Deliverables Contract

for

Mercedes-Benz AI Palette

1. Disclaimer

This document is a course assignment, not a legal document.

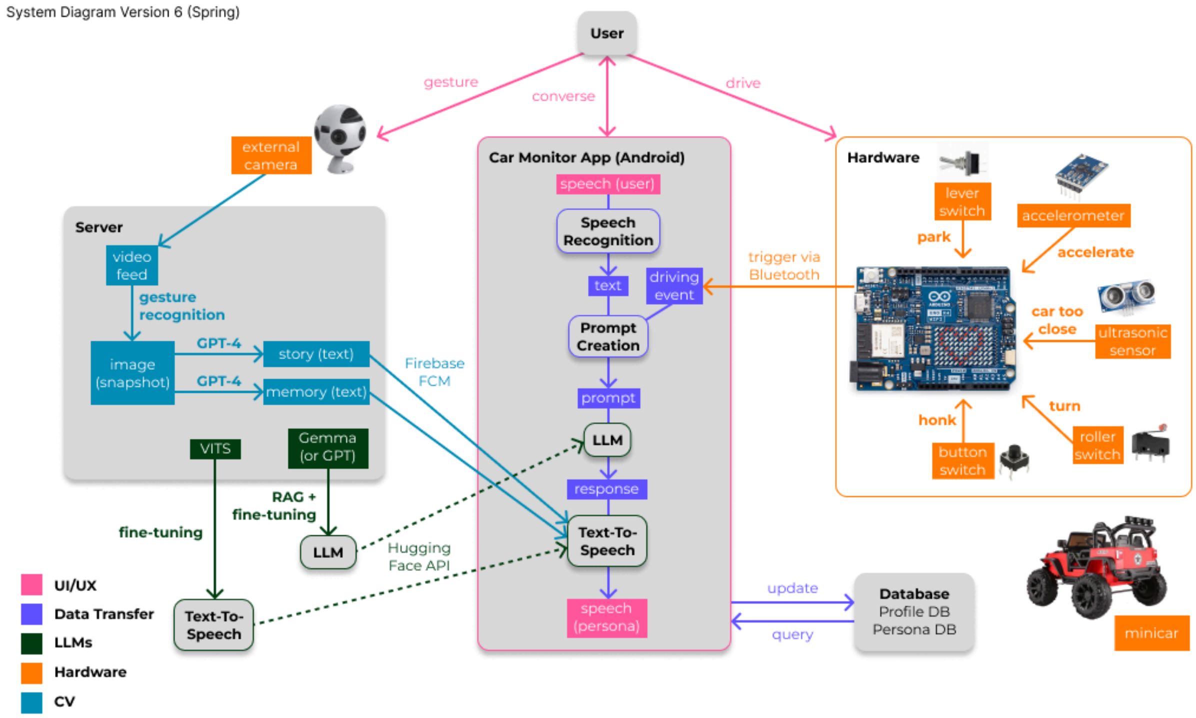
2. Parties Involved

This document outlines an agreement between the KERBY CS210 Team, the Mercedes-Benz® corporate liaison and the Stanford teaching team.

3. Deliverables

The CS210 team will deliver the following:

1. Digital documentation
2. A software prototype consisting of components detailed below that satisfy the validation criteria listed in this document.



The *Mercedes-Benz AI Palette* is a set of conversational AI personas that a driver can interact with. We crafted three personas — Football Commentator, Old Sport, and Art & Nature — to mimic the speech and personality of renowned figures like Peter Drury, Frank Sinatra, and David Attenborough. Powered by GenAI, these personas engage in dialogue, react to events such as turns and accelerations, and offer persona-specific interactions.

We built our personas on the **Android** platform using **conversation-focused LLMs** and **text-to-speech** synthesis. **Fine-tuned** on specific datasets for each real-world figure, our models capture their distinct style, voice, and tone. This ensures that conversations feel authentic and sound like a real friend.

Each persona not only engages in conversations with the driver but also reacts to various **driving events detected with sensors** mounted on a physical minicar. We aim to implement two events (honking and turns) as high-priority goals in our MVP, with three additional events (acceleration, parking, maintaining distance from other vehicles) set as stretch goals.

Additionally, each persona offers unique features: the Football Commentator provides updates on **football news**, the Old Sport **introduces new songs** (a stretch goal), and Art & Nature utilizes computer vision to **narrate landscapes** and **respond to the driver's gestures**.

To optimize performance within the vehicle, we strive to **minimize the system size** while maintaining the aforementioned functionality. Specifically, our **LLMs and text-to-speech models are limited to 2B parameters**, suitable for embedded systems. Additionally, we utilize **LangChain** for conversation retention and **RAG** for domain-specific knowledge insertion to enhance conversation quality.

4. Validation Criteria

Table I provides an overview of our system's key features and functionalities, along with the corresponding validation criteria for each aspect. Table II outlines our OKRs and KPIs to measure user satisfaction, engagement, and system optimization. We plan to conduct live end-user testing and collect feedback through post-test surveys. We will track user engagement metrics and model response time using Google Analytics for Firebase. Additionally, we will monitor the embedded system space usage to maintain optimal performance within the vehicle.

TABLE I - Feature & functionality

Component	Description of Functionality	Validation Criteria
Authentication	Users can sign up, login, logout of their account.	Press the start engine button, and if the user is not logged in, a login / signup page is displayed. From the settings screen, pressing the "logout" button logs the user out.
Persona selection	Users can select one of the three personas.	From the home screen, press the persona selection buttons to switch between the three personas anytime during the ride.
Persona settings	Users can control the persona's proactivity, talkativeness, humor.	From the settings screen, press the "proactive mode" toggle switch to control the persona's proactivity. In proactive mode, the persona may speak without explicit user request (e.g. react to turns). Press the "low," "medium," and "high" buttons to set the talkativeness or humor level of the persona.
Conversation with personas	Users can converse with the 3 personas.	Press the "talk" button or say "hey mercedes" to speak to the persona, and confirm that a cohesive response is heard reflecting the speech patterns of the real-world figure it is modeled after (Peter Drury, Frank Sinatra, or David Attenborough).

Integration with minicar sensors	Personas react to driving events detected by the minicar sensors.	In proactive mode, accelerating or pressing the honk button occasionally triggers the persona to comment on the event.
		Stretch Goal: The persona also responds to turning, parking, maintaining distance from other vehicles.
Persona-specific features	Each persona offers unique features such as football updates, music snippets, and landscape narration.	With the Football Commentator persona selected, when the user asks about "football news", the persona gives relevant news in the domain. Once every session, if in proactive mode, the persona offers to summarize the latest football news.
		When the Art & Nature persona is selected, gesture (e.g. pinch) in front of the camera to trigger narration of the surrounding landscape with computer vision.
		Stretch Goal: With the Old Sport persona selected, once every 5 songs played, if in proactive mode, the persona shares music snippets to introduce new genres.
Customization	A fully-fletched framework to easily incorporate new personas.	**Stretch Goal:** Create a custom persona via the app by inputting the persona's name, the real-world figure's name, profession, speech style, and audio clips of their voice. Once configured, the custom persona is displayed, and can be selected and interacted with.
Onboarding	New users follow an onboarding flow to set up their persona.	Sign up as a new user, and the onboarding flow consisting of initial persona selection / settings is displayed.
		Stretch Goal: The persona greets the user and introduces themself.
System optimization	Response time and embedded system space are minimized.	The total space occupied by our (theoretically) embedded system is minimized to fit within a reasonable magnitude, such as 10 GB. Our LLMs and TTS models are limited to 2 billion parameters, and the outputs are optimized with RAG and fine-tuning. The response time between user input and generated response is minimized to fit within a reasonable magnitude, such as 3 seconds. Ensure stable performance within the vehicle environment under varying conditions (e.g. app won't crash when there's no Wi-Fi).

TABLE II - OKRs & KPIs
(In each category we don't always have a target number, but rather are monitoring the KPI for improvement iteration over iteration)

Measure Category	Relevant KPI	How Will We Measure
User satisfaction	Satisfaction scores (1-10) for: 1. Overall experience 2. Football Commentator persona 3. Old Sport persona 4. Art & Nature persona 5. Intuitiveness of user interface 6. Naturalness of conversation 7. Responsiveness of conversation 8. Gesturing experience 9. Honking experience 10. Accelerating experience	Conduct user testing sessions with a representative sample. After each session, ask users to rate on a scale of 1-10 for each of the defined metrics. Report the average scores across surveys.

	Net promoter score (NPS)	Ask users to rate on a scale of 1-10 how likely they are to recommend our product to others based on their experience.
User engagement	# interactions with each persona	Use app analytics to track the number of total interactions with each persona during a session. An interaction is defined as an input-response pair to TTS, which could be triggered by the user (speaking, gesturing) or by the system in proactive mode (honking, turning, news updates, music recommendations).
	# conversations initiated by the user	Use app analytics to track how frequently a user initiates conversation with each persona during a session.
	# gestures initiated by the user	Use app analytics to track how frequently a user initiates David Attenborough computer vision narration during a session.
	# driving events detected	Use app analytics to track the number of each driving event (e.g., honking, turning) detected by the car sensors during a session.
Product quality (content)	Quality of persona responses evaluated by an external LLM, broken down into: 1. Relevance 2. Authenticity 3. Verbosity 4. Humor	Define a diverse set of possible user queries covering various topics and contexts (using GPT and/or manually). Run Python script to: 1. Generate LLM responses for each persona for each query at different verbosity and humor levels. 2. Evaluate response quality by asking GPT to rate on a scale of 1-10 for each of the defined metrics. 3. Aggregate evaluation results across all queries to calculate an overall score for each metric for each persona.
System optimization	Average response time (milliseconds)	Use app analytics to record timestamps of user input and generated responses, then calculate response time for each interaction. Determine average response time for each type of interaction.
	Embedded system space (bytes)	Track the size of the application's binary file and associated resources, such as AI models. Report the total space occupied by the (theoretically) embedded system.

5. Documentation

Full documentation of the project will be provided by the CS210 team to the corporate liaisons and the teaching team. This documentation will include the following items:

1. Project documentation in a private Github Wiki containing at the following:
    a. Problem description and design criteria
    b. Brainstorming leading to concept generation
    c. Design rationale and design analysis supporting the exploration strategy and decisions
    d. A description of the final software design
    e. Suggested sequence for future extensions of the product
    f. Validation testing description (from this document) and results
    g. Labeled sketches, schematics, drawings and/or other materials used during design and development
2. Detailed flowchart drawings illustrating the software's components and function
3. Github source code address
4. A "Quick Start" one-page guide for getting up and running using the software
5. Videos of demonstrations and user testing

6. Terms

Completion date for all software and testing is: June 8th, 2024
Final presentation to the liaison and teaching team will be held on or before June 8th, 2024
Completion date for all software: June 8th, 2024
Completion date for all documentation: June 8th, 2024
Method of access to the software and documentation will be:
https://github.com/cs210/2024-Mercedes-1/wiki. Email cs210mercedes@gmail.com to be added to our repository.