Honors Capstone: Human Machine Interface Design for Enhanced Accessibility

Team Members: Bradin Zaba (Honors Capstone), Anne Ye (Honors Capstone), Jinxin Li, Bashar Zidan, Rosemary Chen, Zhanchen Huang

December 16, 2022

Arriver

ENGINEERING
HONORS PROGRAM
UNIVERSITY OF MICHIGAN

Section 1: Project Summary

Our industry sponsor, Arriver, is a software company owned by Qualcomm. They were established in 2021 to create advanced driver assistance systems and are focused on sensor perception and drive policy. Arriver is currently working on developing their level 2+ autonomous driving system. A level 2+ system is one that allows the driver to take their hands off the wheel and let the vehicle drive itself when certain conditions are met. Because driving is very dynamic, the vehicle may encounter a scenario where the driver needs to take over. In this case, the vehicle's Human Machine Interface (HMI) is very important. In order for these safety critical driver alert systems to become universally effective, they need to be designed with the broad population in mind. That includes people with accessibility needs. Drivers of autonomous vehicles need to be provided with correct and important information so that they can decide when to take over control of the vehicle. For drivers with accessibility issues, it is important that the information is provided to them in the way that works best for them, so they do not miss an important alert.

Objectives/Scope

The goal of our project was to improve our sponsor Arriver's current in-vehicle HMI that is designed for a level 2+ autonomous driving system. Our objective was to develop accessibility settings and create configuration pages that would customize the HMI based on the user's accessibility needs. For the scope of our project, we targeted five main groups of people with accessibility needs: vision loss, mobility issues, color blindness, hearing loss, and screen motion sensitivity. More specifically, our vision loss target group would include people with partially correctable vision, but who are also not completely blind. For our mobility issues target group, we would include people with hand tremors and who may have difficulty performing button clicks. Our color blindness target group covered four different color profiles: protanopia, deuteranopia, tritanopia, and monochromacy. For our hearing loss accessibility needs group, we targeted people with partial hearing loss, but who are not completely deaf. For people with screen motion sensitivity, we reduced blinking effects and animations on the screen. These five groups were decided based on our team's research on the common accessibility related challenges for drivers and from a critical review of Arriver's current HMI. Our goal was to create five accessibility settings for our sponsor's HMI. To verify our design, we tested our interface in Arriver's test vehicle as well as tried simulating driving environments with users. We also used developed heuristics to evaluate our interfaces and existing accessibility testing applications to check the effectiveness of the interface for people with colorblindness, vision loss, etc. For our project, adding accessibility settings that change how the vehicle drives was out of scope.

Deliverables

At the end of the project, we delivered our modified base HMI with the accessibility settings fully implemented, our Figma designs of the accessibility settings, and supporting documentation

for our changes to the Arriver HMI. To share the code, we added the sponsors to our GitHub repository, and provided them with a zip file containing the Unity project that we worked on. The supporting documentation includes an overview of the accessibility related limitations of their HMI, the settings that we implemented, how they are used, and recommended changes to our design based on our testing.

Value to Sponsors

Our project is very valuable to our sponsors for future development of their interface because they currently do not have any accessibility settings, and they have not started implementing any. This project is their first venture into adding accessibility settings into their interface. From our project, our sponsors are gaining research and recommendations for potential accessibility improvements for their HMI. We have also given them a version of their HMI with five different accessibility settings already implemented. This implementation of their interface serves as a solid foundation for conducting user testing with people who have accessibility needs to get a sense of what their customers need.

Solution Strategy

Our HMI is implemented in Unity, a game development engine that can be used to create 2D applications like user interfaces. The vehicle passes messages containing information from the vehicle's suite of sensors and cameras to Unity using the messaging protocol Lightweight Communications and Marshalling (LCM). Once these messages are in Unity, the HMI handles the processing of the messages to determine the state of the vehicle. In our code, we worked with the many vehicle states and transitions between them to change the HMI based on the accessibility settings that we implemented. The configuration pages that we developed are built on top of the HMI and will allow the user to custom change the accessibility settings based on their needs.

Section 2: Our Work

Over the two semesters that our team worked on the project, we were able to modify Arriver’s base HMI to be simpler and have more obvious alerts, implement 5 accessibility settings, and complete several expert reviews to receive feedback on our design.

Base HMI Changes

Figures 1 and 2 show the baseline changes that we made to Arriver’s HMI. These changes were unrelated to accessibility issues, but were necessary for us to make in order to have a customer facing interface to work with. The first main change we made is removing the toggles on the right side of the interface. We removed them because the buttons were for engineers, not customers. Toggling them put the interface in engineering mode and allowed the engineers to test different settings, which customers would not need to do. The other main change we made to the

interface was shifting the middle panel to the right. We did this because without the buttons, there was empty space on the right, and it gave us more room on the left panel to implement accessibility settings.

Figure 1. Arriver's Original HMI

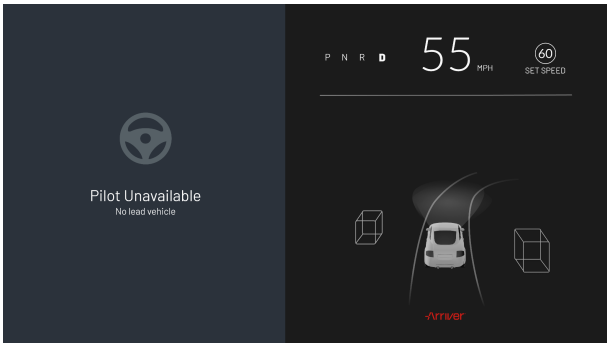

Figure 2. Arriver's HMI after our modifications

Accessibility Settings

In our settings menu, the first setting was for people with vision loss. The setting allows people with vision loss to modify the size of text on the interface by dragging the slider left and right. The setting starts at a high level so that people with vision loss are able to read the settings menu, and people without vision loss are able to decrease the text size. Increasing the size of the text also changes the text size within the settings menu, making the entire settings menu accessible as well. Figure 3 shows the interface for changing the text size, and figure 4 shows the base interface after the text size has been increased. The difference in text size can be seen best when comparing figure 2 with figure 4.

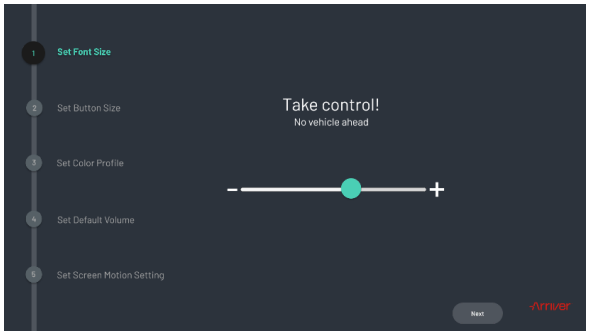

Figure 3. Vision loss setting interface

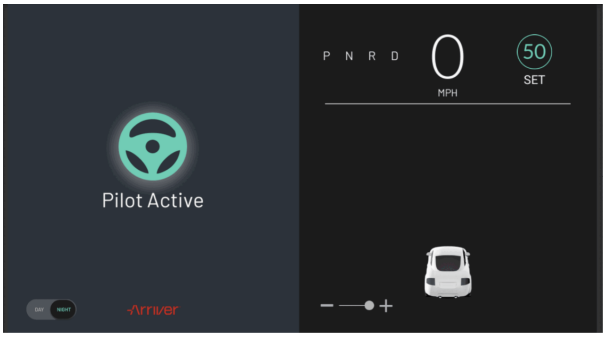

Figure 4. Base interface with larger text applied

The next setting we implemented was for people with mobility issues. This setting allows the driver to increase or decrease the size of buttons and other touch targets. This is useful for people with shaky hands because it decreases the amount of fine motor control necessary when trying to touch buttons with an extended arm. Similar to the vision loss setting, this setting also starts at a higher value, and changes the size of buttons in the settings menu. This is done so that the menu itself is accessible and any driver who gets in the vehicle would be able to use it. Figure 5 shows the interface for changing button sizes in the settings menu we developed. Figure 6 shows the interface with a bigger day/night mode button size in the bottom left corner.

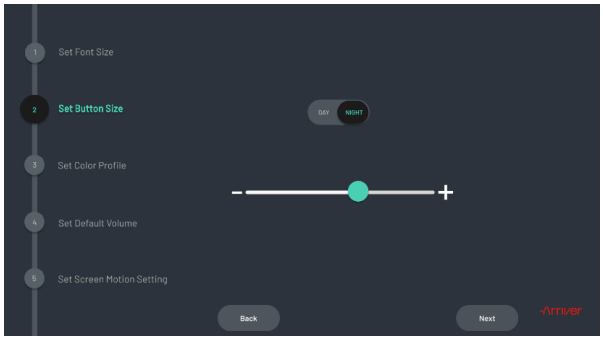

Figure 5. Mobility issues setting interface

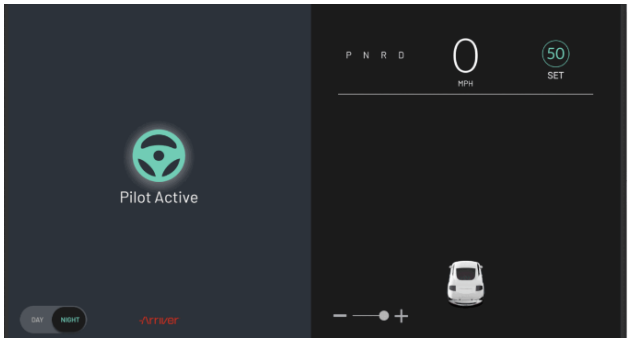

Figure 6. Base interface with larger button applied

The third setting we implemented was for people with color blindness. For this setting, we added four colorblind modes in addition to the original color palette. They are protanopia (red color blindness), deuteranopia (blue color blindness), tritanopia (green colorblindness), and monochromacy (full color blindness). We selected the colors in each color profile such that the contrast between them would be noticeable for people with the respective type of color blindness. Figure 7 shows the interface for selecting color blindness modes. The user is able to see a preview of the color palette for each mode, so that they are able to select the best one, and then they can select the mode by tapping the button under the preview. Figure 8 shows the pilot available state with tritanopia mode enabled. In each of the other colorblind modes, the green color of the square would change to the second color from the top of the selected mode.

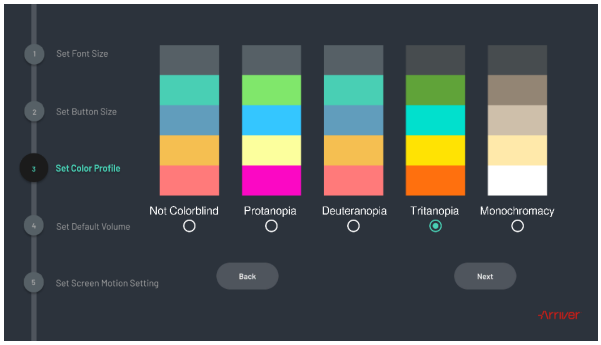

Figure 7. Colorblindness setting interface

Figure 8. Interface after applying Tritanopia mode

The fourth setting we implemented was for people with hearing loss. The setting enables the driver to modify the volume of all auditory alerts from the vehicle by dragging the slider. We also implemented tactile feedback using vibrating motors. The motors vibrate with different frequencies and patterns when the state of the vehicle changes, and it gets more intense when driver action is required. Figure 9 shows the interface for modifying the alert volume. The left side of figure 10 shows the motors attached to an Arduino microcontroller that we used to control them.

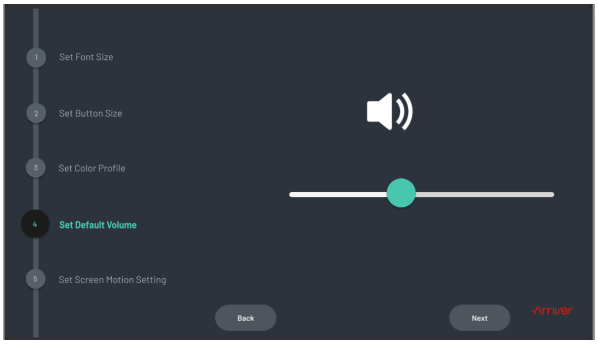

Figure 9. Hearing loss setting interface

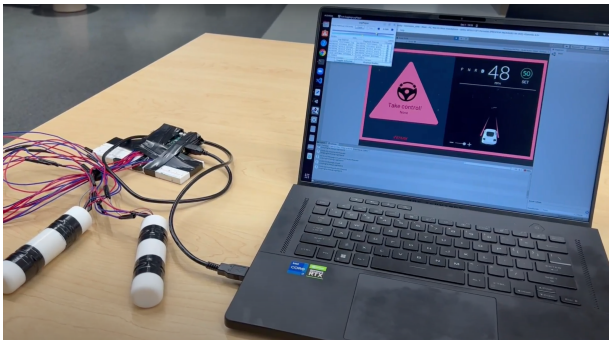

Figure 10. Tactile feedback system using Arduino

The last setting we implemented was for people with screen motion sensitivity. Arriver's current interface has many animations and flashing warnings that could cause discomfort for people with conditions like epilepsy. The interface we created to allow the driver to reduce screen motion can be seen in figure 11. By tapping the button in the middle, the driver is able to turn off all the flashing animations. Figure 12 shows the interface after turning off screen motion. Typically, in the warning state, the yellow square around the interface would be quickly flashing to get the driver's attention. However, with the screen motion reduction applied, it stays solid.

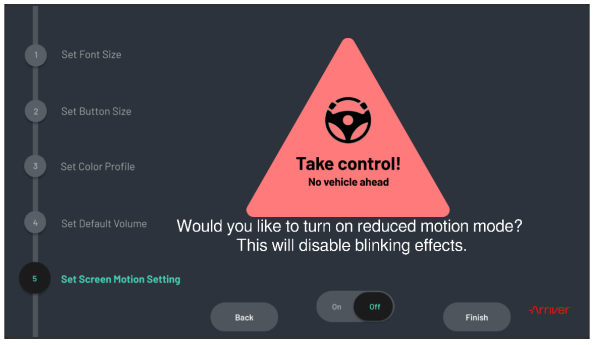

Figure 11. Reduced motion setting interface

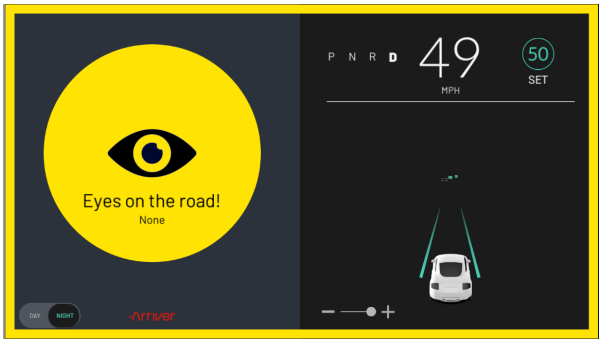

Figure 12. Interface with reduced motion setting applied

Expert Reviews

For this project, we were unable to conduct user testing with people who have accessibility needs due to administrative constraints. After meeting with our sponsors and faculty mentors, we determined that a series of expert reviews would be the best alternative. For these reviews, we set up a meeting with members of Arriver's interface design team and some PhD candidates from the university who do research on accessible user interface design.

The feedback we received from the reviews was overall very positive. They thought that our changes to the base HMI resulted in a very clean and simple interface that was not too information dense, they liked the addition of colored shapes to the alerts, stating that they made it much more obvious when the vehicle changed states, and lastly they found our accessibility settings menu to be intuitive and easy to use. However, there were some places that they found our HMI fell short. The first place was the inability to save the settings as a driver profile. In our

implementation, every time the driver gets in the vehicle, they would need to reset their settings. A driver profile system would allow multiple drivers with accessibility needs to drive the vehicle without needing to change the settings every time. Another thing they found could be improved was the brightness of our interface. We were unable to modify this due to technical limitations, however, they noted that the brightness should be able to change depending on the brightness outside and the time of day. For example, a very bright white screen at night could be very distracting, and could also make it harder for the driver to see the road. Lastly, the experts had difficulty changing their initial settings without restarting the interface. Our intention was not to allow the driver to change the settings while the vehicle was in motion, so we did not add a button to return to the settings menu. We do believe that this functionality could be added, but serious safety measures should be taken into consideration.

Section 3: Detailed Requirements & Status

Completed Requirements

- *Tactile feedback device must be small enough to place into pockets*

  We finished designing the tactile feedback devices and prototyping them with a 3D printer. The devices were two cylinders with some hollow structure in the middle to hold vibration motors, whose CAD and prototype pictures are shown in Figure 3.

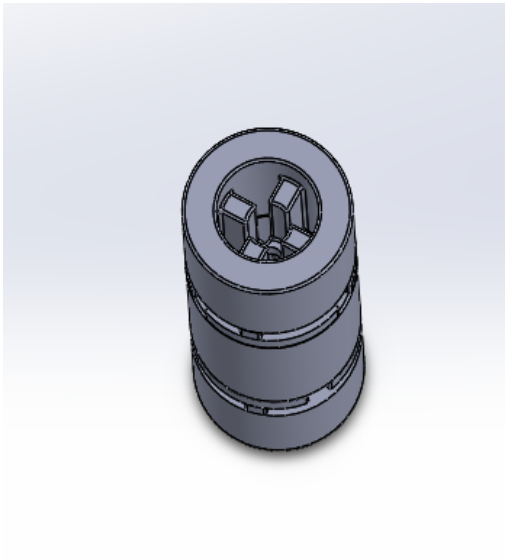

Figure 12: Top view of the device in CAD

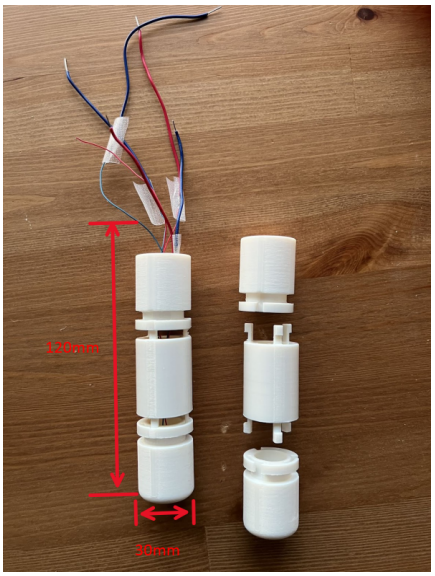

Figure 13: The side view of the 3D-printed device

The dimension of the device is 3x12cm, and it is small enough to put into a driver's pocket without feeling uncomfortable. We tested this by putting the devices into different kinds of pants worn by our teammates.

- *The device is robust enough not to be broken into parts during testing*

  The material we chose to 3D print our prototype was Rigid Opaque (Vero), which had 50 – 65 MPa tensile strength and can ensure our devices were strong enough not to fracture or yield based. This is strong enough not to break in driving situations. In addition, we also designed a lock mechanism to connect parts, which prevents the parts from coming apart while testing.

- *We must be able to run our HMI in Arriver's test vehicle*

  We were able to set up our laptop, receive real time logs from the vehicle, and display the data in our HMI. This was done while the vehicle was driving, and the interface updated in real time.

- *Different modes of vibration generated by the motors should be easy for testers to distinguish and will not negatively impact driving comfort or distract driver*

  We used an Arduino alongside Unity to control the motors and have them vibrate in three different ways. We were unable to test the motors in Arriver's vehicle due to safety concerns, but testing outside the vehicle was successful and people were able to determine the difference between different vibration patterns.

- *The modes of vibration should change along with the vehicle's state shown in HMI*

  The device needs to be able to pass information about vehicle state changes to people with hearing loss because it will be used in addition to audible alerts. We purchased a third party library that we can use through Unity to control the PWM signals to the motors. We programmed the motors to play three different vibration patterns with different intensities when the vehicle was in 'Pilot Available', 'Pilot Active', and the warning states. As the states require more immediate driver input, the motor intensity and vibration pattern become increasingly alarming.

- *HMI updates when changes are made in configuration pages.*

  We have fully implemented the accessibility settings for each of the 5 accessibility needs that we were targeting. The driver is able to select any accessibility settings they desire when the vehicle starts, and their selections are reflected in the interface while they are driving.

Incomplete Requirements

- *Modified HMI can better satisfy accessibility needs of people with: (1) Mobility issues (2) Vision loss, (3) Color Blindness (4) Hearing loss (5) Screen motion sensitivity, compared to the old HMI.*

  To confirm that our HMI better satisfies the accessibility needs of people, we needed to conduct user testing on people with accessibility needs. We had created a user testing plan for our interface. It included simulating a driving environment by showing the vehicle camera feed on one screen and the interface on another. The users would be asked to experience both the original interface and the interface with accessibility and then answer questions about their experience. Our team was going to go through the process of getting IRB approval and conducting proper user testing, however, we ran into some administrative roadblocks and were unable to complete this requirement.

Section 4: Recommendations for future work

As mentioned in Section 2, during the later stages of our project development, we consulted several experts in human machine interface design to help evaluate and give us feedback on the design of our modified HMI. This expert feedback can be summarized into two main categories of our recommendations: designs that require further validation with user testing and features recommended for future development.

Designs That Require Further Validation with User Testing

In general, to further validate our overall HMI design changes, we would need to develop a concrete user testing methodology that could be used to test our interface with people in our five target user groups. Unfortunately, we were not able to do user testing this iteration, but that is something we recommend for future work. Through an established user testing methodology, we would be able to get feedback on the overall design of our accessibility settings and the modified base interface as a whole. From that feedback, we would also be able to understand what we can improve on in our current interface that would make it more user friendly and accessible to our five main target user groups. We envision this to be an iterative process where there would be continuous feedback with every HMI change iteration. In other words, every change we make to the HMI based on user feedback could be tested again with the same user to see if it improves their overall user experience.

During our expert reviews, we also ran into questions and concerns about the layout of the HMI and accessibility setting pages. In some cases, we had experts with differing views. For example, for the text size and button size change accessibility settings, one expert suggested starting the slider in the middle to give the user the impression of being able to customize the setting (between the smallest and largest sizes) while another expert suggested starting the slider at the largest setting. In this case, to figure out the ideal starting position of the slider, we would most likely need to validate these questions and concerns with more user tests/expert review in future iterations. Likewise, some of the other questions and concerns raised by our experts would also be best resolved with user testing.

Features Recommended For Future Development

Based on the feedback we have received from experts, we have also been able to compile a list of the main features that we would recommend developing in future iterations of the project. Table 1 below summarizes our list of features and gives a brief description of how we might envision each feature at this stage of development.

Table 1: A list of features we we recommend developing to improve the overall user experience for future iterations of the project

Feature	Description
Profile saving feature	This feature would allow the user to save their selections for each accessibility setting to a user profile so that they would not need to restart the accessibility settings selection process every time they start the car.
Real-time animation	In the right panel of our interface, we currently have an animation running that would show the current position of the vehicle relative to the road. However, we have noticed that this animation is not always in sync with our vehicle state and does not show the exact location. In future iterations, we would recommend making this animation more real-time and accurate to the road conditions.
Voice commands for enhanced mobility	This feature would allow the user to navigate between pages of the accessibility settings with just their voice. For example, they could say “next” or “back” to go between different pages. Additionally, if the user was driving and they found some features to be distracting, they could also use voice commands to turn off these features (e.g. blinking animation, vibrating motors).
Layout Customization	Our interface in general could use more layout customization. In particular, one of our expert reviewers suggested adding a functionality to the base HMI that would allow for flexibility in swapping between the left and right panels in the interface. That way the user can see the appropriate screen that best matches their current driving needs (depending on whether they are driving or in the autonomous driving mode).

Appendix A: Test methodologies for 3 most critical requirements

Key Requirement 1: HMI must be able to run standalone in Arriver's test vehicle

Requirement:	HMI must be able to run in Arriver's test vehicle standalone
Specific user objective:	Our HMI needs to be able to be run in Arriver's test vehicle without being plugged into our team members laptop. We will need to figure out how to hand over a build of our project that is set up to run properly in their vehicle. This is requirement 3 in our table of requirements.
Define pass/ fail:	In order to pass, our HMI will need to properly display the following: Accurate speed of vehicle, objects surrounding the vehicle, accessibility settings configuration pages, and the gear the vehicle is in. If any of those do not display properly, this requirement will fail.
Status:	COMPLETE
Type of method:	Inspection
Who developed this method?:	Student developed: The criteria that we defined as a pass are the objects on the screen that should change when the vehicle is in different states. If they do not change, then we have created some sort of incompatibility with their vehicle in our implementation.
Experimental Apparatus:	We will need access to Arriver's test vehicle in order to validate this requirement.
Validation Method	These are the steps we will need to complete to validate the requirement: 1. Set up our HMI with the proper settings in order to run in a Linux environment in Arriver's vehicle. 2. Compile the project and provide Arriver with a binary file 3. Go to Arrivers office and run our binary on their vehicle 4. Validate that the HMI responds properly based on our pass/fail criteria
Data Collected	We will collect videos of our HMI running properly in a test vehicle.
Data Analysis and Validation Decision	We will not need to do any post processing, validation will be done in Arriver's test vehicle. If necessary we will be able to review the videos that we will take when testing.

Key Requirement 2: User testing on the accessibility features

Requirement:	User testing on the effectiveness of the accessibility features. This was done with users from Arriver's office, who did not necessarily have accessibility needs.
Specific user objective:	● User Requirements: Modified HMI can better satisfy accessibility needs of people with: (1) Mobility issues (2) Vision loss, (3) Color Blindness (4) Hearing loss (5) Screen motion sensitivity, compared to the old HMI. ● Quantitative Requirements: With limited instructions, users can correctly respond to the pilot status change in 5 seconds.
Define pass/ fail:	Pass: 1. Average score of Likert ratings questions for accessibility mode HMI is higher than that of the old one. 2. Qualitative inputs from users reflect a better experience with the accessibility features Fail: 1. Average score of Likert rating questions for accessibility mode HMI is not greater than that of the old one. 2. Qualitative inputs from users reflect no different or even worse experience with the accessibility features
Status:	COMPLETE
Type of method:	Test
Who developed this method?:	This is a student-developed test.
Experimental Apparatus:	● Touchscreen Tablet to Interact ● Monitor/ TV Screen / Laptop to Play time stamp camera view ● A low-fidelity simulated driving environment (steering wheel with a "button", brake, gas) ● Adaptations: earplugs, gloves, sport bandages ● Users: 5 - 8 abled users or users with mild to moderate disability within UMich community who can manually drive ○ At least 18 years old ○ Be licensed to operate an automobile in the United States ○ Able to understand and communicate in English ○ Have no known disorders or injuries that may affect your ability to use a touchscreen for up to 30 minutes at a time ○ Stretch goal: disabled users from UofM or local disabled communities who are still functional enough for manual driving

Requirement:	User testing on the effectiveness of the accessibility features. This was done with users from Arriver's office, who did not necessarily have accessibility needs.
Validation Method	For each user, they will see the monitor playing camera footage from a driver's view and HMI on the touchscreen. *1. Pre-test* The pre-test questionnaire was delivered verbally. "Before we begin the test, we actually have a few questions as we get warmed up." ● How many years of experience do you have with driving? ● How often do you drive? When's the last time you drove? ● Have you had any experience with autonomous driving systems? If so, how experienced are you (0: not experienced at all; 5: very experienced) ● Do you have any disabilities? We ask this question because that's related to our design goal. *2. Test for baseline and accessibility mode HMI* For abled participants, we will ask them to put on gloves/sport bandages to simulate mobility issues, turn on touchscreen's color filter to simulate color blindness, and wear earplugs to simulate hearing loss throughout the interaction with interfaces. The number of people in each simulation method will be balanced and the assignment will be random. Users will take the same test for baseline and accessibility mode HMI sequentially. The order is randomized and balanced among users to avoid the learning effect.. After each test, they will be asked to answer a short survey based on Likert ratings.
Data Collected	1. Likert rating questions score for baseline and accessibility mode HMI 2. Qualitative answers
Data Analysis and Validation Decision	1. Calculate average score from Likert scale feedback and the standard deviation. Compare the results of the baseline HMI and the accessibility featured HMI. Consider success when the accessibility featured HMI has a higher average score for all three questions and the standard deviation std < 1. 2. Gather all the notes of the qualitative data and form different themes to analyze the positive and negative attitudes participants have towards different themes. Consider success when participants show more positive feedback towards the accessibility featured HMI.

Key Requirement 3: System usability

Requirement:	System usability
Specific user objective:	The new HMI design can better inform users of system status and has higher system usability than the old HMI.
Define pass/ fail:	● **Pass**: Average score of SUS (System Usability Scale) for new HMI is higher than that of old HMI; More than 85% of users can respond to all the status changes in 5 seconds. ● **Fail**: Average score of SUS for new HMI is lower or equal to that of old HMI; More than 15% of users cannot understand and respond to all the status changes in 5 seconds.
Status:	COMPLETE
Type of method:	Test
Who developed this method?:	Recognized Standard (SUS); Sponsor Developed (Response time)
Experimental Apparatus:	● Touchscreen ● Monitor/ TV Screen ● A simulated driving environment (steering wheel with a "button", brake, gas) ● Users: at least 5 users, university students who can manually drive ○ Stretch goal: disabled users from UofM or local disabled communities who are still functional enough for manual driving
Validation Method	For each user, they will see the monitor playing camera footage from a driver's view and HMI on the touchscreen. *1. Pre-test* The pre-test questionnaire was delivered verbally. "Before we begin the test, we actually have a few questions as we get warmed up." ● How many years of experience do you have with driving? ● How often do you drive? When's the last time you drove? ● Have you had any experience with autonomous driving systems? If so, how experienced are you (0: not experienced at all; 5: very experienced) ● Do you have any disabilities? We ask this question because that's related to our design goal.

Requirement:	System usability
	2. Test for before- and after-redesign HMI Users will take the same test for before- and after-redesign HMI sequentially. The order is randomized and balanced among users. After each test, they will be asked to take the SUS. **Test for modified HMI** ● Task 1: Initial HMI setup ○ Instruction: "Now you are sitting in an Arriver vehicle's driver seat for the first time. Please set the HMI to what you feel comfortable." ● Task 2: pilot mode ○ Instruction: "Now you can start to drive. Besides manual driving mode, this vehicle has a pilot mode. In certain cases, the pilot mode will become available. The vehicle will inform you of the status of pilot mode. If pilot becomes available, you can press this button to activate the pilot mode. If you want to deactivate the pilot mode, you can turn the steering wheel slightly or step on the brake. You might also need to respond to other status changes of the vehicle and take actions accordingly. Any questions? If not, let's hit the road now! (start playing video)" ■ Task 2.1: Pilot becomes available and drivers turn on pilot ● Description: In the driver's view, a leading vehicle appears and the pilot becomes available. The HMI and audio system inform users of this status. ■ Task 2.2: Pilot becomes unavailable and driver takes over ● Description: In the driver's view, a leading vehicle leaves. The HMI and audio system inform users of this status. ● Task 3: Button interaction ○ Instruction: "While driving, you can go ahead to switch day/ night mode. And then, can you try to adjust the volume?" ● Task 4: View speed information during "manual driving" ○ Instruction: "Pretend you are driving manually on a highway. Now you need to check the speed of your vehicle just like a driver." ● Post-test questionnaire: SUS ○ Instruction: "Here is a survey for you to reflect on your experience with this system. While finishing this survey, please also think out loud and let me know why you choose a certain option." **Test for Initial HMI** Same as above except task 1.
Data Collected	1. SUS score for old and new HMI 2. Number of errors

Requirement:	System usability
	3. Response time
Data Analysis and Validation Decision	1. Calculate SUS score a. Convert the scale into number for each of the 10 questions (Strongly Disagree: 1 point; Disagree: 2 points; Neutral: 3 points; Agree: 4 points; Strongly Agree: 5 points) b. Calculate: i. X = Sum of the points for all odd-numbered questions – 5 ii. Y = 25 – Sum of the points for all even-numbered questions iii. SUS Score = (X + Y) x 2.5 2. Calculate number of errors a. When users take wrong actions b. When users don't take over the vehicle in 5 seconds 3. Calculate response time a. The time duration between pilot becomes available and users activate pilot b. The time duration between pilot becomes unavailable and users take over