CISC 322/326 - Software Architecture
Assignment #3: Report

Architectural Enhancement

Team Bingus
Haadia Mufti - haadia.mufti@queensu.ca
Emily Poon - 16ejp6@queensu.ca
Kevin Shroff - 17kss@queensu.ca
Oliver Cao -17oyc@queensu.ca
Gregory Secord - 19gdjs@queensu.ca
Connor Colwill - 18cgc5@queensu.ca

Abstract
In this report, we propose an enhancement for the Apollo self-driving cars. In our enhancement we introduce a feature that ensures the driver is alert while the car is on the road. We talk about 2 different ways it can be implemented which is via a notification feature or with a camera. With the help of a SAAM analysis we look at each method's maintainability, testability, evolibility, performance and accuracy, therefore, recognizing the better way of implementing it. The SAAM analysis helped us realize that even though the notification method is easier, it is not necessarily a better method. We talk about the ways the two implementations will affect the high and low level architecture, the potential risks they pose and the ways we can test this implementation. Lastly, we present 2 use case diagrams that show how this enhancement will interact with the architecture in a particular scenario.

Introduction
After an in-depth analysis of the autonomous vehicle software Apollo V7, conceptual and concrete architectures were derived. This report utilizes those architectures to discuss the implementation of a new feature to the system, and how the feature may impact the overall architecture.

The first section of this report explains the proposed enhancement - a system that monitors the user to make sure they are paying attention to the road, even though they may not be manually driving. Two implementations are proposed, a UI based notification screen in which the user must click notifications throughout autonomous driving, and automatic eye detection through the addition of new cameras inside the vehicle. After the proposal of these two implementations, a SAAM analysis is conducted to compare and contrast them on the basis of maintainability, evolvability, testability, performance, and accuracy. The conclusion of this analysis determined that the implementation of the camera system would be more beneficial than the notification system.

The next section details the effects of the enhancement on the architecture. After careful analysis, it was determined that neither implementation would affect the architecture too heavily. The notification and camera systems would not require the introduction of a new subsystem, merely improvements/additions on pre-existing systems.

Sequence diagrams are then used to highlight how the proposed feature would be implemented in a user scenario. The first use case looks at how the notification system would interact during a typical Apollo drive, and the second case examines the interactions needed between perception and guardian for the camera implementation.

The next section examines possible testing requirements for the new feature. It was determined that for the proposed notification feature, testing would need to be done on notification prioritization, dismissal, and recurrence. For the camera feature, testing would need to be done to determine the certainty a driver is not paying attention, along with testing what would happen if the camera was obstructed.

Finally, the last section of the report examines the potential risks and lessons learned throughout the project. For the notification system, human response time, reliability, and dismissal of notifications were all considered risks as they could limit the practicality and effectiveness of the notification system. For the camera, the biggest risks identified include performance issues, the ability to turn the camera off, and blocked line of sight. Invasion of privacy is also a great concern, as an indoor facing camera has access to view any passenger in the car at any time. In regards to the lessons learned during the project, the team thought the toughest part was determining a useful enhancement, as Apollo is thoroughly developed and has many autonomous features already.

Enhancement

When thinking about autonomous vehicles, people often wonder how autonomous a vehicle really is. Of course, they are autonomous in the sense that they do not need any influence from the driver to work as intended. However, they are not autonomous in the sense where the driver never needs to be aware/present in the vehicle. Technology is not perfect by any means, and it may potentially tell the vehicle to take actions that are not correct nor appropriate. There have been stories of fatal situations occurring due to the lack of awareness on the drivers part. In March of 2018, a woman was crossing the road at night and was hit by a car that was using its self-driving system. The driver was looking downwards, watching a video on her phone and not paying attention to what's in front of the car. Because of situations like these, drivers should always be aware of the world around them such that accidents do not occur.

But how does one ensure this? This question is what sparked our interest in coming up with a solution. Our proposed enhancement for Apollo is introducing a mechanism that will make sure the driver is aware at all times. We will discuss two different ways of implementing this idea, as well the changes in the architecture that will be made.

The first implementation is to have notifications pop up at regular intervals through Dreamview. The driver will have to close the notification to allow autonomous driving to continue. In the case of notifications popping up multiple times and not dealt with, Dreamview will notify the Guardian module to take the car off road until the driver is back up and ready. Because of this, a new dependency between these modules, Dreamview->Guardian, will be present in the architecture.

The second implementation is putting a camera inside the car so eye-movement from the driver can be tracked. With help from the Perception module, if it is deemed (through the use of accurate algorithms) that the driver is not paying attention to the road, the Guardian module will be notified and the car will be taken off road until the driver is back up and ready. Because of these new interactions between modules, the dependency Perception->Guardian will be introduced.

SAAM Analysis

Non-Functional Requirements	Implementation 1-Notifications	Implementation 2-Camera	Comparison

Maintainability	Pro: Due to the lack of extra hardware needs, this only needs to be maintained in the code which is fairly easy to do Stakeholders Impacted: *Development engineers*- They will be maintaining the code base for the notifications. *Company executives*- They are the ones who decide to invest the money to maintain this feature. *Drivers*- Drivers reap the benefits of having this added feature.	Con: Harder to maintain as a camera needs to be added to the architecture in order to detect the movements of the driver. Stakeholders Impacted: *Development engineers*- They will be maintaining the code base for the notifications. *Company executives*- They are the ones who decide to invest the money to maintain this feature. *Architects*- They have to maintain the architecture of the added components of this enhancement. *Drivers*- Drivers reap the benefits of having this added feature.	The notification method is easier to maintain as it does not require additional components but easier does not necessarily mean it is the better option.

Evolvability	Pro: The notifications can be evolved for different purposes like giving warnings for low fuel or warnings for traffic etc. since they're just text messages that need to be sent based on the performance of the other components Stakeholders Impacted: *Development engineers*- They will be responsible for the added any new updates to the feature *Company executives*- They are the ones who decide to invest the money to evolve this feature. *Drivers*- Drivers reap the benefits of having this added feature.	Pro: Cameras can also be easily programmed to account for other things in the car like if the passengers are feeling too hot or cold, the temperature can be adjusted accordingly. Stakeholders Impacted: *Development engineers*- They will be responsible for the added any new updates to the feature *Company executives*- They are the ones who decide to invest the money to evolve this feature. *Drivers*- Drivers reap the benefits of having this added feature.	Both of these methods have potential to evolve into something more therefore they are tied in this area.

Testability	Pro: A simple test for the notification to pop up at regular intervals would be needed for this. Stakeholders Impacted: *Development engineers*- They will be responsible for testing this feature *Company executives*- They are the ones who decide to invest the money to include testing of this feature. *Drivers*- Drivers are able to use this feature if it is tested properly	Con: The test for this method is more complex as test for the camera to properly track eye movements would need to be developed. Stakeholders Impacted: *Development engineers*- They will be responsible for testing this feature *Company executives*- They are the ones who decide to invest the money to include testing of this feature. *Drivers*- Drivers are able to use this feature if it is tested properly	The notification method is easier to test compared to the camera method but the camera method fulfills the purpose of enhancement better

Performance	Pro: Due to the simple task of this feature i.e. sending notifications after regular intervals, therefore there are not a lot of system calls involved. Con: It can be annoying for the driver to always acknowledge the message while having to concentrate on the road too. Stakeholders Impacted: *Development engineers*- They will be responsible for ensuring this feature is performing correctly. *Company executives*- They are the ones who decide to invest the money to ensure good performance. *Drivers*- Drivers use the feature.	Pro: As the camera ensure to only take action if the driver is not paying attention to the road, it ensures that the driver and passengers will be safer. Con: It is a longer process as the camera needs to read eye movements, interpret it and take the proper action accordingly. Stakeholders Impacted: *Development engineers*- They will be responsible for ensuring this feature is performing correctly. *Company executives*- They are the ones who decide to invest the money to ensure good performance. *Drivers*- Drivers use the feature.	The notification task is simple but the camera implementation perform the task better so we choose to go with it.

Accuracy	Con: This feature is not necessarily accurate as it sends notifications regardless of if the driver is paying attention on the road or not. Stakeholders Impacted: *Development engineers*- They will be responsible for ensuring the notifications are accurately sent out. *Company executives*- Company gets better reputation if their feature works accurately. *Drivers*- Drivers are safer if the feature is working accurately	Pro: This method is more accurate as the camera only takes action if it detects abnormal behavior from the driver. Stakeholders Impacted: *Development engineers*- They will be responsible for ensuring the camera interprets the eye movements accurately. *Company executives*- Company gets better reputation if their feature works accurately. *Drivers*- Drivers are safer if the feature is working accurately	Camera implementation is more accurate towards the task on hand as it is dependent on whether the eye movements of the driver is such that they are paying attention to the road whereas the notification implementation depends on just notifications at intervals.

Comparison Conclusions

To summarize the SAAM analysis we can see that the 1st implementation method (notifications) is a much simpler way of implementing the enhancement, it is easy to maintain and test and will also not require many changes into the overall architecture of the system. On the other hand, the camera implementation will require more hardware and also change in the conceptual architecture of Apollo. The camera will also need to take the help of other modules within the architecture to interpret the readings it records but it still performs the task on hand better than the notifications method as it is more accurate and more specialized.

After performing the SAAM analysis, we decided to go with the second implementation method. Even though it requires more change and support from the engineers in the company, it will have a better result than the notifications. Not only will the drivers benefit from having this enhancement, but it will make the environment on the road safer too. The notifications method is an easy fix for this problem, but it does not address the actual problem in hand and not addressing the notification promptly might result in unnecessarily actions taken by the car to prevent anything harmful from happening on the road. All in all, the camera method is a much better implementation of this enhancement.

Effects on High and Low Level Architecture

Regarding the high-level architecture of Apollo, the enhancements don't seem to change too much, as it is still a publish-subscribe style. With the addition of driver notifications and a camera that detects the driver's line of sight there wouldn't be any added subsystems, only improvements within each subsystem and some new dependencies between subsystems. For the enhancement of driver notifications, there would be a new dependency from HMI/Dreamview to Guardian.

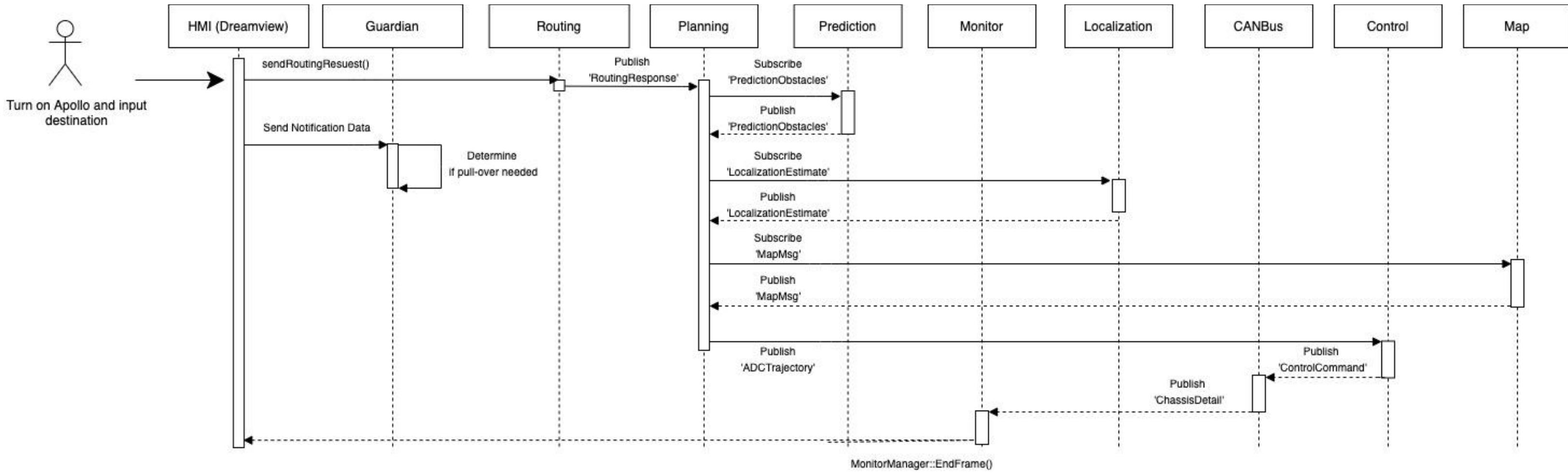
On the other hand, the low level-architecture would be greatly affected by the enhancements. For the enhancement of making sure the driver's eyes are on the road while driving, the enhancement would require a new camera inside the car to detect the driver's eyesight. The following directories and files would be impacted with this new enhancement. In the Perception subsystem, the "perception/camera/app" directory in charge of the different cameras. In the HMI/Dreamview subsystem, the "dreamview/backend" and the "dreamview/frontend" are the directories that would be responsible for the notifications to the driver about their eyes being on the road.

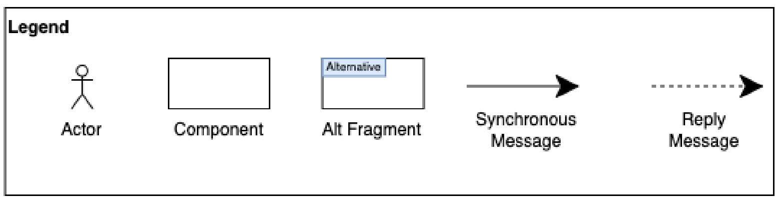
The enhancement of driver notifications requires more communication from other subsystems. The following directories and files would be impacted with this new enhancement. In this enhancement, the HMI/Dreamview subsystem would also impact the "dreamview/backend" and the "dreamview/frontend" directories as they would be responsible for any notifications to the driver. In Guardian, the "guardian/conf" directory in charge of performing the decisions on safety, as well as in Perception, the "perception/onboard" directory that would also be in charge of the onboard systems and messages that would contribute to the notifications to the driver.

Diagrams

Use Case 1:

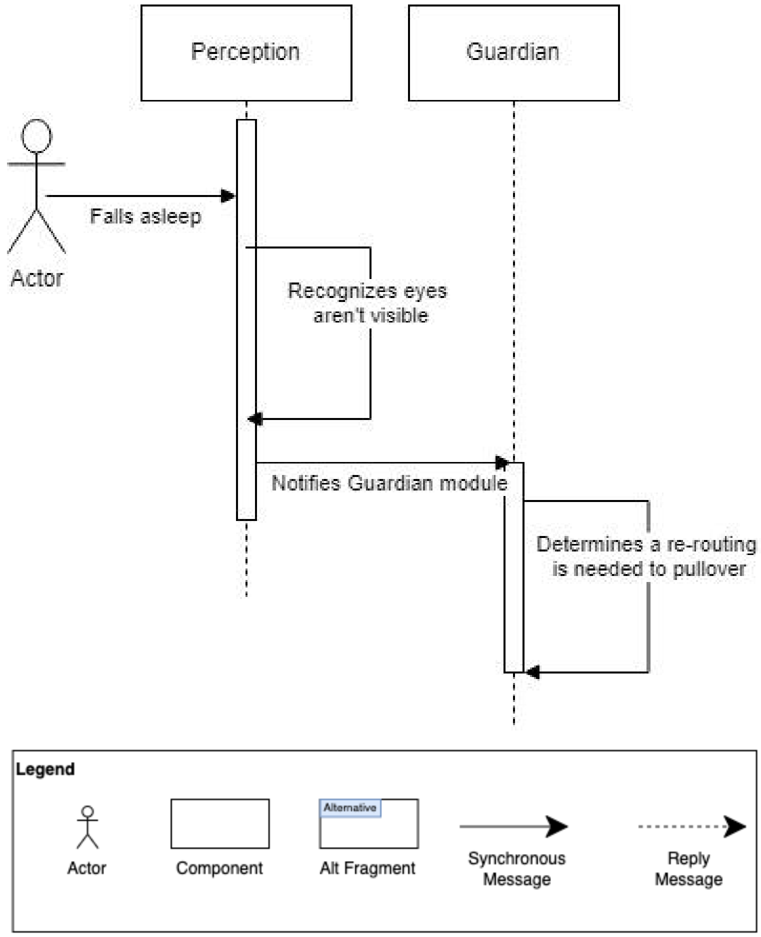
The first use case highlights a typical drive using Apollo - the user inputs a destination and Apollo takes them there. In addition to this, an example of the notification implementation is given. The Dreamview module will send notifications for the user to confirm, and notify the Guardian module if these notifications are not dealt with. For the implementation of the camera system, the perception module would check if the driver is paying attention and notify Guardian if it deems necessary. In both cases, the implementation does not impact the overall sequence diagram of Apollo as the new feature utilizes systems that already exist in the software.





Use case 2

This second use case describes the scenario where the driver falls asleep behind the wheel. The Perception module notices that the driver's eyes are not visible, and deems that action must be made. The Perception module sends the message to the Guardian module, and a re-routing is determined for safety. In the implementation of the notifications, instead of the Perception module discovering action must be taken, Dreamview would realize that the notifications are not being dealt with, and would also send that info to the Guardian module.



Testing

Implementation 1: Notifications

- **Use Case 1: Multiple Notifications - Notification Prioritization**

  In the event that multiple notifications are raised by the vehicle (i.e. comparatively unimportant notifications such as navigation directions, bluetooth music next track notification) on the MID with one of them being the driver attention notification, there needs to be a prioritization system in place to bring the driver attention notification to the front, instead of what would otherwise be a chronological notification system.

  Therefore, the following needs to be tested and verified:
  Does the Multi-Information Display (MID) system correctly prioritize driver attention notifications over other non-essential system notifications?

  This can be verified by testing the prioritization system by raising multiple mock notifications that have varying priority levels assigned to them, and verifying that the driver attention notification is prioritized (brought to the front of the notification stack and displayed) on the MID system above all notifications that it has a greater priority than.

(Note that some few critical notifications will have a greater priority than even the Driver Attention notification, such as the pre-collision warning notification, i.e. BRAKE NOW).

- **Use Case 2: Notification Dismissal and Recurrence**
  When a Driver Attention notification is dismissed, it should correctly remove itself from the notification stack, as well as allow for autonomous operation to continue. It should also reappear at predefined intervals in the future, and be ready to dismiss again. This needs to be tested to ensure the implementation is functioning as expected. The general testing methodology for this use case is straightforward:
  1. Begin autonomous operation of the vehicle
  2. Wait for the Driver Attention notification to appear
  3. Dismiss the notification
     a. Ensure that autonomous systems continue operating working as expected
  4. Verify Driver Attention notification re-appears after predefined interval, and verify steps 3 and 3a again

Implementation 2: Camera

- **Use Case 1: Driver Monitoring - Degree of Accuracy and Certainty of Driver Attention Determination Algorithm**
  Core to implementation 2 is the ability of the proposed Driver Attention system to have high accuracy in its determination of the driver's attention. In order for it to reliably and relevantly prompt the driver for attention, it needs to reliably determine, and with a high degree of accuracy and certainty, on what the driver's attention is currently focused (i.e. the road, or something else).

  Testing needs to be implemented that verifies that the Driver Attention detection algorithm (eye-tracking algorithm) has high accuracy and certainty (>90%) of its determination of the driver's attention at all times.

- **Use Case 2: Driver Monitoring Camera Obstruction**
  In the event that the camera that is required to monitor the driver is obstructed (i.e. dirty, sensor malfunction, physical object in the way) the predefined behaviour to deal with this appropriately is to both:
  1. Not permit autonomous operation until issue with sensor has been corrected
  2. Raise notification to the driver through MID

  Therefore, this behaviour needs to be tested to verify that it is correctly occurring.
  The following needs to be tested and verified:
  1. Notification is correctly raised through MID to driver when:
     a. Sensor issue occurs
     b. Car is started
     c. Autonomous operation is requested
  2. Autonomous operation is correctly disabled and unpermitted until sensor status is corrected

Potential Risks

Implementation 1: Notifications

Driver-notifications systems may prevent risks of collison by reminding passengers to pay attention to the road while using the automated driving features of the car, however there are risks. Such systems may encourage drivers to grow more dependent on the notification system in order to ensure them from reduced distracted driving.

Many driver alert notification systems use cameras and sensors on the exterior of the vehicles in order to remind the drivers when it detects other vehicles approaching too closely or changes in the road conditions. One of the main challenges is implementing real-time alerts as the real time environment surrounding the vehicles changes. While the A.I's ability to successfully predict a collision is a factor, another consideration to take in is human error. The A.I would be able to quickly calculate and respond accordingly while humans may not get the chance to process and react to the notification when they aren't paying attention.

One other factor to consider is the reliability and performance of the software. As autonomous vehicles become more mainstream, it is still considered a relatively new technology. With the feature of pulling over the car when the driver hasn't paid attention by closing the notification may cause issues in light of a software glitch preventing the driver to close the notification.

Finally, a potential hazard is the ability for the driver to easily build a habit of turning off the notification without paying attention to the road ahead, resuming their distracted activities. If the notification system is based on a timer interval, then drivers may have the expectation to only pay attention every 15 minutes or so when the reminder shows up which can dull their reaction time, causing the driver to become discomposed in the event of an emergency.

Implementation 2: Camera

With the implementation of driver-monitoring systems in order to keep drivers in check while using autonomous driving features, there are additional potential risks to consider. The addition of a camera monitoring system will make drivers more dependent on the automated driving features as well as the monitoring system to remind them when they become distracted.

One of the main issues is the performance of the camera system. For instance, on existing Tesla's it detects the drivers eyes to see if they are focused on the road ahead. But if the driver's hands remained on the steering wheel, there was "no difference in warnings whether eyes were on or off the road." Another risk is the ability for the driver to shut off the monitoring system as seen on Subaru's via the menu. On certain vehicles, it is also "not required to be activated" in order to use the driver assistance systems. Another scenario to consider is if the driver is blocking the line of sight of the cameras, whether it's sunglasses or holding a cellular device. This may cause the alert system to be unable to function properly causing unexpected accidents. There already exists testing and research on camera monitoring systems where MIT's researchers found that on Tesla Model 3, when the autopilot was engaged, they were "unable to trigger a warning" with various different tests which included "a driver texting and blocking the view of the road" using a clipboard covering the entire face of the driver. While there may be implementations and restrictions that may help minimize the risks of the performance of these monitoring systems, there are too many scenarios where drivers may render them ineffective.

Another concern for driver-monitoring systems is the security and the invasion of privacy. The technology involves ‘infrared light-emitting diodes (LEDs) or lasers’ which allows the cameras to see the driver's face even at night. Over time the software can collect data and ‘create an initial baseline of the driver's normal, attentive face.’ This brings up certain ethical questions as whether this would be too invasive as many individuals may feel that they are constantly being watched. Through eye-tracking data alone, it can be used to ‘infer someone’s gender, age, body weight, or emotional state and concerns over the data collected of the drivers and if the company were to be hacked would arise.

Finally, the reliability of the driver-monitoring systems need to be regularly checked. The life expectancy of Charge Coupled Device cameras are anywhere between 5 - 25 years, that is without taking into consideration the probability of accidents or how the vehicle is treated by the owners. With such heavy dependency on the CCD in order to ensure the safety of the passengers, the quality and maintenance of these cameras are essential.

Lessons Learned

One of the lessons learned when trying to propose and develop an enhancement to the Apollo autonomous system, is to come up with an enhancement that can genuinely provide value to the system. As Apollo is already thoroughly developed and has already achieved L4 level autonomy, it was quite difficult to think of and propose an enhancement to the system. Taking a look at similar autonomous systems like Tesla’s Autopilot and Cadillac’s Supercruise, driver attention monitoring safety systems similar to our proposed enhancement are common place. Both Tesla and Cadillac’s autonomous vehicle systems are equipped with eye-tracking, camera-based driver monitoring systems that prompt & alert the driver to take control when the systems determine the driver is not paying attention to the road.

Conclusion

In conclusion, the proposed feature of a driver-notification and driver-monitoring systems is another safety net for drivers and passengers as it serves as a reminder to pay attention when using the automated driving feature on autonomous vehicles. While the implementation of the notification system is simpler and reduces costs, the driver-monitoring systems with the use of cameras are more effective and safer. In terms of the effects on the architecture, it is determined that the high level architecture remains as a publish-subscribe style with slight changes except for the new dependency from HMI/Dreamview to the Guardian module while the low level architecture would require changes in the Perception and HMI/Dreamview subsystems with the additions of a new camera for the drivers eye detection. Diagrams were used to demonstrate use cases where in the event that a driver were to fall asleep, how the perception and guardian modules would interact with each other in order to notify and wake the driver. Testing of the implementation of the notification and camera system were discussed where prioritization of different types of notifying sounds are tested and chronologically ordered as well as the reliability, accuracy and obstruction of the cameras.

The potential risks of both the notification system and the driver-alert systems were further explored with how drivers may grow dependent on either system as well as the maintenance and reliability of each system. Finally the lessons learned covers how our group faced the difficulties of proposing and developing a feature that genuinely improved upon a system that has already been thoroughly developed.

References

[1] Tesla's Camera-Based Driver Monitoring Fails to Keep Driver Attention on the Road, CR Test Show
https://www.consumerreports.org/car-safety/tesla-driver-monitoring-fails-to-keep-driver-focus-on-road-a3964813328/

[2] What is a Driver-Monitoring System?
https://www.aptiv.com/en/insights/article/what-is-a-driver-monitoring-system

[3] How Driver Monitoring Systems Can Protect Drivers and Their Privacy
https://www.consumerreports.org/privacy/driver-monitoring-systems-can-protect-drivers-and-privacy-a7714760430/ - oliver

[4] What's the Lifespan Of A Video Surveillance System?
https://www.calgaryalarm.com/blog/video-surveillance/whats-the-lifespan-of-a-video-surveillance-system/

[5] Tesla starts using cabin cameras to make sure drivers are paying attention
https://www.cnbc.com/2021/05/28/tesla-starts-using-cabin-cameras-for-driver-monitoring.html

[6] Cadillac Super Cruise: Hands-Free Driving
https://www.cadillac.com/ownership/vehicle-technology/super-cruise

[7] How Does Cadillac Super Cruise Work?
https://www.jdpower.com/cars/shopping-guides/how-does-cadillac-super-cruise-work

[8] 2018 Accident:
https://www.npr.org/2020/09/16/913530100/backup-driver-of-autonomous-uber-suv-charged-with-negligent-homicide-in-arizona