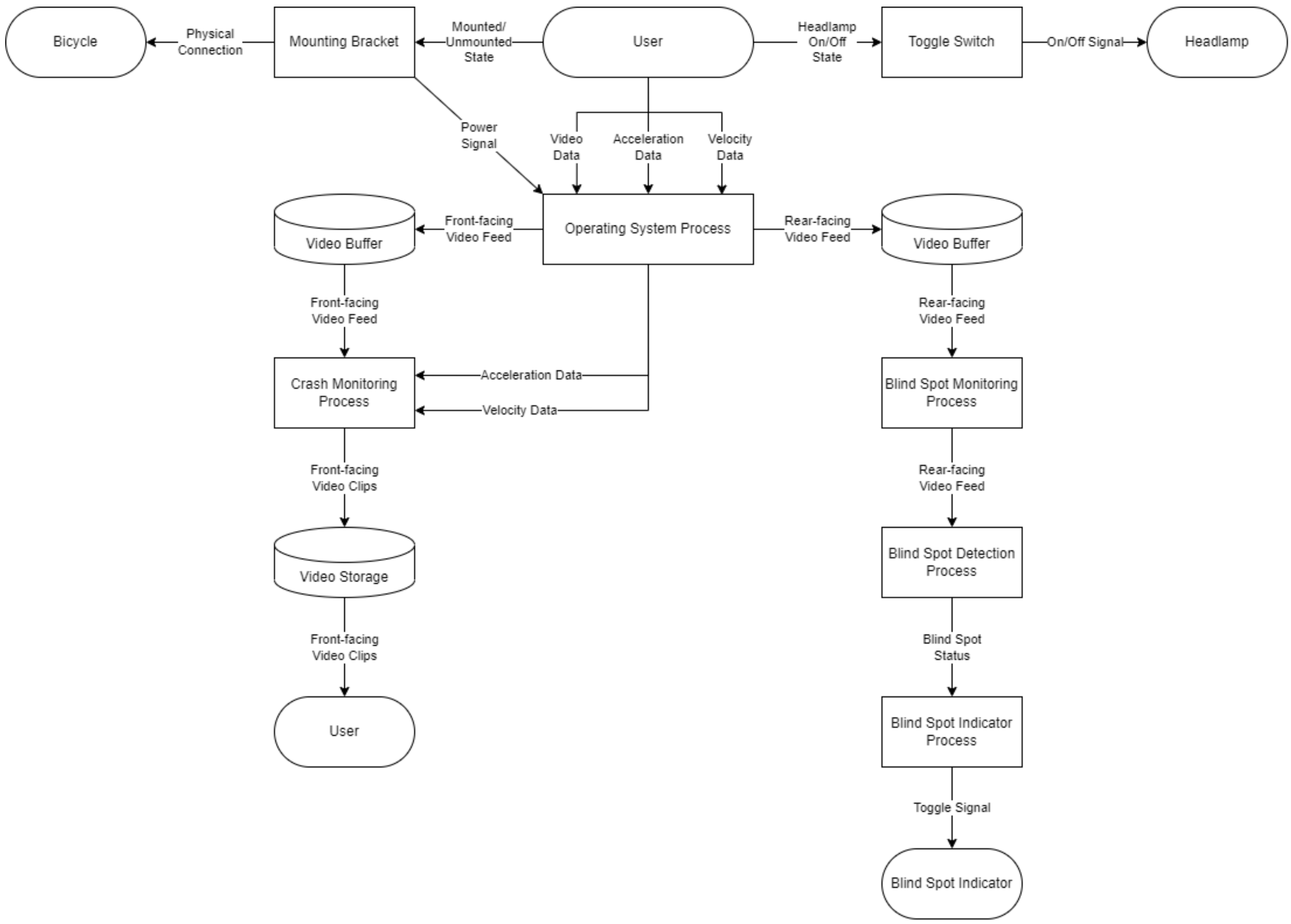
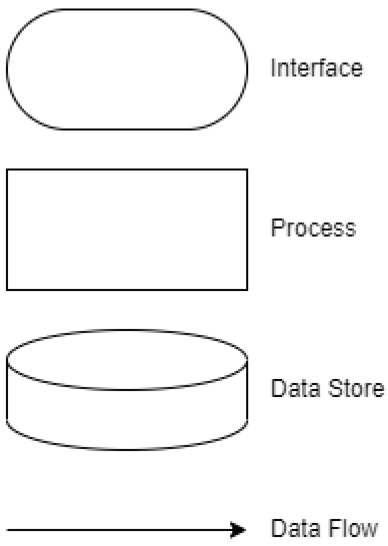


Figure 5.3: Legend for the CRA Data Flow Diagram



6. Functional Requirements

6.1. Scope of Work

CRA will be separated into 3 main components: Microcontroller, Blindspot, Crash Detection. The scope of the Functional requirements will include the mounting, microcontroller, blindspot monitoring, crash monitoring, and forward visibility systems. Each system will be developed as its own module and then brought together with communications between each module facilitated through the microcontroller. Members of the team will be distributed to each member to be completed with the final goal being to merge each for a final product.

6.2. Business Data Model and Data Dictionary

Refer to section 3.2.3 Monitored Variables for the dictionary used throughout the Functional Requirements.

6.3. Scope of the Product

The scope of the product will be a physical enclosure which will contain all of the microcontrollers, sensors, and battery required to facilitate all of CRA's functionalities. The product will be mounted onto a bike where CRA can then monitor a riders blindspot and whether they get into a crash. The blindspot detection will be relayed to the user through a visible LED on the enclosure facing a side which the user can clearly see in their FOV. The product will be capable of clipping/logging the last few seconds of real time video when the user gets into a crash. When visibility becomes an issue during late riding hours, the user will also have a light to turn on at their disposal for increased clarity.

6.4. Functional Requirements

6.4.1. CRA Requirements

CFR1	**CRA must be able to light up LED when a close car/vehicle is recognized. Such that: VehicleDetected = led_blind_spot(VF) where if VF = Vehicle is detected -> LEDLight = 1**
Rationale	CRA should be able to visually inform the rider that there is a car in their blindspot

CFR2	**CRA must be able to provide a live video feed through out users ride**
Rationale	CRA needs the live feed inorder for its computer vision to detect vehicles and for video logging from the crash detection

CFR3	**CRA must be able to take in accurate acceleration information. Such that: A_{bike} = Acceleration(ax, ay, az)**
Rationale	CRA needs the accurate acceleration to determine when a crash has occured

CFR4	**CRA must be able to take in accurate velocity information**
Rationale	CRA needs the accurate velocity to determine when the user is moving to activate both crash and blindspot systems

CFR5	**CRA must know when the user is moving. Such that: Moving = $V_{bike} > 0$**
Rationale	CRA should only log a crash once it has been establish that the user was moving on their bike

CFR6	**CRA must recognize a vehicle within the video feed. Such that: isVehicle = Feed(vrear) where if vrear contains vehicle -> isVehicle = 1**
Rationale	CRA should recognize that there is a car in the users blindspot

CFR7	**CRA must recognize the user has crashed. Such that: crashed = CrashMonitor(A_{bike}) where if A_{bike} > AcceptableG -> crashed = 1**
Rationale	CRA should recognize when to discard buffered video footage and when to save it.

CFR8	**CRA must clip/log the last BUFFER_TIME_MINUTES minutes of the Video Feed when a crash is detected. Such that videoLog(vfront) - > V, SD = {C1, C2 ..}**

CFR8	**CRA must clip/log the last BUFFER_TIME_MINUTES minutes of the Video Feed when a crash is detected. Such that videoLog(vfront) - > V, SD = {C1, C2 ..}**
Rationale	CRA should save the video feed of the moments leading up to the crash on the integrated SD card

CFR9	**CRA must be able to notify when the SD card is full. Such that: sdFull = isSDFull(SD) where if SD.size()/(clipSize * clipCount) $<$ clipSize -> sdFull = 1**
Rationale	CRA should save the video feed of the moments leading up to the crash on the integrated SD card

CFR10	**CRA must only be able to run the crash detection when an SD card is inserted into the system**
Rationale	CRA should only be able to run the crash detection functionality once an SD card is included in the system

CFR11	**CRA shall be able to determine when a componenet is no longer operational due to low power levels. Such that: ErrorLowPower = $V_{battery}$ $<$ V_{min}**
Rationale	CRA should only be able to run the crash detection functionality once an SD card is included in the system

CFR12	**CRA must be able to continue running its video feed after a clip has been logged**
Rationale	CRA should keep the system rolling in the case that the user is able to continue biking after an initial crash

CFR13	**CRA must be able to continue running its crash detection system after a crash has been detected**
Rationale	CRA should keep the system rolling in the case that the user is able to continue biking after an initial crash

CFR14	**CRA must be able turn on front lights when prompted by the user. Such that lightON = illuminationResp(flashLightState)**
Rationale	CRA should keep the system rolling in the case that the user is able to continue biking after an initial crash

7. Non-Functional Requirements

7.1. Look and Feel Requirements

7.1.1. Appearance Requirements

CNFR1	**The appearance of CRA will be white.**
Rationale	This is to act as a safety mechanism to allow the bicycle/motorbike to be more visible at night.

CNFR2	**CRA will be contained in a mechanically created system mounted on the bicycle.**

CNFR2	**CRA will be contained in a mechanically created system mounted on the bicycle.**
Rationale	This is to ensure that the the components will not interfere with the system or with the physical bicycle itself.

CNFR3	**There will be no offensive painting or colours on CRA.**
Rationale	This is to ensure that no one is offended by the design style.

7.1.2. Style Requirements

CNFR4	**CRA will be non-bulky and constructed in a minimalist way.**
Rationale	This is to ensure that the system does not become distracting for the users or others on the road.

7.2. Usability and Humanity Requirements

7.2.1. Ease of Use Requirements

CNFR5	**CRA can be easily attached to the bicycle with minimal effort.**
Rationale	This is to ensure that the user will want to and find it easy to use the product.

CNFR6	**The software application of CRA will allow for a minimal amount of clicks or touches.**
Rationale	This will allow users to easily access the files and videos they require.

CNFR7	**CRA will be designed in a way that can be easily understood and used by teenagers and adults for their own bicycles.**
Rationale	This is to ensure that the system can be understood easily by different age groups.

CNFR8	**CRA will be able to be used by people with minimal education or training.**
Rationale	This is to ensure that the system can be understood easily and taught to different educational groups.

7.2.2. Personalization and Internationalization Requirements

CNFR9	**Users will be able to make small modifications to CRA.**
Rationale	This is needed so that users can make adjustments to allow the system to better fit their own personal bicycle.

7.2.3. Learning Requirements

CNFR10	**CRA shall be easy for anyone to learn within a short time.**
Rationale	This is so that anyone can use the system easily.

CNFR11	**CRA will be able to be created by an engineer with one week of training.**

CNFR11	**CRA will be able to be created by an engineer with one week of training.**
Rationale	This is so that any engineer can upgrade or perform maintenance on the system easily.

7.2.4. Understandability and Politeness Requirements

CNFR12	**The software application of CRA will use lanaguge, words, and symbols that are non-technical and understandable by a regular user.**
Rationale	This is so that it will be understandable by the global community of users.

7.2.5. Accessibility Requirements

CNFR13	**CRA will have signals and cues to alert the user.**
Rationale	This is in the case that a user may have some visual or auditory impairment

CNFR14	**CRA will be able to be mounted to all types of bicycles.**
Rationale	This is to ensure that as many users can benefit.

7.3. Performance Requirements

7.3.1. Speed and Latency Requirements

CNFR15	**CRA will have a maximum response time of RESPONSE_TIME_MILLISECONDS.**
Rationale	This is to ensure that the user is able to use CRA quickly and get on their way.

CNFR16	**CRA will upload the video file to the external storage with a max time of UPLOAD_TIME_SECONDS.**
Rationale	This is to ensure that the user is able to access their files quickly.

CNFR17	**CRA will be able to determine an accident within 1s.**
Rationale	This is to ensure that the camera will keep the recording of before and after the collision.

CNFR18	**CRA can see if a car is nearby to alert the user within 1s with a signal or cue.**
Rationale	This is to allow the user to have enough time to ensure they are in a safe position.

CNFR19	**CRA will have an alert on its software application to let the user if the storage is low.**
Rationale	This is to ensure CRA will store data in the case of a collision.

CNFR20	**CRA will have an alert within LOW_BATTERY_ALERT_TIME_SECONDS of startup on its software application to let the user know if the system components are running out of battery.**
Rationale	This is to ensure CRA does not crash while in use.

7.3.2. Safety-Critical Requirements

CNFR21	CRA will not cause any external damage to the bicycle.
Rationale	This is to ensure the CRA which could result in loss of safety for the rider.

CNFR22	CRA will not emit any harmful toxins to the environment.
Rationale	This is to ensure that there is no risk to the environment.

CNFR23	CRA will have all wiring shielded from human contact.
Rationale	This is to ensure that the user is not negatively affected.

7.3.3. Precision and Accuracy Requirements

CNFR24	The precision of CRA will be to three decimal places.
Rationale	This is to keep as many significant digits to ensure relative precision with other components.

CNFR25	The accuracy of CRA speed reading with be within 1 km/h.
Rationale	This is to ensure that the system is working coherently with the accelerometer subsystem and software application.

CNFR26	The accuracy of CRA timed camera reading with be within 1 second.
Rationale	This is to ensure that the accuracy of the collision is timed correctly.

7.3.4. Reliability and Availability Requirements

CNFR27	The camera on CRA will be able to record once the previous video has finished uploading to storage.
Rationale	This is to ensure that Cyclops can continuously run for the user.

CNFR28	CRA will be able to be used 24 hours per day, 365 days per year.
Rationale	This is to ensure that bikers are covered throughout all times of anyday of any season.

7.3.5. Robustness or Fault-Tolerance Requirements

CNFR29	CRA will be able to work consistently even in the case of user drops when in transport.
Rationale	This is to ensure that CRA is set up to withstand accidents at home or on the road.

7.3.6. Capacity Requirements

CNFR30	CRA will be able to store multiple videos in an external storage system with a time of BUFFER_TIME_MINS.
Rationale	This is to ensure that the user is able to see past videos and continuously record new videos, minimizing downtime.

7.3.7. Scalability and Extensibility Requirements

CNFR31	**CRA will have extra room in its software and hardware storage to allow for additional components.**
Rationale	This is to ensure that CRA is scaleable to additional upgrades.

CNFR32	**CRA will be usable to all bicycle users within the next ten years.**
Rationale	This is to allow for continuous integration into newer bicycles.

7.3.8. Longevity Requirements

CNFR33	**CRA will have a lifespan of five years with expected cleaning and maintenance.**
Rationale	This is expected due to deterioration of the products.

7.4. Operational and Environmental Requirements

7.4.1. Expected Physical Environment

CNFR34	**The expected physical environment will be on a road. Other possible physical environments include trails, sidewalks, pathways.**
Rationale	This is expected as most cyclists use their bicycles outdoors.

CNFR35	**CRA will be used by cyclists in any type of weather.**
Rationale	This is expected as most cyclists use their bicycles outdoors.

7.4.2. Requirements for Interfacing with Adjacent Systems

CNFR36	**Components of the CRA can be interfaced with other software applications and hardware systems such as PCs and adapters.**
Rationale	This will include the video storage cards and CRA software GUI.

7.4.3. Productization Requirements

CNFR37	**CRA's hardware and software will be publicly available for use to those interested in furthering the system.**
Rationale	This will allow for continuous improvement of the product by external developers.

7.4.4. Release Requirements

CNFR38	**CRA will be available as a one-time download and system integration per user**
Rationale	This will also allow for continuous improvement of the product by external developers.

7.5. Maintainability and Support Requirements

7.5.1. Maintenance Requirements

CNFR39	CRA will have crash logs when the software or hardware fails.
Rationale	This is to allow for the developers to work on a fix for the issue.

CNFR40	CRA will be built in several modules.
Rationale	This is so that components can be removed and replaced when need be.

7.5.2. Supportability Requirements

CNFR41	CRA will have an instruction manual included.
Rationale	This is to ensure that any common mistakes or confusion can be fixed easily by the user.

7.5.3. Adaptability Requirements

CNFR42	CRA's external hardware storage will be able to run under any operating system to view files.
Rationale	This is to ensure that CRA is compatible and adaptable with any system for all users.

7.6. Security Requirements

7.6.1. Access Requirements

CNFR43	CRA will allow the users to access their videos freely from an external hardware storage drive.
Rationale	This is to allow the user to connect it to various systems.

7.6.2. Integrity Requirements

N/A	N/A
Rationale	N/A

7.6.3. Privacy Requirements

CNFR44	CRA will not store any data in the cloud to protect the user's personal privacy.
Rationale	The data will only be stored in the user's personal external storage.

7.6.4. Audit Requirements

N/A	N/A
Rationale	N/A

7.6.5. Immunity Requirements

CNFR45	CRA will be connected per user locally.
Rationale	This is to ensure that there will be no malicious interference from unwanted third-parties.

7.7. Cultural and Political Requirements

7.7.1. Cultural Requirements

CNFR46	CRA will have its primary language set as English (Canadian).
Rationale	This is the language of the primary developers.

CNFR47	CRA will not have any offensive language
Rationale	This is to not offend any users.

7.8. Legal Requirements

7.8.1. Compliance Requirements

CNFR48	CRA must follow all safety requirements according to the user's bicycle's standards and requirements.
Rationale	This will include weight and size restrictions to ensure adherance to all types of bicycles.

7.8.2. Standards Requirements

CNFR49	CRA will comply with all product quality standards for automotive products.
Rationale	This will include all processes between idea generation to customer use and satisfaction and will ensure that the industry will not blacklist the product.

8. Project Issues

8.1. Open Issues

1. Creating accurate documentation for the system specifications.
2. Starting to develop the software, purchasing some hardware to begin system creation.

8.2. Off-the-Shelf Solutions

8.2.1. Ready-Made Products

1. INNOVV ThirdEYE

- INNOVV ThirdEYE is a blindspot detection system for motorcycles that uses sensors to determine if there are any objects close to the user. It can be either used through a mirror lens or a watch.
- https://www.innovv.com/innovv-thirdeye

2. Senzar Motorcycle Sensor

- The Senzar Motorcycle Sensor is another system that uses LEDs, vibration, and radar sensors.
- https://meetsenzar.com/pages/senzar-m1-motorcycle-bsm

8.2.2. Reusable Components

1. Cameras
2. Sensors
3. Cables, Connectors, Wires
4. Printed Circuit Boards

8.3. Risks

1. Components (sensor, camera) will deteriorate over time, leading to inaccuracies.
2. Components can be damaged in case of a cycle crash.
3. Incorrect image (car vs a tree) is read through Computer Vision, leading to inaccuracies.
4. Storage space is exceeded or becomes corrupted.

8.4. Costs

1. Raspberry PI Robot
2. Sensors
3. Cameras
4. 3D Printed Mount and Storage System
5. Cables/Wires
6. Bicycle

9. Waiting Room

9.1. Likely Changes

CNFR1	The appearance of CRA will be white.
Rationale	The appearance could be other colors if designers decide to make a change

CNFR24	The precision of CRA will be to three decimal places.
Rationale	Based on the impplementation the precision could change to ensure better accuracy

CNFR25	The accuracy of CRA speed reading with be within 1 km/h.
Rationale	Based on the impplementation the precision could change to ensure better accuracy

CNFR26	The accuracy of CRA timed camera reading with be within 1 second.
Rationale	Based on the impplementation the precision could change to ensure better accuracy

CNFR32	CRA will be usable to all bicycle users within the next ten years.
Rationale	The duration could change based on the method of implementation used during design process

CNFR33	CRA will have a lifespan of five years with expected cleaning and maintenance.
Rationale	The lifespan could increase or decrease when designing the product

9.2. Unlikely Changes

CFR1	**CRA must be able to light up LED when a close car/vehicle is recognized. Such that: VehicleDetected = led_blind_spot(VF) where if VF = Vehicle is detected -> LEDLight = 1**
Rationale	This is one of the core function blind spot detection
CFR2	**CRA must be able to provide a live video feed through out users ride**
Rationale	This is one of the core function crash detection
CFR3	**CRA must be able to take in accurate acceleration information. Such that: A_{bike} = Acceleration(ax, ay, az)**
Rationale	This is one of the core requirement of crash detection
CFR4	**CRA must be able to take in accurate velocity information**
Rationale	This is one of the core requirement of crash detection and blind spot detection
CFR5	**CRA must know when the user is moving. Such that: Moving = $V_{bike} > 0$**
Rationale	This is one of the core requirement of crash detection
CFR7	**CRA must recognize the user has crashed. Such that: crashed = CrashMonitor(A_{bike}) where if A_{bike} > AcceptableG -> crashed = 1**
Rationale	This is one of the core requirement of crash detection
CFR8	**CRA must clip/log the last BUFFER_TIME_MINUTES of the Video Feed when a crash is detected. Such that videoLog(vfront) - > V, SD = {C1, C2 ..}**
Rationale	This is one of the core requirement of crash detection
CFR9	**CRA must be able to notify when the SD card is full. Such that: sdFull = isSDFull(SD) where if SD.size()/(clipSize * clipCount) $<$ clipSize -> sdFull = 1**
Rationale	This is one of the core requirement of crash detection
CFR10	**CRA must only be able to run the crash detection when an SD card is inserted into the system**
Rationale	This is one of the core requirement of crash detection
CFR11	**CRA shall be able to determine when a componenet is no longer operational due to low power levels. Such that: ErrorLowPower= $V_{battery}$ $<$ V_{min}**
Rationale	Components needed to be powered in order to have CRA fully working at all times
CFR12	**CRA must be able to continue running its video feed after a clip has been logged**
Rationale	This is one of the core requirement of crash detection
CFR13	**CRA must be able to continue running its crash detection system after a crash has been detected**
Rationale	This is one of the core requirement of crash detection

CFR14	**CRA must be able turn on front lights when prompted by the user. Such that lightON = illuminationResp(flashLightState)**
Rationale	This is the core function of the headlamp

CNFR2	**CRA will be contained in a mechanically created system mounted on the bicycle.**
Rationale	A container would be needed to put all components together

CNFR3	**There will be no offensive painting or colours on CRA.**
Rationale	This is one of the core value of the designers to keep everyone inclusive and safe while using the product

CNFR4	**CRA will be non-bulky and constructed in a minimalist way.**
Rationale	CRA has to be carried around easily and also not obstructing the user when cycling

CNFR5	**CRA can be easily attached to the bicycle with minimal effort.**
Rationale	This is one of the goal of the product

CNFR6	**The software application of CRA will allow for a minimal amount of clicks or touches.**
Rationale	This is one of the requirement for a user friendly product

CNFR7	**CRA will be designed in a way that can be easily understood and used by teenagers and adults for their own bicycles.**
Rationale	This is one of the requirement for a user friendly product

CNFR8	**CRA will be able to be used by people with minimal education or training.**
Rationale	This is one of the requirement for a user friendly product

CNFR9	**Users will be able to make small modifications to CRA.**
Rationale	Allowing changes to the system will make it more user friendly

CNFR10	**CRA shall be easy for anyone to learn within a short time.**
Rationale	This is one of the requirement for a user friendly product

CNFR11	**CRA will be able to be created by an engineer with one week of training.**
Rationale	This simplifies the maintanence process

CNFR12	**The software application of CRA will use lanaguge, words, and symbols that are non-technical and understandable by a regular user.**
Rationale	This is one of the requirement for a user friendly product

CNFR13	**CRA will have signals and cues to alert the user.**
Rationale	This is to make sure the system is inlcusive to all cyclist even if they might have disabilities

CNFR14	**CRA will be able to be mounted to all types of bicycles.**
Rationale	This is one of the requirement for a user friendly product

CNFR15	CRA will have a maximum response time of 5 seconds.
Rationale	This is one of the requirement for a user friendly product

CNFR16	CRA will upload the video file to the external storage with a max time of 60 seconds.
Rationale	This is one of the requirement for a user friendly product

CNFR17	CRA will be able to determine an accident within 1s.
Rationale	This is a core function of crash detection

CNFR18	CRA can see if a car is nearby to alert the user within 1s with a signal or cue.
Rationale	This is a core requirement for blind spot detection

CNFR19	CRA will have an alert on its software application to let the user know if the storage is low.
Rationale	This is to ensure that the videoing part of crash detection would not be obstructed by other external factors

CNFR20	CRA will have an alert within 5 seconds of startup on its software application to let the user know if the system components are running out of battery.
Rationale	Components needed to be powered in order to have CRA fully working at all times

CNFR21	CRA will not cause any external damage to the bicycle.
Rationale	This is to keep the safety of the cyclist and also to keep the bike in it's original form

CNFR22	CRA will not emit any harmful toxins to the environment.
Rationale	This is to ensure the safety of the users and environment

CNFR23	CRA will have all wiring shielded from human contact.
Rationale	This is to ensure the safety of the users

CNFR27	The camera on CRA will be able to record once the previous video has finished uploading to storage.
Rationale	This is to ensure the product is functioning properly

CNFR28	CRA will be able to be used 24 hours per day, 365 days per year.
Rationale	This is to ensure the product is functioning properly

CNFR29	CRA will be able to work consistently even in the case of user drops when in transport.
Rationale	This is a requirement to ensure product is durable

CNFR30	CRA will be able to store multiple videos in an external storage system with a time of BUFFER_TIME_MINS.
Rationale	This is one of the core function of crash detection

CNFR31	CRA will have extra room in its software and hardware storage to allow for additional components.

CNFR31	CRA will have extra room in its software and hardware storage to allow for additional components.
Rationale	This is needed to achieve strech goals of the project

CNFR34	The expected physical environment will be on a road. Other possible physical environments include trails, sidewalks, pathways.
Rationale	This is required in order for the product to work

CNFR35	CRA will be used by cyclists in any type of weather.
Rationale	This is required in order for the product to work

CNFR36	Components of the CRA can be interfaced with other software applications and hardware systems such as PCs and adapters.
Rationale	This is desired to expand the product's usability

CNFR37	CRA's hardware and software will be publicly available for use to those interested in furthering the system.
Rationale	External help could improve and make the product more robust

CNFR38	CRA will be available as a one-time download and system integration per user
Rationale	External help could improve and make the product more robust

CNFR39	CRA will have crash logs when the software or hardware fails.
Rationale	This will help developers to make changes when nesessary to ensure the product is in good condition

CNFR40	CRA will be built in several modules.
Rationale	This is needed as the product has a lot of componenets and functions

CNFR41	CRA will have an instruction manual included.
Rationale	This is one of the requirement for a user friendly product

CNFR42	CRA's external hardware storage will be able to run under any operating system to view files.
Rationale	This is one of the requirement for a user friendly product

CNFR43	CRA will allow the users to access their videos freely from an external hardware storage drive.
Rationale	This is one of the requirement for a user friendly product

CNFR44	CRA will not store any data in the cloud to protect the user's personal privacy.
Rationale	This is to ensure the privacy of the user

CNFR45	CRA will be connected per user locally.
Rationale	This is to ensure the privacy of the user

CNFR46	**CRA will have its primary language set as English (Canadian).**
Rationale	This is the most well known language around the world to ensure most people can understand and use the product

CNFR47	**CRA will not have any offensive language**
Rationale	This is one of the core value of the designers to keep everyone inclusive and safe while using the product

CNFR48	**CRA must follow all safety requirements according to the user's bicycle's standards and requirements.**
Rationale	This is to ensure the safety of the user

CNFR49	**CRA will comply with all product quality standards for automotive products.**
Rationale	This is to ensure the product is safe for the users and also the surroundings

10. Appendix

10.1. Symbolic Parameters

Table 10.1.1: List of Symbolic Parameters

Parameter	**Description**
BUFFER_TIME_MINUTES	The length of footage that will be saved after an accident occurs (in minutes).
UPLOAD_TIME_SECONDS	The maximum time required to upload a video to the external storage device (in seconds).
RESPONSE_TIME_MILLISECONDS	The maximum time required to update the output upon a change in the input (in milliseconds).
LOW_BATTERY_ALERT_TIME_SECONDS	The maximum time after startup required to indicate if the system is low on battery (in seconds).

10.2. Reflection

There is a lot of knowledge and skills that our team is going to need to acquire to successfully complete our capstone project. From learning how to write formal documentation such as the SRS to pushing to our Github repository, all these tasks have taught our team a lot about the non-coding side of software. As we move along in the project, there will be a lot of ideas and skills that the team are looking forward to learning.

Being an effective and productive team is a multifaceted problem which must be tackled strategically. Perhaps more important than any specific skill, learning to work together and play to one another's strengths will propel Cyclops to success. One important tool in achieving this success is project planning. Every individual in this group will take accountability for project planning. This consists of organizing meetings, scheduling work to meet deadlines, and clearly articulating one's own intentions of completing work to other group members. Frequent communication between team members is crucial in ensuring everyone is on the same page and working towards a common goal. Another critical skill involved in productive teamwork is conflict management. Undoubtedly, at some point during the capstone project, there will be differences in opinion. However, it is how team members react to these disagreements which will be the deciding factor between sinking or swimming. It is expected that team members will support one another and try their best to view problems from the other's point of view, be open-minded and respectful, and most importantly clearly communicate how they feel and what they expect from others. As the team progresses into the latter half of the course, presentations will become more frequent. Successful presentation skills can drastically elevate the perceived success of a project. As such, it is critical that team members practice their presentation skills, present as a united front understanding what and when our teammates will present, and be ready to assist them.

One of the skills that our team is looking forward to is the mechanical side of the project. As mainly mechatronics and software students, we rarely use CAD modeling software; however, we believe that these designs will be key to our capstone. A couple of approaches that we will be taking are learning through practice by watching online videos and tutorials, using coursework from previous courses, as well as consulting our colleagues and peers for their own input. Furthermore, we will be looking at models that exist already to further cement our own designs. Aaron Li will be heading this section as he has a lot of expertise, modeling in both AutoCAD and SolidWorks from his past co-op experiences. Every other team member will have input to the designs, prototypes, and final product. Being able to work together through every step of the mechanical process is going to set our team up for success now and in the future.

Electrical engineering will be the bridge between the software and hardware domains. It is the means by which the hardware domain can collect and send data to be processed, and by which the software is able to interact with the physical world. Amos Cheung will be in charge of electrical engineering. On top of a theoretical understanding of electrical systems, he will need to become comfortable with applying his knowledge in the form of soldering, handling electrical components, noise reduction, and heat management. This may require the team to invest time and money to ensure that Amos Cheung is properly equipped with the skills to reliably read and apply electrical schematics.

This project will also require a significant amount of software development knowledge, as the software is ultimately the brains of the entire system. On top of learning the microcontroller's programming language at a syntactic level, we also need to ensure that the code being written is clean and maintainable. Although a bottleneck when initially beginning development, testing will save time in the long run by identifying bugs while they are small. Testing will also help in the separation of concerns between software and other domains - knowing that the software is functioning as intended will help pinpoint bugs in other areas. Amos Yu will be put in charge of getting software testing up and running for the team. Implementing a strong testing regime will be the best way to ensure that code remains maintainable as it grows and gets more complex.

In order to optimize software processes, leverage APIs, and ultimately complete this project within the time constraints of a single university year, our team will undoubtedly have to take advantage of the many open-source software libraries/packages. For each package that we use, we will need to spend a considerable amount of time reading documentation to understand how to use the package correctly and effectively.

The scope of this project also includes the Computer Vision domain. This includes understanding how to facilitate machine learning and AI’s for the vehicle recognition system. Brian Le will take initiative in this aspect of the system because he has already had exposure to the technologies relating that would be then used to build the required functionalities. Manny Lemos is very interested as well in this field and is looking forward to assisting as well in this area. Approaches for the team to acquire knowledge within this domain would include diving deep into open sources projects with similar goals and technologies. Being able to experiment and play around in these open source projects will give our team understanding on how proper AI’s are trained and deployed as well as what theory is required in order to make an AI successful. Another route our team could take to ramp up on this domain would be to find feature sets online and use them to train models from scratch using different libraries and methods. Getting a feel on each model's use case would be very beneficial to understanding which model would fit the needs of our system the most.