4.2 Data Design (Table Design)

4.2.1 Schema Design

The data design for EcoSweep is structured to define the flow of command and sensor data between the Mobile App, Raspberry Pi, and Arduino Mega. The following schemas describe the structure of data packets exchanged during operation.

☑ Table 1 – Command_Data

Field Name	Data Type	Description	Example Value
command_id	Integer	Unique identifier for the command	1
timestamp	Datetime	Time the command was sent	2025-09-10 12:45:30
mode	String	Mode of operation (‘manual’, ‘semi-auto’)	‘manual’
action_type	String	Type of action (‘move’, ‘arm’, ‘stop’)	‘move’
direction	String	Movement direction (‘forward’, ‘backward’, ‘left’, ‘right’)	‘forward’
speed	Integer	Speed percentage (0–100)	70
arm_position	Integer	Servo motor position (0–180 degrees)	90

▶ Example Row:

command_id	timestamp	mode	action_type	direction	speed	arm_position
1	2025-09-10 12:45:30	manual	move	forward	70	NULL

☑ Table 2 – Sensor_Data

Field Name	Data Type	Description	Example Value
sensor_id	Integer	Unique identifier for the sensor reading	1
timestamp	Datetime	Time of the sensor reading	2025-09-10 12:45:31

sensor_type	String	Type of sensor (‘ultrasonic’, ‘GPS’, ‘IMU’, ‘Compass’, ‘IR’)	‘ultrasonic’
sensor_value	Float	Measured value (e.g., distance in cm or coordinates)	25.3

▶ Example Row:

sensor_id	timestamp	sensor_type	sensor_value
1	2025-09-10 12:45:31	ultrasonic	25.3

4.2.2 Data Integrity and Constraints

To ensure consistent, valid, and reliable operation of the EcoSweep system, the following data integrity rules and constraints are applied to the above schema design:

☑ Primary Key Constraints

- `command_id` is the primary key in the **Command_Data** table.
- `sensor_id` is the primary key in the **Sensor_Data** table.

This ensures every command and sensor reading is uniquely identifiable.

☑ Data Type Constraints

Field	Constraint	Example
speed	Integer between 0 and 100	70
arm_position	Integer between 0 and 180	90
timestamp	Not NULL – Must follow proper datetime format	2025-09-10 12:45:30
mode	Enum: ‘manual’, ‘semi-auto’	‘manual’
action_type	Enum: ‘move’, ‘arm’, ‘stop’	‘move’
sensor_type	Enum: ‘ultrasonic’, ‘GPS’, ‘IMU’, ‘Compass’, ‘IR’	‘ultrasonic’
sensor_value	Float, NOT NULL	25.3

☑ NOT NULL Constraints

- Fields `timestamp`, `mode`, and `action_type` in **Command_Data** must always have a value.
- Fields `timestamp`, `sensor_type`, and `sensor_value` in **Sensor_Data** must never be null.

☑ Logical Data Relationships

- Each command generated by the mobile app is assigned a unique `command_id`.
- Sensor data is captured periodically and assigned a unique `sensor_id`.
- Both data types are time-synchronized using `timestamp` to track the exact sequence of operations and sensor readings for debugging or future AI processing.
- Future expansion can allow commands and sensor readings to be logged in a database for offline analytics.

☑ Final Summary of This Section

- The **Command_Data schema** captures structured commands with explicit control over movement, speed, and robotic arm position.
- The **Sensor_Data schema** provides structured environmental awareness information used by Arduino for basic navigation and future AI expansion.
- Proper constraints (primary keys, data type limits, NOT NULL fields) ensure the system remains robust and fault-tolerant.