# S Dharshini

**2403717624322010**

# Home Appliance Control System

# 1 Introduction

The **Home Appliance Control System (HACS)**, developed for the **23AD313 Software Engineering Principles and Practices coursework**, enables **remote monitoring** and **control of appliances** for **convenient** and **efficient household management**. This document outlines **functional** and **non-functional requirements** along with **system constraints**, serving as the primary output of the **Requirements Elicitation Phase**. It establishes a clear **baseline** for both **instructors (clients)** and the **student team (developers)**, forming the foundation for **system design, implementation, and testing**.

## 1.2 About the Project

## The aim of this project is to perform ****Requirements Elicitation**** and ****Specification**** for the **Home Appliance Control System (HACS)**, focusing on **identifying**, **analyzing**, and **documenting requirements** as per the **23AD313 Software Engineering Principles and Practices** course. The scope is limited to **elicitation** and **documentation of requirements**, while later phases like **design**, **implementation**, and **testing** will be addressed in future work.

## 1.3 Document Scope

## This document defines the **requirements** for the **Home Appliance Control System (HACS)**, covering **functional/non-functional requirements**, **features**, **use cases**, **actors**, and **constraints**. It focuses solely on the **requirements elicitation and specification phase**, excluding **design**, **coding**, **deployment**, and **maintenance**.

## 1.4 Terminology Used

|  |  |
| --- | --- |
| **Term/Acronym** | **Definition/Description** |
| Developers | The team responsible for the complete development of the Home Appliance Control System (HACS). |
| Process | The set of activities followed by students to complete the requirements elicitation, documentation, and submission for the assignment. |

|  |  |
| --- | --- |
| Requirement | A condition or capability needed by a user to solve a problem or achieve an objective within HACS. |
| Specification | A document that prescribes, in a complete, precise, and verifiable manner, the requirements, design, behaviours, or other characteristics of a system or its components. |
| User | The person operating and/or using the HACS system through devices like mobile, desktop, or palm-top. |
| HACS | Home Appliance Control System – a system that enables remote control, monitoring, and coordination of appliances such as microwave oven, sprinklers, pet feeder, etc. |
| Session | A valid communication link established between the remote device and the HACS Appliance Controller after authentication. |
| System Administrator | The authorized person responsible for managing users, adding/modifying appliances, and overseeing system operations. |

## 1.5 Related Documents

The following documents and resources have been referred to in preparing this Software Requirements Specification (SRS) for the Home Appliance Control System (HACS):

1. **Assignment Brief** – Requirement Elicitation and Documentation: Home Appliance Control System (HACS), Department of AI and DS, Course Code: 23AD313.
2. **Document Standards** – Provided by the Department of AI and DS for formatting and structuring the SRS.
3. **Software Engineering Principles and Practices Course Material** – Lecture notes and reference slides related to requirements elicitation, use case modelling, and specification.
4. **IEEE SRS Template (IEEE 830-1998 Standard)** – Used as a reference framework for preparing this document.

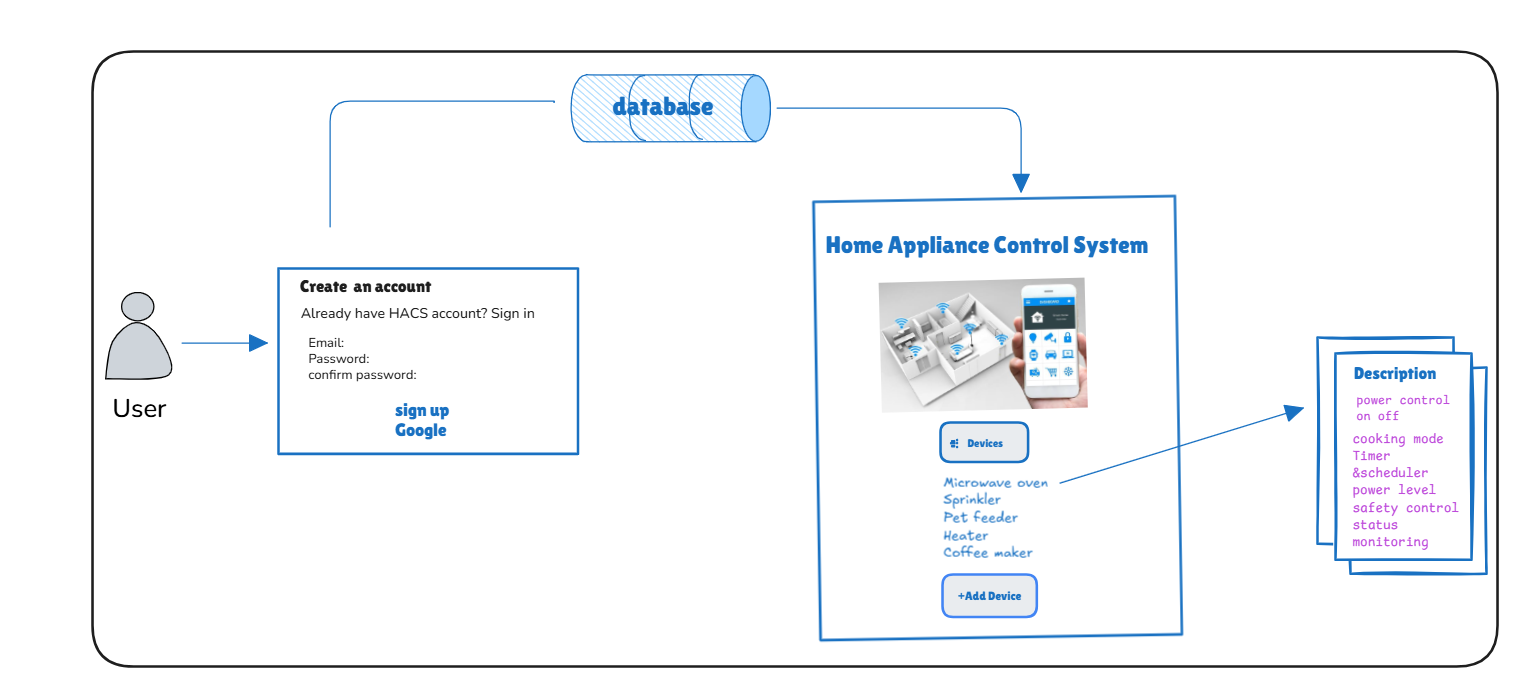
## 1.6 Document Overview

This document is organized into multiple sections to clearly define the requirements of the **Home Appliance Control System (HACS)**.

* **Section 1: Introduction** – Provides the purpose, scope, terminology, related documents, and overview of this SRS.
* **Section 2: System Context** – Identifies system actors, describes interactions, and presents the context diagram.
* **Section 3: Use Case Modeling** – Details the use case diagram, textual descriptions of use cases, exceptions, and variations.
* **Section 4: Functional Requirements** – Specifies the core functions grouped by use cases (e.g., microwave, sprinkler, pet feeder, etc.).
* **Section 5: Non-Functional Requirements** – Defines performance, security, usability, adaptability, and reliability requirements.
* **Section 6: Administrative Features** – Requirements related to system administrator activities such as adding/deleting appliances or users.
* **Section 7: Appendices** – Contains supporting materials like glossary, references, and diagrams.

# 2 Product Overview

The **Home Appliance Control System (HACS)** enables users to **remotely monitor and control appliances** such as **microwaves**, **sprinklers**, **security systems**, **entertainment devices**, **air conditioning**, and **pet feeders**. It is accessible through **smartphones**, **desktops**, and **tablets** using the **Internet** or **WAP**, offering **secure login**, **real-time status**, and **user-friendly controls**. Examples include **turning on sprinklers**, **scheduling a pet feeder**, or **preheating a microwave** before arriving home. Designed for **flexibility** and **scalability**, HACS supports the easy addition of **new appliances**. It ensures **reliable** and **secure communication** for efficient household management.



## 2.2 Business Services Supported

The **Home Appliance Control System (HACS)** shall provide software support for the following key operations:

1. **User Authentication** – Validate users through login credentials before granting access to the system.
2. **Session Management** – Establish and maintain a valid communication session between the remote device and the HACS Appliance Controller.
3. **Appliance List Display** – Provide the user with a list of all available appliances that can be accessed and controlled.
4. **Appliance Status Monitoring** – Allow users to view the current state of an appliance (e.g., ON/OFF, scheduled, running time, etc.).
5. **Appliance Control Operations** – Enable users to perform operations such as switching appliances ON/OFF, adjusting settings, or scheduling activities.

**2.3 Product Characteristics**  
HACS enables remote appliance monitoring and control via smartphones, desktops, and tablets with secure login, real-time status, and user-friendly controls. It is scalable, reliable, and supports easy integration of new devices.

**2.3.1 User Interface**  
The system provides intuitive GUIs across platforms with login screens, appliance menus, control panels (e.g., microwave, sprinkler, pet feeder), and dashboards for real-time status. Design ensures clarity, responsiveness, and easy navigation.

**2.3.2 Operating Environment**  
HACS runs on PCs (min. Pentium 800 MHz, 256 MB RAM, 150 MB disk, Windows 2000) and is platform-independent via JVM, supporting Windows, Linux, and macOS for flexibility.

**2.3.3 Hardware Interfaces**  
Interfaces include a USB module for appliance controller connection, appliance control modules, handheld wireless controller for redundancy, network interface (Wi-Fi/Ethernet) for connectivity, and optional sensor inputs for automation.

**2.4 User Characteristics**  
HACS will be used by homeowners, family members, administrators, and occasionally service technicians. Homeowners and residents need only basic smartphone/computer skills to log in, navigate menus, and control appliances. Administrators require moderate technical knowledge for configuration, networking, troubleshooting, and security management. Service technicians may assist with maintenance or upgrades, requiring strong technical expertise.

**2.4.1 General User Characteristics**  
Users should be able to read English, operate smartphones/PCs, understand household appliances, and follow on-screen instructions. A willingness to use a simple graphical interface is expected.

**2.4.2 Admin Characteristics**  
Admins can configure appliances, manage accounts, set up networking, handle troubleshooting, monitor logs, and apply security practices like strong passwords.

**2.4.3 Homeowner/Resident Characteristics**  
Homeowners/residents only need basic device operation skills and familiarity with appliances. No advanced knowledge is required, as the interface is intuitive. They should be able to follow instructions for tasks like ON/OFF control and timers, with interest in convenience, safety, and energy efficiency.

**2.5 General Constraints**

The following general design and implementation constraints apply to the **Home Appliance Control System (HACS):**

* The software system will be primarily developed on and run under **Windows 2000 or above**.
* All code shall be developed to run on a **Java Virtual Machine (JVM) supporting Java 1.3.1 or above**, ensuring platform independence.
* **Plain text files** shall be used for all text-based data storage and reporting, unless otherwise specified in future upgrades.
* The documentation and source code shall conform to **CITTech documentation and style standards**.
* The system shall require a **continuous power supply and stable internet/network connectivity** for proper appliance monitoring and remote access.
* The system shall comply with **basic security standards**, including authentication and user-level access control.

## 2.6 Priority of requirements

The following requirements are listed in the order from the most critical to the least important:

* **User Authentication & Security** – Ensuring only authorized users (homeowners/admins) can access and control appliances.
* **Appliance Control** – Ability to switch appliances on/off and adjust their settings remotely.
* **Scheduling & Automation** – Support for timers and automation rules to control appliances without manual intervention.
* **System Monitoring** – Real-time status updates of appliances (e.g., ON/OFF state, fault alerts).
* **User Management** – Admin ability to add/remove users and assign roles (homeowner, resident, etc.).
* **Reporting & Logs** – Generation of basic usage reports and activity logs for monitoring.
* **Cross-Platform Compatibility** – Ability to run on multiple operating systems supporting JVM.
* **User Interface Enhancements** – Graphical interface improvements, themes, or customization for better usability.