1 Introduction

This report outlines the key activities, team collaboration efforts, and project progress updates undertaken by **TEAM COSS** while preparing for the **Bosch Future Mobility Challenge**.

Particular Activities

Project Development Planning: Defined team roles and responsibilities clearly while setting overall project timelines and goals.
Vehicle System Initialization: Successfully configured the Raspberry Pi and Nucleo board, performed the vehicle's first ignition, and connected it to the dashboard.
Code Analysis and Research: Each team member studied assigned code modules such as Embedded, Brain, and Simulator, gaining a deeper understanding of the system's functionality.

Team Spirit

Regular team meetings and code reviews enabled knowledge sharing and quick resolution of hardware connection errors, essential package installation issues for running Brain code, and camera sensor malfunctions quickly and efficiently.

Changes

1) Decided to use **Jetson Nano** as the mainboard instead of Raspberry Pi due to its GPU performance, enabling AI-based model training and real-time data processing.
2) Decided to purchase **Intel RealSense Camera** to enhance system capabilities by collecting depth information and obstacle detection data.

These activities and collaborations played a crucial role in achieving the project's initial goals. The entire team responded to challenges with a flexible mindset and proactive attitude.

2 Planned activities

The following are the major activities planned during the reporting period:

1) Project Development Planning

Goal: Establish a comprehensive development plan and assign team roles.
Detailed Tasks:

- Set key development goals and milestones by phases.
- Assign roles and responsibilities to each team member.
- Create a development timeline aligned with Bosch Future Mobility Challenge deadlines and reporting schedules.

Outcome:

- Completed the project plan and main timeline chart.

2) Running Provided Code and System Testing

Goal: Execute the initial code provided by the organizers and evaluate system stability by verifying hardware and software functionality.
Detailed Tasks:

- Run the basic vehicle control code: Verify vehicle ignition, steering, and speed control functionality.

- Validate sensor data collection: Ensure that data from the vehicle's camera and IMU sensors are displayed on the dashboard in real time.

Outcome:

- Verified proper functioning of vehicle ignition and control features.
- Achieved real-time synchronization between the dashboard system and vehicle status data.
- Confirmed expected system behavior after executing the provided code.

3) Initial System Verification and Troubleshooting

Goal: Verify system functionality by setting up the Raspberry Pi OS and vehicle control system, resolving encountered issues to ensure stability.

Detailed Tasks:

- Install Raspberry Pi OS and configure the environment.
- Download required packages to run the vehicle ignition code.
- Inspect vehicle steering and hardware to address initial hardware issues.

Outcome:

- Successfully installed Raspberry Pi OS and ran vehicle ignition code.
- Resolved dependency issues by manually downloading each required package after encountering installation errors.
- Fixed the issue where the camera was not recognized by updating the configuration file.
- Repaired initial hardware issues that prevented proper vehicle steering, restoring basic vehicle control functionality.

4) Future Development and Upgrade Plan

Goal: Plan future development directions and hardware upgrades.

Detailed Tasks:

- **Add Intel RealSense Camera:** Plan to incorporate RealSense camera for depth information and object detection functionality.
- **Jetson Nano Upgrade Configuration:** Use GPU acceleration to run AI models faster and more accurately.

Outcome:

- Completed the system upgrade plan and prepared a list of required equipment.

3 Status of planned activities

Activity Name	Status	Description	Difficulties
Project Planning	Completed ✅	1. Developed timeline 2. Assigned team roles 3. Set milestones	Complexity in setting goals and timelines
Running Provided Code and System Testing	Completed ✅	1. Executed provided code 2. Verified vehicle ignition and dashboard integration	Troubleshooting Python package installation and Camera module configuration complexity
Next Steps for System Improvement	Completed ✅	1. Planned Intel RealSense camera integration 2. Jetson Nano upgrade	Device selection and compatibility review needed
Code Study and Research	In Progress (60%)	1. Studied provided code modules (Embedded, Brain, Simulator) to understand system functions	Understanding complex code structure

4 General status of the project

The project has successfully completed development planning, system initialization, and execution of the provided code. The vehicle ignition and dashboard management system are operational.

Key Progress

Development Planning: A comprehensive project development plan has been established with clearly defined team roles, responsibilities, and timelines. As this was a critical first task, the team is progressing smoothly toward each milestone.
Vehicle Control Status: The vehicle's ignition, speed, and steering controls are functioning properly, with real-time status monitoring enabled through the dashboard.
Code Analysis and Research: The team is studying code modules such as Embedded, Brain, Computer, and Simulator, focusing on understanding the complex code structure and system integration methods. Once the code analysis is complete, the project will move on to the next development phase immediately.

The project has successfully achieved its initial goals, and preparations for the next development stage are complete.

5 Upcoming activities

During the next reporting period, the team will focus on AI model selection, data preparation, sensor data integration, and basic lane-keeping system development. Each task aims to enhance system stability and autonomous driving functionality.

1) AI Model Development and Data Preparation

Goal: Establish initial models and collect data for key AI functions such as lane detection and obstacle recognition.
Detailed Tasks:

- **Model Selection & Dataset Collection:** Train a lightweight lane detection model using pre-trained datasets with data augmentation.

2) Sensor Data Integration

Goal: Build a vehicle environment recognition system by collecting and integrating various sensor data.
Detailed Tasks:

- **Camera Data Processing:** Process depth and visual data from Intel RealSense and develop lane and object detection using OpenCV.
- **IMU Sensor Data Collection:** Collect vehicle motion data from the IMU and enhance accuracy with Kalman filtering.
- **Multi-Sensor Integration:** Synchronize and integrate data from cameras and IMU sensors to update vehicle status and surroundings in real time.

3) Basic Lane-Keeping System Development

Goal: Design a basic lane-keeping system for road driving.
Detailed Tasks:

- **Lane Detection Algorithm Development:** Enhance Road boundary detection using OpenCV filters, ROI settings, and image preprocessing.
- **Lane-Keeping Control Logic Implementation:** Create vehicle steering angle control logic with developing PID Control System
- **Vehicle Testing Environment Setup:** Set up a simulation environment to test and adjust the lane-keeping algorithm on various road conditions.