Group Submission 2: Subsystem Engineering

EEE4113F
Engineering Systems Design

University of Cape Town

Group 24
Taariq Daniels (DNLTAA02)
Inessa Rajah(RJHINE001)
Callum Tilbury(TLBCAL002)
Noel Loxton (LXTNOE001)
Moeketsi Mpooa(MPXMOE001)
Ronak Mehta (MHTRON001)

21 June 2020

Declaration:

1. We are the authors of this work, using our own words (except where attributed to others)
2. We know that plagiarism is to use another's work and pretend that it is one's own, and that this is wrong.
3. We have used *IEEE* convention for citation and referencing. We have provided citations and references in all cases where we have quoted from the work of others, or used other's ideas or reasoning in this essay/project/report.

The writer of each of the sections of this report is listed below:

Name (Student Number)	**Section(s) authored**
Callum Tilbury (TLBCAL002)	1.5. & 2.5. Application Back-end Subsystem
Ronak Mehta (MHTRON001)	1.4. & 2.4. Application Front-end Subsystem
Inessa Rajah (RJHINE001)	1.2 & 2.2 Alarm Subsystem
Noel Loxton (LXTNOE001)	1.3. & 2.3 Automatic Check in Subsystem
Moeketsi Mpooa (MPXMOE001)	1.6 & 2.6 Navigation and Communication Subsystem
Taariq Daniels (DNLTAA002)	1.1. & 2.1 Power supply Subsystem

Signatures of all authors:
Noel Loxton (NJL)
Inessa Rajah (IR)
Callum Tilbury (CRT)
Ronak Mehta (RM)
Taariq Daniels (TD)
Moeketsi Mpooa (MM)

Contents

Section 1

1.1 Power Supply Subsystem

User Requirements

UR1	**Reliability**
Requirement	The user requires that the hardware's source of power is reliable
Rationale	Given that all of the other subsystems will rely greatly on the power supply subsystem, it is imperative that, to ensure the system's successful operation, the power supply system is sufficiently robust and not easily prone to failure.
Refined by	UR1:FR1 UR1:FR2
Verification	ATP1, ATP2, ATP3, ATP4, ATP5, ATP6

UR2	**Simple maintenance**
Requirement	The user requires that the power supply can be easily replaced/maintained
Rationale	It is important that users do not need to expend unnecessary effort/time regarding maintenance of power supply. The system must be designed, such that individuals with no engineering background may maintain and potentially troubleshoot the system.
Refined by	UR2:FR1 UR2:FR2
Verification	ATP7, ATP8

Functional Requirements

UR1:FR1	**Power regulation**
Requirement	Voltage regulators are required to ensure correct voltages are supplied to the hardware (subsystems).
Refines	UR1
Refined by	UR1:FR1:DSP1 UR1:FR1:DSP2
Verification	ATP1, ATP2

UR1:FR2	**Protection Circuitry**
Requirement	Protection circuitry (in the form of diodes, resistors, etc.) and correct biasing are required to reduce chances of potential damage to hardware.
Refines	UR1
Refined by	UR1:FR2:DSP1 UR1:FR2:DSP2
Verification	ATP3, ATP4

UR1:FR3	**Redundancy**
Requirement	To further increase the reliability and robustness of the system, a secondary battery supply that is decoupled from the primary battery is required. In the event that the primary battery fails, the secondary battery will step in and continue system operations in its place.
Refines	UR1
Refined by	UR1:FR3:DSP1 UR2:FR1:DSP1 UR2:FR2:DSP2
Verification	ATP5

UR1:FR4	**Heat regulation**
Requirement	Heat regulation is essential in preventing performance degradation due to excessive heat build up. As such, heat sinks are required.
Refines	UR1
Refined by	UR1:FR4:DSP1
Verification	ATP6

UR2:FR1	**Removable power supply**
Requirement	System requires power supply to be modular (i.e. needs to be easily accessible and removable from the system as whole).
Refines	UR2
Refined by	UR2:FR1:DSP1
Verification	ATP7

UR2:FR2	**Rechargeable Battery**
Requirement	The battery must support recharging functionalities.
Refines	UR2
Refined by	UR2:FR2:DSP2
Verification	ATP8

Design Specifications

UR1:FR1:DS1	**Regulators**
Requirement	The main power source (battery) must be regulated to voltages of to 3V3 (GPS) (dash camera) (bio metric sensors) (alarm). Power will flow from the battery to multiple separate regulators, each acting as the required voltage source for the hardware. The regulators must be rated for 1.5A. (The battery will be connected directly to LCD screen and has no need for regulation).
Refines	UR1:FR1 (Power Regulation)
Verification	ATP1

UR1:FR1:DS2	**Decoupling capacitors**
Requirement	Decoupling capacitors are to reduce potential noise emanating from the supply. 2x 0.33uF and 2x 0.1uF Capacitors are required for the power supply and each regulator.
Refines	UR1:FR1 (Power regulation)
Verification	ATP2

UR1:FR2:DS1	**Safety diodes**
Requirement	Safety diodes are required to prevent the incorrect flow (reverse flow) of current. The diodes reverse breakdown voltage must be rated for at least 9V.
Refines	UR1:FR2 (Protection Circuitry)
Verification	ATP3

UR1:FR2:DS2	**Fuses**
Requirement	A fuse is required to protect the circuit from high currents in the event of a short circuit. The fuse needs to be rated for for a nominal operating current of 1.5A and a breaking capacity rating of 10000A.
Refines	UR1:FR2 (Protection Circuitry)
Verification	ATP4

UR1:FR3:DS1	**Redundancy module**
Requirement	A dual input, single output redundancy module that is rated for at least 1.5A is required.
Refines	UR1:FR3 (Redundancy)
Verification	ATP5

UR1:FR4:DS1	**Heat sink**
Requirement	A heat sink of size 35 x 50 x 30mm with a thermal resistance of 12.5°C is required, to ensure proper heat regulation.
Refines	UR1:FR4 (Heat regulation)
Verification	ATP6

UR2:FR1:DS1	**Battery enclosure**
Requirement	The enclosure which holds the battery must be easily accessible. An enclosure capable of holding 6 x AAA/AA batteries is required. The dimensions of the enclosure are required to be 53 X 55 X 26mm.
Refines	UR2:FR1 (Removable Power supply)
Verification	ATP7

UR2:FR2:DS1	**Batteries**
Requirement	Rechargeable battery pack (potentially consisting of multiple single battery cells) with nominal voltage of 7V is required.
Refines	UR2:FR2 (Rechargeable Battery)
Verification	ATP8

Acceptance Test Protocols

Code	**Refines**	**Description**
ATP1	UR1:FR1:DS1	Each regulator must be subjected to an uninterrupted input voltage range of 7-8V for duration of 20 minutes. A multi meter must be used to measure the output voltage. Pass Case: for the given input voltage range, the output of the corresponding regulator (as displayed on the multi meter) is $(3.3 \pm 0.1V)$.
ATP2	UR1:FR1:DS2	A simple regulator configuration needs to be setup with a capacitor placed at the input to regulator and GND, and a capacitor placed at the output of the regulator and GND. The regulator(3V3) must be subjected to an uninterrupted 7V DC input produced by a signal generator for a duration of 2 minutes. Pass Case: The variations in the output of the regulator are within 2% of 3V3.
ATP3	UR1:FR2:DS1	For a duration of 1 minute, subject the diode to a reverse biased voltage of 9V where the higher potential of the supply is connected to the diode's cathode and the GND potential is connected to the diode's anode. Using a multi meter, measure the current flow in the circuit. Pass Case: The current reading on the multi meter is less than 0.1mA.
ATP4	UR1:FR2:DS2	Configure a simple circuit where a fixed load is connected to a power supply. Connect the fuse in series with the load. Short the load for a moment (0.5s). Using a multi meter, check for continuity between either end of the fuse. Repeat test (with new components) 3 times. Pass Case: For each of the 3 tests, the continuity check is negative (fuse is blown).
ATP5	UR1:FR3:DS1	Configure a simple circuit where the redundancy module's inputs are each connected to separate power supplies of 7V. Connect the output of the module to a load that draws a current of 1.5A. Simulate a 'failure' by disconnecting the primary supply. Pass Case: The load remained powered despite the disconnect of the primary supply.
ATP6	UR1:FR4:DS1	Configure a circuit as described in ATP1 but with the heat sink mounted against the regulator. Use a multi meter (with temperature sensing capabilities) to measure the temperature across the regulator. Pass Case: The reading on the multi meter is less than 30°C.
ATP7	UR2:FR1:DS1	Drop the enclosure from a 1 meter height a total of 2 times. Pass case: Coils of enclosure are remain intact and can operate as normal.
ATP8	UR2:FR2:DS1	Configure a simple circuit where the battery pack acts as a supply for a fixed load that draws 1.5A. Using a timer, record the duration for which the load is powered. Pass case: Load is powered for a duration of 5 hours.

OPM Diagram

See figure 1 for an *Object Process Methodology* (OPM) diagram (i.e. a functional flow block diagram), representing the important components of the power supply subsystem. The power supply subsystem forms the cornerstone of the entire system, as all of the other subsystems rely on it for correct operation.

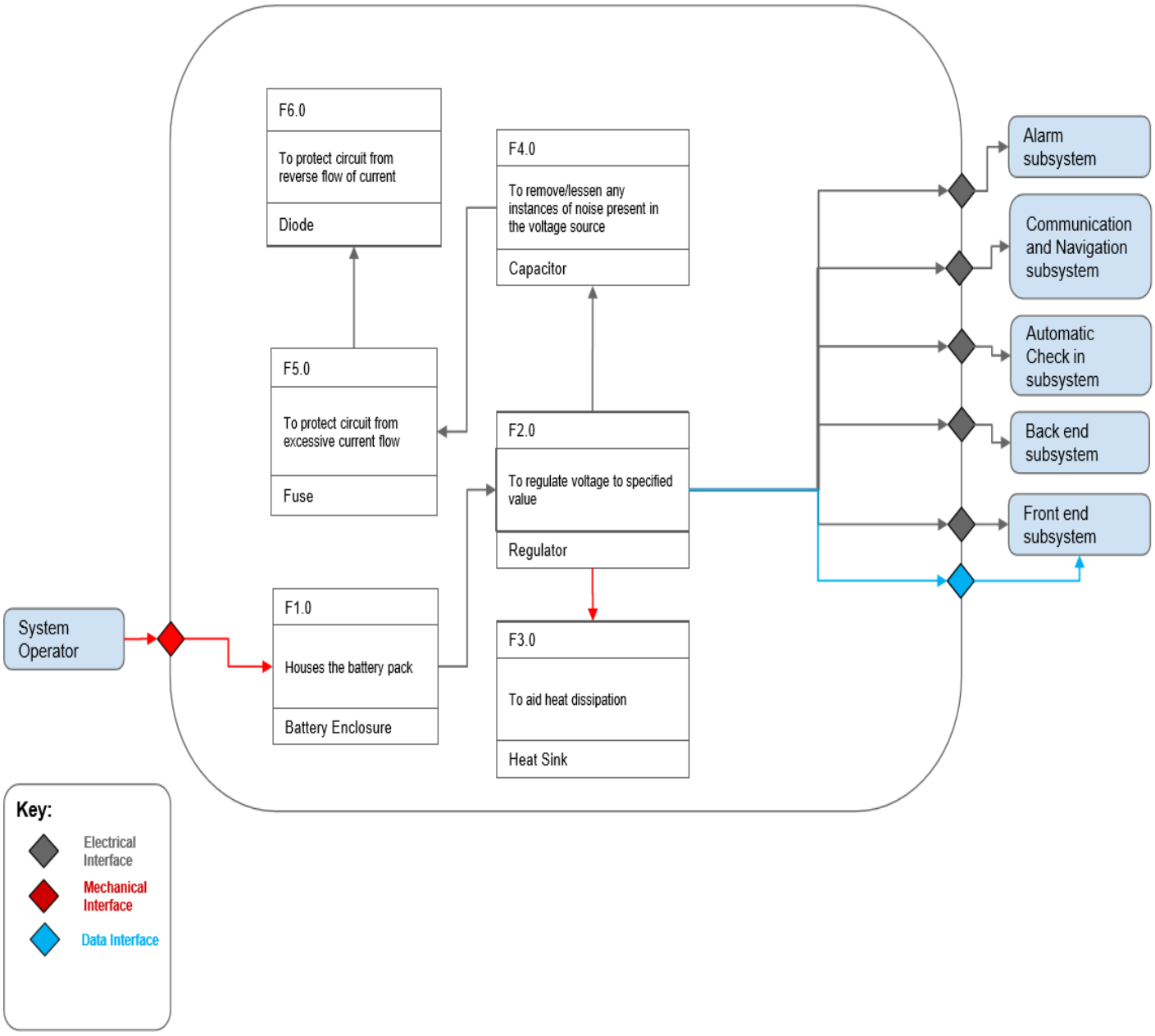


Figure 1: OPM diagram for power supply subsystem

1.2 Alarm Subsystem

User Requirements

UR1	**Comprehensive Alarm Triggering**
Requirement	The alarm must trigger in any possible emergency situation.
Rationale	A key aim of the overall system is to increase the safety of the driver. To this end, the alarm needs to be triggered in the event of any emergency situation. This includes situations where the driver unable to trigger the alarm manually.
Refined by	UR1:FR1, UR1:FR2
Verification	ATP1, ATP6, ATP7

UR2	**Alert Employer**
Requirement	The employer must be alerted when the alarm is triggered.
Rationale	In the event of an emergency, the driver will require assistance. By alerting the driver's employer in the event of emergency, the employer is able to assess the situation via the camera and microphone modules embedded on the bike dashboard and notify the relevant emergency services accordingly. This provides the driver with the best opportunity to receive the help they require in the most efficient way.
Refined by	UR2+UR4:FR3
Verification	ATP2, ATP3

UR3	**Alert Other Drivers**
Requirement	All the drivers using the same network must be alerted when a fellow driver's alarm is triggered.
Rationale	Alerting all drivers of all emergency situation promotes safety. It informs the drivers of adverse situations that happening around them which could lead to safer behaviour e.g. alerting a driver that another driver has been involved in a robbery in an area could inform that driver to stay out of the vicinity for safety reasons. On the other hand, the other drivers could also offer assistance to the driver whose alarm triggered if they are able to.
Refined by	UR3:FR4, UR3:FR5
Verification	ATP5, ATP8

UR4	Reliability
Requirement	The alarm triggering and alarm communication sub-subsystems need to operate reliably.
Rationale	In order for the alarm subsystem to perform its purpose of promoting driver safety, the alarm needs to trigger reliably and communicate with the other drivers and, most importantly, the employer reliably in order to facilitate action to assist a driver in jeopardy. The alarm needs to reliably trigger only in the event of emergency, stop triggering if that emergency resolves itself and the status of the alarm for each driver needs to be reliably communicated to the employer and the other drivers.
Refined by	UR4:F6, UR4:F7
Verification	ATP2, ATP4, ATP6, ATP7, ATP8, ATP9

Functional Requirements

UR1:FR1	Manual Alarm Triggering
Requirement	The driver needs a manual needs a means of manually triggering the alarm in the event of an emergency. This will be achieved by the interface between a panic button and an interrupt handled by the processor.
Refines	UR1
Refined by	UR1:FR1:DS1, UR1:FR1:DS2
Verification	ATP1, ATP6

UR1:FR2	Automatic Alarm Triggering
Requirement	The alarm needs to be automatically triggered if the driver's does not meet the "check-in" criteria for a certain amount of time (indicating that driver is incapacitated in some way). This will be controlled by an interrupt handled by the system processor.
Refines	UR1
Refined by	UR1:FR2:DS3
Verification	ATP9

UR2+UR4:F3	Communication Channel to Employer
Requirement	A communication channel with reliable, constant availability needs to exist between the alarm subsystem and the employer front-end subsystem.
Refines	UR2, UR4
Refined by	UR2+UR4:FR3:DS4
Verification	ATP2, ATP3

UR3:FR4	Communication Channel to Other Drivers.
Requirement	A communication channel needs to be available between all drivers on the network.
Refines	UR3
Refined by	UR3:FR4:DS5
Verification	ATP5

UR3:FR5	LCD Position Update
Requirement	The alarm subsystem needs to interface with the navigation and communications subsystem to indicate on all drivers LCD screen modules the location of the driver whose alarm is triggered. This will be controlled by an interrupt handled by the system processor.
Refines	UR3
Refined by	UR3:FR5:DS6
Verification	ATP8

UR4:FR6	Fast Response Time
Requirement	The both alarm sub-subsystems (alarm triggering and communication subsystems) need to have fast response times to changes in alarm status.
Refines	UR4
Refined by	UR4:FR6:DS8, UR4:FR6:DS9
Verification	ATP2, ATP4, ATP6, ATP7, ATP8, ATP9

UR4:FR7	Disarm Functionality
Requirement	Once the alarm is triggered, it needs to be disarmed based on driver or employer commands. This will be controlled by interrupts handled by the system processor.
Refines	UR4
Refined by	UR4:FR7:DS7
Verification	ATP9

Design Specifications

UR1:F1:DS1	**Detachable Panic Button**
Requirement	A panic button is required on the bike's handle bars. The button should be integrated into an easily detachable module. This button should be a push button with a robust protective layer and a maximum bounce time of 5ms. Pressing the panic button will trigger the ManualAlarm ISR.
Refines	UR1:F1
Verification	ATP1, ATP6

UR1:F1:DS2	**ISR (Name: ManualAlarm)**
Requirement	The ISR named ManualAlarm will be called when an interrupt is triggered by the panic button being pressed. The ISR will set $alarm = True$ and activate the alarm communication channel to the employer front-end to inform the employer of the change in alarm status. While $alarm = True$ the employer will have access to the camera module and microphone embedded in the dashboard in order to assess the emergency situation.
Refines	UR1:FR1
Verification	ATP7

UR2:FR2:DS3	**ISR (Name: AutomaticAlarm)**
Requirement	Automatic alarm triggering will be handled by a software controlled interrupt. The interrupt is triggered by the driver's $check - instatus = False$ after the prescribed 20 minutes (interfacing with the Check-in Subsystem). The interrupt will call the ISR named AutomaticAlarm. AutomaticAlarm will set the $alarm = True$ and activate the alarm communication channel to the employer front-end to inform the employer of the change in alarm status. While $alarm = True$ the employer will have access to the camera module and microphone embedded in the dashboard in order to assess the emergency situation.
Refines	UR1:F2 (Automatic Alarm Triggering)
Verification	ATP9

UR2+UR4: FR3:DS4	**Dual Alarm Communication Channel**
Requirement	A designated dual IP and cellular communication channel with 128-bit AEC encryption is required to facilitate communications between the alarm subsystem and the employer front-end subsystem. This ensures constant, reliable communications regarding driver safety and ensures the security of that information. The IP communication channel is provided by the use of an RPi2 Single Board Computer.
Refines	UR2+UR4:F3
Verification	ATP2, ATP3

UR3:FR4:DS5	**IP Communication Channel to Other Drivers**
Requirement	An IP communication channel with 128-bit AEC encryption is required to facilitate communications between the drivers on the network. This ensures constant communications between drivers regarding their safety and position. This IP communication channel is provided by the use of an RPi2 Single Board Computer.
Refines	UR3:FR4 (Communication Channel to Other Drivers)
Verification	ATP5

UR3:FR5:DS6	**ISR (Name: NotifyDrivers)**
Requirement	Drivers will notified that a fellow driver is in trouble via an interrupt. The interrupt will be triggered when $alarm = True$ status is set. The interrupt will call the ISR names NotifyDrivers. The ISR will interface with the Communication and Navigation subsystem to indicate on all drivers screen the position of the driver who triggered the alarm in red.
Refines	UR3:F5
Verification	ATP8

UR4:FR7:DS7	**ISR (Name: Disarm)**
Requirement	The disarming of the alarm is controlled by an interrupt. This interrupt is triggered when the employer declares the emergency resolved or the driver check's. In these events, the interrupt will call the ISR named Disarm which will set $alarm = False$, notify the other drivers that the issue has been resolved and end the employer's ability to access the camera and microphone embedded on the dashboard.
Refines	UR4:F7 (Disarm Functionality) UR4:FR6 (Fast Response Time)
Verification	ATP9

UR4:FR6:DS8	**Fast ISR Response Time**
Requirement	All ISR's must be called and executed with a latency time of $< 2s$.
Refines	UR4:FR6 (Fast Response Time)
Verification	ATP6, ATP7, ATP8, ATP9

UR4:FR7:DS7	**Fast Communication Time**
Requirement	Alarm updates must be transmitted to and from employer in a time of $< 2s$.
Refines	UR4:F7 (Disarm Functionality) UR4:FR6 (Fast Response Time)
Verification	ATP2, ATP4

Acceptance Test Protocols

Code	Refines	Pre-requisites	Description
ATP1	UR1:FR1:DS1	N/A	The panic button must be pressed 50 times and the voltage change across it must be measured. Success metric: Voltage when unpressed should be GND and voltage when pressed should be VDD. % Tests that should be successful $\geq 99.5\%$
ATP2	UR2:FR2:DS4	N/A	The cellular communication line between the alarm subsystem on the bike and the employer front-end needs to be tested. An artificial 24- bit bitstream must be created and transmitted via the cellular communication channel to the employer front-end. The bitstream received by the employer front-end server needs to be compared to the originally transmitted bitstream. This test must be repeated every 2-5s for 24 hours. Success metric: Transmitted and received bitstream must agree $\geq 90.5\%$. % Tests that should be successful $\geq 99.5\%$
ATP3	UR2:FR2:DS4	N/A	The IP communication channel between the alarm subsystem on the bike and the employer front-end needs to be tested. The test description and success metrics are the same as ATP2, applied to the transmission of the bitstream across the IP communication channel.
ATP4	UR4:FR6:DS9	ATP2 Passed	ATP2 repeated with the additional success metric that transmission of bitstream needs to be successfully completed in $< 2s$.
ATP 5	UR3:FR4:DS5	N/A	The IP communication line between the alarm subsystem on one bike and the navigation and communication subsytems of all the other bikes needs to be tested. An artificial 24- bit bitstream must be created and transmitted via the IP communication channel to a test set of 100 bikes on the same network in a 50km radius. The bitstream received by the test set of bikes fneeds to be compared to the originally transmitted bitstream. This test must be repeated every 2-5s, 24 hours, alternating the test set of bikes used every 2 hours. Success metric: Transmitted and received bitstream must agree $\geq 90.5\%$. % Tests that should be successful $\geq 90.5\%$
ATP6	UR1:FR1:DS1 UR4:FR6:DS9	N/A	Aritifical trigger test needs to be conducted to test that the ManualAlarm interrupt and ISR functionality. This test should be repeated in 5 minute intervals for 12 hours. Success metric: ISR should perform the following; update $alarm = True$, set flag to allow employer employer front-end system access to camera and microphone on bike dashboard, execute in $< 2s$. Success rate should be $\geq 99.5\%$

Code	Refines	Pre-requisites	Description
ATP7	UR2:FR2:DS2 UR4:FR6:DS9	N/A	Aritifical trigger test needs to be conducted to test that the AutomaticManual interrupt and ISR functionality. This test should be repeated in 5 minute intervals for 12 hours. Success metric: ISR should perform the following; update $alarm = True$, set flag to employer front-end system access to camera and microphone on bike dashboard, should execute in $< 2s$. Success rate should be $\geq 99.5\%$
ATP8	UR3:FR5:DS6 UR4:FR6:DS9	Pass ATP5, ATP6, ATP7	Artifical trigger test should be conducted to trigger execution of either AutomaticAlarm or ManualAlarm ISR. The $alarm = true$ should, in turn, should trigger the NotifyDriversISR. The test set of bikes should be in 50km radius of the testing facility. The test should be repeated every 30min for 5 hours. The trigger ISR should be split 50/50 between ManualISR and AutomaticISR. Success metric: NotifyDriversISR should perform the following; interface with Navigations and Communications subsystem to indicate GPS location of testing facility as a red dot on test set of 100 other functional bike systems' LCD screen modules, should execute in $< 2s$. Success rate should be $\geq 99.5\%$
ATP9	UR4:FR7:DS7 UR1:FR2:DS3	Pass ATP 2, ATP5, ATP6	The check-in status flag should be changed from False to True. This, in turn, should trigger the Disarm ISR. Success metric: Disarm ISR should perform the following; set $alarm = False$, should set flag to prohibit employer access to camera and microphone on bike dashboard, should execute in $< 2s$.Success rate should be $\geq 99.5\%$

OPM Diagram

See figure 3 for an *Object Process Methodology* (OPM) diagram (i.e. a functional flow block diagram), representing the important components of the alarm subsystem as well as how the alarm subsystem interacts with relevant other subsystems mentioned in this report (Namely, the *Check-In, Navigation and Communication and Employer Front-end Subsystems.*

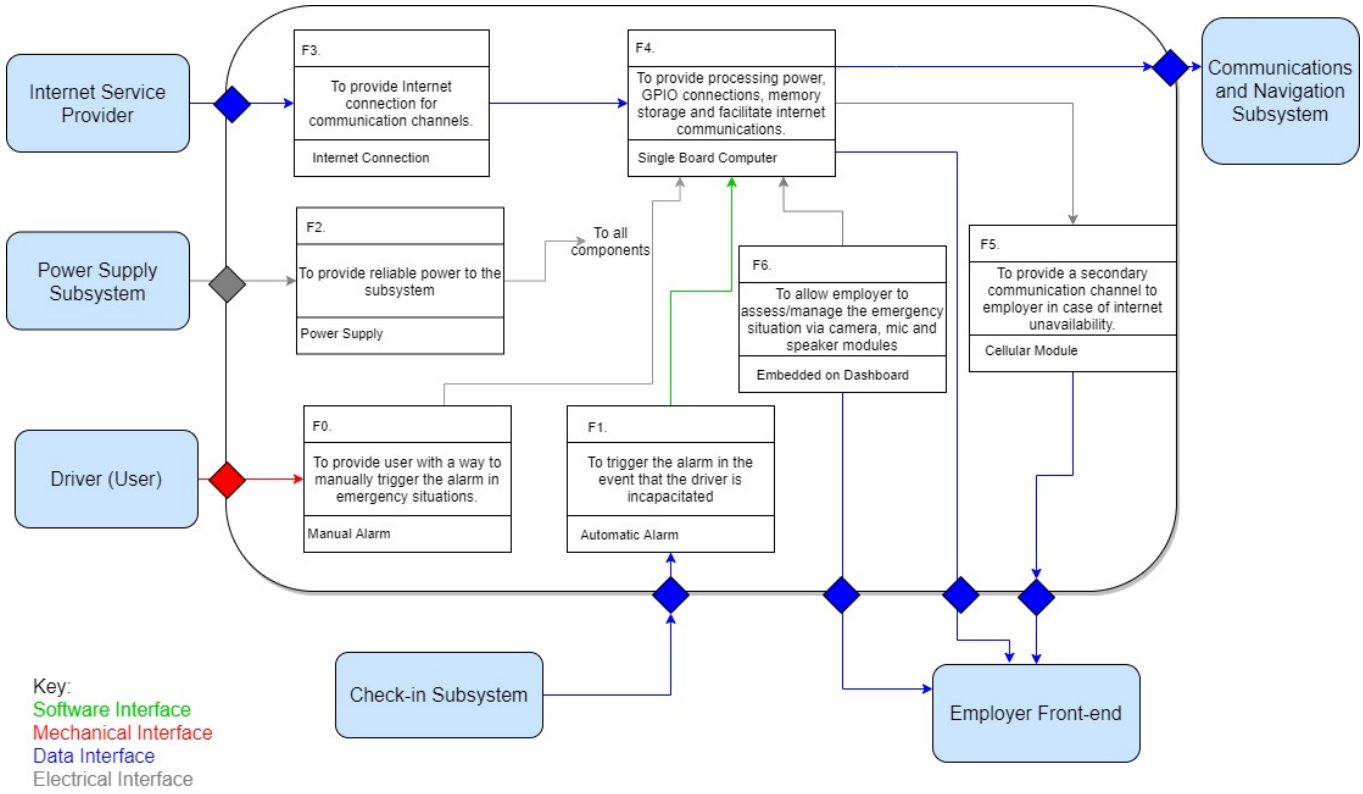


Figure 2: OPM diagram for Alarm Subsystem

1.3 Automatic Check-in Subsystem

User Requirements

UR1	**Check-in**
Requirement	The user requires the ability to "check-in" with the system on their bike.
Rationale	The driver will face certain situations/dangers which could render them unconscious or injured. The check-in function will allow the system a way to automatically register when they are driving or 'safe' on their bike. If the are in an accident or knocked unconscious they will not be checked-in and other subsystems will be alerted.
Refined by	UR1:FR1, UR1:FR2
Verification	ATP10, ATP11, ATP12

UR2	**Reliability**
Requirement	The user requires that the check-in function is reliable and works consistently.
Rationale	The check-in system needs to work reliably in order to ensure driver safety. The system must be able to allow quick and efficient checking-in without hardware/software errors.
Refined by	UR2:FR1, UR2:FR2
Verification	ATP1, ATP2, ATP3, ATP4, ATP5, ATP13, ATP14, ATP15, ATP16, ATP17, ATP18

UR3	**Conspicuousness**
Requirement	The user requires the check-in function to be inconspicuous and easily accessible.
Rationale	The ability to check in should not interfere with the drivers ability to drive i.e. it should not be obtrusive on the handle bars or bike. It should also be comfortable and accessible for the driver since it should be easy to use and once again not change the handles of the bike drastically. It is also important to be inconspicuous since one would want an unwanted person tampering with it.
Refined by	UR3:FR1, UR3:FR2
Verification	ATP6, ATP7, ATP8, ATP9

UR4	**Integration**
Requirement	The user requires this system to communicate with other subsystems and be easily integrated onto the bike system as a whole.
Rationale	The check-in function in isolation is useless without the ability to effectively merge with the other bike subsystems such as the alarm and the network. What good is a failed check-in if it does not alert anyone?
Refined by	UR4:FR1
Verification	ATP1, ATP2, ATP3, ATP4, ATP5

Functional Requirements

UR1:FR1	**Bio-metric Sensing**
Requirement	A form of bio-metric sensing/scanning can allow the driver to check-in to the bike system and ultimately the server.
Refines	UR1
Refined by	UR1:FR1:DS1, UR1:FR1:DS2
Verification	ATP11, ATP10

UR1:FR2	**Timed check-in**
Requirement	The check-in function needs to be timed. Not too short as to hassle the driver and not too long as it needs to actually assist with safety for the driver.
Refines	UR1
Refined by	UR1:FR2:DS1
Verification	ATP12

UR2:FR1	**Speed of check-in**
Requirement	It is required that the ability to check in is both consistent and fast enough to allow for effective checking-in.
Refines	UR2
Refined by	UR2:FR1:DS1
Verification	ATP1, ATP2, ATP3, ATP4, ATP5

UR2:FR2	**Reliable hardware**
Requirement	It is required that the hardware (and software) used to check in the driver is reliable and efficient to meet the user requirement.
Refines	UR2
Refined by	UR1:FR1:DS1, UR2:FR1:DS1
, **Verification**	ATP11, ATP1, ATP2, ATP3, ATP4, ATP5

UR3:FR1	**Built into Bike Handles**
Requirement	In order to be inconspicuous the sensing for check-in should be built into the bike handle.
Refines	UR3
Refined by	UR3:FR1:DS1
Verification	ATP6, ATP7, ATP8, ATP9

UR3:FR2	**Thumb fingerprint Scanning**
Requirement	The sensors should be comfortably fitted to where the drivers thumbs will sit under the handle to be both out of sight and easily accessible to the driver.
Refines	UR3
Refined by	UR1:FR1:DS1, UR3:FR1:DS1
Verification	ATP11, ATP6, ATP7, ATP8, ATP9

UR4:FR1	**Communication with other subsystems**
Requirement	The check-in function needs to be integrated onto the system as a whole so that it communicates effectively with the alarm, network, power as well as back-end and front end servers.
Refines	UR4
Refined by	UR2:FR1:DS1
Verification	ATP1, ATP2, ATP3, ATP4, ATP5

Design Specifications

UR1:FR1:DS1	**RPi Fingerprint sensor**
Requirement	The system should have the Raspberry Pi fingerprint sensor installed on both handles at the bottom of the handle where the driver thumb rests.
Refines	UR1:FR1, UR2:FR2, UR3:FR2
Verification	ATP11

UR1:FR1:DS2	**Serial USB converter**
Requirement	The system should have a serial USB converter with both a 3.3V and 5V connection to link the fingerprint scanner to the RPi and allow for future adaptations.
Refines	UR1:FR1
Verification	ATP10

UR1:FR2:DS1	**Time of check-in process**
Requirement	The system should register a fingerprint scan successful or unsuccesful in no more than 0.5s after the thumb is applied.
Refines	UR1:FR2
Verification	ATP12

UR1:FR2:DS1	**Time between check-ins**
Requirement	The system should require the user to scan their finger at least once every 20minutes
Refines	UR1:FR2
Verification	ATP19

UR2:FR1:DS1	**RPi processor**
Requirement	The system should use a Rapsberry Pi 2b for processing of fingerprint results and communication with other subsystems.
Refines	UR1:FR1
Verification	ATP1, ATP2, ATP3, ATP4, ATP5

UR2:FR2:DS1	**Success of Scan**
Requirement	The fingerprint scanner should be successful 99.5% of the time when attempting a fingerprint scan which is registered on the system.
Refines	UR2:FR2
Verification	ATP13, ATP14, ATP15

UR2:FR2:DS2	**Rejection of Scan**
Requirement	The fingerprint scanner should reject an unregistered fingerprint 100% of the time.
Refines	UR2:FR2
Verification	ATP16, ATP17, ATP18

UR3:FR1:DS1	**Casing for sensor/handle**
Requirement	The fingerprint scanner should have a hard casing surrounding it when connected onto the bike handle which blends into the bike handle and is comfortable for the driver. 3D printing is acceptable.
Refines	UR3:FR1
Verification	ATP6, ATP7, ATP8, ATP9

Acceptance Test Protocols

Code	Refines	Description
ATP1	UR2:FR1:DS1	Raspberry Pi (RPi) is present on system (bike system as a whole)
ATP2	UR2:FR1:DS1	RPi is connected to power (see section above [Power subsystem])
ATP3	UR2:FR1:DS1	Fingerprint module is connected to the RPi
ATP4	UR2:FR1:DS1	Alarm subsystem is connected to the RPi (see section above [Alarm subsystem])
ATP5	UR2:FR1:DS1	The Raspberry Pi is connected to the internet/network server (see section below [Communications and Navigation])
ATP6	UR3:FR1:DS1	Raspberry Pi fingerprint scanner is placed on bike handle where thumb rests when driving
ATP7	UR3:FR1:DS1	The sensor is enclosed on the handle and is not easily removed or tampered with i.e. wiring is not exposed
ATP8	UR3:FR1:DS1	Sensor is out of view to the naked eye i.e. well hidden to someone unaware of its presence
ATP9	UR3:FR1:DS1	Sensor does not cause discomfort when using it for at least a 10 minute period
ATP10	UR1:FR1:DS1	The RPi has a serial usb connection which can take either a 3.3V input or 5V input (this allows upgrades/updates to fingerprint module if neccessary in future)
ATP11	UR1:FR1:DS1	Raspberry Pi acknowledges presence of fingerprint when finger on scanner
ATP12	UR1:FR2:DS1	Raspberry acknowledges fingerprint presence in less than 0.5seconds after fingerprint applied
ATP13	UR2:FR2:DS1	The Raspberry Pi can store a specific fingerprint in memory to be compared later and this fingerprint can be accessed after powering off
ATP14	UR2:FR2:DS1	The RPi can correctly compare that same fingerprint when it is present on the scanner and identify it as the one stored in memory
ATP15	UR2:FR2:DS1	The RPi can correctly identify the same fingerprint as per ATP14 100 times in 100 attempts
ATP16	UR2:FR2:DS2	The RPi can register a fingerprint on the scanner even though it is not in the system
ATP17	UR2:FR2:DS2	The RPi can compare the fingerprint on the scanner to the one in memory and reject it as not being a match correctly when the fingerprint is not the one that was stored in memory
ATP18	UR2:FR2:DS2	The RPi can reject the incorrect fingerprint 100 times out of 100 tries.
ATP19	UR2:FR1:DS2	The RPi triggers an output high when there has been 20minutes since the last fingerprint scan (this output is linked to the alarm subsystem [see subsystem above [alarm])

OPM Diagram

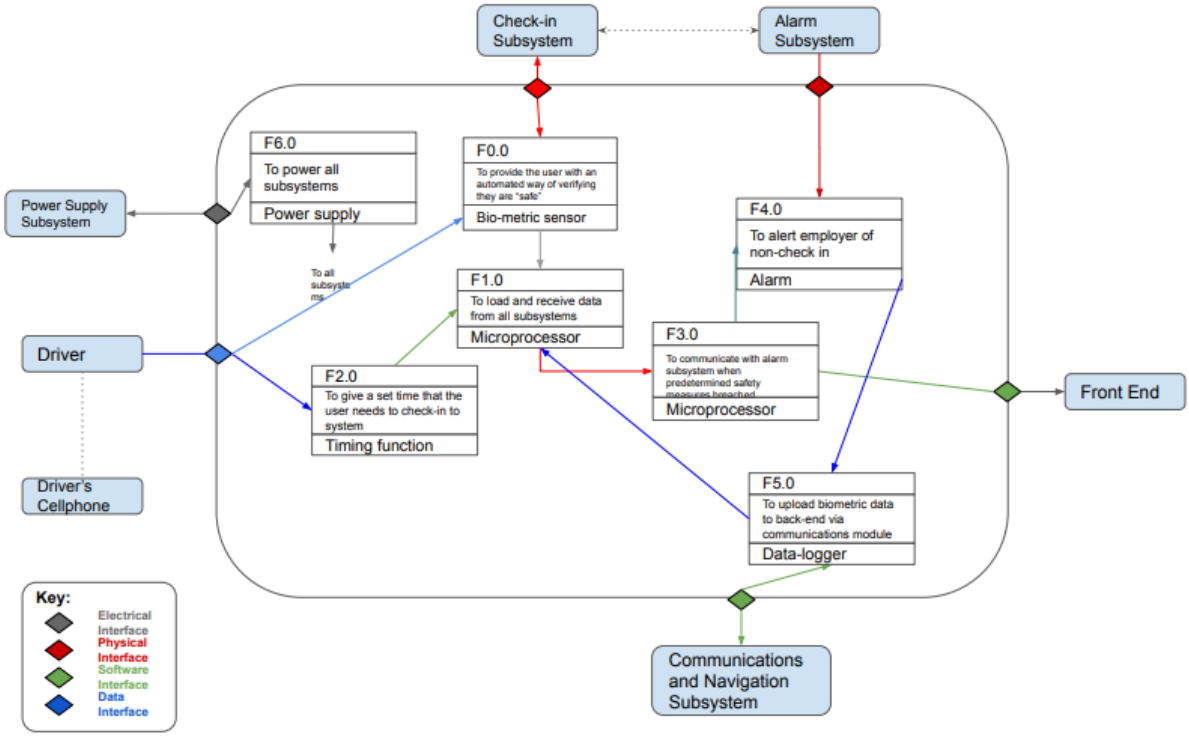


Figure 3: OPM diagram for Check-in Subsystem

1.4 Application Frontend Subsystem

User Requirements

UR1	**Ease of Use**
Requirement	The system needs to be simple and user friendly.
Rationale	It is important that the user feels connected with the system, and this can be achieved by making the user interface a simple, elegant and easy to understand system, that can be operated by anyone with little or no knowledge of the system prior to using it.
Refined by	UR1:FR1 (GUI) UR3:FR2 (Debugging)
Verification	ATP1; ATP2; ATP9

UR2	**Integration**
Requirement	The front-end application needs to collaborate with other subsystems.
Rationale	This subsystem needs to communicate and integrate with all the other subsystems either directly or indirectly to ensure a smooth process flow of the system as a whole.
Refined by	UR2:FR1 (Integration with Alarm) UR2:FR2 (Integration with Back-end Applications) UR2:FR3 (Integration with Automatic Check-in) UR2:FR4 (Integration with Communication and Navigation) UR2:FR5 (Integrated Dashboard) UR3:FR2 (Debugging)
Verification	ATP3; ATP4; ATP5; ATP6; ATP7; ATP9

UR3	**Reliable**
Requirement	The subsystem needs to be reliable ensuring little or no bugs.
Rationale	The front-end application along with all the subsystems needs to be reliable to ensure user safety as this is the utmost priority of the system in general. The front-end must reliably deliver and send information to and from the user using its UI.
Refined by	UR3:FR1 (Quick Updates) UR3:FR2 (Debugging)
Verification	ATP8; ATP9

UR4	**Compatibility**
Requirement	The front-end application must be compatible with other devices.
Rationale	Since our users are motorbike drivers, who are constantly on their toes moving from one location to the other, it is extremely important to have the front-end application compatible with other devices to ensure constant communication and safety is maintained throughout their journey day in and day out.
Refined by	UR3:FR1 (Quick Updates) UR3:FR2 (Debugging) UR4:FR1 (Compatibility with mobile phones)
Verification	ATP8; ATP9; ATP10

Functional Requirements

UR1:FR1	**GUI**
Requirement	The front-end application needs to have an enhanced Graphical User interface (GUI) with clear instructions and guidelines helping to create a simple yet elegant user-friendly environment.
Refines	UR1 (Ease of Use)
Refined by	UR1:FR1:DS1 (Enhanced GUI)
Verification	ATP1; ATP2

UR2:FR1	**Integration with Alarm**
Requirement	A communication channel needs to exist between the alarm subsystem and the employer front-end subsystem.
Refines	UR2 (Integration)
Refined by	UR2:FR1:DS1 (Alarm Connection) UR4:FR1:DS1 (Mobile phone OS)
Verification	ATP3; ATP10

UR2:FR2	**Integration with Back-end Applications**
Requirement	A communication channel needs to exist between the employer back-end subsystem and the employer front-end subsystem.
Refines	UR2 (Integration)
Refined by	UR2:FR2:DS1 (Back-end Connection)
Verification	ATP4

UR2:FR3	**Integration with Automatic Check-in**
Requirement	A communication channel needs to exist between the Automatic Check-in subsystem and the employer front-end subsystem.
Refines	UR2 (Integration)
Refined by	UR2:FR3:DS1 (Automatic Check-in Connection) UR3:FR1:DS1 (Refresh Rates) UR4:FR1:DS1 (Mobile phone OS)
Verification	ATP5; ATP8; ATP10

UR2:FR4	**Integration with Communication and Navigation**
Requirement	A communication channel needs to exist between the Communication and Navigation subsystem and the employer front-end subsystem.
Refines	UR2 (Integration)
Refined by	UR2:FR4:DS1 (Communication and Navigation Connection) UR3:FR1:DS1 (Refresh Rates) UR4:FR1:DS1 (Mobile phone OS)
Verification	ATP6; ATP8; ATP10

UR2:FR5	**Integrated Dashboard**
Requirement	The integrated dashboard comprises of a camera, speaker, microphone and a power bar. This allows the user to interact with the employee or another driver and thus ensures safety, reliability and punctuality.
Refines	UR2 (Integration)
Refined by	UR2:FR5:DS1 (Embedded Dash) UR3:FR1:DS1 (Refresh Rates) UR4:FR1:DS1 (Mobile phone OS)
Verification	ATP7; ATP8; ATP10

UR3:FR1	**Quick Updates**
Requirement	The front-end application must rapidly refresh its contents a number of times per second to provide the most up-to-date information to the user.
Refines	UR3 (Reliable) UR4 (Compatibility)
Refined by	UR3:FR1:DS1 (Refresh Rates) UR4:FR1:DS1 (Mobile phone OS)
Verification	ATP8; ATP10

UR3:FR2	**Debugging**
Requirement	The subsystem must frequently look out for bugs and fix them to ensure the operations performed are reliable. It is extremely important to regularly solve issues regarding this system as a whole because user safety is our number one priority, and by running debugging processes for every subsystem involved, greatly improves the reliability and efficiency of the entire system.
Refines	UR1 (Ease of Use) UR2 (Integration) UR3 (Reliable) UR4 (Compatibility)
Refined by	UR3:FR2:DS1 (Intensive Debugging) UR4:FR1:DS1 (Mobile phone OS)
Verification	ATP9; ATP10

UR4:FR1	**Compatibility with mobile phones**
Requirement	To ensure constant communication and safety of the motorbike drivers, the front-end application needs to compatible with the drivers' mobile phones.
Refines	UR4 (Compatibility)
Refined by	UR3:FR1:DS1 (Refresh Rates) UR3:FR2:DS1 (Intensive Debugging) UR4:FR1:DS1 (Mobile phone OS)
Verification	ATP8; ATP9; ATP10

Design Specifications

UR1:FR1:DS1	Enhanced GUI
Requirement	Daily enhancement/updating of the Graphical User Interface for improved quality in user experience. GUI elements consists of size, position, width, alignment, fonts etc which needs to be updated on a daily basis in order to be up-to-date with the changing world.
Refines	UR1:FR1 (GUI)
Verification	ATP1; ATP2

UR2:FR1:DS1	Alarm connection
Requirement	An IP communication channel with 128-bit AEC encryption using a Raspberry Pi 2b microprocessor is required to facilitate reliable and constant communications between the alarm subsystem and the employer front-end application subsystem.
Refines	UR2:FR1 (Integration with Alarm)
Verification	ATP3; ATP9

UR2:FR2:DS1	Back-end connection
Requirement	A secure connection with the back-end servers is ensured by connecting the back-end worker pods using the Domain Name System (DNS) name given to the back-end service. The DNS name used at the front-end should be 100% accurate with the one given for the back-end.
Refines	UR2:FR2 (Integration with Back-end Applications)
Verification	ATP4

UR2:FR3:DS1	Automatic Check-in connection
Requirement	An IP communication channel with 128-bit AEC encryption using a Raspberry Pi 2b microprocessor is required to facilitate reliable and constant communications with the Automatic Check-in subsystem.
Refines	UR2:FR3 (Integration with Automatic Check-in)
Verification	ATP5; ATP8; ATP10

UR2:FR4:DS1	Communication and Navigation Connection
Requirement	A wireless connection using Wi-Fi on the RPi2b microprocessor must be set to ensure a reliable connection with the Communication and Navigation subsystem.
Refines	UR2:FR4 (Integration with Communication and Navigation)
Verification	ATP6; ATP8; ATP10

UR2:FR5:DS1	Embedded dash
Requirement	The dash camera must be at least of 12 megapixel or greater. The Speaker and microphone should be able to produce and receive sounds in the range of 80-90 dB.
Refines	UR2:FR5 (Integrated Dashboard)
Verification	ATP7; ATP8; ATP10

UR3:FR1:DS1	Refresh Rates
Requirement	The refresh rate must be locked at 60 fps to ensure reliable quick updates are made possible.
Refines	UR3:FR1 (Quick Updates)
Verification	ATP8; ATP10

UR3:FR2:DS1	**Intensive Debugging**
Requirement	Frequent testing and debugging every 3 days to identify and fix all logged bugs from the user.
Refines	UR3:FR2 (Debugging)
Verification	ATP9; ATP10

UR4:FR1:DS1	**Mobile phone OS**
Requirement	The front-end must be 100% compatible with all IOS and Android devices.
Refines	UR4:FR1 (Compatibility with mobile phones)
Verification	ATP8; ATP9; ATP10

Acceptance Test Protocols

Code	Refines	Description
ATP1	UR1:FR1:DS1	The system must undergo a GUI model based testing procedure using the Sikuli product licensed under MIT for at least 5 desirable GUI states with a positive confirmation percent of greater than 99%
ATP2	UR1:FR1:DS1	The screen testing must be carried out using different resolutions with the help of zooming in and zooming out at 640X480 , 600X800 etc. and ensure that the screen bias with every resolution tested is within 5% range.
ATP3	UR2:FR1:DS1	An artificial 24 bit bitstream must be created and transmitted from the alarm subsystem via the established communication to the front-end application. This bitstream received at the front-end server must be 90% or more similar with the original bitstream. The whole test must be further repeated every 2.5s for 24 hours.
ATP4	UR2:FR2:DS1	An artificial 24 bit bitstream must be created and transmitted from the back-end subsystem via the established communication to the front-end application. This bitstream received at the front-end server must be 90% or more similar with the original bitstream. The whole test must be further repeated every 2.5s for 24 hours..
ATP5	UR2:FR3:DS1	Conduct fingerprint tests as mentioned in the above section (Automatic Check-in subsystem) and verify those results with the results obtained in the employer front-end. The tests must be repeated for at least 100 samples and the success rate should be greater than 99.5%.
ATP6	UR2:FR4:DS1	Connect to the Wi-Fi and ensure connectivity is maintained for a continuous 15 hours duration. Allow disconnects of not more 5 minutes per session. Conduct at least 5 such sessions to ensure a stable connection. Also, keep checking that the connection stays above 0.5mbps for all sessions.
ATP7	UR2:FR5:DS1	Check for the megapixel rating of the dash camera from the camera manual. Use a decibel meter with a resolution of less than 6 dB to record the sound range. Conduct the sound range test at least 10 times with a success rate of greater than 95%.
ATP8	UR3:FR1:DS1	Conduct a UFO Motion test by monitoring the refresh rate of the system. The refresh rate value should be between 59.5 - 60.5 fps for every reading obtained. Repeat this test every hour for 24 hours.
ATP9	UR3:FR2:DS1	Daily, a bug-review process must be undertaken, and the time-based response to various bug severities must be calculated, and ensured to be in accordance with the specifications.
ATP10	UR4:FR1:DS1	Use the system on both IOS and Android and ensure that it runs on both these platforms 100% of the time. The system must be compatible with IOS version 12.0 or greater and with Android Oreo version 8.0 or greater.

OPM Diagram

See figure 4 for an *Object Process Methodology* (OPM) diagram (i.e. a functional flow block diagram), representing the important components of the front-end subsystem. It also represents all the subsystems related to the employer front-end application.

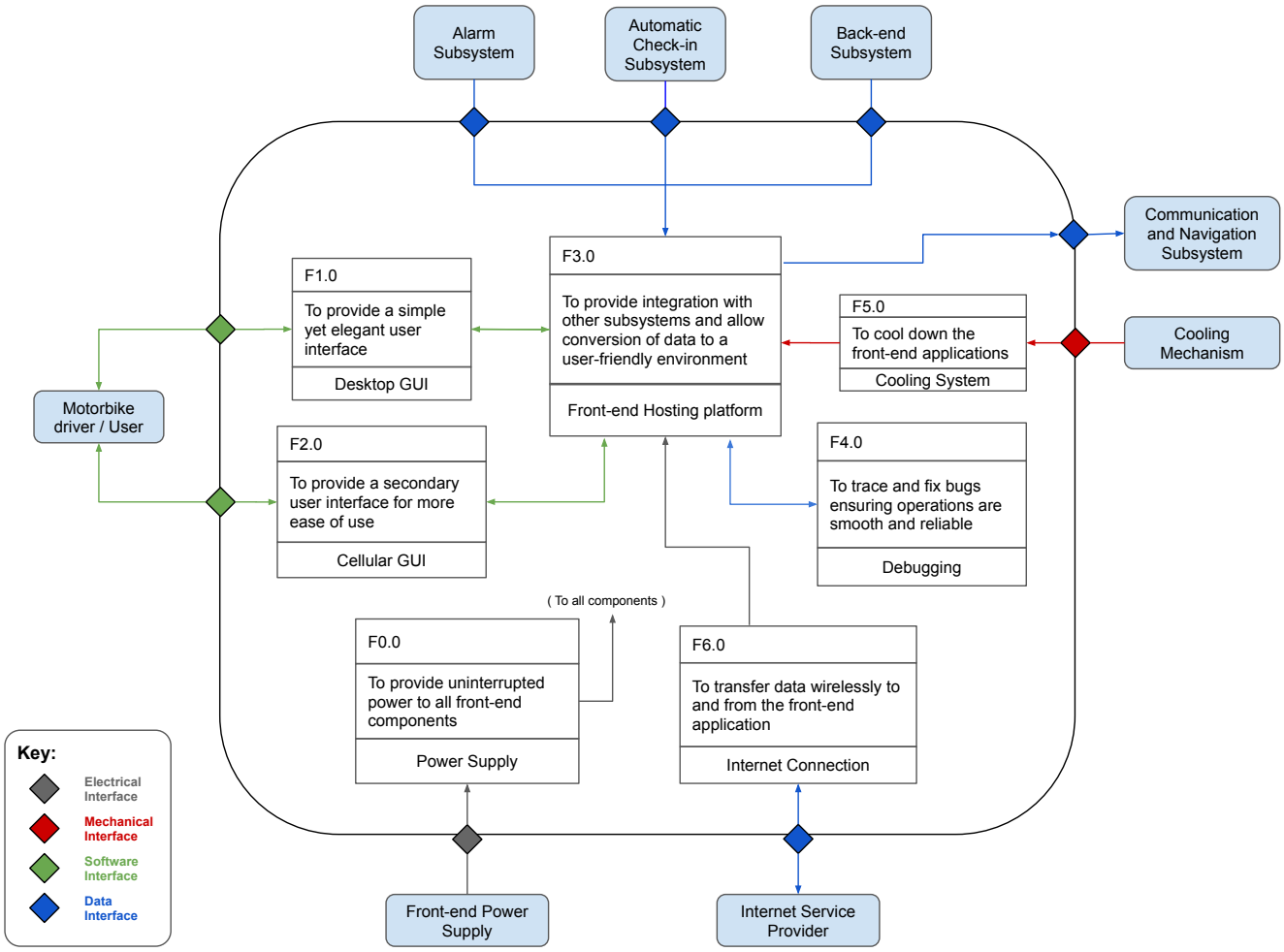


Figure 4: OPM diagram for front-end subsystem

1.5 Application Back-end Subsystem

User Requirements

UR1	Reliability
Requirement	The back-end subsystem must be reliable.
Rationale	The intention of the overall system is to improve the safety of bike drivers. Unless the system is reliable, however, these intentions are pointless—for it is in the critical conditions of operation when the system reveals its true value. More than anything else—such as the aesthetics and user interface—a user must have peace of mind that if things go wrong or if the system is urgently needed, the back-end processing will deliver.
Refined by	UR1:FR1 (High uptime) UR1:FR2 (Automatic backups) UR1:FR3 (Simultaneous Multi-Node Processing) UR1:FR4 (Reliable Power Source) UR1:FR5 (Appropriate Temperatures)
Verification	ATP1; ATP2; ATP3; ATP4; ATP11; ATP12

UR2	Integration Compatibility
Requirement	The back-end application must be compatible for integration with existing software tools.
Rationale	Many companies are already committed to a particular suite of software systems, from handling bookkeeping and inventory control to point-of-sale applications. Employers will thus be hesitant to change to using a tool that does not integrate well with their existing processes—since it may expensive and time-consuming to do so. Ideally, the new back-end system will integrate smoothly into the organization's software stack.
Refined by	UR2:FR1 (Connection to 3rd party APIs) UR2:FR2 (Adherence to Standards)
Verification	ATP5; ATP6

UR3	Scalability
Requirement	The back-end server structure must be scalable.
Rationale	Before adopting a new piece of software, a business will want to know that the tool will grow with the company. It is no use if the back-end application can handle rigid amounts and types of traffic, outside of which it becomes useless to the employer. Instead, it must be able to scale with- and match operational requirements—from low-load, simple processes for small user bases, to handling high, demanding loads for large companies, efficiently and effectively.
Refined by	UR3:FR1 (Active Development) UR3:FR2 (Distributed Processing)
Verification	ATP7; ATP8

UR4	Security
Requirement	The back-end server system must be secure.
Rationale	In the current digital era, hacking and cybercrime are massive concerns for any company. There have been countless attacks on businesses—big and small—ranging from spyware, to malware, to ransomware. The effects of these can be catastrophic, and thus any networked software system must be well-guarded against possible threats, including hackers and other malicious agents.
Refined by	UR4:FR1 (Regular Software Patches) UR4:FR2 (Good Cybersecurity Practices)
Verification	ATP9; ATP10

Functional Requirements

UR1:FR1	High uptime
Requirement	The back-end subsystem must have a high uptime—that is, the percentage of time for which the system is active and responding to requests.
Refines	UR1 (Reliability)
Refined by	UR1:FR1:DS1 (Percentage uptime)
Verification	ATP1

UR1:FR2	Automatic backups
Requirement	The back-end subsystem must perform automatic, periodic backups of the generated employer and employee data.
Refines	UR1 (Reliability)
Refined by	UR1:FR2:DS1 (Daily Backups)
Verification	ATP2

UR1:FR3	Simultaneous Multi-Node Processing
Requirement	The back-end servers must be able to acknowledge and respond to multiple user nodes at any given time.
Refines	UR1 (Reliability)
Refined by	UR1:FR3:DS1 (Separate Processor Units) UR1:FR3:DS2 (Thread safety)
Verification	ATP3; ATP4

UR1:FR4	Reliable Power Source
Requirement	The back-end system should have a constant source of electricity to all components, all the time—even during adverse conditions, such as loadshedding.
Refines	UR1 (Reliability)
Refined by	UR1:FR4:DS1 (Uninterruptible Power Supply)
Verification	ATP11

UR1:FR5	Appropriate Temperature
Requirement	The back-end system should have a cooling mechanism to ensure that none of the components are damaged due to high temperatures.
Refines	UR1 (Reliability)
Refined by	UR1:FR5:DS1 (Implementation of Cooling Systems)
Verification	ATP12

UR2:FR1	Connection to 3rd party APIs
Requirement	The back-end servers should be able to connect seamlessly with the Application Programming Interfaces of common software tools for business administration, database handling, and employee communications.
Refines	UR2 (Integration Compatibility)
Refined by	UR2:FR1:DS1 (API Vendors)
Verification	ATP5

UR2:FR2	Adherence to Standards
Requirement	All applications running on the back-end servers must adhere to the relevant international standards for coding and for documentation.
Refines	UR2 (Integration Compatibility)
Refined by	UR2:FR2:DS1 (Relevant Standards)
Verification	ATP6

UR3:FR1	Active Development
Requirement	The code running on the back-end servers must be in active development.
Refines	UR3 (Scalability)
Refined by	UR3:FR1:DS1 (Critical Bug Fixes) UR3:FR1:DS2 (Biannual major version bump)
Verification	ATP7

UR3:FR2	Distributed Processing
Requirement	There should exist structures in place for the back-end servers to run over a distributed mesh network, rather than on a single processor.
Refines	UR3 (Scalability)
Refined by	UR3:FR2:DS1 (RAID system)
Verification	ATP8

UR4:FR1	Regular Software Patches
Requirement	All applications running on the back-end system should be able to receive updated software and patches via the internet (directly or indirectly) at frequent intervals.
Refines	UR4 (Security)
Refined by	UR4:FR1:DS1 (Common Vulnerabilities List)
Verification	ATP9

UR4:FR2	Good Cybersecurity Practices
Requirement	All access tools such as passwords must follow the industry standard level of security.
Refines	UR4 (Security)
Refined by	UR4:FR2:DS1 (Two-factor Authentication)
Verification	ATP10

Design Specifications

UR1:FR1:DS1	Percentage uptime
Requirement	The system should have a system uptime of more than 99.5% over a 7-day stress-period.
Refines	UR1:FR1 (High uptime)
Verification	ATP1

UR1:FR2:DS1	Daily Backups
Requirement	Daily backups of all generated back-end system data (including video recordings) must be made automatically, and kept for at least two weeks on a completely separate RAID1 system, except the driver emergency logs, which must be kept in perpetuity.
Refines	UR1:FR2 (Automatic backups)
Verification	ATP2

UR1:FR3:DS1	Separate processor units
Requirement	The back-end system should contain at least 4 isolated processor units, thus capable of serving at least 4 users at any instant in time, after which the user requests should be held in a priority-FIFO queue.
Refines	UR1:FR3 (Simultaneous Multi-Node Processing)
Verification	ATP3

UR1:FR3:DS2	Thread safety
Requirement	Through the use of semaphores and atomic mutual exclusion objects, critical sections of code must be locked/unlocked in order to guarantee thread saftey at all points in time.
Refines	UR1:FR3 (Simultaneous Multi-Node Processing)
Verification	ATP4

UR1:FR4:DS1	Uninterruptible Power Supply
Requirement	The back-end system should use a commercially available, APC BR900GI Back-UPS Pro 540W uninterruptible power supply (UPS) unit.
Refines	UR1:FR4 (Reliable Power Supply)
Verification	ATP11

UR1:FR5:DS1	Implementation of Cooling Systems
Requirement	The back-end system should use a combination of heatsinks, fans, and liquid cooling solutions to ensure that the temperature of all components remains below $85°C$ at all times.
Refines	UR1:FR5 (Appropriate Temperature)
Verification	ATP12

UR2:FR1:DS1	API Vendors
Requirement	The subsystem must conform to the requirements set out in the 'Microsoft Graph Unified API' endpoint documentation, the 'Sage One API' specification, and the 'Amazon API Gateway web service' that uses the HAL (Hypertext Application Language).
Refines	UR2:FR1 (Connection to 3rd party APIs)
Verification	ATP5

UR2:FR2:DS1	Relevant Standards
Requirement	All backend C++ code must adhere to the ISO/IEC JTC 1/SC 22 standard, which is specifically clarified in the ISO/IEC 14882 document, ensuring adherence to the latest version of the standard within at least 4 years of its publication.
Refines	UR2:FR2 (Adherence to Standards)
Verification	ATP6

UR3:FR1:DS1	Critical bug fixes
Requirement	Developers must be available to work on high-intensity, 'critical' bugs within 2 working days of the bug being logged.
Refines	UR3:FR1 (Active Development)
Verification	ATP7

UR3:FR1:DS2	Biannual major version bump
Requirement	All back-end software running must be continuously developed, with version increments happening at least every 6 months—reflecting fixes to all low- to medium-intensity bugs that have occurred over that period.
Refines	UR3:FR1 (Active Development)
Verification	ATP7

UR3:FR2:DS1	RAID system
Requirement	Non-backup server storage must be configured as a RAID0 system, with a minimum combined capacity of 2 TB.
Refines	UR3:FR2 (Distributed Processing)
Verification	ATP8

UR4:FR1:DS1	Common Vulnerabilities List
Requirement	The Common Vulnerabilities and Exposures (CVEs) list must be checked every week for new identified security flaws in existing code libraries, and associated security patches must be written and delivered within fortnight (14 days) thereafter.
Refines	UR4:FR1 (Regular Software Patches)
Verification	ATP9

UR4:FR2:DS1	Two-factor Authentication
Requirement	All back-end system operators must adhere to a two-factor authentication system which expires daily. This process should utilise a AirPrime HL6528RDx cellular module, which can send an SMS request to the operator's cellphone.
Refines	UR4:FR1 (Good Cybersecurity Practices)
Verification	ATP10

Acceptance Test Protocols

Code	Refines	Description
ATP1	UR1:FR1:DS1	The system must be stress-tested via a continuous operation for 7-days straight, with requests made every 2-5 seconds, and the proportion of successful attempts must be confirmed to be $\geq 99.5\%$
ATP2	UR1:FR2:DS1	A day of normal system operation must be conducted with additional external recording of data, and this data should be compared to that found on the RAID1 backup drive. Moreover, after 14 days, one must confirm that the data is automatically deleted. Finally, the emergency log data must be checked that it is never deleted, even after the 14 day period.
ATP3	UR1:FR3:DS1	The *sysbench* testbench suite must be run on the back-end server system, and confirmation of the four-processing units must be noted. Moreover, a simple script test must be written to test the handling of the priority-FIFO queue. These tests should be exectuted every 3 months.
ATP4	UR1:FR3:DS2	A meta-analysis of the codebase must be undertaken every 6-12 months to ensure that all functions are thread-safe, using simulated race-conditions; as well as a general thread test, again using *sysbench*.
ATP5	UR2:FR1:DS1	The documentation of the mentioned API vendors must be considered and by inspection, the back-end system's codebase must be confirmed to be compliant. This should take place every 12 months.
ATP6	UR2:FR2:DS1	The codebase must be periodically spot-checked for compliance with the latest C++ standards, at least every 3 months. Moreover, nightly builds of the codebase must be executed and analysed for errors.
ATP7	UR3:FR1:DS1/2	Annually, a version control and bug-review process must be undertaken, and the time-based response to various bug severities must be calculated, and ensured to be in accordance with the specifications.
ATP8	UR3:FR2:DS1	Every 3 months, the RAID configuration must be checked using the *mdadm.conf* Linux configuration file, and by inspection, the status of the hard drive setup must be verified.
ATP9	UR4:FR1:DS1	A senior developer in the team must confirm that the identified vulnerabilities are indeed fixed using various industry-standard (confidential) penetration tests. All of such tests must be passed perfectly.
ATP10	UR4:FR2:DS1	The two-factor authentication system must be tested by inspection, both for the positive/correct and negative/incorrect situations—with each evoking the appropriate measures of logging in or blocking the user respectively.
ATP11	UR1:FR4:DS1	The back-up power supply must be activated at 10 random intervals over a 7-day period, and the system should successfully restart in all tests within 15 seconds.
ATP12	UR1:FR5:DS1	The system should be stress-tested at full load for 4 days straight, and the internal temperature must remain below $85 \pm 5°C$ at all times.

OPM Diagram

See figure 5 for an *Object Process Methodology* (OPM) diagram (i.e. a functional flow block diagram), representing the important components of the back-end subsystem.

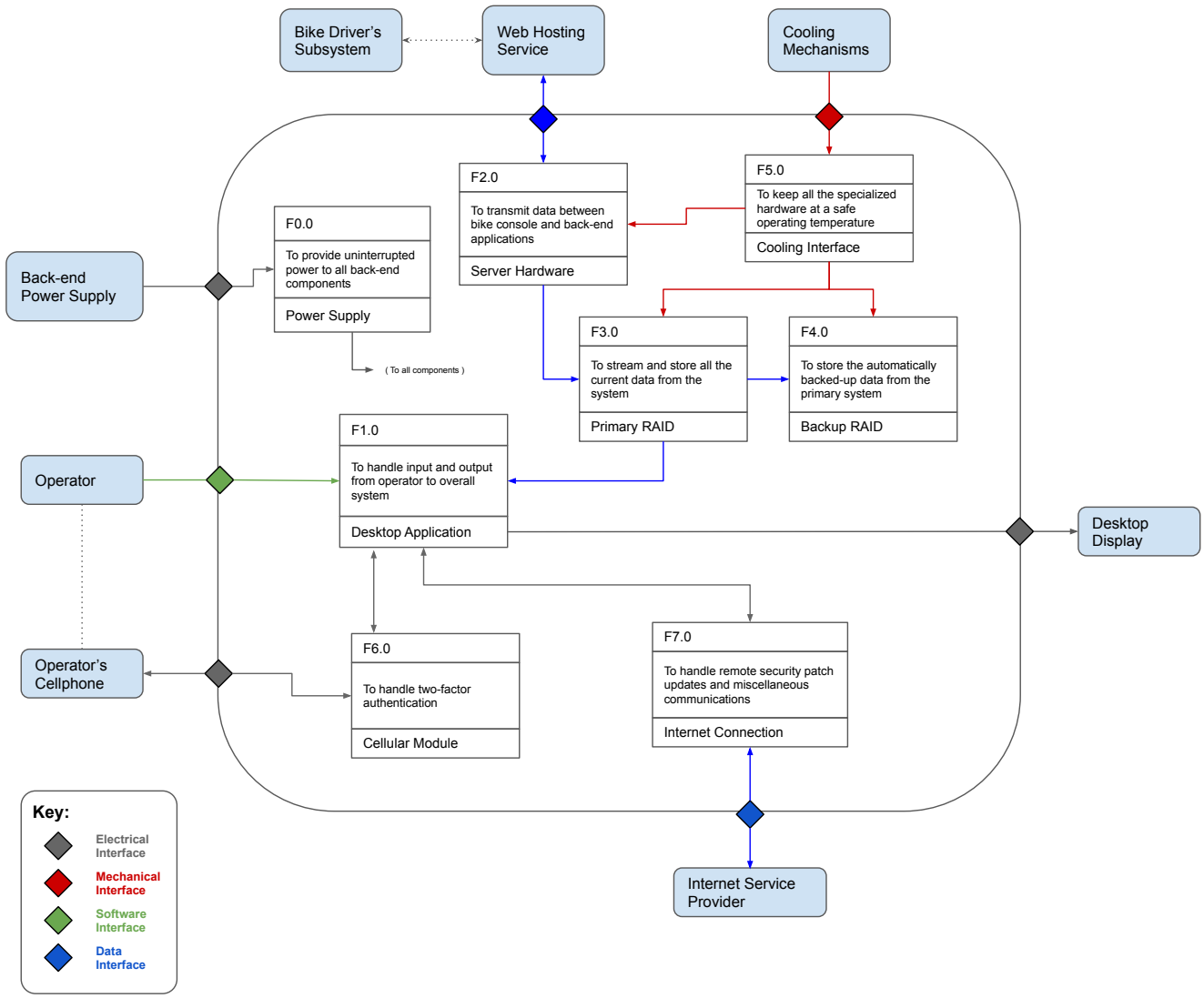


Figure 5: OPM diagram for back-end subsystem

1.6 Communications & Navigation Subsystem

User Requirements

UR1	**Other Users Security Status**
Requirement	All users including the employer must be able to see the real-time security status of other users.
Rationale	The system needs to be updated on real-time basis to enable other driver to see if a nearby driver is in danger. The employer should also be able to see the status of each and every driver in the network.
Refined by	UR1:FR1 UR1:FR3 UR1:FR4
Verification	ATP1 & ATP3

UR2	**Other Users Location**
Requirement	All users including the employer must be able to see the real-time Location of nearby other users.
Rationale	Similar to UR1 above, the location of the nearby drivers should be visible and be updated on real-time basis. The employer should be able to see all active drivers locations.
Refined by	UR2:FR2 UR2:FR3 UR2:FR4
Verification	ATP2-3

UR3	**Delivery route**
Requirement	The user must be able to see their own route for which they are to deliver.
Rationale	As a main concern for most drivers is that their phones get snatched especially at night when delivering. The system needs to show the route for the driver to follow when delivering. This would enable the driver to put away his/her phone away from being snatched.
Refined by	UR3:FR2 UR3:FR3
Verification	ATP2

UR4	**Communication with Driver's Phone**
Requirement	The information on the driver's delivery should be shared to the system display
Rationale	The delivery industry already has phone applications to enable the delivery process to be easier. It is easier and cheaper to have the information from the app to be shared to the system to be build than to have new app to be build to accommodate this system. Therefore all information on the delivery app screen will be sent to the system.
Refined by	UR4:FR3
Verification	ATP2 & ATP4

Functional Requirements

UR1:FR1	**Send and Receive Safety Status**
Requirement	Text here The real-time safety status of each driver should be send to the employer back-end system. And each driver should receive nearby drivers safety status.
Refines	UR1
Refined by	UR1-2:FR1+FR4:DS1 UR1-4:FR1-4:DS3 UR1-2:FR1-4:DS5
Verification	ATP1

UR2-3:FR2	**Copy information from Driver's Phone Delivery App**
Requirement	The Driver's delivery app already has delivery route and driver's current location information. That information should be copied from the app.
Refines	UR2 UR3
Refined by	UR1:FR2-3:DS2 UR1-4:FR1-4:DS3 UR2-3:FR2-3:DS4 UR1-2:FR1-4:DS5
Verification	ATP2

UR1-4:FR3	**Communication Channel with Driver Phone App and Employer back end System**
Requirement	There should be a communication channel between driver's phone app, employer back-end system and the devise to enable communication.
Refines	UR1 UR2 UR3 UR4
Refined by	UR1:FR2-3:DS2 UR1-4:FR1-4:DS3 UR2-3:FR2-3:DS4 UR1-2:FR1-4:DS5
Verification	ATP2 & ATP4

UR1-2:FR4	**Proximity Determination**
Requirement	There should be a way to determine how close one driver is to another in other to sent safety and location information to drivers that can do something should nearby driver be in danger
Refines	UR1 UR2
Refined by	UR1-2:FR1+FR4:DS1 UR1-4:FR1-4:DS3 UR1-2:FR1-4:DS5
Verification	ATP3

Design Specifications

UR1-2:FR1+FR4:DS1	**Back-end connection channel**
Requirement	See section 1.4 DS3 and section 1.2 DS5
Refines	UR1-2:FR1 UR1-2:FR4
Verification	ATP1

UR1: FR2-3: DS2	**Phone App communication channel**
Requirement	See section 1.4 DS5
Refines	UR1:FR2 UR1-2:FR3
Verification	ATP2 ATP4

UR1-4:FR1-4:DS3	**Real-time capability**
Requirement	See section 1.4 DS7
Refines	UR1-4:FR1-4
Verification	ATP1

UR2-3:FR2-3:DS4	**Copy information**
Requirement	Copy driver's delivery App screen information as a HTML file and sent information to appropriate channels.
Refines	UR2-3:FR2-3
Verification	ATP2

UR1-2:FR1-4:DS5	**Phone copy and send App**
Requirement	Exchanges drivers' safety information and location with employer back-end on a TCP BBR connection. Also sends other drivers information to front-end if the other driver is within 5km radius..
Refines	UR1-2:FR1-4
Verification	ATP3

Acceptance Test Protocols

Code	**Refines**	**Description**
ATP1	UR1-2:FR1:DS1+DS3	The system must update employer back-end spreadsheet with field of driver unique employment number, location and safety information when driver has done a check in each time there is a change in location (1 second) or safety information(less than 5 seconds). 98% of all data should be present on the spreadsheet with 24 hours test time
ATP2	UR2-4:FR2-3:DS2 + DS4	Enabling the network connection and starting the delivery app on the driver's phone. Performing ping from RaspberryPi should return all the required information on the display of a driver front-end module. The screen should also have identical information to the delivery app information, for top delivery apps(Takealot,Mr. D, All popular restaurants delivery apps).
ATP3	UR1-2:FR4:DS5	Visual inspection on the system to determine if only drivers within 5km radius can see each other's information
ATP4	UR4:FR3:DS2	Enabling the driver's front-end device WIFI and enabling driver's phone connection then ping the phone should be reachable each time.

OPM Diagram

Figure 6 below represents functional block diagram of Communications and Navigation subsystem as well as interaction with other subsystems.

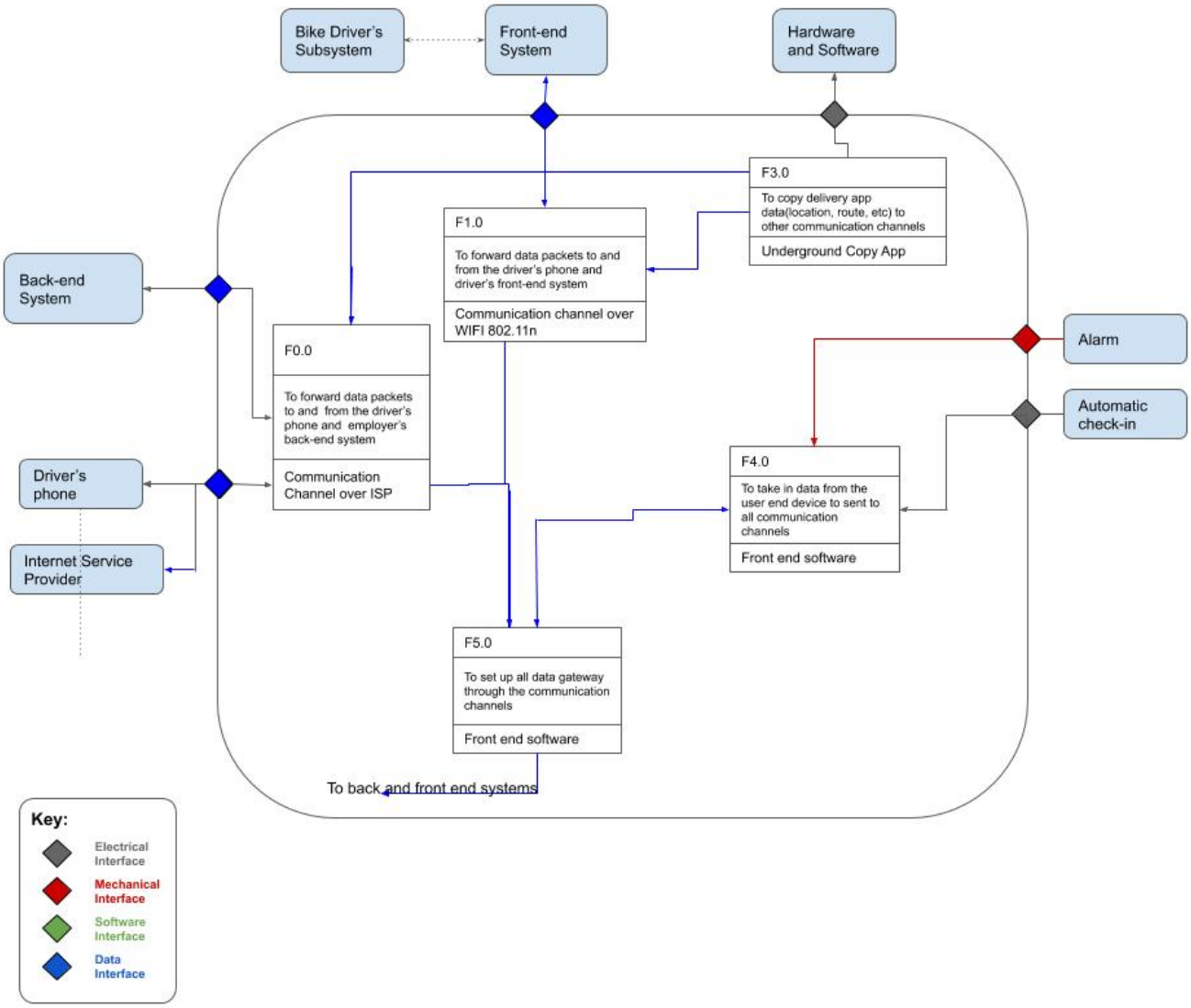


Figure 6: OPM diagram Communications and Navigation

Section 2

2.1 Power Supply Subsystem

Atomic Level Solutions

The power supply subsystem can be further refined into atomic level components, namely battery, regulator, capacitors, diodes, resistors, heat sink, fuses, battery enclosure and redundancy module.

Figure 7 shows the breakdown of the power supply subsystem into sub-subsystems (shown in cyan) and, further, into 'atomic' level components (shown in light green). The 3 atomic solutions that will be discussed are battery, regulator and diode.

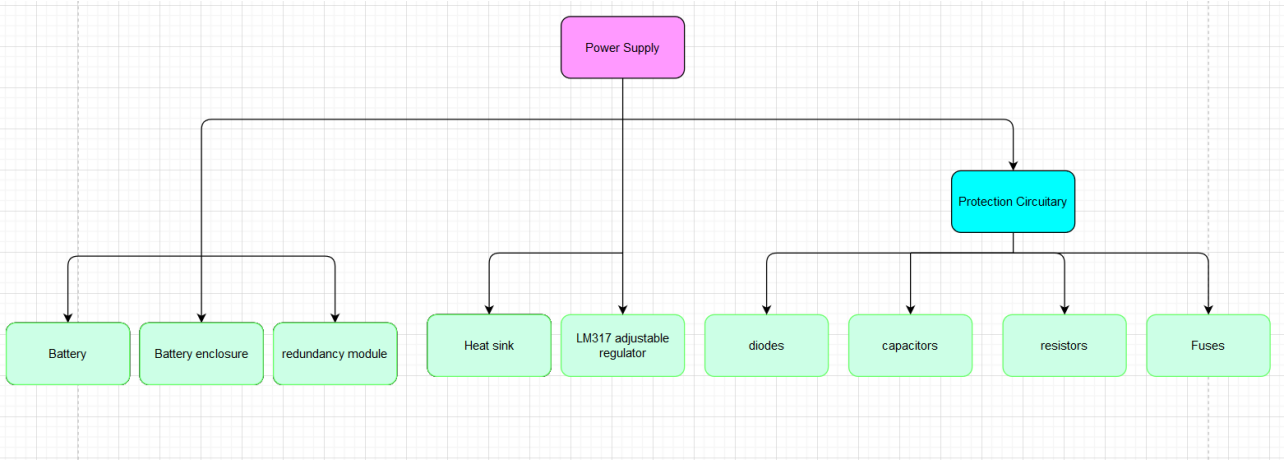


Figure 7: Atomic Level Breakdown for Power Supply Subsystem

Table 1: Atomic level component comparison (Battery)

Component	Value	Battery chemistry	Operating Temperature	Rechargeable	Capacity	Price
1. **RS Pro Battery**	9V	NiCd	-40 °C -> 60 °C	Yes	1300mAh	R1300
• [https://docs.rs-online.com/eecb/0900766b8157ee2d.pdf]						
2. **RS AA battery**	1.2V	NiMh	-10°C -> 60 °C	Yes	2300mAh	R46 (pack of 4)
• [https://docs.rs-online.com/c5cc/0900766b8164f568.pdf]						
3. **RS Pro Battery**	3.7V	LiPo	0°C -> 45 °C	Yes	2000mAh	R364
• [https://docs.rs-online.com/aadf/0900766b8163bee5.pdf]						

The RS Pro 1300mAh NiCd 9V Rechargeable Battery (1) is convenient in that it is a single cell battery with nominal voltage that is suitable for our application. The battery also possesses recharging capabilities. Nickel Cadmium batteries are rather simple to charge and rugged; however, they are toxic to the environment and must disposed in a responsible manner. The capacity of the battery falls short of what is anticipated to be required for the system's operation. The price is extremely expensive and therefore difficult to replace if irreparable damage occurs.

RS Pro 3.7V Li-Po Rechargeable Battery, 2000mAh (3) possesses the capacity that will meet our system's design requirements. It is also a single component that can provide a stable voltage source and is rechargeable. Lithium Polymer batteries, however, are potentially dangerous and care must be taken when storing and charging them. Its chemistry can also be potentially harmful to the environment if not handled correctly. Additionally, given its construction, it may prove to be somewhat difficult to implement multiple cells in series configuration (to meet the system's voltage requirement).

Selected solution: The 2300mAh NiMh 1.2V Rechargeable Battery (2) possesses the required capacity, as detailed in the system's design specifications. Its structural nature allows for the usage of a simple battery enclosure that is familiar amongst most (non-engineering) individuals. This will reduce the chance that a user might, for instance, place the battery in the incorrect manner (incorrect orientation). The Nickel Metal Hydride battery is the least harmful to the environment (compared to the other batteries) and disposal/recycling is simple. The batteries can be easily placed in series configuration to meet the nominal voltage requirement of the system. Finally, these batteries offer the most affordable price out of the potential selection and can be replaced with relative ease.

Table 2: Atomic level component comparison (regulators)

Component	Output Voltage	Input Voltage	Operating Temperature	Output Type	Max Current Output	Price	Mounting Type
1. **1.5A Fixed Output LDO Regulator**	3V3	4.5V->8V	-25 °C -> 60 °C	Fixed	1.5A	R10 (pack of 20)	Surface mount
• [https://docs.rs-online.com/39a3/0900766b81676e62.pdf]							
2. **LM3173-Terminal Adjustable Regulator**	3V3	4.2V->60V	-0°C -> 125 °C	Adjustable	1.5A	R46 (pack of 5)	Through hole
• [http://www.ti.com/lit/ds/slvs044y/slvs044y.pdf?HQS=slvs044-aaj&ts=1591434729877&ref_url=https://www.google.com/]							

Both above regulators can satisfy the system requirements.

1.5A Fixed Output LDO Regulator (1) is convenient in that it has a fixed output type; no additional components are required, other then regulator itself, to achieve a stable output voltage of 3V3. The main gripe with this regulator is that is a surface mount component; such components cannot be easily soldered using conventional means and the ability to solder such a component falls out of the scope of our skills.

Selected solution: LM3173-Terminal Adjustable Regulator (2) is an adjustable regulator and requires a few additional components to achieve the desired voltage. The addition of more components results in an increase of circuit complexity, which leads to an increase in uncertainty; as such, extra care must be taken to ensure that the performance variations is within an acceptable range. The mounting type is through hole, allowing for easy (and reliable) soldering. The structure of the LM317 is also somewhat robust (less fragile) compared to the previous regulator. The input voltage range is also extremely forgiving and allows for batteries that may not be of the exact type defined in the specifications, to be used. Additionally, an advantage offered by this regulator can be found in its flexibility; if power requirements of subsystems potentially change in the future, with a few modifications to the configuration, this same regulator can meet those new demands.

Table 3: Atomic level component comparison (diode)

Component	Max Reverse Voltage	Max Forward Current	Max Forward Voltage Drop	Operating Temperature	Price	Mounting Type
1. **Ultrafast recovery diode**	200V	1.5A	1.2V	-25 °C -> 175 °C	R7 (pack of 25)	Through hole
• [https://za.rs-online.com/web/p/rectifier-diodes-schottky-diodes/7958593/]						
2. **Ultra-Fast Avalanche Sinter glass Diode**	1000V	3A	1.7V	-55 °C -> 175 °C	R46 (pack of 5)	Through hole
• [https://docs.rs-online.com/0074/0900766b81691c66.pdf]						

Both above diodes can satisfy the system requirements.

Ultra-Fast Avalanche Sinter Glass Diode (2) boasts great capabilities that can easily meet our system requirements. However, our system requirements can still be met without such extreme capabilities, therefore using this diode will result in over design.

Selected solution: Ultrafast recovery diode (1) is much better suited to our system requirements. It is considerably cheaper, has a lower forward voltage drop (although the voltage drop of the previous diode wouldn't affect performance, as the voltage would still fall within the operation range of the selected regulator). Despite it having 'worse' specs compared to the Avalanche, it is still able to comfortably meet the system requirements.

Simulation

A simple simulation was conducted to demonstrate suitable voltage regulation of a 7V-8V battery to the required voltage of 3V3.

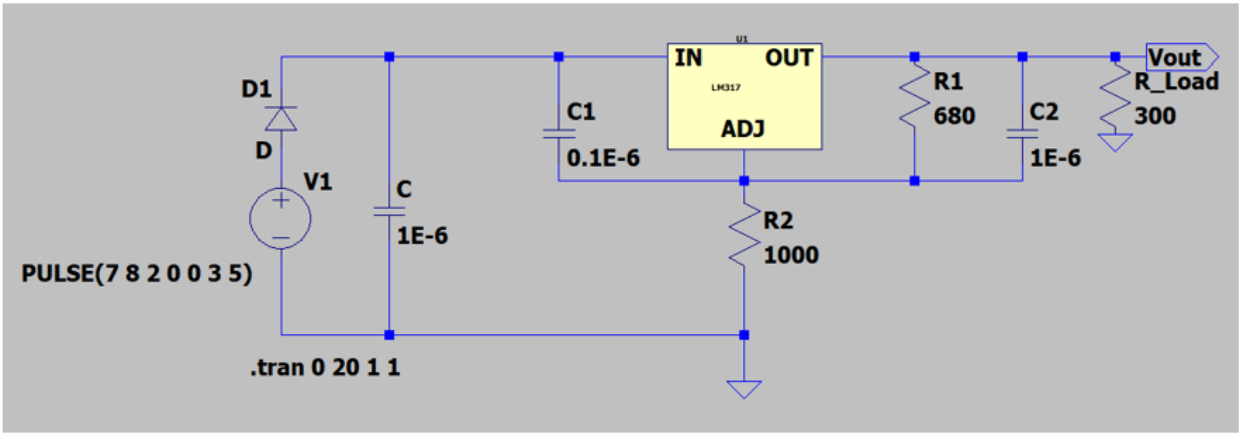


Figure 1: Circuit schematic describing power supply sub system.

As seen in the above Figure 1, is a simple circuit configuration describing the general operation of the power supply system.

- The battery pack is modelled as V1; voltage source varying between 7 and 8V
- The safety diode is modelled as D1; a diode with a forward voltage drop of 1.2V
- The decoupling capacitor is modelled as C; a 1uF capacitor
- The voltage regulator is modelled as U1; the LM317 adjustable voltage regulator configured such that it outputs 3V3.
- The load is modelled as RLoad; a 300Ω resistor
- Capacitors C1 and C2 are required for the proper functioning of the regulator, as described in the LM317 data sheet.
- Resistors R1 and R2 were calculated according the equation Vout=Vref*(R2/R1) where Vout and Vref were 3V3 and 2V2 respectively

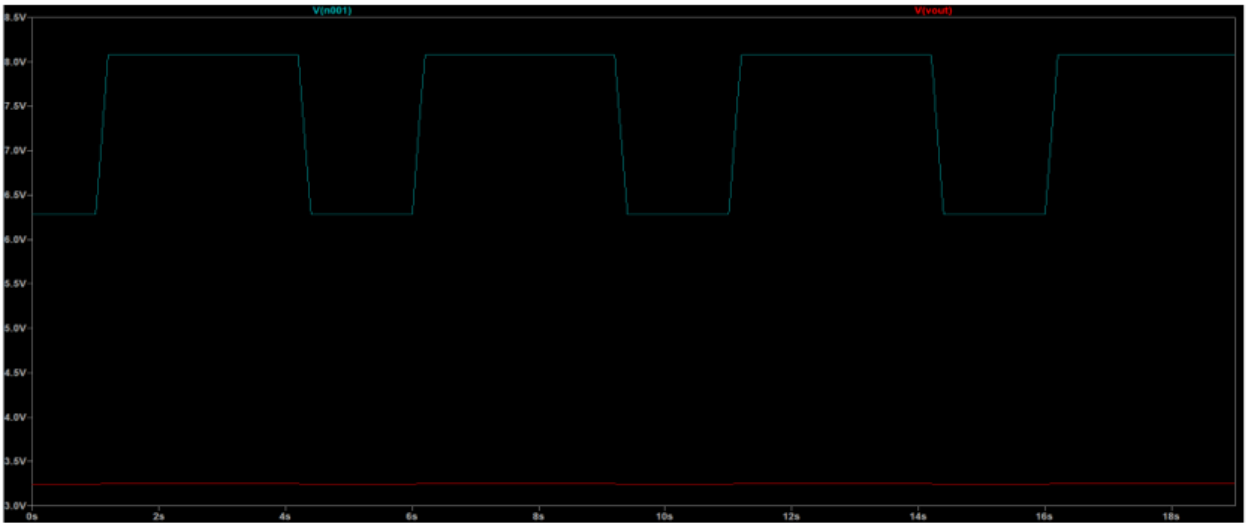


Figure 2: The simulation results of Figure 1 circuit schematic. Given a varying input between 7V and 8V (cyan), the output of the regulator (red) is approx. 3.2V. This is in the acceptable range of regulated voltage as described by ATP2 (Section 1: Power supply subsystem).

2.2 Alarm Subsystem

Atomic Level Solutions

Figure 8 shows the breakdown of the alarm subsystem into sub-subsystems (shown in blue) and, further, into 'atomic' level components (shown in orange).

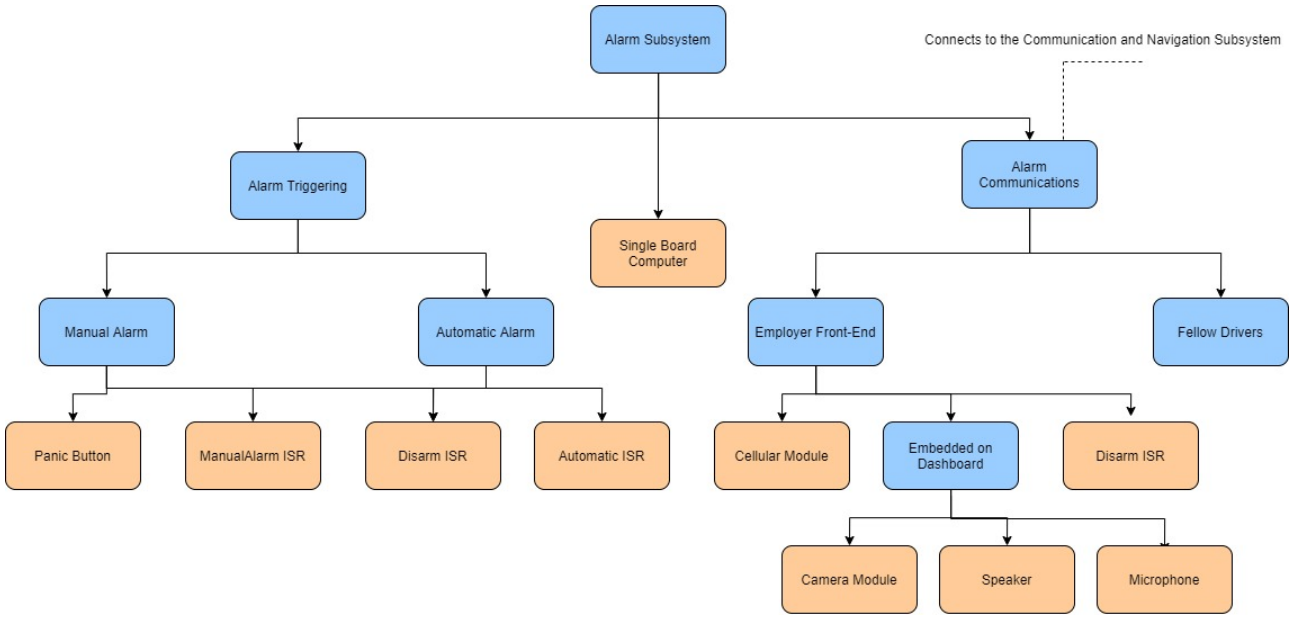


Figure 8: Atomic Level Breakdown for Alarm Subsystem

The atomic level solutions to be discussed are: the single board computer, the camera module and the panic button.

Single Board Computer

When choosing the processor for the alarm subsystem, the overall system needs to be considered as the system's functionality relies inter-subsystem communication. Considering very fast processing speeds are required, the STM and Arduino microprocessors were not chosen as their latency times were too long. The RPi presented itself as the best solution to overcome this issue. Additionally, the RPi is a Single Board Computer. This means it provides everything the alarm subsystem, and overall system, would require in terms of processing power, GPIO ports and RAM, while still maintaining a compact environment that can be easily integrated onto a bike where space is limited. The Raspberry Pi Model B was chosen for price purposes.

Camera Module

The camera module is required to be integrated onto the bike dashboard to allow the employer to assess/manage the emergency situation when the alarm is triggered. Various camera modules were considered.
The first option was the Sony MCB1152 camera module. This module has a high frame rate for 720p HD quality, is compact and high definition but is very expensive and provides many complex features that are unnecessary for this application.
Another option considered was the lower grade CMU ARDUI CMOS Camera module. The price is more competitive and it is proven to be easily compatible with and RPi but its low resolution makes it ill-suited to this application where clear assessment of the situation is required.
Finally, the SparkFun's CMOS Camera Module was considered. This is a CM-32 module which is easy to hook up to any LCD screen. The resolution is high at 976 x 592 and the operating voltage range is robust (6V-26V). The module's output is an RCA signal meaning that it would interface easily with and RPi. In light of this, SparkFun's CMOS Camera module was chosen.

Panic Button

There were many options for the panic button: from push buttons to toggle switches. Considering the panic button needed to be part of compact, detachable module integrated onto the bike's handle bars, small size was imperative. Additionally, the panic button needed to be unobtrusive and easy to use. To this end, a 1 pole on-off switch Momentary Miniature, panel mount push button switch was chosen as it is small, easy to integrate onto a detachable module and intuitive to use.

Simulation

Panic Button

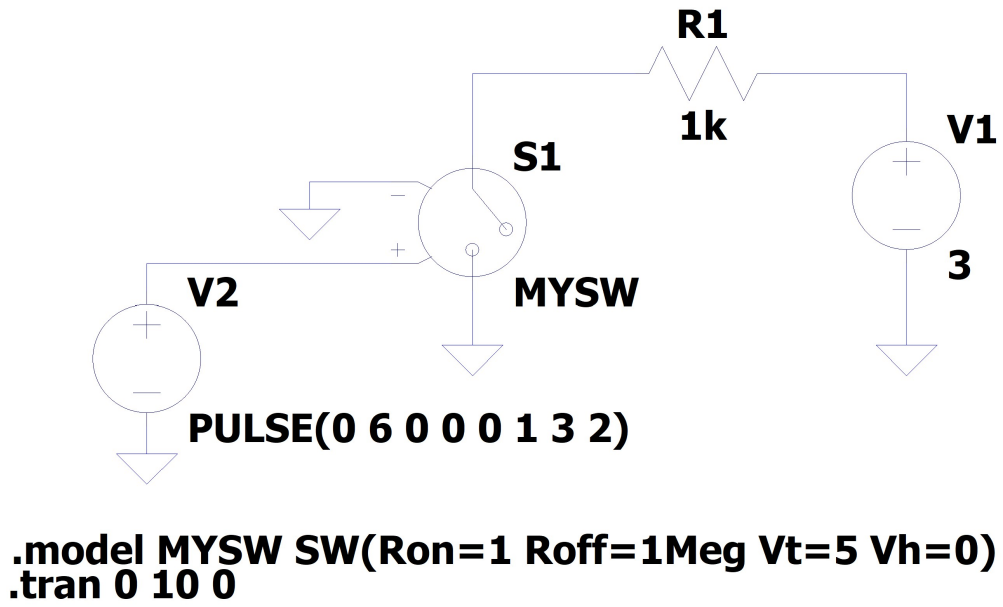


Figure 9: Circuit Mimicking Panic Button Functionality for Simulation

The functionality of the panic button itself was simulated in LT Spice using the circuit shown in Figure 9. The panic button was chosen to be normally open. V1 was chosen to be 3V to mimic the voltage output provided by the RPi2. This push button itself was modelled using a LT Spice voltage controlled switch component labelled MYSW. V2 controls the switching application. The model is configured such that when V2 $> 6V$, the switch closes. Thus, V2's pulse to 6V models the driver pushing the panic button. The length of the pulse is limited to 1s to try to mimic this action. The voltage across the switch should change in this event, triggering the relevant Manual Alarm protocols. For this to occur correctly, there needs to be a distinct voltage change across the switch when it is "pressed".

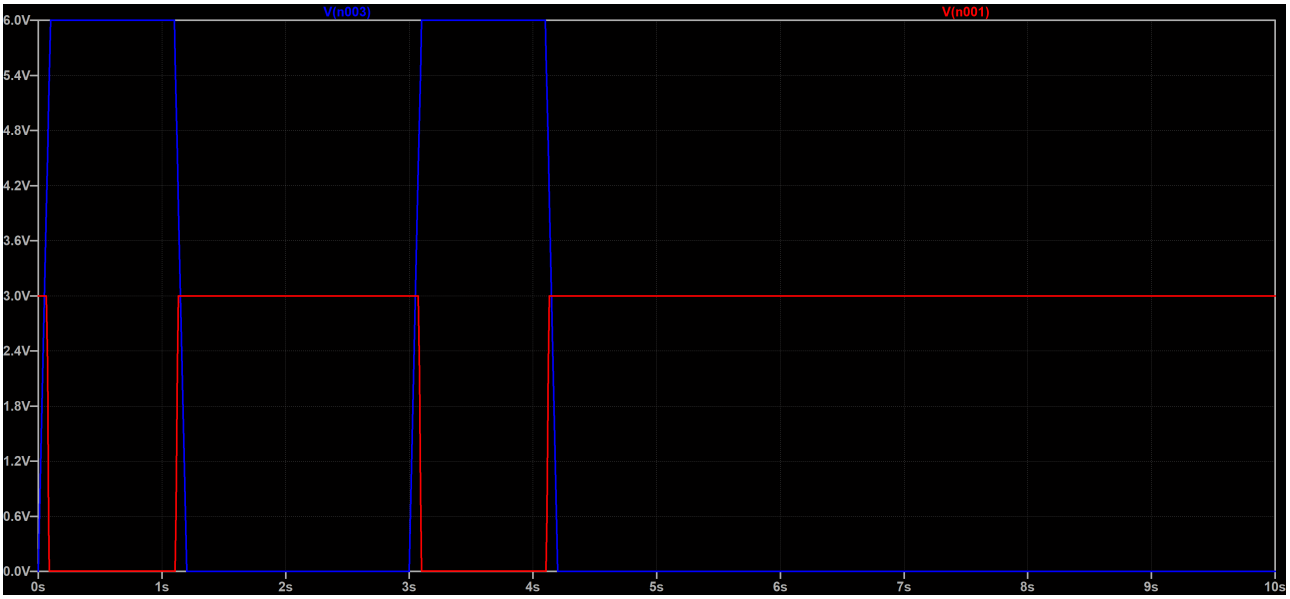


Figure 10: Results of Panic Button Simulation

Figure 10 illustrates the results of the panic button simulation. The blue trace is the V2 and the red trace is the voltage across the switch MYSW. Correct functionality is exhibited as the red trace is 3V when the panic button isn't pressed (i.e. the blue trace is below 6v) and the red trace drops to zero when the panic button is pressed (i.e. the blue trace is 6V). The change in the red trace happens almost instantaneously. This sudden, drastic change in voltage would signal a ManualAlarm trigger.

RPi 2 and Camera Module

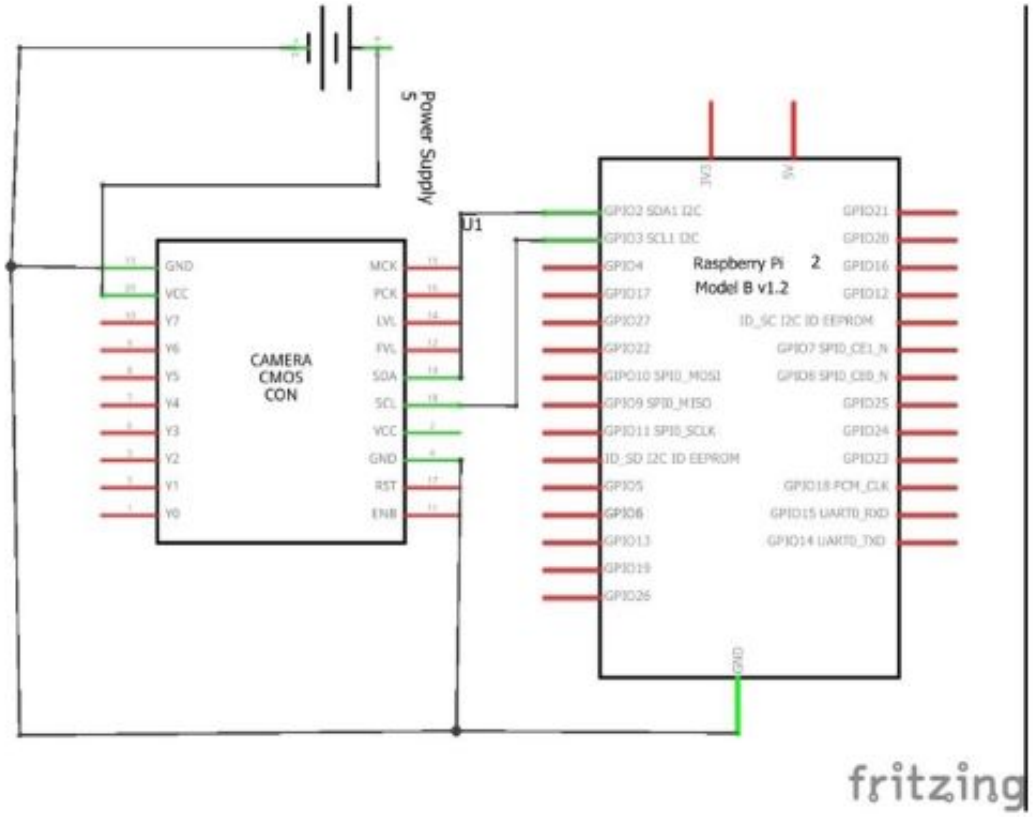


Figure 11: RPi2 Model B and Camera Module Interface Simulation

The functionality of the RPi2 Model B and the chosen camera module is difficult to simulate. However,Figure 11 simulates the connections required between the camera module and RPi via a schematic. This illustrates the wiring and connectivity required for correct implementation as well as illustrating the I2C communication channel between the two components.

2.3 Automatic Check-in Subsystem

Atomic Level Solutions

The check-in functionality of this system has been identified as one of the key subsystems and was broken down into various requirements in section 1 above. This section reviews how this subsystem can be further broken down into atomic level solutions, which when combined can make up this subsystem. This is shown in figure 12

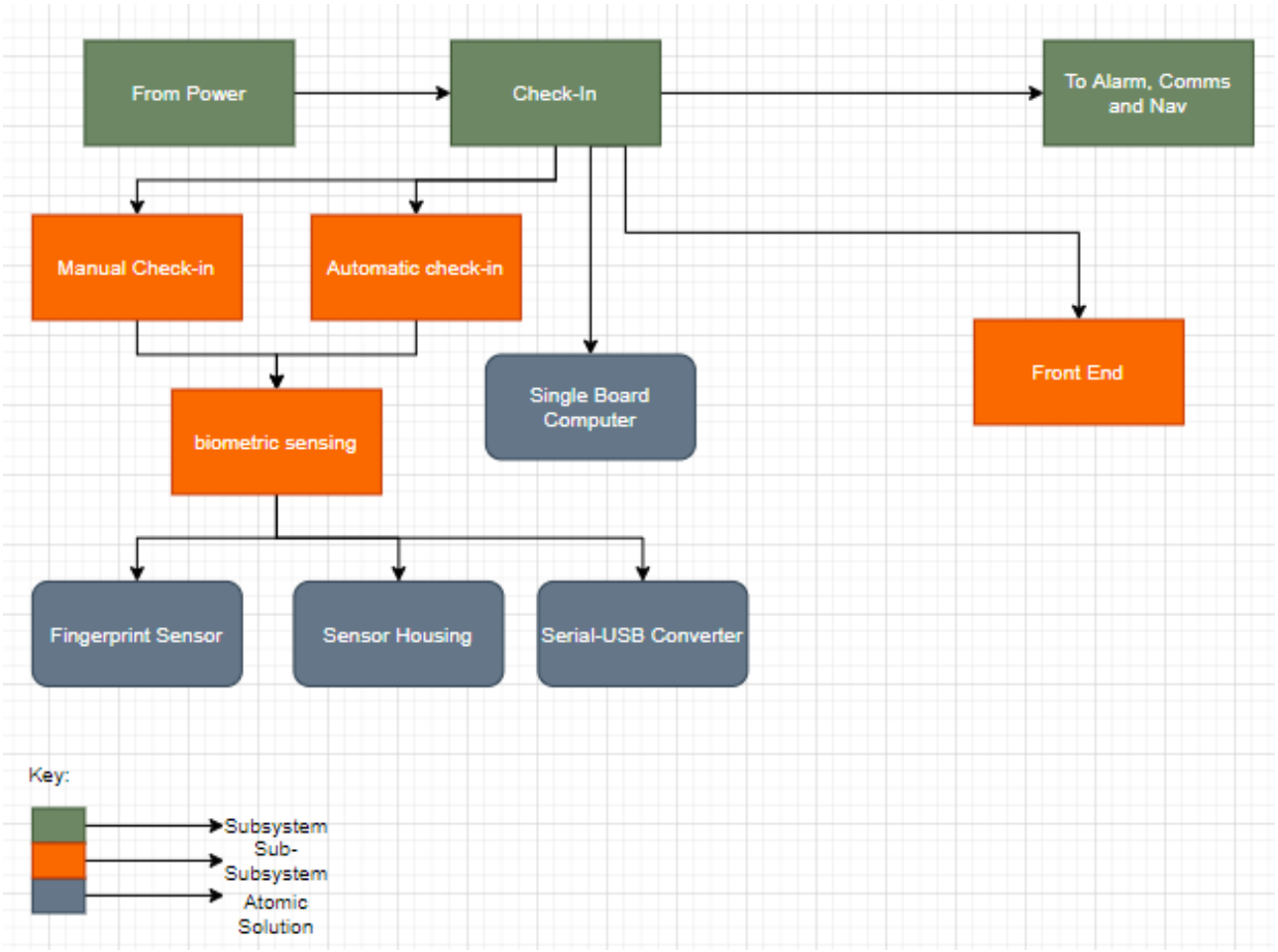


Figure 12: Diagram showing the Check-in subsystem broken down into atomic level solutions

The identified atomic level solutions are:

1. A microprocessor/single board computer (refer to table I below)
   - To allow subsystems to communicate with one another see figure below
   - To process fingerprint samples and compare them to one another
   - To store fingerprint samples
   - To upload and download data to the central server via the network
2. A fingerprint scanner (refer to table II below)
   - To allow bio-metric scanning of the user (driver)
   - To fit onto the handle bars allowing comfort and inconspicuousness
3. A serial to USB connector
   - To connect the fingerprint scanner to the microprocessor
4. Housing for the scanner/sensor
   - To connect the scanner to the bike handle

- To conceal the scanner
- To protect the scanner

Table I: Table showing the comparison of Microprocessors and Board Types

Board	Version	Processor	RAM	USB (#ports)	Ethernet	Wifi	Bluetooth	HDMI	MicroSD	Cost (Rands)
Raspberry Pi	A+	700MHz ARM11	512MB	1	none	none	none	yes	yes	R521.73@Digi-Key SA
	B+	700MHz ARM11	512MB	4	10/100Mbps	none	none	yes	yes	R828.84@Digi-Key SA
	2B	900MHz Quad-Core ARM Cortex-A7	1GB	4	10/100Mbps	none	none	yes	yes	R688.09@Digi-Key SA
	3B	1.2GHz Quad-Core 64-bit ARM Cortex A53	1GB	4	10/100Mbps	802.11n	4.1	yes	yes	R720.00@DIY Geek
	3B+	1.4GHz 64-bit ARM Cortex A53	1GB	4	300/Mbps?PoE	802.11ac	4.2	yes	yes	R622.00@Communica
	Zero	1GHz single-core ARM11	512MB	1	none	none	none	mini	yes	R544.00@Digi-Key SA
	Zero W	1GHz single-core ARM11	512MB	1	none	none	4.1	mini	yes	R959.00@Digi-Key SA
Arduino	Yun	400MHz AR9331 Linux	16MB	1	10/100Mbps	802.11b/g/n	none	none	none	retired
STM32	F030C6	40MHz ARM Cortex M0	4KB	none	none	none	none	none	yes	R154.69@Digi-Key SA

The subsystem requirements need to be considered as well as the system as a whole since the processor will be connecting multiple subsystems. The processing requirements of the system in terms of the navigation and fingerprint scanning remove the Arduino and STM boards since their processors simply do not have the speed to cope with the processing requirements of the entire system. Furthermore the most powerful Arduino board was put in table I and it cannot even be purchased anymore as it was retired. Thus the RPi is the solution. Looking at the different variations only the B models have the number of USB ports required. The 2B has a lot of RAM and processing power and is cheaper than the RPi 3 variations which is why we have chosen this board.
Solution: Raspberry Pi 2 Model B.

Table II: Table of Fingerprint Scanners

Name	Voltage (V)	Connection Type	Cost
CMU FPM10a Fingerprint Module	3.3	Serial UART & USB	R600 Communica
CMU AS608 Mini Fingerprint Module	3.3	Serial UART	R420.00 Communica
Raspberry Pi Fingerprint Scanner	3.3	Serial UART & USB	R599.00 Communica

All three of these solutions meet the requirments and will be capable of scanning fingerprints and storing them and/or processing them on the RPi. We will therefore go with the cheapest option and hence need a serial to USB converter.
Solution: CMU AS608 Mini Fingerprint Module

The converter is easy as there is already a RPi compatible serial TTL - USB converter which provides a solution still cheaper than the scanners in table II with built in USB ports. The USB to Serial CP2102 TTL UART Converter can be purchased at Zasttra for R125.99.
Solution: USB to Serial CP2102 TTL UART Converter

Finally the sensor housing will be 3d printed to meet the size specifications of the sensor itself and installed on the bike handle. The specifications are 56 x 20 x21.5mm as shown in the AS608 data-sheet.

Simulation

The circuit diagram for the connection of the sensor to the serial-usb converter to the Raspberry Pi is shown below in figure 13.

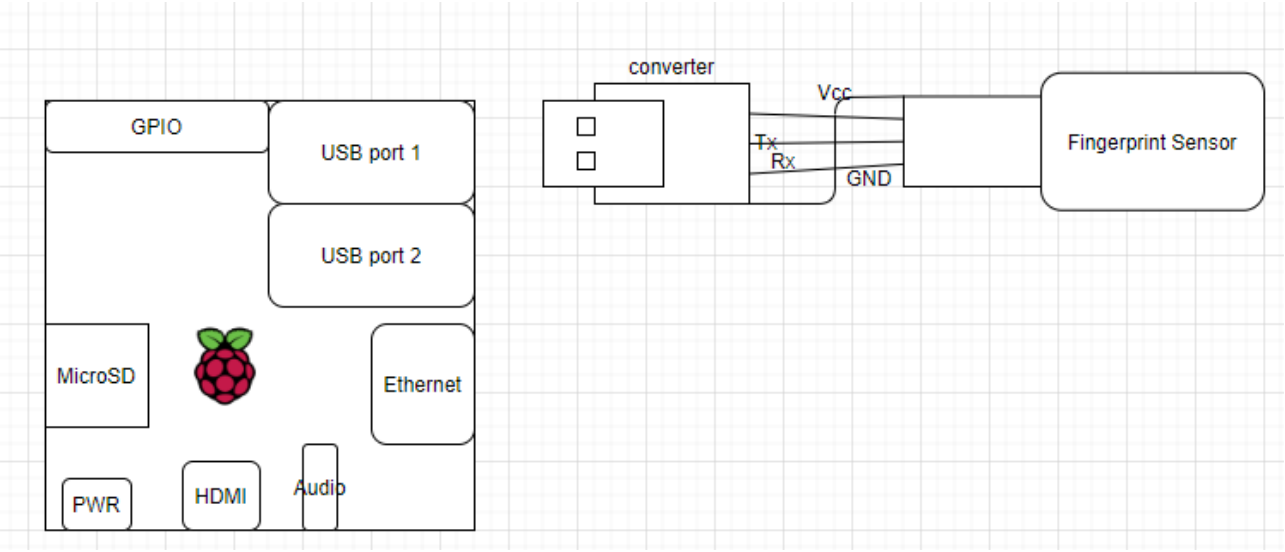


Figure 13: Circuit diagram of the Fingerprint Sensor connected via the Serial-USB converter to the RPi

Since the actual circuitry inside the fingerprint sensor cannot be accessed and is not given in any datasheet it is assumed that the sensor itself will act like a "black-box". This "black-box" is modelled as a comparator whereby it is assumed that there is capacitive sensing occurring and that when the fingerprint is applied, the capacitance changes thereby varying the voltage at one input of the comparator. The other input is a reference voltage and when the voltage due to the capacitive sensor changes enough, the reference voltage is overcome and the output goes high i.e. there is a fingerprint. There will then more processing as it images the exact location of the fingerprint based on which capacitors are being altered by the external fingerprint capacitance but for the purpose of this simulation - just a basic comparator is shown with reference voltage and applied voltage where applied voltage is the voltage due to the capacitor.
The output will be regulated to be 3v3 for TTL logic. This is all shown below in figure 14.

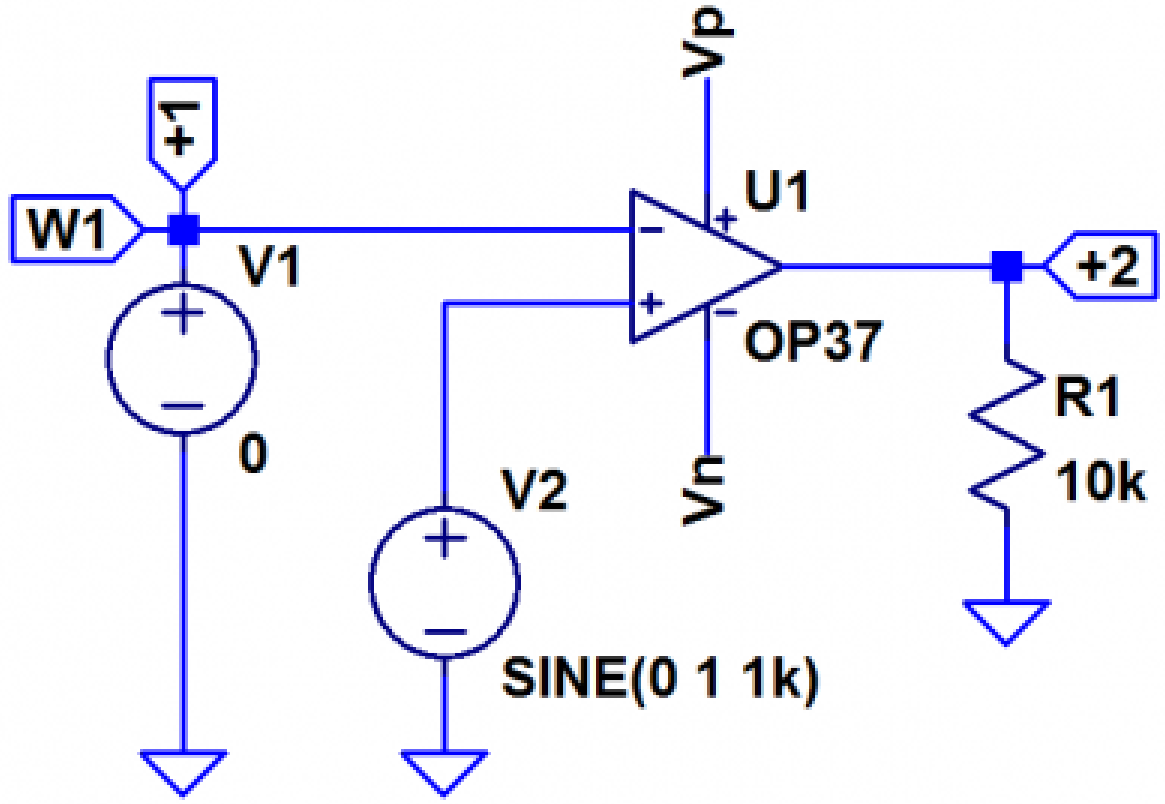


Figure 14: Circuit diagram used for simulation where V2 is the applied voltage, W1 is the reference voltage, Vn is GND, Vp is 3v3 and output2 is the output in TTL logic

The waveform is shown below in 15 where the applied voltage is a sinusoid and it can be seen that the output only goes high to 3v3 when the voltage exceeds the refernce voltage.

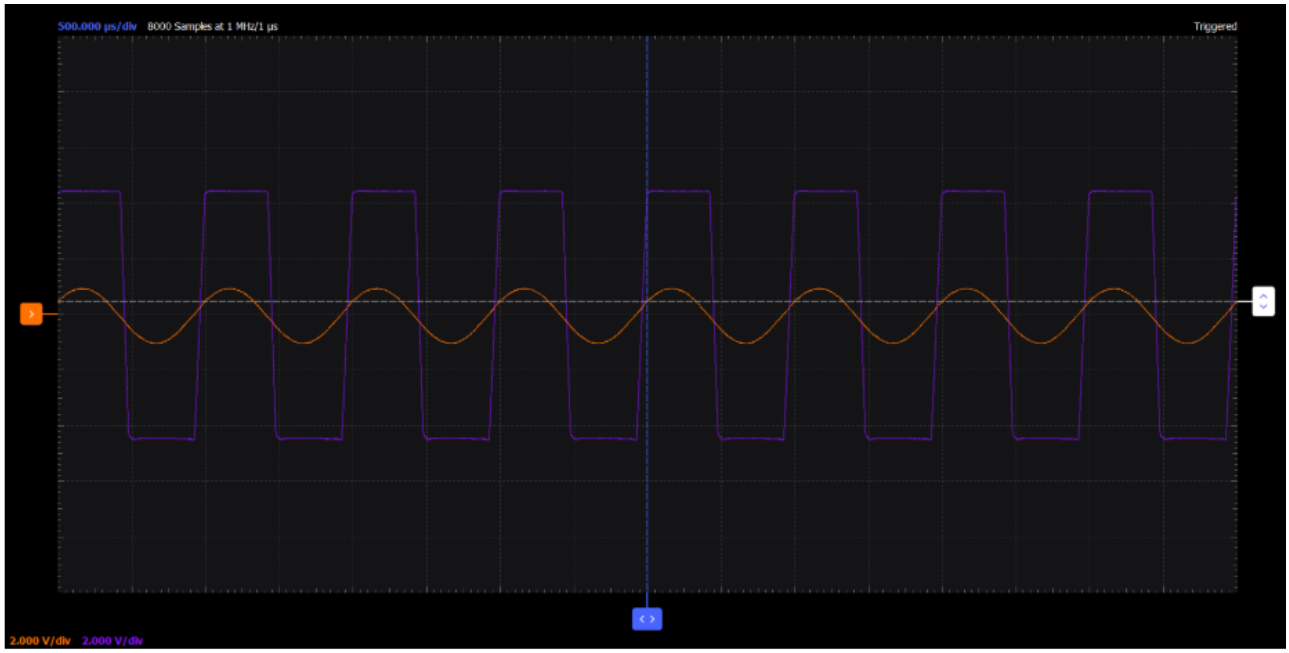


Figure 15: Simulation for the modelled fingerprint sensor

The 3v3 would be converted to a USB signal to transmit and receive data from the RPi processor.

2.4 Application Frontend Subsystem

Atomic Level Solutions

Figure 16 shows the breakdown of the front-end application subsystem into its atomic level components.

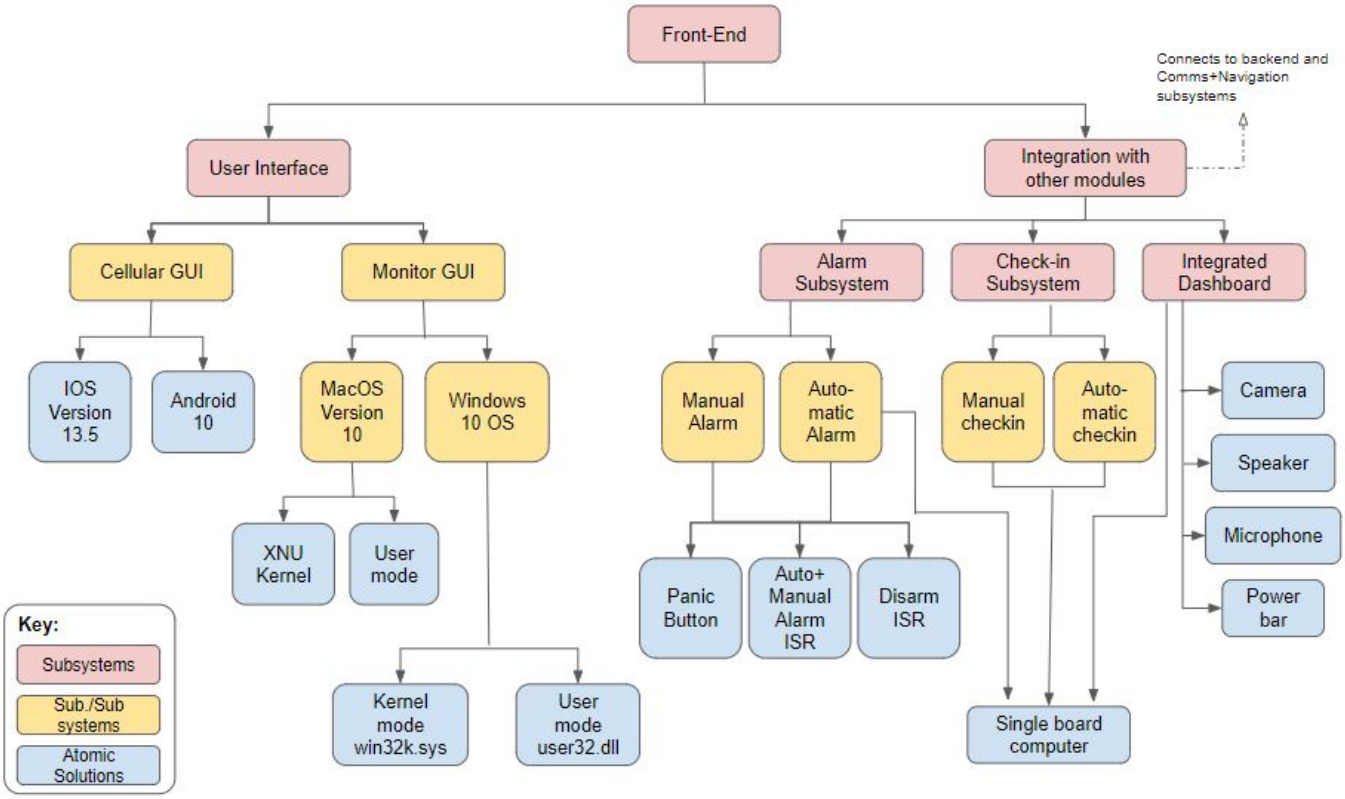


Figure 16: Atomic Level Breakdown for front-end subsystem

Atomic Level Analysis

From the numerous atomic level components mentioned in Figure 16, three were chosen for further detailed analysis and are discussed below:

Integrated Dashboard: Speaker

The speakers form one of the key elements for communication between fellow drivers and also between drivers and employee. According to the design flowcharts, the speaker is placed on the integrated dashboard alongside the microphone, camera and the power bar.

There are some key specifications such as THD, Sound levels, price and frequency range to mention a few, that needs to be considered before selecting the optimum speaker. Three speakers that were close to our design specifications were chosen and analyzed in table III below.

Specifications	UG Wireless speaker	RoHS Composite speaker	Dynamic metal speaker
Weight	1.3g	1.5g	3.2g
Rated sound level	[91.0 +- 3] dB	[88.0 +- 2] dB	[86.0 +- 3] dB
THD	<10% at 1kHz	<10% at 1kHz	<5% at 1kHz
Frequency range	0Hz - 10kHz	0Hz - 20kHz	300Hz - 7kHz
Rated impedance	8 ohms	8 ohms	32 ohms
Rated Power	0.7W	1W	2W
Price per piece	0.35 USD	0.56 USD	2.00 USD

Table III: Comparison between different speaker brands

After comparing these different speaker brands, the RoHS Composite Speaker was selected as the final choice. It was a close call between RoHS and UG speakers. The Dynamic metal speaker was the first to get eliminated from the list due to its low Frequency range, high rated impedance and high price as compared to the other two.

Now it was a decision between the remaining two speakers; both had some advantage over the other. i.e. the UG speakers were light weight and cheap but the RoHS speakers had higher frequency ranges and its rated sound level fell perfectly under the design specifications of 80-90 dB range.

Thus, there was clear trade off between Price Vs. Functionality. Here we went for functionality [i.e. RoHS speakers] only because the price difference wasn't much and we only had to purchase a single component. However, if the quantity was to increase ten folds then a closer inspection would be conducted to decide how much difference would a 2-3 dB sound level affect the overall communication experience.

Single board computer

A single board computer/microprocessor was another key system which integrates most of the subsystems with each other allowing them to safely communicate and transmit data for numerous functionalities. Since our system is a real time operation which requires rapid processing speeds, the STM and Arduino microcontrollers were eliminated leaving behind Raspberry Pi as the best choice. Also, according to table I, we clearly see that RPi 2B is the best choice from the Raspberry Pi family due to its RAM, processing power and price.

User Interface: Android 10

For the cellular GUI as a secondary user interface, we mentioned earlier in the design specifications section that our system must be compatible with both IOS and Android Operating Systems. We have chosen to analyze the atomic level component of the Android OS. From the atomic level breakdown flowchart, it was clear that the version chosen was Android 10. This was because:

- V10 introduced better UI designs and themes including a system-wide dark mode and excess of themes helping the users to customize their screen according to their taste.
- Android 10 had OLED screens which burnt fewer pixels by basically turning off pixels in deliberately black regions of the display.
- Android 10 allowed users to create a QR code for Wi-Fi or can a scan a QR code to join a network directly.
- It also allowed a 5G network support and better battery consumption than the past Android versions.

In a nutshell, Android 10 offered a simple to use yet an elegant Graphical User Interface to its users as compared to its previous family versions. It allowed for an enhanced user experience, which was one of the requirements mentioned earlier in the report.

Atomic Level Simulation

The atomic level components discussed above are simulated to show that they meet the design requirements. These simulations are in the form of a CAD design, statistical model and test codes.

Integrated Dashboard: Speaker

In order to measure accurate sound level ranges, the following circuit diagram as shown in figure 17 needs to be set up. The microphone is used as in input and must be placed at a distance of 10 cm from the speaker unit under test. The speaker is placed inside a baffle box and is connected to the generator at 890mV. This converts sound waves to electric pulses and it is then graphed as shown in figure 18.

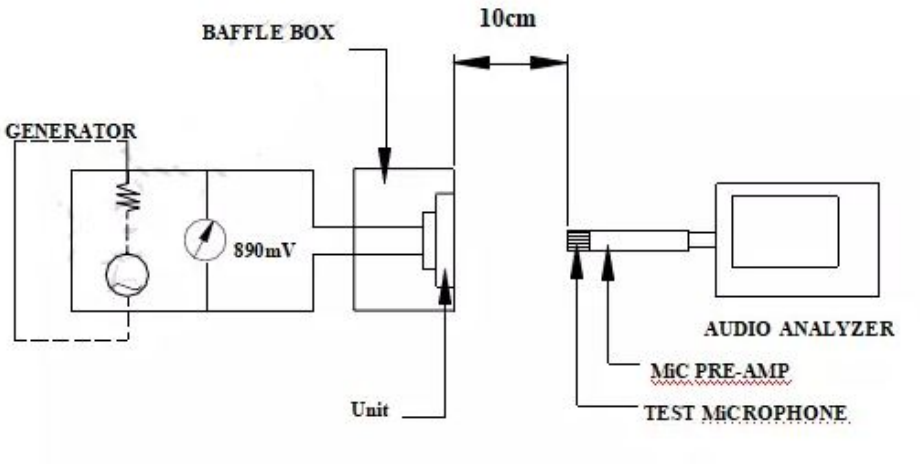


Figure 17: Set up for measuring Sound level

The graph below clearly shows that the output from the speaker starting at audible frequencies [i.e. from around 600Hz to 20kHz] has a Sound level range varying between 80-90 dB as per the design requirements.

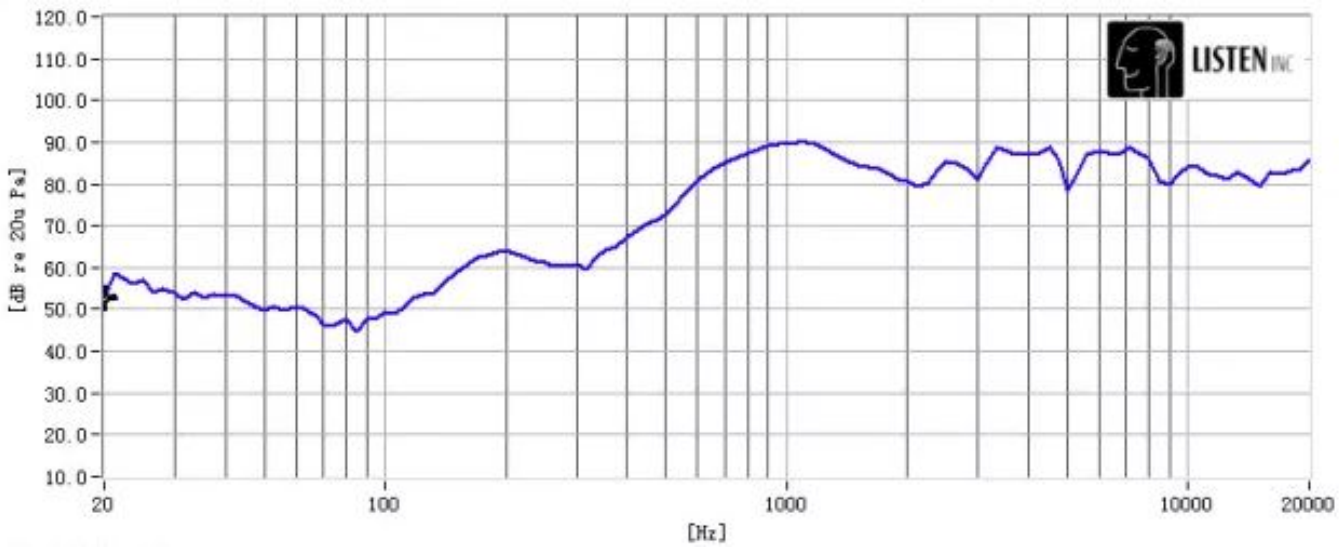


Figure 18: Graph of Sound Levels in dB Vs. Frequency in Hz

Single board computer

The simulation of a RPi2B in the front-end subsystem is difficult to implement as it connects with three other subsystems, namely with the Alarm subsystem, the Automatic Check-in subsystem and the Integrated Dashboard.
However, Figure 11 simulates the connections required between the camera module and the RPi via a schematic, and Figure 13 simulates the connections required between the Fingerprint sensor, the Serial-USB converter and the RPi via a schematic.
These connections do serve as a validation that the RPi used in the system would indeed connect as such with the above mentioned subsystems and ensures that the simulated connections are reliable and scalable to the overall system.

User Interface: Android 10

The following code describes the testing of a GUI simulation of an application in an Android Operating System. By using the IntentsTestRule mentioned in 19, the testing framework launches the activity under test before each test method annotated with @Test. The test class provides a simple test for an explicit intent. It tests the activities and intents created in the application.

```
@Large
@RunWith(AndroidJUnit4.class)
public class SimpleIntentTest {

    private static final String MESSAGE = "This is a test";
    private static final String PACKAGE_NAME = "com.example.myfirstapp";

    /* Instantiate an IntentsTestRule object. */
    @Rule
    public IntentsTestRule<MainActivity> intentsRule =
            new IntentsTestRule<>(MainActivity.class);

    @Test
    public void verifyMessageSentToMessageActivity() {

        // Types a message into a EditText element.
        onView(withId(R.id.edit_message))
                .perform(typeText(MESSAGE), closeSoftKeyboard());

        // Clicks a button to send the message to another
        // activity through an explicit intent.
        onView(withId(R.id.send_message)).perform(click());

        // Verifies that the DisplayMessageActivity received an intent
        // with the correct package name and message.
        intended(allOf(
                hasComponent(hasShortClassName(".DisplayMessageActivity")),
                toPackage(PACKAGE_NAME),
                hasExtra(MainActivity.EXTRA_MESSAGE, MESSAGE)));

    }
}
```

Figure 19: Code for testing GUI simulation of an app

Figure 20 shows the system-wide dark mode with interesting themes and an enhanced user interface. It also shows a large battery life indicated by the 6h 41m screen on time.

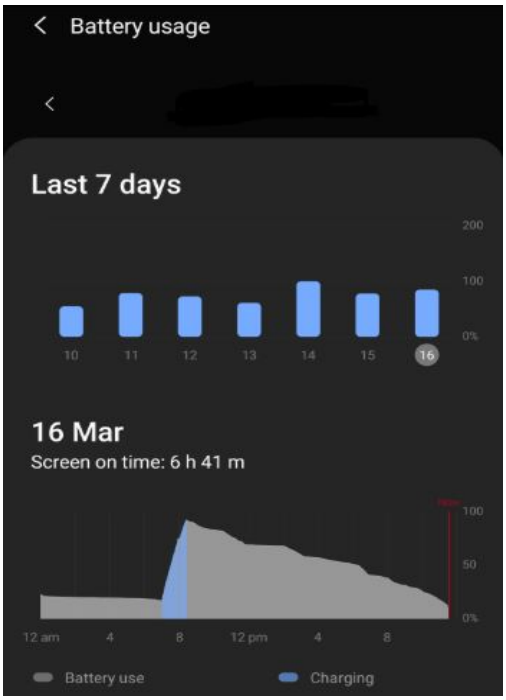


Figure 20: Android 10 features

2.5 Application Back-end Subsystem

Atomic Level Solutions

Figure 21 shows the breakdown of the subsystem into further sub-components, until eventually arriving at 'atomic' level components.

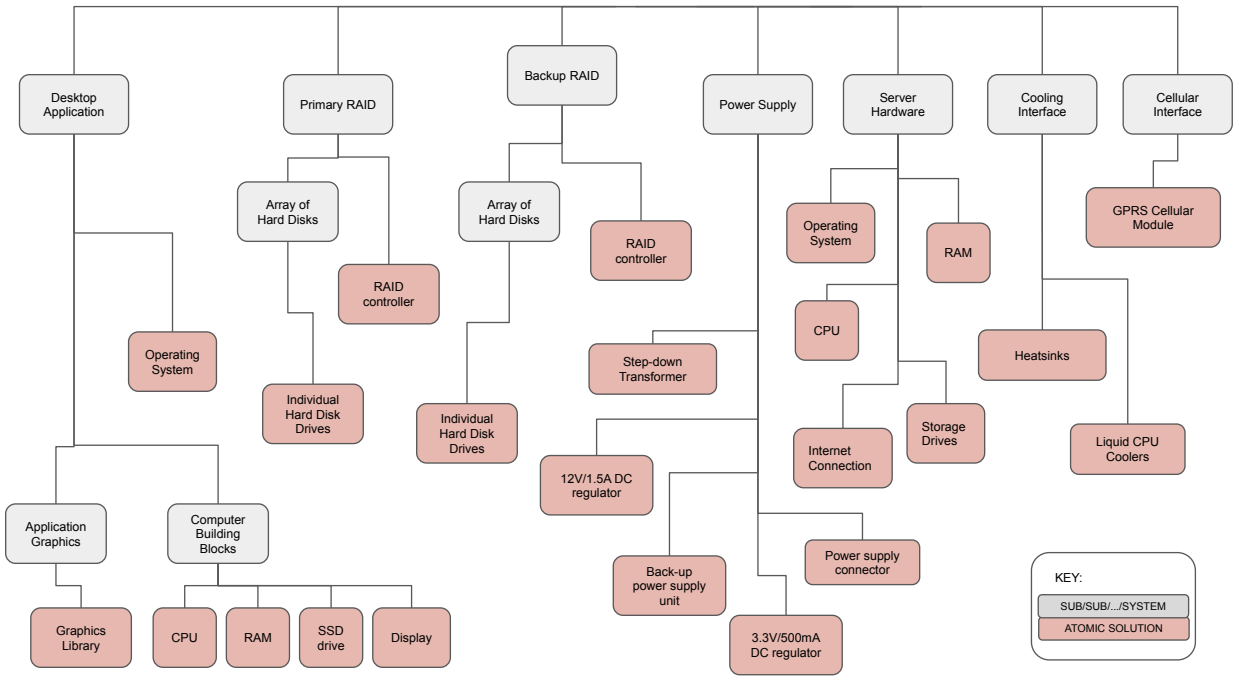


Figure 21: Atomic Level Breakdown for Back-end Subsystem

Atomic Level Analysis

Of the multitude of identified atomic components, a handful will be analysed in detail. These analyses follow.

- Desktop Application: Graphics Library
For the graphics running on the back-end application, several libraries were considered. Recall that according to the user requirements, the subsystem needs to be reliable, scalable, compatible, and secure. The application running on the operator's computer is a key component to the back-end system, and thus is strongly coupled with these requirements. Table IV shows a comparison between some of the leading options.

Application	Features	Drawbacks	License	Community
SDL 2.0	Cross-platform; multi-functional; good track-record	Written in C; Requires platform-specific tweaks	zlib	Used in over 700 games, 180 applications, and 120 demos.
Dear ImGui	Active community; self-contained; platform-agnostic	Recently developed; lack of some features	MIT	204 contributors, 3.5k forks, many recent commits
Qt	Massive community; good documentation	Initial learning curve	GPL 2.0/3.0, LGPL 3.0	~1 million developers worldwide
SFML	Simple interface; good documentation	No native support for C++	zlib/png	119 contributors, 1.2k forks, active forum

Table IV: Comparison of atomic level solutions for the back-end application interface

After considering the results in the table, it was decided to implement the Dear ImGui library. While Qt has massive worldwide support, Dear ImGui is an incredibly simple tool, allowing fast throughput from concept to reality. Moreover, it is being actively grown, and the community is thus receptive to suggestions and product decisions—this can be a strategic advantage.

- Power Supply: 12V/1.5A Regulator

The 12V voltage regulator is a key component of the back-end's power supply, as it forms part of the ATX configuration required for the motherboard's power. There are, of course, a host of options when shopping for a voltage regulator chip. Each has its own technical merits, and these should be taken into account. Having said that, it is also important to consider factors like availability, simulation data, and so on.

Four devices were considered for the 12V regulator circuit, and their salient details are shown in table V.

Code	Manufacturer	Unit Price	Available in ZA	Voltage Output	Current Output	Voltage Dropout (Max)	Quiescent Current
L7812CD2T-TR	STMicroelectronics	R 12.37	10 026	12V	1.5A	2V @ 1A	8mA
LM340SX-12/NOPB	Texas Instruments	R 25.26	2 257	12V	1.5A	2V @ 1A	8mA
NJM7812FA	NJR Corporation	R 14.46	1 433	12V	1.5A	not specified	6mA
L7812ACD2T-TR	STMicroelectronics	R 13.76	1 461	12V	1.5A	2V @ 1A	6mA

Table V: Comparison between the considered regulators for the back-end power supply

Notice that all these regulators are very similar on paper—the differences are mainly found in the amount available in South Africa, and the unit prices. Especially in the current global COVID-19 crisis, availability of components is massively important, which sets apart the L7812CD12T-TR from STMicroelectronics as a good choice. Moreover, this component is actually the cheapest of the lot. It was thus chosen for the 12V/1.5A regulator in the back-end power supply.

- Cooling System: Heat Sinks

Heat sinks are a great way to add passive cooling to a project, effectively distributing heat away from core processor components. Once again, there are a myriad of options, and choosing between them can be tricky. Important to the selection is the sink's thermal resistance, and the material from which it is made. Table VI shows a comparison between some of the leading choices.

Code	Manufacturer	Unit Price	Available in ZA	Thermal Resistance @ Air Flow	Material
ATS1195-ND	Advanced Thermal Solutions Inc.	R 87.80	9 863	$26.20°C/W$ @ 200 LFM forced	Aluminium
ATS1516-ND	Advanced Thermal Solutions Inc.	R 57.66	1 802	$12.18°C/W$ @ 100 LFM forced	Aluminium
HS107-ND	Aavid, Boyd Corporation	R 6.27	10 342	$10.00°C/W$ @ 200 LFM forced	Aluminium
AE10774CT-ND	Assmann WSE Components	R 13.24	11 631	$25.00°C/W$ @ Natural	Copper

Table VI: Comparison between the considered heat sinks for the back-end cooling system

With the subsystem requirements in mind, the ATS1516-ND is an ideal choice—it balances good thermal resistance characteristics, with a decent price and availability. Though some of the other components may be cheaper, the Advanced Thermal Solutions' offerings are made of high-quality aluminium, and feature a unique 'pushPin' technology.

Atomic Level Simulation

Different atomic level solutions require different modes of simulation. Whilst some components easily lend themselves to rigid mathematical or statistical analysis, others may require more of a meta-analysis.

- Graphics Library: Dear ImGui (C++)

To simulate the graphics library, many of the other components in the back-end system would first need to exist. It was thus hard to emulate the application in isolation. Instead, a standard example was implemented—a simple profiling tool that runs a map, some search algorithms, and other graphics features, simultaneously. This problem, while not depicting the specifics of the current project, nevertheless served as a proof-of-concept simulation for scalability, reliability, and so on, of the library. See figure 22 for the example implementation, which is built on the library.

The library build-times were fast and consistent, and the graphics elements were easily added. Integration into an existing project is as simple as including the relevant .cpp and .hpp files, and referring to them correctly. From this, it can be concluded that the library meets the requirements of the back-end application system.

- Power Supply 12V regulator: L7812CD2T-TR

To simulate the 12V regulator circuit, LTspice was used. Figure 23a shows the simple circuit used, along with the relevant SPICE directive.

Figure 23b shows the output of the regulator circuit, clearly indicating its performance in regulating the 12V supply correctly. The results of this simulation show that the chosen regulator chip is appropriate for the back-end power supply.

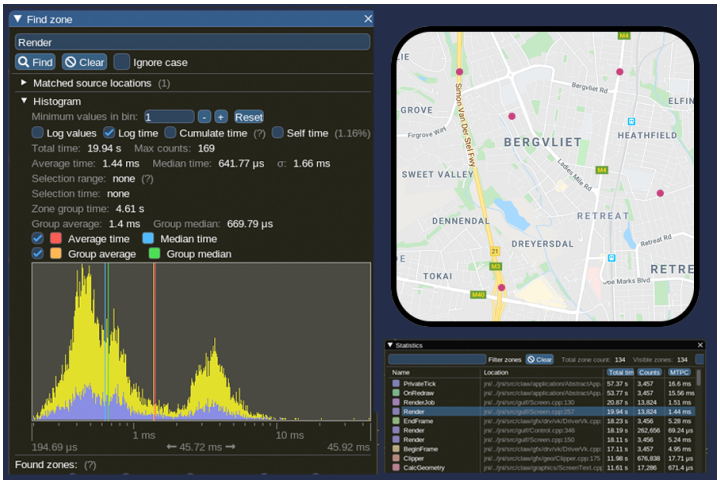
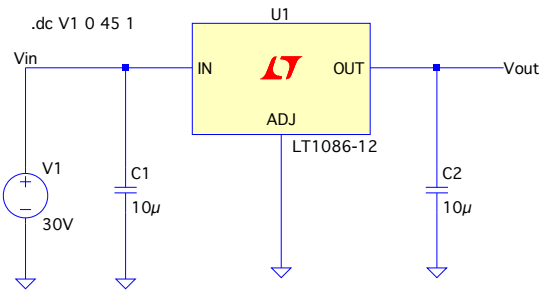
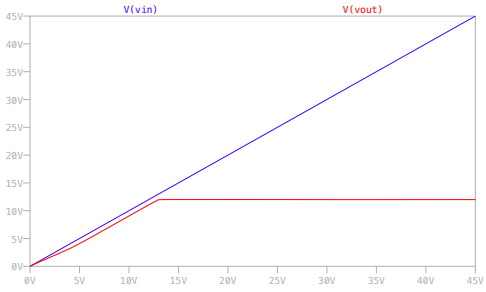


Figure 22: Sample application profiler to test GUI library



(a) Back-end 12V regulator circuit SPICE diagram



(b) Back-end 12V regulator circuit SPICE simulation graph

Figure 23

- Heatsink: ATS1516-ND

The dimensions and thermal characteristics of the ATS1516-ND heatsink were recorded in a CFD simulation tool, and a simple, preliminary analysis was run. Figure 24 shows a screenshot midway through the simulation, with the fluid flow surrounding the black rectangular fins of the heatsink clearly shown.

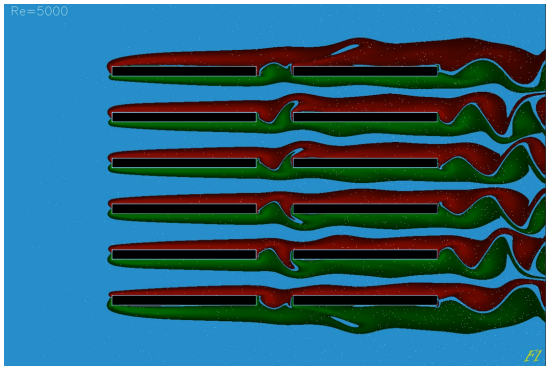


Figure 24: Fluid simulation for the heatsinks used in the back-end system.

Notice how the heatsink is effectively able to move warm air away from itself, thus dissipating its heat outwards. This analysis shows that the heatsink would be appropriate for the simple, passive cooling that some of the back-end solutions would require.

2.6 Communications & Navigation Subsystem

Atomic Level Solutions

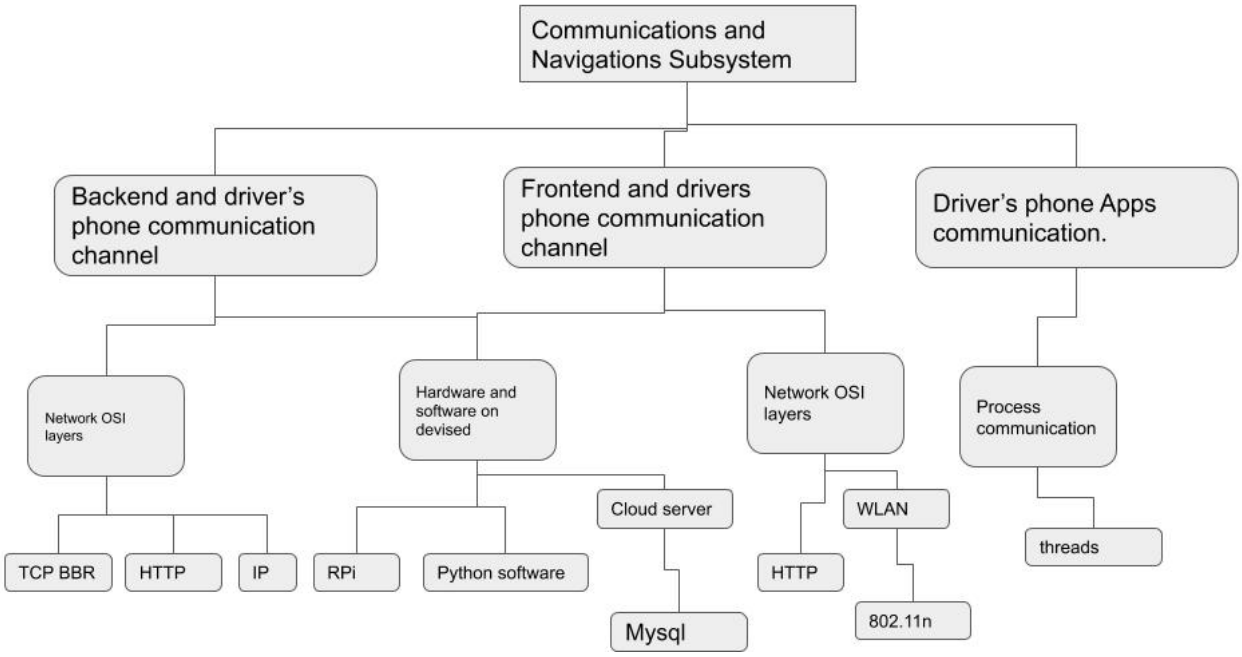


Figure 25: Diagram showing the Communications and Navigation subsystem broken down into atomic level solutions

- TCP BBR
Because we want the user security status to be secure and accurate over the internet and also avoid congestion as much as possible. TCP BBR is the best chose over other transport layer protocols.

- HTTP
Most of the traffic to be sent over the Internet is to and from the driver's phone and the Employer's back-end system. The traffic will mostly be driver id, location and safety status. Therefore HTTP file will be optimum for this type and is also reliable.

- MySQL
For database query from the employer back end we need a MySQL for its open source and easy to use. and it can also sent users accurate information with regards to sending locations and safety information of close by drivers.

Simulation

The three host should be able to communicate and be reachable at all times. The figure bellow is the network simulation of the three host to show reach ability. Where h1 is the employer back-end System, h2 is the Driver's phone and h3 is the drive's front device.

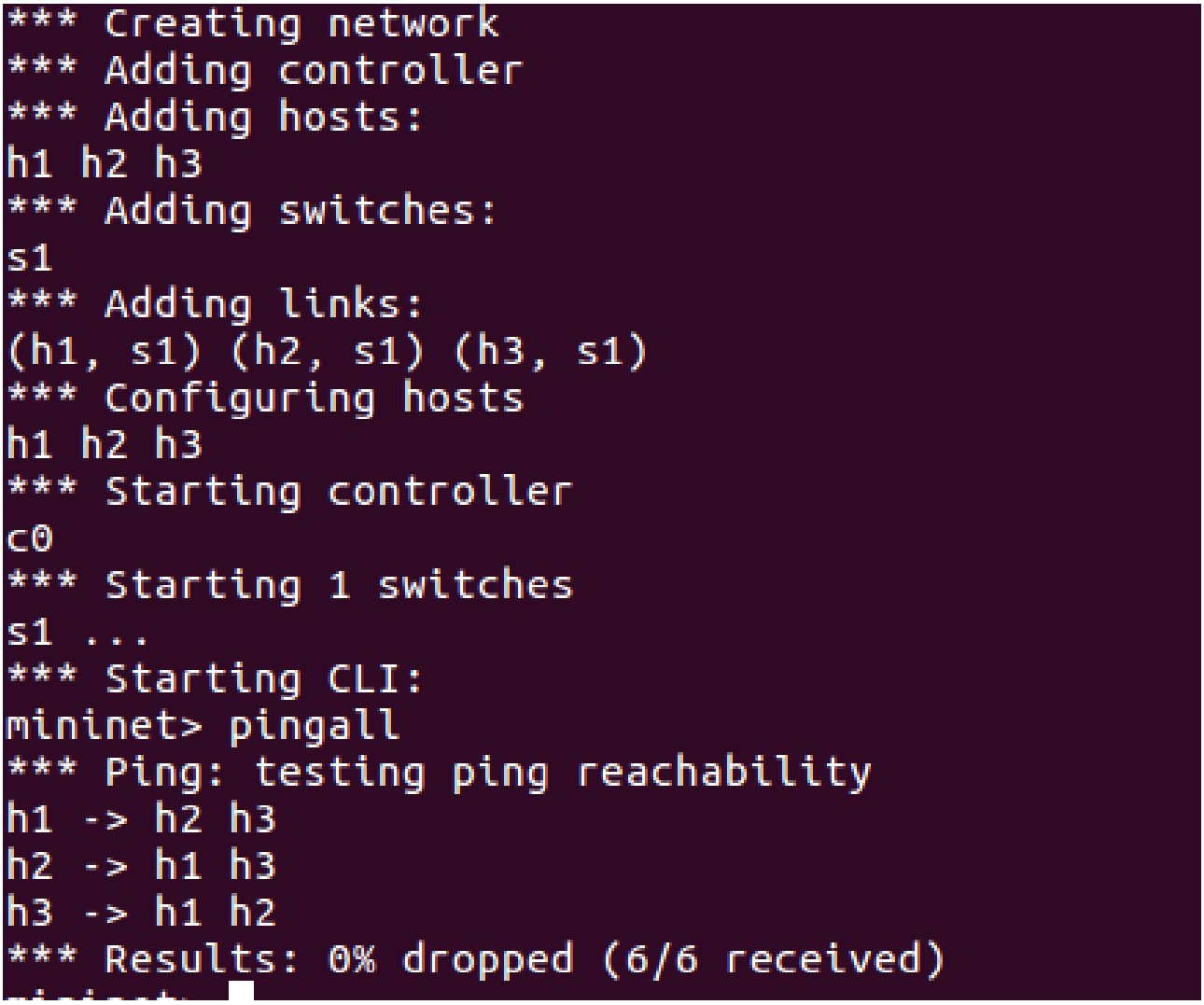


Figure 26: :Communications and Navigation simulation of network test

Section 3

Analysis: Taariq Daniels

Power Supply Subsystem

Batteries and their impact on the environment

The composition of batteries vary from one type to the next; each possessing unique advantages and disadvantages. Certain batteries are deemed more harmful than others and may require certain regulations to be followed when using or disposing of them.

An advantage of the Nickel Metal Hydride battery that was selected for the power supply subsystem, is that it is mostly* not harmful to the environment and can be simply disposed of in the bin. *If disposed of in large quantities (10 or more), the risk of it impacting the environment increases; however, given that the power supply subsystem makes use of rechargeable batteries, the disposal/replacement of batteries will be infrequent.
Nickel is an essential component in many rechargeable batteries but can also be toxic to the environment and individuals if administered in large quantities. A few complications associated with an abnormal intake of Nickel include the development of various cancers, heart disorders, failure in respiratory functions and high blood pressure.

In South Africa, majority of batteries are disposed with domestic refuse. The batteries eventually end up at landfill site where often no care is taken to separate the refuse. These batteries can then leak and poison the surrounding environment.

Fortunately, there are organisations in South Africa that collect used batteries from your home and extract the raw materials for reuse. Virgin Earth is such an organisation; they specialise in recycling electronic waste and are based in Cape Town, South Africa.

While Nickel is considered to be relatively safe (when compared to other extremely toxic elements that can be found in batteries such Cadmium and Mercury), the actual mining of Nickel reveals greater environmental consequences. The metal, Nickel, is not extracted in its pure form; rather, ore containing Nickel (and other metals) are mined. The process of extracting the Nickel from the ore (smelting) results in high concentrations of various metals being released into the air in the form of dust [1]. The metals in the polluted air eventually settle and are absorbed by the soil and rivers, which ultimately affect wild life and humans. The process of smelting also produces large amounts of Sulphor Dioxide; a compound that, in addition to causing respiratory complications in humans, can combine with water found in the atmosphere to produce acid rain. Acid rain can prove detrimental to vegetation and may potentially contaminate water supplies.

Sustainability and impact on user

The user requirements for this subsystem was that it needed to be reliable and simple for the user to maintain. As described in the functional requirements/ design specifications, multiple forms of protection circuitry were designed for. This was all to ensure that the system as a whole, from the perspective of the power supply subsystem, would be sufficiently robust. Additionally, the subsystem was designed such that the batteries required, would not only be rechargeable but also easily acquirable (affordable) in the event of failure.

The NiMh battery are also well suited to the system's application, as they boast large capacities and long cycle. There ability to reach full charge after continuous use is not impeded as greatly as some other battery types; this will allow for users to make long use of the batteries before having to eventually replace them. NiMh batteries also relatively safe compared to its Lithium Polymer (LiPo) counterpart, as there is little risk of the battery exploding or being set alight in the event of abuse or failure.

In order to get the most out of the battery, certain procedures need to be adhered to by the user. These procedures dictate things regarding how one should charge/discharge the battery, precautions that need to be taken when storing, amongst others. These rules may seem initially convoluted but are relatively reasonable and should cease to be a source of confusion after minimal instances of application.

The inclusion of rechargeable batteries will result in the final product being more expensive than if regular non-rechargeable batteries were used. The need for a battery charger also contributes to the initial investment made when purchasing the product. However, in the medium to long run, the benefits of having reliable and robust batteries that do not require frequent replacement will definitely justify the initial investment cost.

Analysis: Inessa Rajah

Alarm Subsystem

There are numerous issues to take into account when considering the impact the Alarm Subsystem has on the user (the delivery bike driver), the environment and the larger societal landspace in which the subsystem exists.

Firstly, the overall aim of the subsystem is to provide the user with an alarm service that enables the user to receive the necessary help they require in an emergency situation. In this way, the subsystem aims to meet the overall system user requirement of improving the safety of delivery bike drivers. The design specifications related to the reliability of the subsystem such as fast ISR response time, fast communication time and dual redundant modes of communication in emergency situations, aim to ensure that the user is consistently provided with alarm functionality. The overall impact of the Alarm Subsystem on the user should be to aid the user's driving experience by providing an additional safety measure and thereby also affording the user some peace of mind.

Although the aim of the Alarm Subsystem is to help the user, there is the issue of the user's privacy to consider. The Alarm Subsystem functionality includes that, in the event of an emergency when the alarm is triggered, the driver's employer will gain access to the camera module embedded in the dashboard in order to assess/manage the emergency situation and notifying the relevant emergency services. It is imperative that the employer only gain access to this dashboard camera service in the event of an emergency. The purpose of this functionality is to provide the user with the highest probablity of receiving the help they would require in any emergency, not to private the user's employer with the means to survey the user. Although measures are in place to prevent the employer from having continous access to the camera, it is noted that in the event of an accidental automatic alarm trigger- the employer might be provided with camera access in non-emergent situations. In this way, there is the possibility of the employer invading the user's right to privacy. However, the "check-in" subsystem is integrated with the alarm subsystem to ensure that accidental triggers of the automatic alarm are unlikely.

Considering the Alarm Subsystem contains numerous electronic components ranging in complexity from a Raspberry Pi to a push button. there is also the negative environmental impact caused by each component's lifecycle to take into account. Firstly, there is the power consumption required to manufacture all electrical components. Considering the use of sustainable, renewable energy sources is not extremely widespread, it is very likely that coal or gas was burned to generate the necessary power. This indicates that the the manufacturing of these components contributed directly to the emission of greenhouse gases. Moreoever, the extraction of the raw metals required to manufacture electrical components requires even more power, as well as degredating the environment and disrupting nearby eco-systems. It is also pertinent to consider the consequences of the eventual disposal of the electrical components used in the Alarm Subsystem. The incorrect disposal of electronics contributes to e-waste. E-waste contains high levels of toxic chemicals and other substances which have the potential to contaminate soil, running water sources and groundwater sources. In South Africa in particular, infrastructure required for safe e-waste disposal is not very advanced, meaning that chances of contamination due to e-waste is high. Considering the lack of universal access to vetted, safe drinking water in South Africa- the chance of numerous members of the South African population consuming water tainted with chemicals from e-waste is also high. Animals are also likely to be harmed by this contamination. In light of this, correct, safe disposal of any electronics involved in the Alarm Subsystem is required to implement the subsystem ethically and sustainably.

The societal consequences of the use of imported electrical components is also of relevance. The Raspberry Pi is manufactured by Sony either in factories in Wales or China. Either way, the components would need to be imported to South Africa for the construction of the Alarm Subsystem. There are numerous negative environmental effects associated with importing goods. Moreover, the conditions for factory workers in Sony factories have been described as abusive and unethical. Additionally, Sony factories in China profit off paying factory workers menial wages despite horrific working conditions. Purchasing products from Sony perpetuates this abuse and fuels the disenfranchisement of factory workers.

Analysis: Noel Loxton

Check-in Subsystem

To consider the impact of this subsystem, one must consider the various atomic solutions and their impacts on both the user and the environment.

The main thing to consider with regards to the user is the ergonomic impact that a thumb sensor would have on the driver. It is imperative that the ATPs which deal with comfort with respects to the user are passed. If the driver has to press the thumb sensor and it is an awkward angle for their hand, their could be various impacts such as Repetitive Strain Injury or prolonged discomfort thereby impacting their work, health and therefore happiness. Furthermore one needs to consider the social impact of storing the fingerprints of the driver on a device. Does this data get stored on a central server and if so is the user comfortable having this data kept in a remote location for other people to potentially access? The user could feel that this impacts their privacy. One must ensure that the driver knows about this data being kept on the Raspberry Pi and a central server and consents to such.

The environmental impact of the solution is easier to analyse. Obviously any electrical device is manufactured and as such contributes to global pollution. The Raspberry Pi as well the converter and fingerprint scanner would have been manufactured in a factory and, as aforementioned, contribute to this problem. It is such a small device and as they are made in mass batches it is a very small impact but obviously if this system begins to be made in mass, the demand for the atomic solutions goes up and the pollution goes up. One needs to understand and acknowledge the trade-off between the positive social impact of increased driver safety and environmental impact.

The usual lifespan of a Raspberry Pi is around 25-30 years when being overclocked and 30-35 years running at normal clock speed so a safe assumption is that the Raspberry Pi will last 30 years in this system. The connector will assumedly last much the same amount of time if left untouched in the system i.e. not being disconnected and reconnected excessively. The maintenance of the fingerprint sensor will be more so since it will be susceptible to wear with extensive use from the users fingers. The usual lifespan of a USB scanner is around 5-10 years. The housing will most likely be the component needing maintenance since it will be exposed to the environment and like the fingerprint sensor will be handled by the user. The usual lifespan can last anywhere from six months to fifteen years depending on the use and exposure to external factors. It is safe to assume that the environmental impact will be felt most by replacements to the scanner and housing around every two to five years. At least this should be the maintenance goal of us - the developers.

The social impact of this subsystem is the main thing focus however and the point of this system is to enhance safety and security for the user. If this subsystem performs, the system as a whole will provide the driver with a safe and reliable check-in function that will alert the employer and/or security services if something goes wrong. The impact on the driver and the motorcyclist industry as a whole will ultimately be felt in years to come - this is the goal of this subsystem and its optimistic contribution on the system as a whole.
It must be considered however that this device will not perform the way one would hope and extensive post production studies must be conducted to ensure that the system is performing acceptably and ultimately providing safety for the driver as per the user requirements.

One should also however acknowledge that not only will safety for delivery drivers (our target market) be increased but safety for motorcyclists as a whole and perhaps safer road practices for driving in general will be an outcome further down the road.

This goal all depends on each subsystem performing its sub-task in the larger system. The check-in functionality, as aforementioned, as little impact on the environment and socially its impact on the user can be mitigated by proper disclosure to the user so ultimately it can provide safety for the driver.

Analysis: Ronak Mehta

Application Front-end Subsystem

In an increasingly online world, the user interface is generally the users' first interaction and first impression towards a new upcoming technology. The front-end developers need to take this into account for a strong front-end application as its aim is to attract users and provide the necessary functionality they claim to promote. The front-end application usually reflects the business profile because the first impression is usually what makes or breaks an agreement.

The front-end subsystem for this project is an extremely vital system as it not only has to provide a smooth, simple and elegant user interface but also has to integrate and communicate with all the other major and minor subsystems. The designed front-end system is set to operate on both cellular and monitor Graphical User Interface with IOS, Android, macOS and windows compatibility. The front-end subsystem integrates with the Alarm subsystem, the Automatic Check-in subsystem, the power subsystem, back-end subsystem, the dashboard and communication and navigation subsystem. We have to look in depth up until the atomic level structures, to analyze the impacts which the entire system has on the society at large. Since the success of the front-end has a direct impact on the success of the entire system, it is extremely important to have a sustainable and impactful footprint on the user as well as the environment. Some of these social, economical and environmental impacts are discussed below in unison to the role of engineering in society.

To start on a positive note, the demand for front-end development has increased ten folds with an ever increasing number of online users. This not only improves the quality of the current online systems due to competition [benefit to the user] but it also opens a new job market by providing opportunities to budding developers. This further improves the overall employment rate of the country and helps to maintain a source of stable income for the developers [economic benefit to the society at large]. Due to high demands of front-end development, it also allows developers to collaborate with foreign nationals and increase their social relations which further propagates globalization [cultural benefit]. These international relations can also bring foreign investments to the country market increasing the GDP of the country.

Since most of the framework on the entire web system is electrical based, the trail left behind does have a negative impact on the environment in the form of global warming. Currently, most of the electrical tools used for communication and data transfer from place to place is not based off of renewable energy, and thus most of it uses fossil fuels for generation. This takes a toll on the environment and increases the concentration of greenhouse emissions resulting in the depletion of the ozone layer. For this project, we have used Raspberry Pi as the microprocessor which needs a power utility to run and relay information from one subsystem to the other. The manufacturing industry which produces the electrical components used either directly or indirectly in the front-end subsystems [such as the fingerprint sensors, biometric alarms, camera, microphones, speakers etc] are manufactured in bulk which also increases the emissions and harm the environment.

Despite the fact that a front-end application leads to an increase in income as discussed above, it also creates a dependency on foreign products. Most of the front-end hosting platforms are not South African based; they are either built on American or Chinese soil. In this new era of the fourth industrial revolution, these two giants are in a constant battle of taking charge of the big data wave. So, having too much dependence on an international entity could result in a breach of privacy as witnessed by the Facebook-Cambridge Analytica Data Scandal. The solution to such a problem is not straight forward as all of us are data driven in this current era and thus we either have to strike a balance between accumulation and exposure of data or have locally built platforms which are under the government jurisdiction.

All in all, the front-end subsystem, like all of us, has both the yin and the yang which makes it that much interesting. It is clearly evident that through correct engineering practices like developing systems which are recyclable and dependent on renewable energies, we can definitely improve the sustainability and reliability of the front-end systems and all other systems in general. Also, by having more local citizens enganged in development and having more locally manufactured products, we can help better the social, cultural and economic aspects of the society at large. Finally, we have to strike a commanding balance between privacy and dependency to manage risks within the system.

Analysis: Callum Tilbury – Back-end Subsystem

In the modern technological era—an epoch of artificial intelligence, blisteringly fast internet, and cloud computing—it is easy to be a blissfully ignorant consumer. In fact, this is perhaps exactly what the big technology companies desire: to create a black box that is wonderfully convenient, and sell it for massive profits. If you asked someone how Google always shows you the right things, what Facebook does with your selfies, or where Amazon stores all of its data, you may be disappointed with the feeble response. And yet, these questions matter: they have an undeniable impact on the user, the environment, the economy, and society as a whole. This is why there is a responsibility for engineers and scientists alike to be transparent with their creations: we cannot simply abuse the fact that many people do not understand the mystical world of technology.

Indeed, this project, too, has an impact on society. While such an impact is far more localised, it is nevertheless real and it must be made clear. It is important to interrogate critically every atomic solution, and ensure that each one aligns with an overarching vision of a better future for all. It is certainly no use for the system to help some people—*e.g.* bike drivers—and yet destroy the livelihoods of many others. However, the considerations are not guaranteed to be clear, and it is vital to notice the nuances of the problem. Few solutions exist spatiotemporally in isolation; technology is evolving constantly, and the complex interplay between new and old innovations must be taken into account.

What, then, can one say about the impact of a back-end subsystem for a bike driver safety application? The subsystem is made up of several key parts, which includes an uninterruptible power supply, an operator's computer, two Redundant Arrays of Inexpensive Disks (RAIDs), and a high-uptime web server. The former two components are fairly simple to deal with: computers and power supplies undoubtedly have strong influences on society and noticeable impacts on the environment, but they form an essential backbone to the overall subsystem and project. To exist, they both require resources—be it ecological, intellectual, economic—and these should not be ignored. However, there is little ethical room in which to move when considering these components. Of course, we must aim—on a broad level—to reduce the footprint of our computers and power supplies, but for now, our choice is limited. For example, while the operation of a power supply requires electrical energy, an innovative *digital* solution for improving the safety of bike drivers cannot exist without the said power supply—it is a necessary evil, perhaps.

On the other hand, the latter two of the mentioned components—the RAID drives and the web server—present a unique problem. Whereas the computer and power supply form a backbone to the project, these two are arguably not entirely necessary. Through some creative work, an alternative solution would likely exist without these two components. This raises the question: do they have a net positive impact? And thus, are they justifiable inclusions?

Hard drive technology is improving rapidly: simultaneously becoming faster, more reliable, and more 'green', largely thanks to progress made with solid-state drive (SSD) technology [2]. An SSD may not be an environmentally *perfect* candidate for digital data storage, but compared to an old hard-disk drive, it actually presents a more sustainable solution [3]. Though once prohibitively expensive, these drives are now much cheaper, and are quickly becoming the preferred choice for many applications [4] [5]. However, these factors do not guarantee that the RAID arrays in this subsystem are necessarily good for society—there are many other factors involved. Even a modern hard drive has a finite lifespan [6], and through its creation, technological waste is generated. Granted, much of this can be recycled, but that, too, will consume energy [7]. The problem is deeper than that, though. Socially, the advancement in data storage technology ushers in a new fetish for data *collection* [8] [9]. Is this a good thing? In this project, some of the collected data is designed to remain in perpetuity on the back-up storage media. The purpose of this is to assist in the safety of the bike drivers, but is it appropriate?—how do the drivers feel? What if, for example, insurance companies solicited the accident records from a company to raise the premiums of a particular bike driver involved in several incidents? Of course, both companies would be acting unethically, but the drivers involved certainly deserve a say in where and how their data is stored [10]. How will a company act if a driver refuses for their data to be collected? These things need to be understood first, before any decisions are made.

Now consider the system requirement of high uptime. The reason for this is cited as *reliability*—it is vitally important that if/when something goes wrong, the system does not fail. These specifications necessitate spending additional money and energy on web servers which are permanently connected, and highly dependable. While such considerations come from a good ethical justification—out of interest for the often-vulnerable bike drivers—perpetual server connection has an undeniable environmental impact [11]. Notably, renewable energy sources often cannot supply electricity around-the-clock [12], and fossil fuel sources have to be used instead. Moreover, the mere energy cost for running and cooling these servers is frankly enormous [13], thus adding to the dire ecological condition of the planet. Is the benefit of extremely high reliability—a proxy for driver safety—sufficient to justify the ecological damage of the aforementioned web servers? Are we ignoring the creative solutions by jumping to the easy, high-bandwidth, high-energy implementations which we call 'innovative' [14]?

In the same way that engineering has a close relationship with the physical sciences, it too has an intimate—often forgotten—relationship with the social sciences. This important fact cannot be overlooked. The impact of our technological innovations is a complex interplay of a myriad of factors. This is not to say that we should never try innovate, but when we do, we must not accept things at face value—we must also interrogate our justifications, our biases, our blind spots.

Analysis: Moeketsi Mpooa

Communication and Navigation

The communication and navigation systems are one of the most important subsystem in the whole design process. If badly designed the design will not work as indented. Also, the security of the whole systems will be compromised if this part is not secure enough. In other for this subsystem to be sustainable, it should respond fast enough with accuracy and reliability to not miss any danger on the driver's life. The subsystem should also have the lower running cost as it has capacity to increase the running cost of the design tremendously. The system should also be secure.

Reducing running cost
In other to reduce running cost in the subsystem, the information running through the ISP is made as minimum as possible. This is achieved by sending low data byte over HTTP. This is only the data enough to identity drivers in danger, their location and their identification, allowing the users to send the least data as possible of a few bytes per second. Which is enough to maintain the security of the users while cutting the operation cost greatly. The data through the ISP is significantly lowered by connecting driver's front-end device and his/her phone with wireless local area network. Removing the need for ISP data connection. This allows the system to not add a significant running cost above what the drivers are already paying for themselves.

The subsystem should also be fast, accurate and reliable. In order to achieve this, the algorithm or protocol to be used must have the capabilities mentioned above. The TCP BBR was chosen as it offers these qualities. It offers more stable throughput and responds more aggressively to attempt to transmit more data as possible when there has been issues in transmission. Also, TCP is a reliable connection meaning data will not be transmitted altered. This feature is most needed when dealing with safety issues.

The information to be sent over this sub system is sensitive nature and is too personal. It can be seen that if the data reaches wrong people it may put the drivers in far worse danger than they are currently in. To minimize the risk secure connections has to be used between the communicating channels. To achieve this the HTTP and TCP algorithms will be used to ensure that only people who are authorized to receive the information have access to it. This also offers some sort of security over the data.

In this design sub system, the focus is also innovation to improving lives of the drivers' society. The human problems of safety are therefore turned around and solved by an engineering innovation that increases safety of the delivery bikes drivers.

This engineering solution through communication and navigation brings about collaboration of individuals working together to solve issues that affect all of them. They could possibly help each other out in ways that were not possible before as they can see the safety status of each other when they are at proximity. By having access to each other's safety information means they could also prevent crime as more people including the criminals will eventually know of the networking between the drivers leading to a more secure work environment.

One might be curious why bike drivers can only see each other's location and safety status only when they are within five-kilometer radius. The assessment with this regard is that it is not of significant use to a far way driver to see what's happening too far as there are way too many bike drivers. Seeing way too far could lead to drivers ignoring each other when there are problems due to frequency the alarm button may go off on their screens. Also, it is quite possible that if this technology could fall in the wrong hands it could be more dangerous than safe. It could possibly lead to criminals targeting the drivers as they could see everyone. Therefore, the limit on visibility on the drivers' side makes it harder to trace someone while making it easier to someone random to help if there are problems

References

[1] M. Opray, "Nickel mining: the hidden environmental cost of electric cars," *The Guardian*, Aug 2017. [Online]. Available: https://www.theguardian.com/sustainable-business/2017/aug/24/nickel-mining-hidden-environmental-cost-electric-cars-batteries

[2] S.-H. Kim, S.-J. Lee, S.-H. Kim, S.-E. Kang, D. S. Lee, and S.-R. Lim, "Environmental effects of the technology transformation from hard-disk to solid-state drives from resource depletion and toxicity management perspectives," *Integrated Environmental Assessment and Management*, vol. 15, no. 2, pp. 292–298, feb 2019.

[3] M. Rouse. [Online]. Available: https://searchstorage.techtarget.com/definition/SSD-RAID-solid-state-drive-RAID

[4] V. Kasavajhala, "Solid state drive vs. hard disk drive price and performance study," *Proc. Dell Tech. White Paper*, pp. 8–9, 2011.

[5] G. Wong, "SSD market overview," in *Inside Solid State Drives (SSDs)*. Springer Netherlands, aug 2012, pp. 1–17.

[6] B. Schroeder, R. Lagisetty, and A. Merchant, "Flash reliability in production: The expected and the unexpected," in *14th USENIX Conference on File and Storage Technologies (FAST 16)*. Santa Clara, CA: USENIX Association, Feb. 2016, pp. 67–80. [Online]. Available: https://www.usenix.org/conference/fast16/technical-sessions/presentation/schroeder

[7] A. Micks. (2012, Dec.). [Online]. Available: http://large.stanford.edu/courses/2012/ph240/micks2/

[8] D. Kifer and A. Machanavajjhala, "No free lunch in data privacy," in *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, 2011, pp. 193–204.

[9] D. Fletcher, "How facebook is redefining privacy," 2010.

[10] Y. N. Harari. [Online]. Available: https://www.npr.org/2018/11/02/662619612/yuval-noah-harari-could-big-data-destroy-liberal-democracy

[11] R. Matthews. [Online]. Available: https://earthmaven.io/planetwatch/energy-economics/how-environmentally-sustainable-is-cloud-computing-and-storage-QF6qx7l9L0-e4uh7aE39sw

[12] K. Thoubboron. [Online]. Available: https://news.energysage.com/advantages-and-disadvantages-of-renewable-energy/

[13] A. Taylor. [Online]. Available: https://www.coronatimes.net/going-digital-not-as-green-covid-19/?fbclid=IwAR2qLi_bF4yD0MyWICqRyF7ivdSBUsA4htLvZVgAmaQctWuFVQFqQAKSmt8

[14] N. Shirazi and A. Johnson. [Online]. Available: https://medium.com/@CitationsPodcst/episode-110-the-shiny-object-psychology-of-american-capitalist-innovation-c0dc39af81c9