GROUP NAME: NEUTRO

Group Members:

No.	Name	Position
1	PATRICK WONG JUN WEN	Team Leader

DOMAIN 3:

ECONOMIC EMPOWERMENT THROUGH AI

GRABBUDDY – A CONTEXT-AWARE VOICE ASSISTANT FOR SAFER AND SMARTER DRIVING

Table of Contents

1.0 Introduction

GrabBuddy is a hands-free, voice-controlled AI assistant for Grab drivers (DAX). It allows drivers to operate the system using just voice, creating safer and more convenient trips. GrabBuddy is fin-tuned for Southeast Asian environments and takes in typical driving commands with voice responses that do not require physical contact. Although assistant branding incorporates the words "Hey Buddy," the current prototype invokes voice input via button press, as opposed to passive voice detection.

2.0 Problem Statements & Objectives

Problem Statement:

- Safety risks from physically interacting with the app while driving.
- Not being able to use voice commands in heavy traffic or during rain.
- Failure to understand local accents such as Malaysian English, Singlish, or Taglish.
- Lack of personalized support based on driving behaviour or context (e.g., pickup vs. idle).
- Lack of intelligent voice assistant that adapts to the different stages of a trip.

These limitations affect driver productivity and road safety, especially in Southeast Asian conditions.

Objectives:

- Enables drivers to make safe conversation using voice during trips
- Offers contextually appropriate information while driving (e.g., pickup, on the way)
- Processes incomplete voice commands and Southeast Asian accents
- Gives positive feedback, performance advice, and safety reminders

3.0 Solution Overview

GrabBuddy contains the following modules:

• **ASR**: Voice Input – captures and speech-writes out driver voice utilizing Vosk speech recognition engine.

• **Intent Recognition**: Identifies driver intent based on transcribed text (e.g., "How much income today?

• **Context Awareness**: Dynamically adjusts accessible commands and responses as a function of trip stage.

• **TTS**: Voice Output – responds via natural speech by means of text-to-speech.

• **Smart Companion Logic**: Comprises performance summaries, reminders, and safety alerts.

• **User Interface**: Simulated ususing Kivy, with buttons, status indicators, and label-based feedback.

4.0 Architecture Diagram & User Interaction Flow

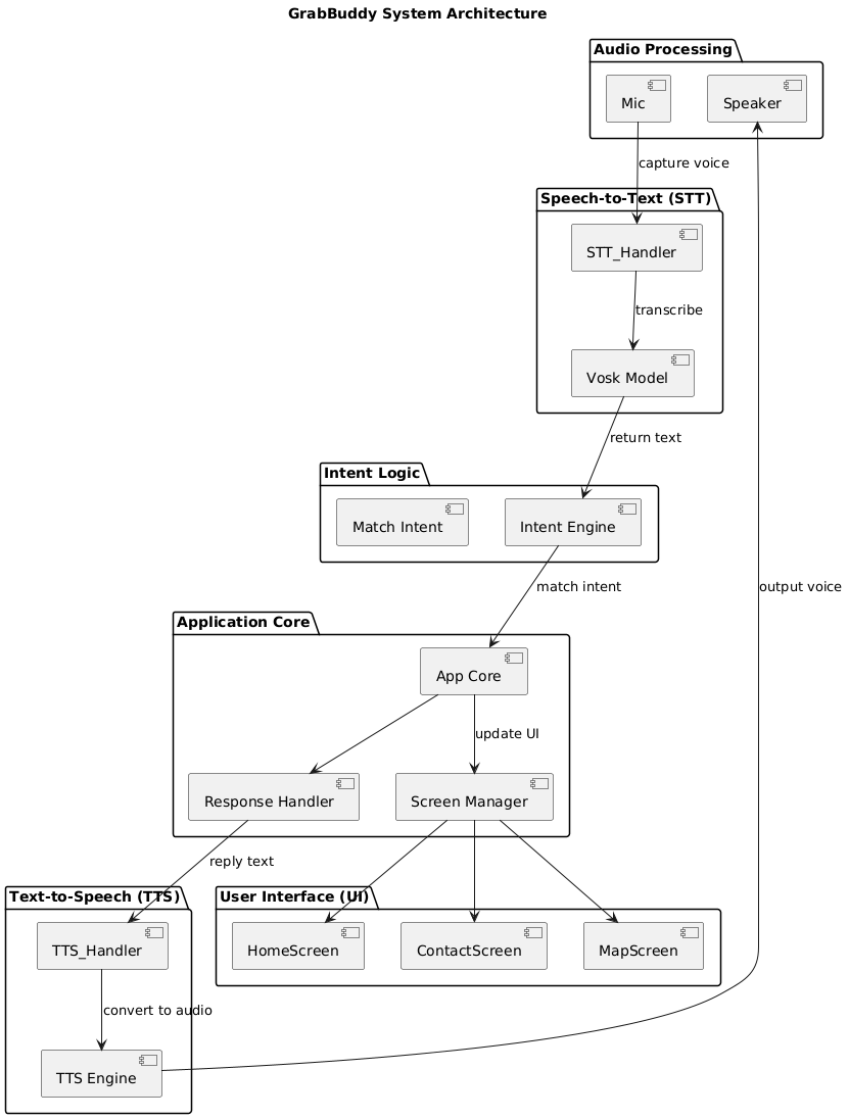

Figure 1: System Architecture

The GrabBuddy system architecture is a lean voice interaction pipeline. It starts from the microphone, where the voice of the driver is detected and processed by the Speech-to-Text (STT) module using Vosk into text. This text is forwarded to the Intent Logic, where the assistant resolves the driver's intent. The Application Core processes this intent, updates the UI, and sets a response. That response is then converted to speech by the Text-to-Speech (TTS) engine and read aloud through the speaker. The User Interface (UI) ties all layers together by providing real-time feedback on different screens like Home, Contact, and Map. This modular flow causes GrabBuddy to operate in a natural, responsive, and hands-free way.

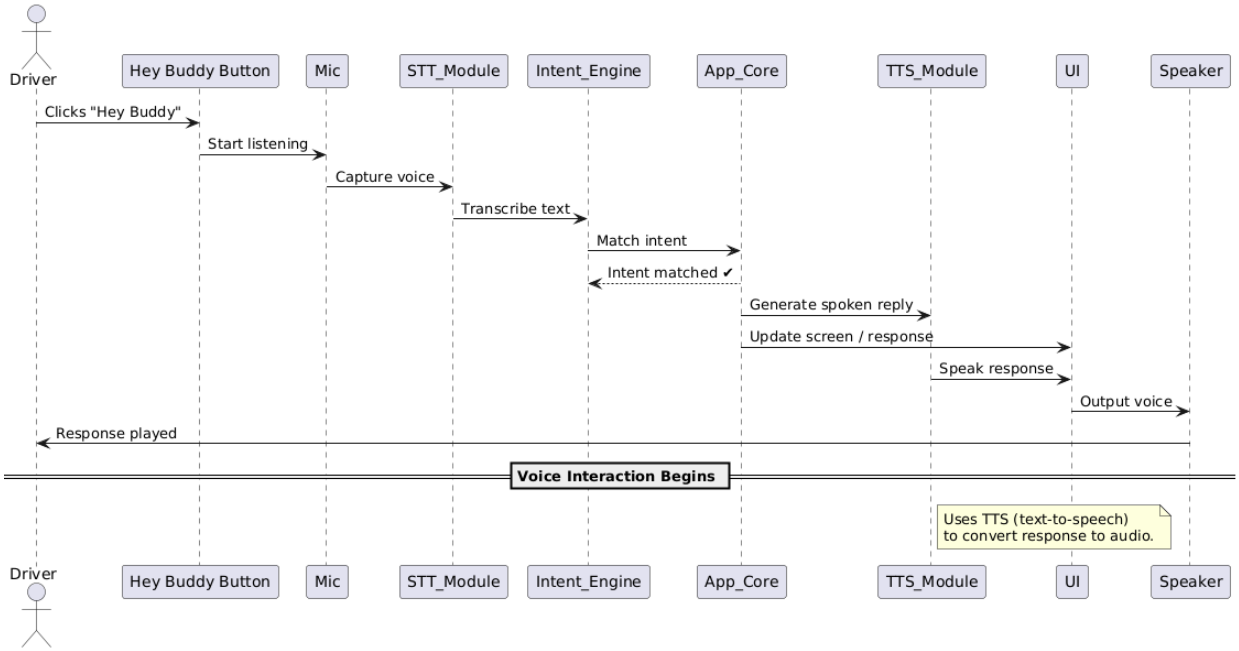

Figure 2: User Interaction Flow

The conversation begins when the driver touches the "Hey Buddy" button on the screen. It causes the system to start listening using the microphone. The "Call Hey Buddy" button mimics a wake word by hand. The assistant won't begin to listen until the user clicks on the button, rather than capturing the phrase by ear. The captured voice is sent to the Speech-to-Text (STT) module, which transcribes it into text. The text is interpreted by the Intent Engine to discover what the user wishes to command. Following the pairing, the App Core generates an ensuing response, refreshes the user interface, and sends the message to the Text-to-Speech module, which verbalizes it through the speaker. Finally, the voiced response is played out, completing the hands-free process.

5.0 Data Utilization

GrabBuddy handles user data to enable real-time voice interaction. The assistant takes in microphone input and converts it to text through Vosk, and this is processed by an intent recognition engine to determine appropriate actions. System state such as the stage of the trip (e.g., idle, pickup) is tracked internally to restrict or permit context-dependent commands. Driver performance reports (e.g., completed trips, revenue) are fake data now, but future releases can dynamically pull this. All are done locally with saved preferences and trip reports in JSON format. No user information is saved or transmitted remotely in this version.

6.0 Component Descriptions

1. Voice Input Layer

•Captures real-time voice commands through a microphone.

•Captures audio through sounddevice.

2. Speech Recognition Module

•Transcribes spoken input to text through:

- ❖ Vosk for multilingual & accent-robust transcription
- ❖ lightweight, offline operation

3. Intent Detection Engine

•Maps phrases to pre-defined commands through:

- ❖ Rule-based pattern matching
- ❖ Fuzzy logic (e.g., fuzzywuzzy) for partial input recovery

4. Trip Context Manager

•Detects current trip status of driver (idle, pickup, enroute, drop-off)

•Suppresses permitted commands depending on trip status

•Provides system reminders (e.g., "Take a break" after X trips)

5. Noise-Aware Listener

•Passively detects decibel level while in the background

•Reaction to changes in system behavior:

- ❖ Slows down TTS
- ❖ Makes things louder
- ❖ Confirms intent amid loud noises

6. Companion Logic & Feedback Engine

•Imparts human-like behaviour:

- ❖ Encouragement positivity
- ❖ Learning summaries about performance
- ❖ Safety advise depending on duration or time of day

7. Text-to-Speech (TTS) Engine

• Converts system output to voice output

- ❖ pyttsx3 (offline, cross-platform)
- ❖ Optional fallback: gTTS (Google TTS, online)

8. Front-End Simulation

- Kivy UI

Provides:

- Button-based quick actions
- Dynamic text display for guidance and response
- Listening status bar
- Integrated map and contact view per phase

7.0 Key Features & Modules

1. Trip-Aware Voice Assistant

Purpose: Adjusts assistant behaviour based on the driver's current trip phase.

States: idle, pickup, enroute, drop-off

- During pickup allows "ETA to passenger", blocks "drop-off review"
- During enroute: supports "traffic update", "re-route", etc.
- After drop-off offers trip summary or rest suggestion

Example: "You're on the way to your passenger. It's a 6-minute drive with light traffic."

2. Accent Personalization

Purpose: Supports drivers with different SEA backgrounds by allowing the assistant to support multiple English dialects (e.g., Malaysian English, Singlish, Taglish).

• Accent or auto-detect accent mode (e.g., accent_mode = "my_en")

• Intent dictionary and phrasing adaptation

• Colloquial phrase matching like:

- ❖ "Pickup liao?" → "Passenger has been picked up."

3. Noise-Aware Listening Mode

Purpose: Capturing high ambient noise (traffic, rain) and adapting interaction style.

• Monitors real-time decibel level through sounddevice and NumPy

• When noise is above threshold (e.g., 80 dB):

- ❖ Shortens responses
- ❖ Slows TTS
- ❖ Asks to repeat unclear inputs

Example: "It's a bit noisy—can you repeat that?"

4. Partial Input Recovery (Smart Guessing)

Purpose: Ensures assistant to guess noisy or incomplete inputs.

• Leverages keywords and fuzzy logic (fuzzywuzzy)

• Context-sensitive guesser (takes advantage of current trip phase to determine meaning)

Example: Input: "ETA. destination?"

Response: "Estimated time to your drop-off is 9 minutes."

5. Companion Personality

Purpose: Makes the assistant sound friendly, supportive, and human.

- Random supportive phrases
- Personalized greetings based on time or trip count
- Custom voice identity (e.g., named “Buddy” or “GrabPal”)

***Example*:** "Morning! Let’s start the day strong."

6. Driver Performance Tips

Purpose: Gives a motivational summary of driving stats after a few trips.

Data (mock):

• Number of trips

• Average ETA

• Passenger ratings

• Tips earned

Example: "You’ve completed 3 rides today! Great job! Your average trip time was 6 minutes."

7. Safety Voice Alerts

Purpose: Prevents fatigue or burnout by monitoring driving duration and trip count.

Triggers:

• 3 consecutive trips without break

• Extended driving session (e.g., >2 hours)

• Voice fatigue patterns (optional advanced feature)

Example: "You’ve been driving for a while. Would you like to rest?"

8. UI

Purpose: Simulates how the assistant would look and behave in a real car system.

Kivy app

Features:

- Voice input
- Text display
- Trip status toggle
- Real-time mic noise level
- Assistant response box

Demo Example: An animated "Start Trip" button toggles state to pickup → voice command initiates ETA check → output is vocalized + displayed on screen.

8.0 Core Functions Summary

Functions	**Description**
Voice Command Recognition (Quiet)	Provide commands like "Where is my next pickup?" in a soundproof room
Intent Detection	Verify that commands are being properly matched to intended actions
Text-to-Speech Output	After command is processed, assistant speaks the response aloud
Trip Phase Logic	Change trip phase (Idle, Pickup, Enroute, Drop-off) and provide context-based commands
Noise Resilience	Replay rain traffic sound on command input
Accent Handling	Test normal local speech patterns ("pickup liao", "ETA my place")
Partial Input Recovery	Provide incomplete sentence like "ETA. destination?"

9.0 Tech Stack

Layer	Tool/Library	Purpose
Audio Input	sounddevice	Captures and processes microphone input
ASR (Speech-to-Text)	Vosk	Transcribes spoken commands into text
NLP / Intent Matching	re, Python dicts	Recognizes user intent from partial or accented input
Noise Detection	sounddevice, NumPy	Analyses ambient noise to trigger adjustments
Text-to-Speech	pyttsx3, gTTS (fallback)	Speaks assistant responses aloud
UI (optional)	Kivy	Simulated car dashboard interface
Data Handling	JSON	Stores user trips, summaries, preferences

10.0 Current & Figma Prototype

11.0 Future Work

To evolve GrabBuddy into a production-ready voice assistant, the following enhancements are proposed:

Note: The following features are proposed for future development and are **not implemented** in the current preliminary round prototype.

1. **Wake Word Activation ("Hey Buddy")**
   - Enables true hands-free functionality
   - Firmly aligned with your safety-first mission

2. **Multilingual/Mixed Input Support (e.g., English + BM)**
   - Even straightforward phrase recognition in Bahasa can show local relevance

- Update intent dictionary accordingly

3. **UI Refinement Based on Figma**
   - Enhanced usability
   - Streamline buttons, response display, and instructions

12.0 Demo Video

https://youtube.com/shorts/RdnuEmRYY3E?si=dxF2B2sA2pQeffng