Position: Developer / Programmer

Appliance: Smart Vacuum Cleaner in HACS

1. Introduction

In our HACS (Home Appliance Control System) project, I contributed as the Developer. My duty was to transform the requirements and design of the Smart Vacuum Cleaner module into a functioning system. In simple words, I was responsible for bringing the appliance “to life” within our project.

The Smart Vacuum Cleaner was chosen because it represents a modern, intelligent appliance that fits perfectly into a smart home. People today want convenience, automation, and remote control, and a smart vacuum cleaner provides all of these. It can clean without human effort, follow schedules, avoid obstacles, and notify the user if something is wrong.

My role was to make sure all of these features could be realized inside the HACS system. While the requirements engineer told us what features were needed, and the designer showed how the modules would connect, my responsibility was to make it actually work.

2. Role of a Developer (Theory)

In software engineering, the developer is central to the system-building process. Theories in software development state that:

The analyst identifies what the system should do.

The designer shows how the system should look and behave internally.

The developer actually creates the system based on these guidelines.

The tester checks the developer’s work.

The developer’s role includes several important responsibilities:

1. Implementation – converting diagrams and requirements into working features.

2. Error handling – making sure the system works even when something goes wrong.

3. Collaboration – working with other team members to solve issues.

4. Integration – ensuring that the module works not only by itself but also as a part of the whole system.

This means a developer is not just a coder. A developer is someone who makes sure that the vision of the project becomes a real product.

3. Purpose of My Contribution

The purpose of my work was clear: to ensure the Smart Vacuum Cleaner could be controlled smoothly within HACS. My tasks involved:

Turning the listed functional requirements into working features.

Making sure the appliance could respond correctly to commands like ON, OFF, Auto Clean, Spot Clean, Pause, Resume, and Cancel.

Ensuring that the scheduling feature worked so the vacuum could clean automatically at chosen times.

Building the part of the system that could send alerts for battery and dust bin status.

My contribution connected the theoretical parts of the project to its practical output. Without this step, the vacuum cleaner would remain just an idea, not an actual module.

4. Implementation Work I Carried Out

To carry out my work, I first studied the requirement document. The analyst had described each function in simple terms, and I used that as my checklist. Then I went through the diagrams created by the system designer. These showed me how the user, HACS server, and vacuum cleaner would communicate.

Once I understood the big picture, I started breaking the task into smaller parts:

Remote ON/OFF: Ensuring the user could start or stop the vacuum cleaner through HACS.

Auto Clean Mode: Allowing the vacuum to clean all available areas without user direction.

Spot Clean Mode: Enabling the vacuum to focus only on one area of the house.

Scheduling: Storing the time chosen by the user and starting the vacuum automatically at that time.

Pause and Resume: Giving the user the option to temporarily stop the vacuum and then continue.

Cancel Operation: Stopping the cleaning completely and bringing the vacuum back to idle state.

Alerts: Sending notifications if the battery was low or if the dust bin was full.

Each feature was carefully thought out so that it matched both the design and the real-life use of a smart vacuum cleaner.

5. Challenges I Faced

Working as a developer was not always easy. I faced several challenges during this process:

State Management: The vacuum cleaner had many possible states: ON, OFF, cleaning, paused, resumed, or cancelled. Making sure the system behaved correctly when moving between these states was sometimes confusing.

Scheduling Logic: Scheduling was one of the most challenging features, because the system had to keep track of time and start the operation exactly when required.

Alerts: Setting the right conditions for alerts was another challenge. For example, deciding when the dust bin should be considered “full” or when the battery should be marked as “low.”

Integration with HACS: I also had to ensure that the vacuum cleaner module worked well with the rest of the appliances in the system. This meant extra care in communication between the appliance and the server.

Even though these were challenges, solving them gave me deeper learning.

6. Team Collaboration

One of the most important parts of being a developer is team collaboration. I did not work alone. My contribution was possible only because of the efforts of other team members.

The Requirements Engineer gave me a list of features. Without this, I would not know what to build.

The System Designer prepared diagrams. These diagrams guided me in structuring my work.

The Tester later verified my work and pointed out if anything went wrong.

This teamwork taught me that software development is never a one-person job. Every member’s contribution is linked to the others.

7. Outcome of My Contribution

The final outcome of my contribution was a working Smart Vacuum Cleaner module that met the given requirements. The system responded to remote commands, supported multiple cleaning modes, allowed scheduling, and generated alerts.

Because of my implementation:

Users could control the appliance easily.

The system performed cleaning operations automatically.

Alerts made the appliance safer and more reliable.

The vacuum cleaner was successfully integrated into the overall HACS system.

My work transformed the theoretical parts of the project into a practical result.

8. Learning and Reflection

This project was a valuable learning experience for me. From the theoretical side, I learned about the Software Development Life Cycle (SDLC), where implementation comes after design. I also learned about the importance of structured thinking in system development.

From the practical side, I experienced the real challenges of being a developer. I learned that even simple features need careful planning. For example, a pause button may look easy for the user, but behind the scenes, it requires careful management of system states.

I also realized the importance of communication. Whenever I was unsure about a requirement, I asked the analyst for clarification. Whenever I found a design unclear, I consulted the system designer. This teamwork helped me complete my task successfully.

9. Conclusion

As the Developer, my contribution was to implement the Smart Vacuum Cleaner module of the HACS project. I converted requirements and designs into working functions, solved challenges during implementation, and collaborated with the team. My work ensured that the appliance was not just an idea but a real part of the system.

This role taught me patience, problem-solving, and the importance of teamwork. It showed me how a project moves from paper to practice and how every feature has to be carefully thought out. My contribution helped make the HACS project successful, with the vacuum cleaner module functioning smoothly within the system.