Software Engineering (SE) Lab 1 Report

Faisal Ahemmed (Partner A), Mir Md Redwon Sagor (Partner B)

12th April 2025

Authors

Family name	First name(s)	Partner A or B
Ahemmed	Faisal	A
Sagor	Mir Md Redwon	B

Project Details

- **Project Name:** DustDash
- **GitLab Project ID:** 68920804
- **GitLab Repository:** `https://gitlab.com/robotvac-lab1-se/vacuum-cleaner`

Glossary

Term	Acronym	Description
Robot Vacuum	RV	Robotic Vacuum Cleaner
Firmware	FW	Embedded software running on the RV
User Story	US	Short, structured requirement description
Wi-Fi	WIFI	Wireless network interface
Cloud System	CS	Online service to register and sync RV

1. Market Analysis for Robot Vacuum

To understand the current landscape of smart robotic vacuum systems, we conducted a comparative market analysis of three leading products: **Xiaomi Mi Home**, **iRobot Roomba**, and **Ecovacs Deebot**. These systems were selected based on their popularity, advanced feature sets, and integration with mobile platforms.

The evaluation focused on the functionality and usability of their corresponding mobile applications, particularly their support for:

- Smart navigation and mapping using sensors (LiDAR, IR, etc.)
- Cleaning mode flexibility (auto, room-based, zone/region selection)

- User interaction and remote control via Wi-Fi
- Cloud connectivity for multi-device access
- Update and maintenance workflows (e.g., OTA firmware updates)

Each app was assessed for ease of use, responsiveness, customization options, and system feedback mechanisms. Xiaomi's Mi Home app, for example, provides a robust interface with real-time mapping and region-based cleaning, while iRobot's app offers seamless room management and voice assistant integration. Ecovacs, although simpler, features effective virtual wall settings and clear job summaries.

Screenshots and feature breakdowns from each product are included in **Appendix A**.

2. Functional Requirements

- The app shall start the robot in auto, room, or region mode.
- The app shall allow the user to view the live map of the cleaned area.
- The robot shall return to the dock when the cleaning is completed or battery is low.
- The robot shall avoid obstacles and detect staircases using sensors.
- The app shall maintain a history log of past cleaning jobs.
- The robot firmware shall process and execute commands sent by the app.
- The app shall allow robot registration via Wi-Fi and cloud synchronization.

3. Non-Functional Requirements

- The app shall update the robot status (map, battery, job) every 30 seconds.
- The communication between app and robot shall be secured using encrypted protocols (e.g., TLS).
- The app interface shall allow any operation in no more than three taps (usability).
- The system shall be available with 99.9% uptime annually.
- The robot shall be able to recover from faults or disconnections within 10 seconds.
- Firmware updates shall be delivered over-the-air (OTA) with minimal user interaction.

4. Stakeholder List

Stakeholder	Role
End User	Operates the app and RV
App Developer	Builds and maintains mobile application
Embedded Developer	Programs and tests the firmware
Product Manager	Defines requirements and priorities
UX Designer	Designs intuitive app UI
QA Tester	Validates features and performance
Security Analyst	Ensures data safety and access control
Customer Support	Assists users with issues or bugs

5. Kano Classification

Feature	Kano Category
Auto Mode	Must-be
Room Mode	Performance
Region Mode	Performance
Live Map	Performance
Job History	Performance
Robot Discovery	Must-be
OTA Updates	Must-be
Obstacle Avoidance	Must-be
Fall Detection	Must-be
Voice Assistant Support	Delighter

6. App User Stories (Partner A)

- As a user, I want to start the RV in auto mode so that it cleans the whole area.
- As a user, I want to define and clean custom zones via the app so that I can target specific areas.
- As a user, I want to view the RV's progress on a live map so that I know what's cleaned.
- As a user, I want to register my RV in the cloud so that I can access it from multiple devices.
- As a user, I want to receive error notifications from the RV so that I can take action.

7. Firmware User Stories (Partner B)

- As a system, I must scan and map the environment using LiDAR so that the RV can navigate.
- As a system, I must detect obstacles in real time so that the RV can avoid them.

- As a system, I must stop the RV from falling down steps to prevent damage.
- As a system, I must execute start/pause/home commands received from the app.
- As a system, I must report cleaning status every 30 seconds to the app.

References

- SE_01a_Renz_Introduction.pdf
- SE_01b_Renz_ReqEng-1.pdf
  - Functional and Non-Functional Requirements – Pages 9–14
  - Stakeholder Identification – Pages 21–24
  - Kano Model – Pages 33–36
  - User Stories and INVEST – Pages 39–42
- `https://www.mi.com/global/product-list/vacuum-cleaner/robot-vacuum/`
- `https://www.irobot.com/en_US/home`
- `https://www.ecovacs.com/de/deebot-robotic-vacuum-cleaner`

Appendix A

(Market analysis screenshots, links to studied product pages, raw notes from Xiaomi Mi Home and Roomba apps.)

Appendix A: Market Analysis Screenshots

Figure 1: Xiaomi Mi Home App - Home Interface

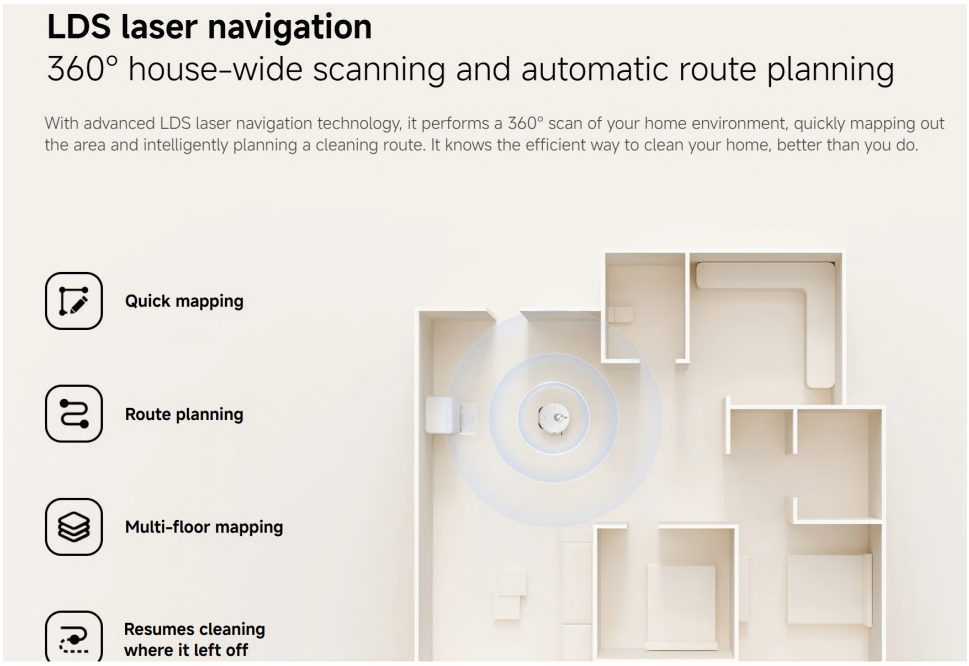

Figure 2: Xiaomi Mi Home App - Cleaning Mode Options

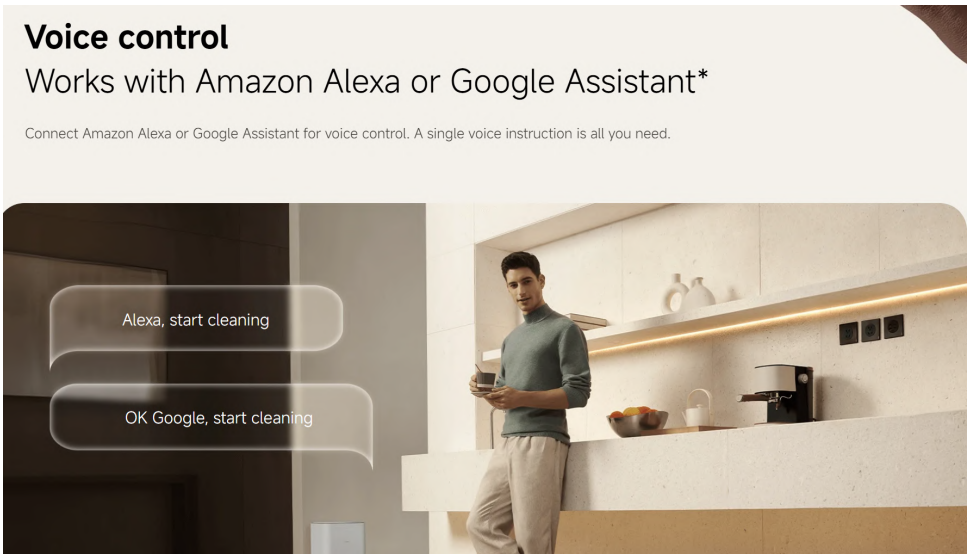

Figure 3: Xiaomi Mi Home App - Map View

Figure 4: iRobot App - Room Mapping Interface

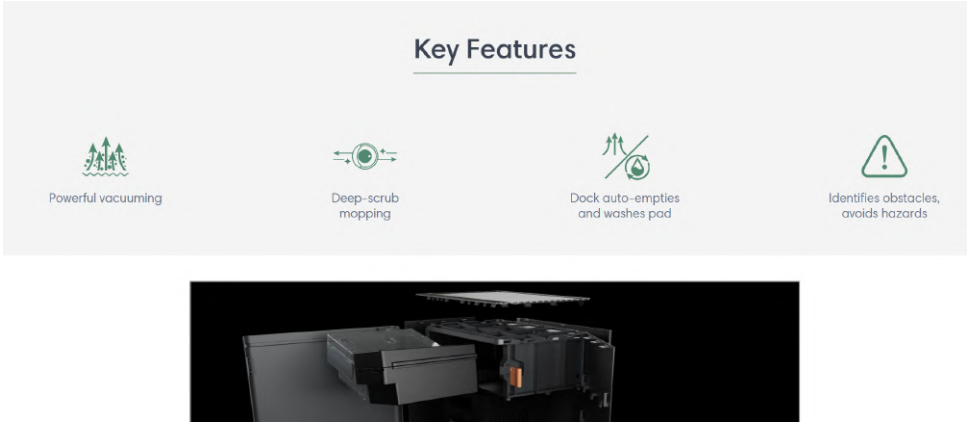

Figure 5: iRobot App - Cleaning History

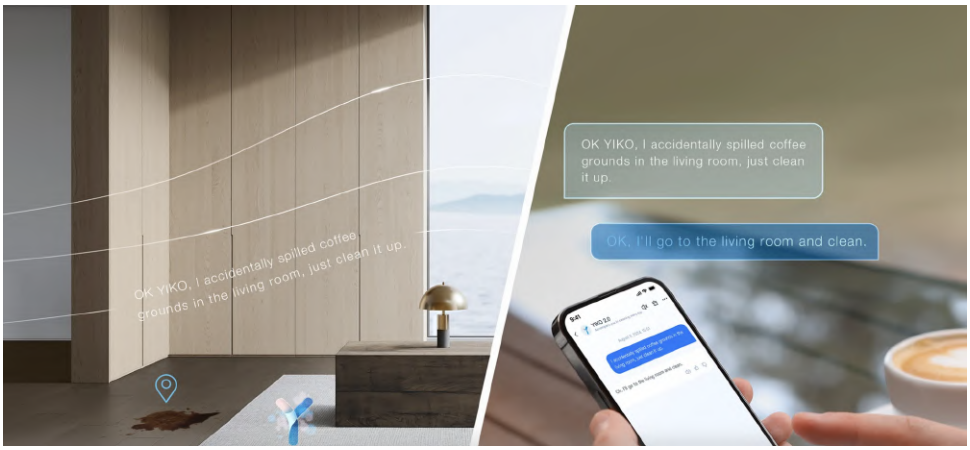

Figure 6: Ecovacs App - Virtual Wall Setup