

**Department of Computer Science and Engineering
The University of Texas at Arlington**

Team: Auto-Climatix

Project: Automotive Climate Controller

Team Members:

Raju Karki

Robert Lopez

Bishal Shrestha

Hai Nguyen

Anthony Vecera

Table of Contents

Table of Contents.....	2
Document Revision History	3
List of Figures.....	4
List of Tables	5
1. Product Concept	6
1.1 Purpose and Use.....	6
1.2 Intended Audience	7
2. Product Description and Functional Overview	8
2.1. Features and Functions.....	8
2.2. External Inputs and Outputs.....	8
2.3. Product Interfaces	8
3. Customer Requirements	10
4. Microcontroller Requirements	15
5. Packaging Requirements	18
6. Connectivity Requirements	21
7. Performance Requirements	27
8. Safety Requirements.....	30
9. Maintenance and Support Requirements.....	32
10. Other Requirements	35
11. Acceptance Criteria	37
12. Use Cases.....	40

Document Revision History

Revision Number	Revision Date	Description	Rationale
0	3/19/12	First Draft Completion	First Draft Completion
1	4/03/12	Revised Draft Completion	Revised Draft Completion
2	4/12/2012	Final Draft Completion	Final Draft Completion

List of Figures

Figure 1: System Concept.....6

Figure 2: Automatic Mode GUI Mockup9

Figure 3: Manual Mode GUI Mockup9

Figure 4: Debug Mode GUI Mockup9

Figure 5: Individual Component Debug9

Figure 7: Automatic Mode Use Cases45

Figure 8: Manual Mode Use Cases.....46

Figure 9: Debugging Mode Use Cases47

List of Tables

Table 1: Function Point Analysis50

Table 2: Size Estimation Summary52

1. Product Concept

This section describes the purpose, use and intended user audience for the Automotive Climate Controller. Automotive Climate Controller is the system that performs the function to detect the current climate condition inside and outside of a vehicle so that it can allow the user to manually or automatically adjust available hardware devices of the Air Conditioning system of the vehicle. Users of the Automotive Climate Controller will be able to take advantage of the system by using their portable device to control the Air Conditioning system in their vehicle. The old, traditional AC system with inconvenient buttons will be replaced by user- friendly and convenient, Automotive Climate Control System.



Figure 1: System Concept

1.1 Purpose and Use

Mobile devices are developing rapidly and have a crucial role in our society. They contribute to the process of economic development and raise quality of life. People are extremely familiar with these

devices and are now willing to use them to aid in their own businesses. Automotive Climate Controller is a man-in-the-middle system meant to modernize the air conditioning systems of older vehicles by combining the older system with a mobile device. The aim of this marriage of old and new technology is to provide users of the Automotive Climate Controller the most convenient and comfortable solution for delivering luxurious climate control.

Auto-Climatix will give drivers of older vehicles another method to manually or automatically control the air conditioning system in their vehicle by interfacing the existing system with a mobile device. Users can choose the automatic mode of the system, delivering a classic automatic climate controlled environment and eliminating for user input to achieve desired comfort. To perform this function, the system processes information from sensors attached to the vehicle through a microcontroller. The microcontroller is heart of the system, which will be introduced in more detail later in this document.

The Automotive Climate Controller is used whenever the user desires climate control in their vehicle. The system will be compatible with any mobile device that provides root access and implements a common mobile OS. The device should have the ACC program installed and running. It should be connected to the microcontroller.

1.2 Intended Audience

People who own a mobile device and a vehicle can take advantages of the Automotive Climate Controller. A large portion of the US adult population owns a mobile device and similarly own vehicles. With this in mind, people of any socioeconomic class may use the Auto-Climatix Climate Controller and expect to see an improvement in their vehicle's climate controller's performance. However, the target audience is actually those with older vehicles seeking to retrofit their A/C systems to include proper climate controls without compromising styling. Given the abundance of vehicles without luxury climate controls and the lack of current affordable solutions for the problem, the product could establish itself as a viable solution or implementation of A/C controls for vehicle manufacturers.

2. Product Description and Functional Overview

This section provides the reader with an overview of the Automotive Climate Controller. The primary operational aspects of the product include the manipulation of the temperature of the interior of older vehicles. The initial implementation will focus largely on an early model 1980's E-Class Mercedes Benz. Additionally, the system shall be capable of obtaining diagnostic information about vehicle components that are connected to the Automotive Climate Controller. End users will operate the vehicle's AC using the mobile device. Maintenance will be performed through necessary hardware and software upgrades.

As mentioned before, the key feature of this system is the ability to manipulate the temperature of the cabin of the vehicle the system is integrated into. This is accomplished by connecting various portions of the old climate control system to the Automotive Climate Controller's microcontroller unit. From there, the microcontroller will send its signals to the various portions, adjusting the portions as needed to bring the temperature of the cabin of the car to the temperature that was selected by the end user through his or her interactions with the touch-screen interface. The ability to obtain diagnostic information about vehicle components will be acquired in a similar manner as the adjusting of the temperature of the cabin.

2.1. Features and Functions

The product will allow users to control the climate control system of the early model 1980's E-Class Mercedes Benz car through a mobile device. This mobile device will be connected to the microcontroller of the product.

The product will have three modes of operation. The modes of operation are the automatic mode, the manual mode, and the debug mode. The automatic mode will allow users to specify a temperature into the system in order to have the car automatically attempt to adjust the temperature in the car. The manual mode will allow users to control the disparate portions of the car's climate control system through the mobile device. The debug mode will allow users to verify the status of all of the components of the system that are connected to the microcontroller of the system.

2.2. External Inputs and Outputs

The product requires user input from the touch-screen interface. The product requires connections to the microcontroller from the various portions of the old climate control system. All portions of the system require power. The product should adjust the temperature of the cabin of the car the system is installed in.

2.3. Product Interfaces

The product will have a touch-screen interface. On the interface there will be a picture of the car, with indicators for the status of the various vents. The temperature of the car will be displayed as well.

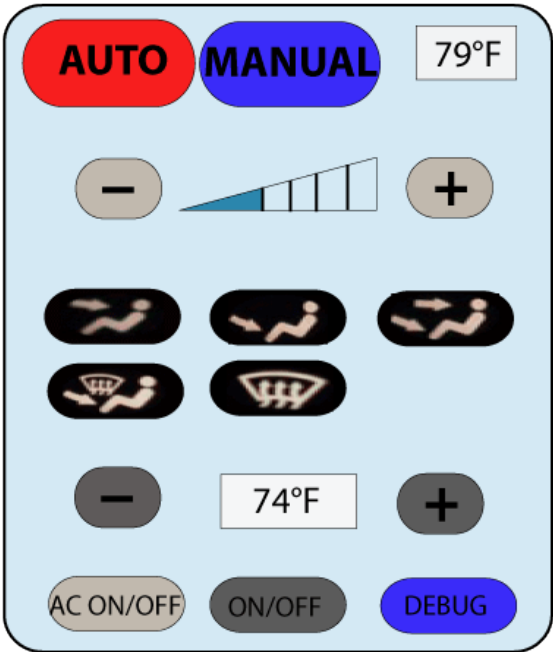


Figure 2: Automatic Mode GUI Mockup

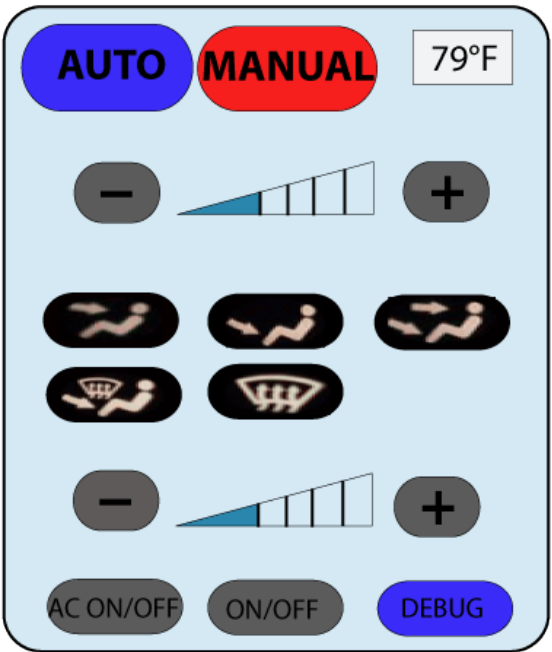


Figure 3: Manual Mode GUI Mockup

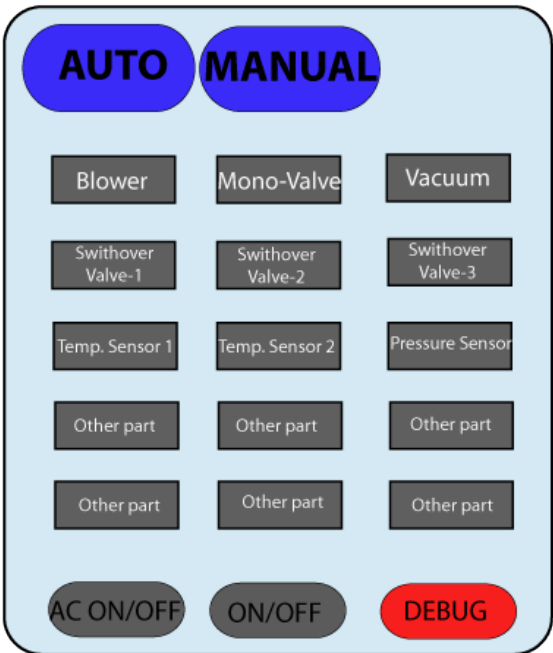


Figure 4: Debug Mode GUI Mockup

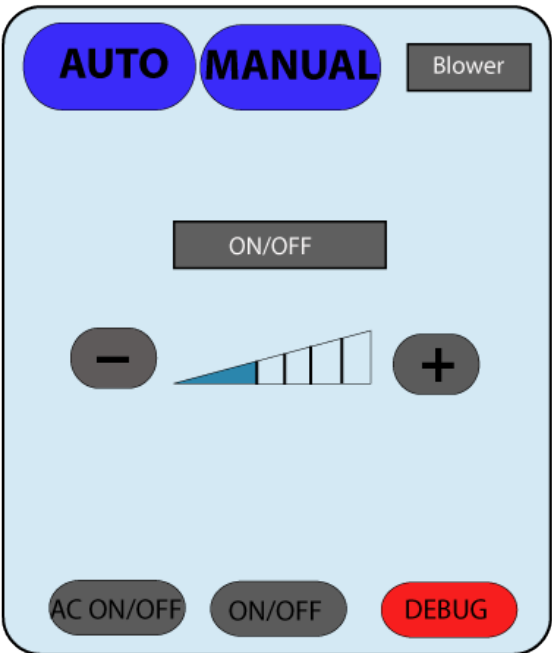


Figure 5: Individual Component Debug

3. Customer Requirements

Customer requirements describe the basic functionalities, usability, and appearance of the product as required by the customer. Team Auto-Climatix gathered the following requirements in collaboration with project sponsor, Dr. Gergely Zaruba. These requirements define the core functionalities of the Automotive Climate Controller as specified by the project sponsor and Team Auto Climatix. These requirements specify what each end-user should expect the product to do and/or not do, as well as the “look and feel” of the system. It should be noted that these customer requirements are end-user display specific as the customer is primarily concerned with the system’s output. Requirements in this section must not be changed without specific agreement with the project sponsor.

3.1. The system shall use a mobile device.

3.1.1. Description: The system shall use a mobile device. The mobile device shall have a capacitive touch screen. The mobile device shall have at least a single button.

3.1.2. Source: Dr. Gergely Zaruba

3.1.3. Constraints: The mobile device shall utilize a common mobile OS having a wide release no older than 2009. The operating system of the mobile device shall not prohibit access to root controls. The mobile device must not exceed budgetary constraints.

3.1.4. Standards: (*IOS and Android Programming Standards*) *Android 2.X, IOS 3.X*

3.1.5. Priority: Critical

3.2. The mobile device shall be connected to the microcontroller.

3.2.1. Description: The mobile device shall always be connected to the microcontroller wirelessly. The application shall communicate with the microcontroller.

3.2.2. Source: Dr. Gergely Zaruba

3.2.3. Constraints: None

3.2.4. Standards: TIA-232-F, Bluetooth v1.0-3.0

3.2.5. Priority: Critical

3.3. The mobile application shall have a standard wake-up procedure.

3.3.1. Description: The mobile device shall wake up when the key is moved from the off position to the ACC/start position. The application shall initiate previous user settings. If no user setting is available, the system shall assume default settings.

3.3.2. Source: Dr. Gergely Zaruba

3.3.3. Constraints: In the ACC position, the compressor doesn't turn on. So, the A/C doesn't work. Only the airflow and the fan speed can be controlled in ACC position.

3.3.4. Standards: None

3.3.5. Priority: High

3.4. The mobile application shall have a sleep mode.

3.4.1. Description: Sleep mode shall disable all system functions except for power delivery to the mobile device. The mobile device shall go to sleep when the vehicle is turned off. The application shall allow the user to set the application to sleep mode from any screen.

3.4.2. Source: Dr. Gergely Zaruba

3.4.3. Constraints: In the ACC position, the compressor doesn't turn on. So, the A/C doesn't work. Only the airflow and the fan speed can be controlled in ACC position.

3.4.4. Standards: None

3.4.5. Priority: High

3.5. The mobile application shall wake up when the user touches the screen.

3.5.1. Description: The mobile application shall wake up when the user provides system recognizable input.

3.5.2. Source: Dr. Gergely Zaruba

3.5.3. Constraints: The mobile application shall be in sleep mode.

3.5.4. Standards: None

3.5.5. Priority: High

3.6. The system shall have an automatic mode.

3.6.1. Description: The automatic mode shall manipulate cabin temperature to the user's desired temperature. This mode shall calculate fan speed, heating and cooling output and vent settings using input from the system temperature. The system shall go to the manual mode when the user modifies anything from the automatic mode setting other than the temperature setting. The system shall escape to manual mode when the user modifies anything from the automatic mode setting other than the temperature setting.

3.6.2. Source: Dr. Gergely Zaruba

3.6.3. Constraints: None.

3.6.4. Standards: None

3.6.5. Priority: Critical

3.7. The system shall have manual mode.

3.7.1. Description: The application shall implement a manual mode that shall allow the user to manually select a preset AC output temperature, select a preset air speed and determine vent settings. The user shall be able to disable or enable the AC manually.

3.7.2. Source: Dr. Gergely Zaruba

3.7.3. Constraints: The engine of the automobile should be on in order for the application to be able to turn the AC on.

3.7.4. Standards: None

3.7.5. Priority: Critical

3.8. The application shall have a debugging mode.

3.8.1. Description: The debugging mode shall show information of all vehicle hardware input to the system. The debugging mode shall allow the user to manually adjust individual vehicle hardware components.

3.8.2. Source: Dr. Gergely Zaruba

3.8.3. Constraints: Some vehicle components are interdependent and can't be individually manipulated.

3.8.4. Standards: None

3.8.5. Priority: Critical

3.9. The debugging mode shall show the problems that the system is able to identify.

3.9.1. Description: The debugging mode shall perform system diagnostics and display a short message about the problem that the system was able to identify.

3.9.2. Source: Dr. Gergely Zaruba

3.9.3. Constraints: None

3.9.4. Standards: None

3.9.5. Priority: Future

3.10. The mobile application shall generate a log file of the system.

3.10.1. Description: The mobile application shall generate a log file of the system that records the system events like the installation of some inputs devices, output devices, system startup, shutdown and also the hardware failure information. The user shall be able to download the log file.

3.10.2. Source: Dr. Gergely Zaruba

3.10.3. Constraints: Storage capacity of the mobile device.

3.10.4. Standards: None

3.10.5. Priority: Future

3.11. The application shall have a GUI that is clearly visible.

3.11.1. Description: The GUI of the application shall have visible components such as buttons, scrolls, labels, etc. that are visible to the people with in corrective vision from a distance of 4 feet.

3.11.2. Source: Dr. Gergely Zaruba

3.11.3. Constraints: None

3.11.4. Standards: None

3.11.5. Priority: Critical

3.12. The application GUI shall have meaningful icons and symbols.

3.12.1. Description: The application GUI shall have icons and symbols related to the functionality of the icon/symbols and following the standards.

3.12.2. Source: Dr. Gergely Zaruba

3.12.3. Constraints: None

3.12.4. Standards: None

3.12.5. Priority: High.

3.13. The application GUI shall recognize the straight-line gesture as the input.

3.13.1. Description: The applications GUI shall recognize the straight-line gesture as the input like swipe, slide, push and hold. It shall not have complicated gestures like panning and rotation, etc.

3.13.2. Source: Dr. Gergely Zaruba

3.13.3. Constraints: None

3.13.4. Standards: None

3.13.5. Priority: High.

3.14. The application shall display a message in case of system failure.

3.14.1. Description: The application shall detect the failure of the system and display the application failure information if the system is not working at all.

3.14.2. Source: Dr. Gergely Zaruba

3.14.3. Constraints: None

3.14.4. Standards: None

3.14.5. Priority: High

4. Microcontroller Requirements

Microcontrollers are designed by Auto-Climatix team for embedded applications that will work as a small computer on a single integrated circuit. It is will be used to control the a/c unit in the car. The microcontroller will contain processor core, memory, and programmable input/output peripherals. The basic functionalities of microcontroller will be described below at the requirement section.

Requirements in this section must not be changed without specific agreement with the project sponsor.

4.1. The system shall have a microcontroller unit.

4.1.1. Description: The system shall have a microcontroller unit with 40 I/O pins. The pins shall be software configurable to either an input or output state. When these pins are configured to an input state, they shall read sensors or external signals. When these pins are configured to the output state, they shall drive external devices such as LEDs or motors.

4.1.2. Source: Dr. Gergely Zaruba

4.1.3. Constraints: None

4.1.4. Standards: None

4.1.5. Priority: High

4.2. The microcontroller shall provide real time response.

4.2.1. Description: The microcontroller shall provide real time response to events in the system. Certain events or user inputs shall trigger an interrupt system, signaling the processor to suspend the current instruction sequence and to begin a new service routine. This new service routine shall perform any processing required based on the source of the interrupt before returning to the original instruction sequence. Possible interrupt source shall be an internal timer overflow, completing an analog to digital conversion, a logic level change on an input such as from a button being pressed, and data received on a communication link.

4.2.2. Source: Dr. Gergely Zaruba

4.2.3. Constraints: Interrupts may also wake a microcontroller from a low power sleep state where the processor is halted until required to do something by a peripheral event.

4.2.4. Standards: None

4.2.5. Priority: Critical

4.3. The microcontroller shall be programmable.

4.3.1. Description: The microcontroller shall be programmable. The program shall fit in on-chip program memory. The program shall be field-alterable flash or erasable read-only memory. The field-alterable flash shall allow field updating of the firmware.

4.3.2. Source: Dr. Gergely Zaruba

4.3.3. Constraints: None

4.3.4. Standards: None

4.3.5. Priority: Critical

4.4. The microcontroller shall include a timer.

4.4.1. Description: Microcontroller shall include a variety of timers as well like Programmable Interval Timer. The timer shall either count down from some value to zero, or up to the capacity of the count register, overflowing to zero. The timer shall be useful for devices such as thermostats.

4.4.2. Source: Dr. Gergely Zaruba

4.4.3. Constraints: None

4.4.4. Standards: None

4.4.5. Priority: High

4.5. The microcontroller shall use a high level programming language.

4.5.1. Description: The microcontroller shall use high-level programming languages such as the C programming language. The compiler shall make tools freely available to make it easier to adopt their hardware.

4.5.2. Source: Dr. Gergely Zaruba

4.5.3. Constraints: The developed program may not adopt the entire hardware environment.

4.5.4. Standards: None

4.5.5. Priority: High

4.6. The microcontroller shall use simulators.

4.6.1. Description: Microcontroller shall use simulators to analyze what the behavior of the microcontroller and their program should be (as they were using the actual part). The simulator shall show the internal processor state and also that of the outputs, as well as allowing input signals to be generated.

4.6.2. Source: Dr. Gergely Zaruba

4.6.3. Constraints: None

4.6.4. Standards: None

4.6.5. Priority: High

5. Packaging Requirements

These packaging requirements will determine how the finished Automotive Climate Controller will look and feel prior to delivery for the users. The Automotive Climate Controller needs two separated parts: the software on mobile devices and a microcontroller. Other devices that are required are enclosed in the package.

5.1. The microcontroller hardware shall be fully assembled.

5.1.1. **Description:** All the parts of the system shall be connected to the microcontroller.

5.1.2. **Source:** Dr. Gergely Zaruba

5.1.3. **Constraints:** May need to connect to A/C system of a vehicle to boot it up.

5.1.4. **Standards:** None

5.1.5. **Priority:** High

5.2. The firmware shall be preloaded onto the microcontroller.

5.2.1. **Description:** The firmware that is integrated in a microcontroller shall be automatically loaded by the microcontroller whenever the system boots up.

5.2.2. **Source:** Dr. Gergely Zaruba

5.2.3. **Constraints:** The microcontroller does not have an individual battery, and system boots up when the engine starts.

5.2.4. **Standards:** None

5.2.5. **Priority:** High

5.3. The Automotive Climate Controller application shall be pre-installed onto the mobile device.

5.3.1. **Description:** The Automotive Climate Controller application shall be installable, updateable and a newest version that can run on platforms of common mobile operating systems. Users can request for a new version or download a free new version via Internet.

5.3.2. **Source:** Team Auto-Climatix

5.3.3. Constraints: A software CD must be enclosed in the Automotive Climate Controller package.

5.3.4. Standards: None

5.3.5. Priority: High

5.4. The Automotive Climate Controller shall be a single package.

5.4.1. Description: The microcontroller and mobile device shall be delivered in a single package. The package shall include a user manual showing product installation on an early model 1980's Mercedes.

5.4.2. Source: Team Auto-Climatix

5.4.3. Constraints: The manual shall only provide instruction for connection for the specified vehicle. The manual may not be indicative of all vehicle configurations.

5.4.4. Standards: None

5.4.5. Priority: High

5.5. The Automotive Climate Controller shall be installable and configurable by an end user with fair mechanical knowledge.

5.5.1. Description: The system should be installed by following the steps as described in the user's manual of the system. The process will first involve ripping open the chassis of the vehicle that the system is to be installed on. Then, the original air conditioning system is to be disconnected from the vehicle. The Automotive Climate Controller will then be connected to the vehicle's air conditioning components as described in the user's manual. Then, the Automotive Climate Controller will then be mounted to the vehicle in the manner as described in the user's manual of the system. Finally, the mobile device will be affixed to the dashboard of the vehicle as described by the user's manual of the system.

5.5.2. Source: Team Auto-Climatix

5.5.3. Constraints: The manual shall only provide instruction for connection for the specified vehicle. The manual may not be indicative of all vehicle configurations.

5.5.4. Standards: None

5.5.5. Priority: High

5.6. The Automotive Climate Controller shall come with the necessary components to provide power to the system.

5.6.1. Description: The system shall come with the various cables to be used to connect the microcontroller of the system with the car battery of the vehicle of which the system is to be installed on. The microcontroller will receive power from the car battery of the vehicle. The microcontroller will then provide power to all components that are not given power by other vehicle systems. These steps will be explained to the user through the user's manual of the system.

5.6.2. Source: Team Auto-Climatix

5.6.3. Constraints: The manual shall only provide instruction for connection for the specified vehicle. The manual may not be indicative of all vehicle configurations.

5.6.4. Standards: None

5.6.5. Priority: High

6. Connectivity Requirements

Automotive climate control systems are not universal among vehicles. It is beyond the scope of the project to make a truly universal controller for all automobiles. The following requirements are meant to address the immediate requirements for implementation in the climate controlling system specified by the sponsor. The requirements directly address the communications between all devices connected to the microcontroller either wired or wirelessly.

6.1. The mobile device and microcontroller shall actively communicate.

6.1.1. Description: The mobile device shall transmit data to the microcontroller at least once per second. The microcontroller shall transmit system status data to the mobile device at least once per second.

6.1.2. Source: Team Auto-Climatix

6.1.3. Constraints: None

6.1.4. Standards: None

6.1.5. Priority: Critical

6.2. The microcontroller shall supply power to the mobile device.

6.2.1. Description: The microcontroller shall constantly supply 5v to the mobile device. The microcontroller shall not output in excess of 5v to the mobile device. The microcontroller shall persist in delivering power to the mobile device after vehicle shut-off. The mobile device shall terminate displays upon vehicle shutoff. The mobile device shall resume data displays upon vehicle start-up.

6.2.2. Source: Team Auto-Climatix

6.2.3. Constraints: The vehicle battery may be incapable of supplying needed power to the microcontroller, resulting in output below 5V.

6.2.4. Standards: TIA-232-F

6.2.5. Priority: High

6.3. The microcontroller shall supply power to the components that require minimal current.

6.3.1. Description: The microcontroller shall supply power to the switchover valves. The microcontroller shall supply power to the mono- valve. The microcontroller shall supply current to the temperature sensors. The microcontroller shall supply current to the compressor clutch. Components requiring minimal current are those components which do not require persistent current or require less equal current to the mono-valve.

6.3.2. Source: Team Auto-Climatix

6.3.3. Constraints: The vehicle battery may be incapable of supply needed power to the microcontroller, resulting in inadequate power supply being delivered to the vehicle hardware components.

6.3.4. Standards: None

6.3.5. Priority: High

6.4. The vehicle shall supply power to the components that require significant amperage.

6.4.1. Description: The vehicle battery shall supply power to the microcontroller. The vehicle battery shall supply power to the blower motor. The vehicle battery shall supply power to the vacuum source. The microcontroller shall be capable of monitoring vehicle battery status. The microcontroller shall be capable of adjusting the power supply of the battery to the blower motor. The power supply to the vacuum source shall be triggered by the microcontroller. Significant amperage shall be defined as that which exceeds or meets the power supply of the blower motor.

6.4.2. Source: Team Auto-Climatix

6.4.3. Constraints: The vehicle battery may be incapable of supplying sufficient power to the vehicle hardware components.

6.4.4. Standards: None

6.4.5. Priority: High

6.5. The application shall control the vehicle hardware that is interfaced with the system.

6.5.1. Description: The microcontroller shall be capable of accepting and delivering analog and digital signals. The microcontroller shall be capable of delivering output to and receiving input from at least twelve vehicle hardware components.

6.5.2. Source: Team Auto-Climatix

6.5.3. Constraints: The vehicle battery may be incapable of supplying sufficient power to the microcontroller, resulting in altered control signals.

6.5.4. Standards: None

6.5.5. Priority: Critical

6.6. The system shall control switchover valves.

6.6.1. Description: The system shall be capable of controlling a dual mode switchover valve. The microcontroller shall be capable of determining the current status of the switchover valve. The microcontroller shall be capable of determining if a fault has occurred at the switchover valve.

6.6.2. Source: Team Auto-Climatix

6.6.3. Constraints: The vehicle battery may be incapable of supplying sufficient power to the microcontroller, resulting in altered control signals. A hardware malfunction at the microcontroller may yield faulty status conditions for the switchover valves.

6.6.4. Standards: None

6.6.5. Priority: Critical

6.7. The system shall control the blower motor.

6.7.1. Description: The system shall be capable of controlling airflow magnitude by adjusting output of the blower motor. The system shall be capable of altering power delivered from the battery to the blower motor. The system shall be capable of monitoring current flowing to the blower motor.

6.7.2. Source: Team Auto-Climatix

6.7.3. Constraints: The vehicle battery may be incapable of supplying sufficient power to the microcontroller, resulting in altered control signals. A hardware malfunction at the microcontroller may yield faulty status conditions for the blower motor.

6.7.4. Standards: None

6.7.5. Priority: Critical

6.8. The system shall monitor the temperature sensors.

6.8.1. Description: The system shall accept input from resistance based temperature sensors. The system shall correlate temperature to resistance with a resolution of 1° F. The system shall persistently monitor temperature sensor status while the vehicle is running.

6.8.2. Source: Team Auto-Climatix

6.8.3. Constraints: The vehicle battery may be incapable of supplying sufficient power to the microcontroller, resulting in altered control signals. A hardware malfunction at the microcontroller may yield faulty status conditions for temperature. Oxidation of the temperature sensor control wires may result in increased resistance values.

6.8.4. Standards: None

6.8.5. Priority: Critical

6.9. The system shall control the vehicle's mono-valve.

6.9.1. Description: The system shall deliver a PWM signal to the mono-valve. The system shall persistently monitor mono-valve status while the vehicle is running.

6.9.2. Source: Team Auto-Climatix

6.9.3. Constraints: The vehicle battery may be incapable of supplying sufficient power to the microcontroller, resulting in altered control signals. A hardware malfunction at the microcontroller may yield faulty status conditions for the mono-valve. Mono-valve diaphragm failure may result in unexpected mono-valve behavior.

6.9.4. Standards: None

6.9.5. Priority: Critical

6.10. The system shall control the vacuum source of the system.

6.10.1. Description: The system shall control the vacuum source. The system shall trigger power delivery from the battery to the vacuum source. The system shall monitor power input to the vacuum while the vacuum motor is active.

6.10.2. Source: Team Auto-Climatix

6.10.3. Constraints: The vehicle battery may be incapable of supplying sufficient power to the microcontroller, resulting in altered control signals. The vehicle battery may be incapable of delivering sufficient power to the vacuum source, yielding insufficient vacuum. A hardware malfunction at the microcontroller may yield faulty status conditions for the vacuum. Vacuum line failure may result in unexpected vacuum behavior.

6.10.4. Standards: None

6.10.5. Priority: High

6.11. The system shall control the compressor clutch of the system.

6.11.1. Description: The system shall control the compressor clutch. The system shall monitor the compressor clutch's status. The system shall trigger activation of the clutch when the user desires cooling capability from the AC.

6.11.2. Source: Team Auto-Climatix

6.11.3. Constraints: The vehicle battery may be incapable of supplying sufficient power to the microcontroller, resulting in altered control signals. A faulty pressure reading may result in unexpected compressor clutch behavior. High or low-pressure readings may prevent the compression clutch from engaging.

6.11.4. Standards: None

6.11.5. Priority: Critical

6.12. The system shall monitor the pressure sensors of the system.

6.12.1. Description: The system shall accept readings from vehicle pressure sensors. The system shall monitor vehicle pressure sensors persistently while the vehicle is on.

6.12.2. Source: Team Auto-Climatix

6.12.3. Constraints: The vehicle battery may be incapable of supplying sufficient power to the microcontroller, resulting in altered control signals. A hardware malfunction at the microcontroller may yield faulty status conditions for pressure sensors. This system must be implemented only with the compressor clutch.

6.12.4. Standards: None

6.12.5. Priority: Future

6.13. The system shall monitor the vehicle's power status.

6.13.1. Description: The system shall accept input differentiating between vehicle off, vehicle battery power, engine active, and engine idle.

6.13.2. Source: Team Auto-Climatix

6.13.3. Constraints: The vehicle battery may be incapable of supplying sufficient power to the microcontroller, resulting in altered control signals. A hardware malfunction at the microcontroller may yield faulty status conditions for pressure sensors.

6.13.4. Standards: None

6.13.5. Priority: Moderate

6.14. The system shall control the vehicle's water pump.

6.14.1. Description: The system shall trigger power supply from the vehicle battery to the vehicle water pump. The system shall initiate this action upon engine idling.

6.14.2. Source: Team Auto-Climatix

6.14.3. Constraints: The vehicle battery may be incapable of supplying sufficient power to the microcontroller, resulting in altered control signals. A faulty vehicle status line could yield unexpected behavior from the water pump.

6.14.4. Standards: None

6.14.5. Priority: Low

6.15. The system shall control the power seating.

6.15.1. Description: The system shall allow the user to manipulate electronic seat settings including forward-backward position, seat tilt and seat heating.

6.15.2. Source: Team Auto-Climatix

6.15.3. Constraints: The vehicle battery may be incapable of supplying sufficient power to the microcontroller, resulting in altered control signals. A faulty vehicle status line could yield unexpected behavior from the seat.

6.15.4. Standards: None

6.15.5. Priority: Future

7. Performance Requirements

Although the Automotive Climate Controller is not going to be engaged in critical operations, the product needs to get the temperature of the car cabin to a specified temperature within a reasonable time frame.

7.1. The system shall alter the temperature of the cabin to within a few degrees of the temperature that was specified by the user.

7.1.1. Description: The temperature output to the cabin of the vehicle should be within $\pm 2^{\circ}$ F degrees of what the target temperature of the system depicts.

7.1.2. Source: Team Auto-Climatix

7.1.3. Constraints: A hardware malfunction can give faulty data to the temperature sensors, resulting in improper actions taken to set the temperature. Engine and extreme outside conditions may prevent desired temperature output.

7.1.4. Standards: None

7.1.5. Priority: Moderate

7.2. The system shall achieve specified temperature on automatic mode promptly.

7.2.1. Description: The cabin should achieve the target temperature when the system is set to automatic mode in under ten minutes.

7.2.2. Source: Team Auto-Climatix

7.2.3. Constraints: A hardware malfunction can give faulty data to the temperature sensors, resulting in improper actions taken to set the temperature. Extreme conditions from the engine or outside may prevent the temperature from being achievable in ten minutes.

7.2.4. Standards: None

7.2.5. Priority: Moderate

7.3. The mobile application shall be responsive to the user.

7.3.1. Description: The mobile application shall respond to the identifiable gestures within 2 seconds of the action.

7.3.2. Source: Dr. Gergely Zaruba

7.3.3. Constraints: Processing speed of the mobile device.

7.3.4. Standards: None

7.3.5. Priority: High

7.4. The application shall not have memory leak problem.

7.4.1. Description: The application shall be built with proper garbage collection strategy in order to avoid memory leak problem.

7.4.2. Source: Team Auto Climatix

7.4.3. Constraints: The mobile platform we use might not have appropriate garbage collection functionality.

7.4.4. Standards: None

7.4.5. Priority: Moderate

7.5. The system shall be physically rugged.

7.5.1. Description: The system shall be able to withstand the oscillation rate of 10 Hertz.

7.5.2. Source: Team Auto Climatix

7.5.3. Constraints: Verification depends on the resources available to the team.

7.5.4. Standards: None

7.5.5. Priority: Moderate

7.6. The failure rate of the mobile application shall be less than a 1/10,000 uses.

7.6.1. Description: The system shall not fail more than once for 10,000 uses.

7.6.2. Source: Dr. Gergely Zaruba

7.6.3. Constraints: Vehicle hardware failure may be significantly less reliable than the Automotive Climate Controller. The mobile device may be delivered in a faulty condition from the factory. Hardware components may be delivered to team Auto-Climatix in a faulty condition.

7.6.4. Standards: None

7.6.5. Priority: Future

8. Safety Requirements

These safety requirements are meant to address user and technician safety during regular operation or installation. These requirements largely address damage prevention for both the vehicle and the microcontroller.

8.1. Electrical hardware shall implement fly back circuits.

8.1.1. Description: The fly back current shall be controlled in the system for connections requiring high amperage or voltage.

8.1.2. Source: Dr. Gergely Zaruba

8.1.3. Constraints: None.

8.1.4. Standards: None.

8.1.5. Priority: Critical

8.2. The microcontroller shall implement fuses.

8.2.1. Description: The micro controller board shall implement appropriate fuses.

8.2.2. Source: Dr. Gergely Zaruba

8.2.3. Constraints: None.

8.2.4. Standards: None

8.2.5. Priority: High

8.3. Electrical wiring shall be insulated.

8.3.1. Description: The wirings of the system shall be insulated properly, so as to prevent interference with and from other electronic devices within the vehicle.

8.3.2. Source: Team Auto-Climatix

8.3.3. Constraints: None.

8.3.4. Standards: None

8.3.5. Priority: Critical

8.4. The system shall warn users of extremely high temperature input.

8.4.1. Description: The system shall stay passive and display a warning to the user of extremely high temperature input.

8.4.2. Source: Team Auto-Climatix

8.4.3. Constraints: None.

8.4.4. Standards: None

8.4.5. Priority: High

8.5. The system shall warn users of extremely low temperature input.

8.5.1. Description: The system shall stay passive and display a warning to the user of extremely low temperature input

8.5.2. Source: Team Auto-Climatix

8.5.3. Constraints: None.

8.5.4. Standards: None

8.5.5. Priority: High

9. Maintenance and Support Requirements

The major maintenance areas of the Automotive Climate Controller address hardware failure and software failure. Team Auto-Climatix will bear no responsibility for the maintenance or support of vehicle air-conditioning components. The requirements contained below are meant to address ease of maintenance and support.

9.1. Team Auto-Climatix shall provide direct maintenance and support through August 2012.

9.1.1. Description: Team Auto-Climatix shall provide any maintenance or support requiring hardware replacement, hardware redesign and software updates or fixes through August 2012.

9.1.2. Source: Team Auto-Climatix

9.1.3. Constraints: None

9.1.4. Standards: None

9.1.5. Priority: Critical

9.2. All major hardware components used in the system shall be replaceable and not permanent.

9.2.1. Description: If any major components stops working due to any unforeseen circumstances like damaged parts, power surges, etc., then that component must be easily replaceable and not permanent as to ensure further operation of the system.

9.2.2. Source: Team Auto-Climatix

9.2.3. Constraints: None

9.2.4. Standards: None

9.2.5. Priority: Low

9.3. The system shall be delivered with all associated documents.

9.3.1. Description: The system shall be deliverable with all associated documents like System Requirements Specification, Architecture Design Specification, Detailed Design Specification, and the Project Charter.

9.3.2. Source: Team Auto-Climatix

9.3.3. Constraints: None

9.3.4. Standards: None

9.3.5. Priority: High

9.4. Team Auto-Climatix shall be available to fix any hardware and/or software issues.

9.4.1. Description: If any hardware or software issues exist in the system, team Auto-Climatix shall be available anytime to fix them. This guarantee shall remain valid through August 2012

9.4.2. Source: Team Auto-Climatix

9.4.3. Constraints: None

9.4.4. Standards: None

9.4.5. Priority: High

9.5. The system's software and hardware shall be tested frequently.

9.5.1. Description: The software use to communicate with the system's hardware and microcontroller shall be tested frequently to ensure it is functioning properly. Diagnoses screen shall be able to notify any identified irregularities, so that the problem can be fixed. Frequently is defined as at least once per function. All changes to working functions shall be tested immediately upon implementation.

9.5.2. Source: Team Auto-Climatix

9.5.3. Constraints: None

9.5.4. Standards: None

9.5.5. Priority: High

9.6. Team Auto-Climatix shall provide upgrades for any software issues.

9.6.1. Description: Team Auto-Climatix shall come with upgrade for major software issues until August 2012.

9.6.2. Source: Team Auto-Climatix

9.6.3. Constraints: None

9.6.4. Standards: None

9.6.5. Priority: Moderate

9.7. Team Auto-Climatix shall provide schematics and source code to customers after product support has expired.

9.7.1. Description: Team Auto-Climatix shall make available product documentation, source code and schematics readily available for a period of two years following August 2012. After this time period, all warranties and guarantees provided shall expire and the user shall maintain full responsibility for maintenance and support.

9.7.2. Source: Team Auto-Climatix

9.7.3. Constraints: None

9.7.4. Standards: None

9.7.5. Priority: Moderate

9.8. The debugging mode of the mobile application shall assist in detection of major problems in the system.

9.8.1. Description: The debugging mode shall assist the user to detect the major problems that arise during the operation of the system. The debugging mode shall allow the user to individually control each parts of the system and look for the problems.

9.8.2. Source: Team Auto-Climatix

9.8.3. Constraints: None

9.8.4. Standards: None

9.8.5. Priority: High

10. Other Requirements

This section contains requirements that can be followed to develop more functions of the product after the main requirements have been accomplished and in order to extending scope of the product. These requirements have lower priority than other requirements and they are only developed when all of other requirements are complete perfectly.

10.1. The application shall provide radio controls in the vehicle.

10.1.1. Description: The application shall be able to provide standard radio controls (Turn on, turn off, volume adjustment, FM band controls, AM band controls).

10.1.2. Source: Team Auto-Climatix

10.1.3. Constraints: The mobile device must support multi-programming. The radio functionality must not hinder any processing done to support the air-conditioning systems.

10.1.4. Standards: None

10.1.5. Priority: Future

10.2. The application shall provide audio output through the vehicle speakers.

10.2.1. Description: The application shall output audio cues through the vehicle's speakers.

10.2.2. Source: Team Auto-Climatix

10.2.3. Constraints: The mobile device must not interfere with existing audio-output.

10.2.4. Standards: None

10.2.5. Priority: Future

10.3. The mobile device shall be removable from the vehicle.

10.3.1. Description: The system shall allow for the mobile device to be secured by the user externally from the microcontroller.

10.3.2. Source: Team Auto-Climatix

10.3.3. Constraints: The system must immediately recognize and restore settings promptly upon the reintroduction of the mobile device to the system.

10.3.4. Standards: None

10.3.5. Priority: Future

10.4. The AC mockup shall simulate the standard functionality of an AC system.

10.4.1. Description: The mockup shall simulate the same functions of air conditioning system of a vehicle. It shall be a prototype to supply customers to simulate operation of the product in their vehicle before they decide to assemble the product for using.

10.4.2. Source: Team Auto-Climatix

10.4.3. Constraints: None

10.4.4. Standards: None

10.4.5. Priority: Critical

11. Acceptance Criteria

After several discussions with the primary stakeholders, it was agreed that the Automotive Climate Controller would be accepted as a completed product once the system being designed met the following criteria.

11.1. The system shall use a mobile device.

11.1.1. Requirement(s) addressed: 3.1

11.1.2. Verification Procedure: The sponsor shall verify that the device may be removed from the system and meets sponsor aesthetic standards.

11.2. The mobile device shall be connected to the microcontroller.

11.2.1. Requirement(s) addressed: 3.2

11.2.2. Verification Procedure: For wireless connectivity, Bluetooth recognition of the microcontroller shall be apparent on the mobile device's display. A visual inspection by the sponsor may verify the wireless connection or a hardwired connection.

11.3. The mobile application shall have an automatic mode.

11.3.1. Requirement(s) addressed: 3.6

11.3.2. Verification Procedure: When the user specifies desired temperature in the application, the environment of the cabin should change to the specified temperature within ten minutes. When the user presses any other control in the automatic mode, the application shall highlight the manual mode, which means the application went to the manual mode from the automatic mode.

11.4. The mobile application shall have a manual mode.

11.4.1. Requirement(s) addressed: 3.7

11.4.2. Verification Procedure: When the user specifies desired temperature level in the application, the environment of the cabin should change to the specified temperature within ten minutes. When the user adjusts the fan speed using the +/- button on the GUI, the fan speed should change accordingly. When the user picks the air direction from the presets, the respective vents and the air flaps should open/close respectively.

11.5. The application shall have a debugging mode.

11.5.1. Description: The debugging mode shall show information of all vehicle hardware input to the system. The debugging mode shall allow the user to manually adjust individual vehicle hardware components.

11.5.2. Requirement(s) addressed: 3.8

11.5.3. Verification Procedure: When the user touches the individual parts on the debug menu, it should show the information about that individual component. When the user picks the functionality that is specific to that hardware, that functionality should be performed by the respective hardware part.

11.6. The application shall have a meaningful GUI.

11.6.1. Requirement(s) addressed: 3.12

11.6.2. Verification Procedure: The sponsor shall approve of the GUI after a visual inspection. The sponsor shall verify that the GUI recognizes straight-line commands by interacting with the GUI sliders.

11.7. The microcontroller shall be programmable and shall provide real time response.

11.7.1. Requirement(s) addressed: 3.17, 3.18

11.7.2. Verification Procedure: The sponsor shall verify that the microcontroller exhibits real-time response by adjusting individual hardware components from the software ‘debug’ mode. The sponsor shall then approve or disapprove of the exhibited response time seen during a visual inspection. The response of the hardware shall exhibit that the microcontroller has been programmed to function within the system.

11.8. The microcontroller shall control the vehicle hardware interfaced with the system.

11.8.1. Requirement(s) addressed: 5.5-5.9

11.8.2. Verification Procedure: The sponsor shall verify that the system can control all interfaced vehicle hardware by adjusting individual component settings within the ‘debug’ GUI.

11.9. Electrical hardware shall implement fly back circuits for connections requiring high amperage or voltage.

11.9.1. Requirement(s) addressed: 7.1

11.9.2. Verification Procedure: The sponsor shall verify the existence of fly back circuits in circuit schematics and on the physical circuits. The sponsor shall verify implementation by demonstrating that the microcontroller does not overload after system shutoff.

11.10. Electrical wiring shall be insulated.

11.10.1. Requirement(s) addressed: 7.3

11.10.2. Verification Procedure: The sponsor shall verify that performing a visual inspection of the prototype installation insulates wiring.

12. Use Cases

The following section describes the use cases of the Automotive Climate Controller. These use cases cover all the major processes that the system will be designed to accomplish. Those tasks are categorized under four major modes that the system will be operating under. The first one is the general mode where the system is not in any particular mode and the system is expected to go to a particular mode and perform general operations like starting and shutting down the system. The next mode is the automatic mode where the system will only take the input from the user and adjust the temperature of the vehicle by taking the temperature input from the user. Manual mode is little more sophisticated than automatic mode in the sense that the user gets to pick more choices and the user has more control over the system. The last mode, meaningful GUI is the mode to check the system hardware and the system status. So, the followings are the individual tasks that the user of the system will be able to do. The description of those use cases is followed by the categorized visual representation of the use cases in form of use case diagram.

12.1. The user starts the system.

12.1.1. Scenario: The user wants to activate vehicle climate controls using the installed Automotive Climate Controller.

12.1.2. Actor(s): The user

12.1.3. THE USE CASE BEGINS WITH: The user switches the key position from off to ACC/Start position.

12.1.4. THE USE CASE ENDS WITH: The user sees the startup screen on the mobile device.

12.2. The user stops the system.

12.2.1. Scenario: The user wants to stop vehicle climate controls using the installed Automotive Climate Controller.

12.2.2. Actor(s): The user

12.2.3. THE USE CASE BEGINS WITH: The user switches the position of the key from ACC/start position to off position or presses the 'OFF' button in the application.

12.2.4. THE USE CASE ENDS WITH: The user sees the blank screen on the mobile device.

12.3. The user enables automatic climate control.

12.3.1. Scenario: The user wants to enable automatic climate control mode using the installed Automotive Climate Controller.

12.3.2. Actor(s): The user

12.3.3. THE USE CASE BEGINS WITH: The user touches the “Auto” button on the “Manual,” “Debug” or main page of the GUI.

12.3.4. THE USE CASE ENDS WITH: The user sees the “Auto” title highlighted on the title of the GUI.

12.4. The user enables manual mode.

12.4.1. Scenario: The user wants to enable manual control of the vents, fan and temperature range using the installed Automotive Climate Controller.

12.4.2. Actor(s): The user

12.4.3. THE USE CASE BEGINS WITH: The user touches the “Manual” button on the “Auto,” “Debug” or main page of the GUI or the user specifies manual settings in “auto” mode.

12.4.4. THE USE CASE ENDS WITH: The user sees the “Manual” title highlighted on the title of the GUI.

12.5. The user specifies desired cabin temperature.

12.5.1. Scenario: The user wants to set the cabin to a specific temperature using the installed Automotive Climate Controller.

12.5.2. Actor(s): The user

12.5.3. THE USE CASE BEGINS WITH: The user touches either the “+” or the “-” button in automatic mode to adjust the desired temperature.

12.5.4. THE USE CASE ENDS WITH: The user sees the temperature number change.

12.6. The user specifies magnitude of cooling or heating.

12.6.1. Scenario: The user wants to manually adjust the ‘hotness’ or ‘coolness’ of the cabin without specifying temperature using the installed Automotive Climate Controller.

12.6.2. Actor(s): The user

12.6.3. THE USE CASE BEGINS WITH: The user touches either the “+” or the “-” button in manual mode to adjust the desired temperature level that are in the form of fixed presets.

12.6.4. THE USE CASE ENDS WITH: The user sees the adjusted indicator on the “Air Temperature Range” indicator.

12.7. The user adjusts fan speed.

12.7.1. Scenario: The user wants to adjust the airflow output of the blower using the installed Automotive Climate Controller.

12.7.2. Actor(s): The user

12.7.3. THE USE CASE BEGINS WITH: The user touches either the “+” or the “-” button in manual/ debug mode to adjust the desired fan speed level that are in the form of fixed presets.

12.7.4. THE USE CASE ENDS WITH: The user sees the adjusted indicator on the “Fan Speed” indicator.

12.8. The user specifies the air direction.

12.8.1. Scenario: The user wants to specify the airflow output locations using the installed Automotive Climate Controller.

12.8.2. Actor(s): The user

12.8.3. THE USE CASE BEGINS WITH: The user touches the appropriate button from the available buttons that have icons representing different preset vent flap direction in the GUI.

12.8.4. THE USE CASE ENDS WITH: The user sees the touched button highlighted on the GUI.

12.9. The user enables debug mode.

12.9.1. Scenario: The user wants to debug automatic climate control mode using the installed Automotive Climate Controller.

12.9.2. Actor(s): The user

12.9.3. THE USE CASE BEGINS WITH: The user touches the “Debug” button on the “Auto,” or “Manual” mode.

12.9.4. THE USE CASE ENDS WITH: The user sees the “Debug” title highlighted on the bottom title bar of the GUI.

12.10. The user gets diagnostic information about the individual vehicle hardware components.

12.10.1. Scenario: The user wants to view individual component status of the vehicle hardware components using the installed Automotive Climate Controller.

12.10.2. Actor(s): The user

12.10.3. THE USE CASE BEGINS WITH: The user picks the hardware that he/she wants to debug from the list of the hardware components displayed on the GUI.

12.10.4. THE USE CASE ENDS WITH: The user sees the information about the individual hardware components the GUI.

12.11. The user checks the individual hardware components.

12.11.1. Scenario: The user wants to check the individual hardware component using the installed Automotive Climate Controller.

12.11.2. Actor(s): The user

12.11.3. THE USE CASE BEGINS WITH: The user picks the actions related to the particular part displayed on the GUI.

12.11.4. THE USE CASE ENDS WITH: The user sees the actual hardware performing the specified action.

12.12. The user turns on the AC of the system.

12.12.1. Scenario: The user wants to turn on the ac of the vehicle using the installed Automotive Climate Controller for fuel economy.

12.12.2. Actor(s): The user

12.12.3. THE USE CASE BEGINS WITH: The user touches the ‘Turn on/Off AC’ button on GUI.

12.12.4. THE USE CASE ENDS WITH: The user sees the ‘turn on/off AC’ button getting highlighted on the GUI.

12.13. The user turns off the AC of the system.

- 12.13.1. Scenario:** The user wants to turn off the ac of the vehicle using the installed Automotive Climate Controller for fuel economy.
- 12.13.2. Actor(s):** The user
- 12.13.3. THE USE CASE BEGINS WITH:** The user touches the 'Turn on/Off AC' button on GUI
- 12.13.4. THE USE CASE ENDS WITH:** The user sees the 'turn on/off AC' button getting un-highlighted on the GUI.

12.14. General Use Cases Diagram

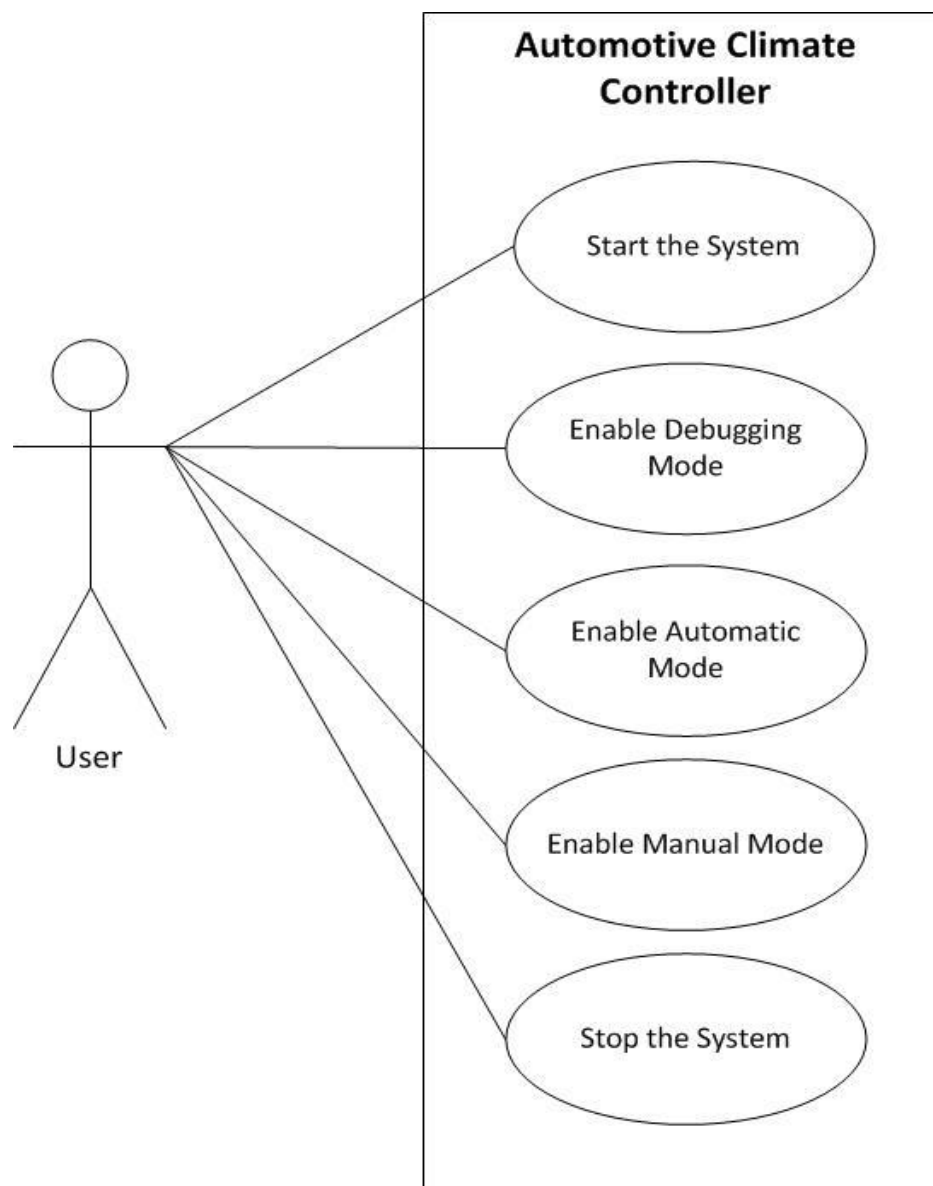


Figure 6: General Use Cases

12.15. Automatic Mode Use Cases Diagram

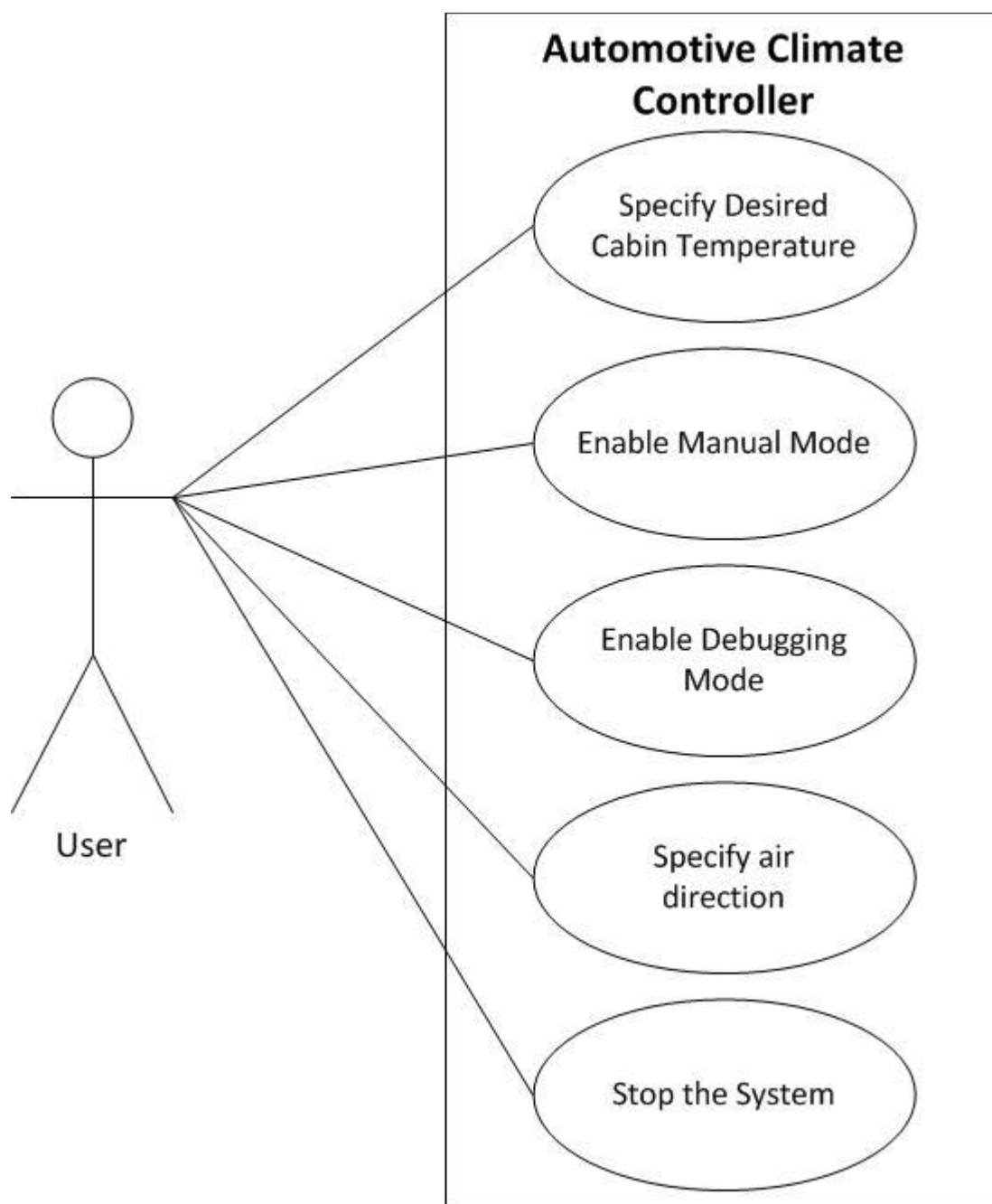


Figure 7: Automatic Mode Use Cases

12.16. Manual Mode Use Cases Diagram

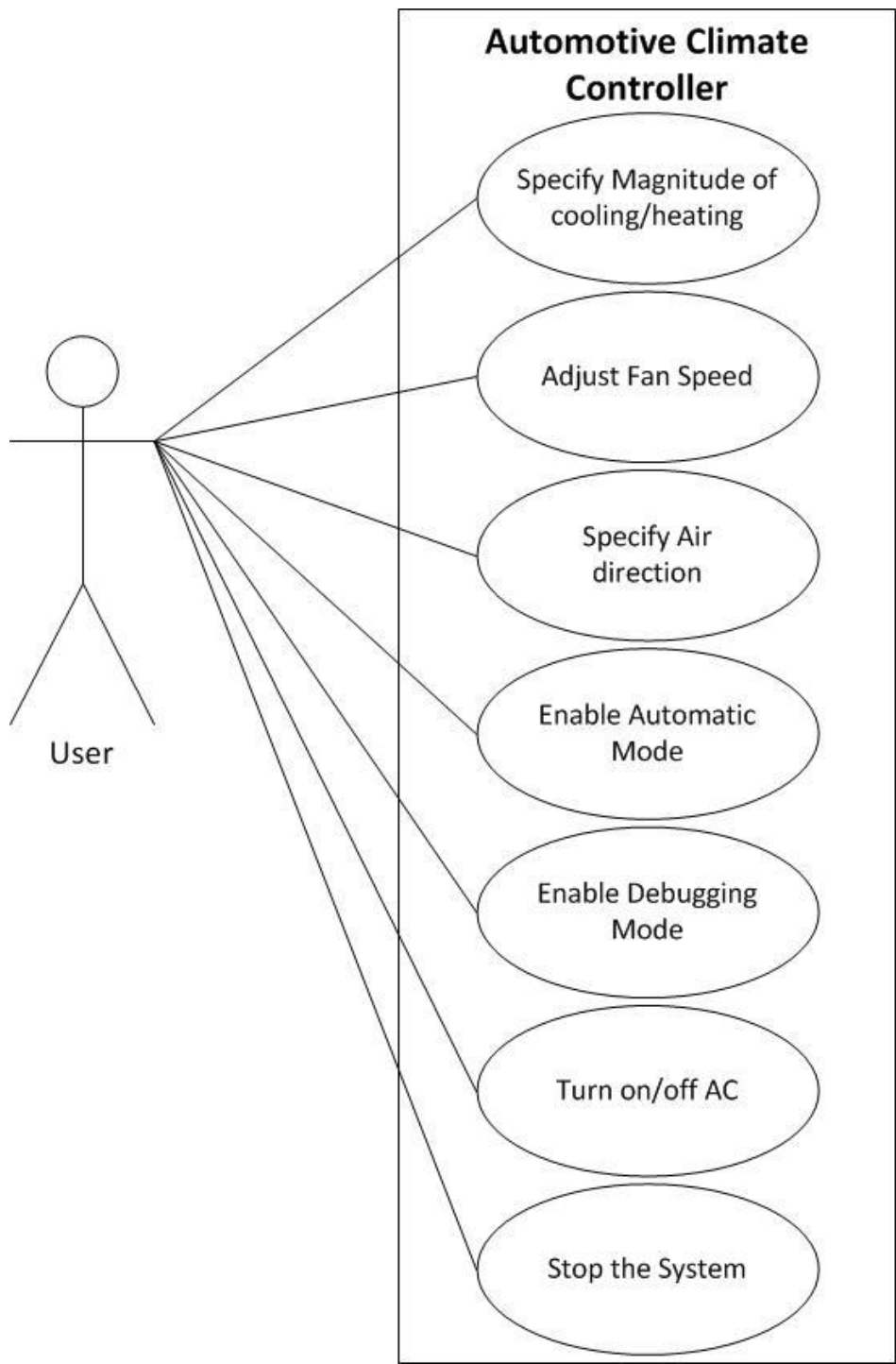


Figure 8: Manual Mode Use Cases

12.17. Debugging Mode Use Cases Diagram

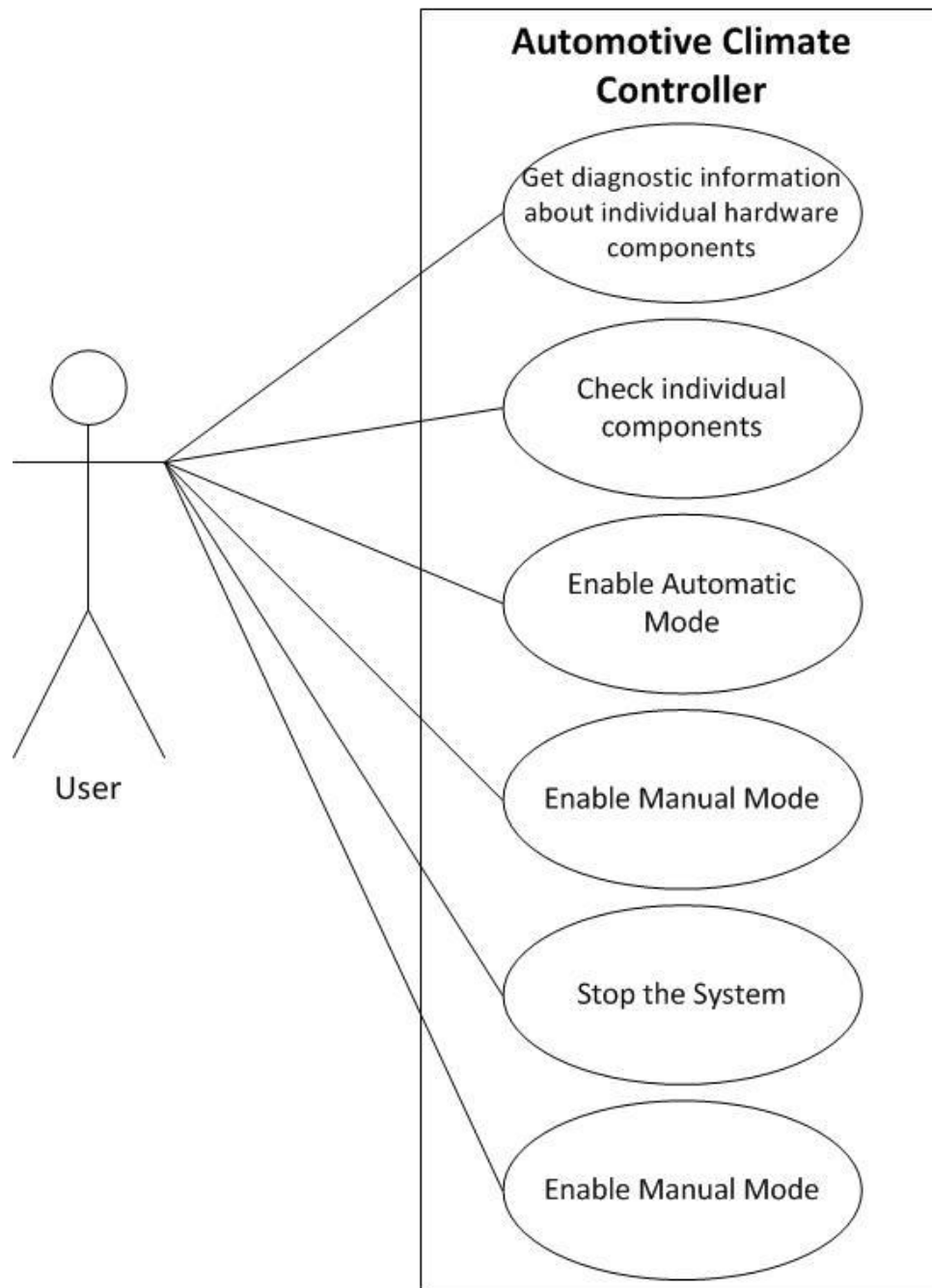


Figure 9: Debugging Mode Use Cases

13. Feasibility Assessment

The Automotive Climate Controller has many components that require considerable knowledge of various topics such as circuit design and the car cooling cycle. Thus, it is necessary to conduct a preliminary feasibility assessment of the product. The results have been broken up into six components and have been discussed below.

13.1. Scope Analysis

The nature of the project is very feasible for the team Auto Climatix to accomplish. However, the project size, as determined by different estimation techniques is very huge. So, it is almost impossible for the team to complete the project on allocated time. The team is very excited to complete the project by implementing all the requirements discussed in this document. However, the team realizes that there is not enough time and expertise with the team to accomplish all the requirements. So, the team has concluded that all the critical priority and the high priority requirements will be implemented. The moderate requirements will be considered for design and implemented if the team has enough time left after completing the high priority requirements. The low priority requirements will be considered for the future items. The team reached the above decision after wide research on the project, frequent meeting with the stakeholders and group discussion among the team members.

13.2. Research

Team Auto-Climatix has performed significant research to identify the technological and practical feasibility of this project. The primary focus was to determine how components that must be implemented work in functioning systems, what controls are required and what controls are beyond the scope or scheduling constraints of the project.

The primary technological areas of research were done by investigating the existing wiring diagram of the vehicle, a physical air conditioning console from a 1983 E-Class Mercedes Benz, and a more modern air conditioning diagram of a Nissan 350Z. In this process, Auto-Climatix identified mechanically critical components of the vehicle that do not lend themselves to be integrated into the system while maintaining appropriate testing standards. The compressor clutch was identified in this process as one such item that could not be interfaced with the system without compromising the mechanical integrity of the air-conditioning system and testing standards required for implementation. Additionally, the team identified numerous redundant or arrayed systems that would reduce the initial project size estimates dramatically.

An additional research area was financial feasibility. In the team's research, it was determined that the most significant cost would be the mobile device. Adequate mobile devices fell within a reasonable range for the team's budget; premium devices can easily extend beyond the budget. This limits the team to very specific operating system constraints for the development of the end-user application.

13.3. Technical Analysis

The automotive climate controller demands technical ability in three major areas: Hardware design, software design, and cross-platform communications.

Cross-platform communication enables the mobile device to interact with the primary hardware component of the system and enables that system to control vehicle hardware. The data transmitted between the microcontroller and tablet must be reliable and timely. If the information is not timely, control signals to critical systems could result in mechanical failure of the hardware or inadequate system performance. Similarly, unreliable signals between the devices could result in mechanical failure of the components or inadequate system performance. Options available for handling cross-platform communications include Blue-Tooth wireless communications and serial connections.

Hardware design refers to the development of the circuitry unique to the microcontroller in addition to the signal conditioning that must be handled for the microcontroller to adequately interpret hardware status and respond with appropriate controls. The microprocessor must be designed with handling multiple real-time inputs and handling input and output to and from the mobile device. The most significant challenge is designing a microprocessor that is modular and may take multiples of varying inputs. These inputs may have different signal types, complicating the signal conditioning required. An option for handling this would be designing a central microcontroller handling communications to the hardware with independent modules managing signal conditioning. Finally, the hardware design must be made in such a way that the microcontroller is protected from power surges from normal vehicle use. An available option for this is implementing a blowback circuit, which would mitigate instances of this to occurrences outside of vehicle shutoff.

Software design consists of three categories: The GUI, signal processing, and control output. Signal processing is likely to be the most challenging component of the software design as it entails interpreting digital input to the microcontroller into meaningful values for the user. Similarly, system control output must be processed into digital and analog output. An option available to mitigate the required processing load caused by signal conditioning is offloading the processing to the mobile device which will have an abundance of processing power compared to the onboard microprocessor. Control output will have to be determined through algorithms taking the entire system status into account.

13.4. Cost Analysis

Team Auto-Climatix has a maximum internal budget of \$800 for the Automotive Climate Controller project. This budget must only account for the mobile device, the microcontroller, connections between the vehicle hardware and the microcontroller, and components exclusive to the mock-up. As the sponsor has provided vehicle hardware components, the cost of these components has been excluded from the team's budget considerations. Auto-Climatix has arrived at the conclusion that the required components for the mock-up may be delivered under budget based on current market costs. The availability of used and working condition items, particularly touch-screen mobile devices, from previous senior design

projects or the availability of said items at a discount from online retailers might drastically diminish the required cost of the automotive climate controller mockup.

13.5. Resource Analysis

Team Auto-Climatix comprises of two software engineering students, two computer science students and one computer-engineering student. The analysis of these resources indicates that the team has significant advantages in developing stand-alone applications and architectural and detailed design. The analysis also shows that the team has marked deficiencies in the development of embedded systems, circuit design, programming embedded systems, and understanding of the mechanics of the system.

The complete lack of experience in mobile programming is not seen as a considerable risk as the most prevalent mobile operating systems utilize programming languages derived from languages seen as relative strengths within the team (specifically Java, C and C++). The largest risk associated with the project is derived from our reliance on a single team member for embedded system expertise and another for understanding the mechanics of the system. The team's strength in programming is seen as an advantage in the development of a modular application for deployment in multiple car platforms. Additionally, the team is familiar with widely used industry tools. Additionally, the team has strong leadership and organizational skills.

13.6. Schedule Analysis

Team Auto Climatix has analyzed several estimation methods and accommodated our system to each one to determine the size, effort, and schedule of the project.

Size Estimates

The first method used to determine the size of the project was function point estimation. Table 11-1 gives an overview of the function point estimation performed on our system:

Table 1: Function Point Analysis

Program Characteristic	Low Complexity	Moderate Complexity	High Complexity	Function Point Totals
Number of inputs	6*3	0* 4	4* 6	42
Number of outputs	8* 4	0* 5	4* 7	60
Inquiries	0* 3	5* 4	0* 6	20
Logical internal files	0* 7	1* 10	0* 15	10
External interface files	0* 5	0* 7	0* 10	0
Unadjusted Function Points Total:				132

Adjustment Factor:	0.95
Total:	125.4

The bold numbers indicate the number of function points and then they are multiplied by a function multiplier determined by using *McConnell's table 8-2. The second method used to determine the size of the project was the COCOMO "lines of code" estimate.

Effort Estimation:

Jones First Order of Effort Analysis

125.4 Function points $\approx$ 126

Using the table 8-7 from *McConnell

Software Type: Systems Software

Skill level Type: Worst in class

$126^{.45} \approx 8.8 \sim 9$ calendar months.

COCOMO MODEL

This model continues from the function point analysis. Here, we take the function points calculated and convert them to the lines of code using the multipliers for that particular language. Most of the code of our application will be written in object- oriented language Java/Objective that will be for the mobile application in the mobile device. Some part of the coding will be in the PIC micro-controller, which will be PIC Assembly (Macro Assembly) and C.

Since this model is using the function Points from the Function Points Analysis, again this model gives us the complexity of the system and the estimated time that will be consumed in terms of the input, outputs, inquiries, logical files and interface files.

FPS from function Point Analysis: 146

Unmodified Lines of Code

C: $126 * 128 = 16128$

Objective C/Java: $126 * 53 = 6678$

PIC Assembly (Macro Assembly): $126 * 213 = 26838$

Modified Lines of Code

C: $126 * 128 = 16128 * 30\% = 4838.4 \Rightarrow 4839$

Objective C/Java: $126 * 53 = 6678 * 65\% = 4340.7 \Rightarrow 4341$

PIC Assembly (Macro Assembly): $126 * 213 = 26838 * 5\% = 1341.9 \Rightarrow 1342$

Summation:

C + Objective C/Java + PIC Assembly = 4839+ 4341+ 1342= 10522 Lines of Code

Effort Applied: = $a \cdot (KLOC)^{bi} = 3.0 \cdot (10.5)^{1.12} = 41.76 = > 42$ people months

Development Time: = $2.5 \cdot (42)^{0.35} = 9.24 \Rightarrow 10$ calendar months

Acronyms

FP = Function Point

FP count $\Rightarrow$ Function Point count

Summarized Table:

Table 2: Size Estimation Summary

Method	Function Point	Jones First Order	CoCoMo Model
Size Estimate	126		10522 Lines of code
Effort Estimate			42 people months
Schedule Estimate		9 Calendar Months	10 calendar months

14. Future Items

The following requirements are items that may be implemented at a later date. These items currently extend beyond the schedule, technical skills and budget allowed for the project. These items additionally are not critical to the acceptance or performance of the Automotive Climate Controller.

14.1. The microcontroller shall monitor pressure sensors.

14.1.1. Description: The microcontroller shall accept readings from vehicle pressure sensors. The system shall monitor vehicle pressure sensors persistently while the vehicle is on.

14.1.2. Constraints: Schedule: The sensors are in series with the control signal to the compressor clutch and therefore do not require manipulation. Safety: It is inadvisable to set all points of failure within the application.

14.2. The system shall control the power seating.

14.2.1. Description: The microcontroller shall allow the user to manipulate electronic seat settings including forward-backward position, seat tilt and seat heating.

14.2.2. Constraints: Schedule: This requirement needs a separate UI.

14.3. The mobile device shall provide radio controls in the vehicle.

14.3.1. Description: The mobile device shall be able to provide standard radio controls (Turn on, turn off, volume adjustment, FM band controls, AM band controls).

14.3.2. Constraints: Schedule: Inadequate time to design retrofitted stereo system in vehicle.

14.4. The mobile device shall provide audio output through the vehicle speakers.

14.4.1. Description: The mobile device shall output audio cues through the vehicle's speakers.

14.4.2. Constraints: Budget: Inadequate budget to retrofit stereo system in vehicle. Schedule: Inadequate time to design retrofitted stereo system in vehicle.

14.5. The mobile device shall be removable from the vehicle.

14.5.1. Description: The system shall allow for the mobile device to be secured by the user externally from the microcontroller.

14.5.2. Constraints: Schedule: Allowing the mobile device to save state is not trivially done. Additionally, safety must take into account that this could only be done when the vehicle is turned off.

14.6. The debugging mode shall show the problems that the system is able to identify.

14.6.1. Description: The debugging mode shall display a short message about the problem that the system was able to identify.

14.6.2. Constraints: Schedule: Identifying specific signals that correspond to error states for each hardware item is well beyond the scope of the team's schedule.

14.7. The mobile application shall generate a log file of the system.

14.7.1. Description: The mobile application shall write a log file of the system that records the permanent events like the installation of some inputs devices, output devices, system startup, shutdown and also the hardware failure information.

14.7.2. Constraints: Budget: Allocating a mobile device with sufficient storage is beyond the scope of the budget.

14.8. The failure rate of the mobile application shall be less than a 1/10,000 uses.

14.8.1. Description: The system shall not fail more than once for 10,000 uses.

14.8.2. Constraints: Schedule: Inadequate time to properly develop simulation for device. Resources: Impossible to allocate multiple devices to verify reliability of system.