Unified Home Automation Interface

Software Requirements Specification

Version 2

5th November 2019

Submitted in partial fulfillment of the requirements of
Software Engineering (IT- 3003)
Instructor: Dr. H. K. Tripathy

Made by: Team int_elligence

1706040	Himanshu	Project Designer
1706042	Ishani Aich	Software Tester/Maintainer
1706044	Keshav Agarwal	Technical Analyst
1706046	Krishnanya Swami	Systems Analyst
1706048	Kushal Agarwal	Project Manager

School of Computer Engineering
Kalinga Institute of Industrial Technology
Deemed to be University
Bhubaneswar-751024

Autumn 2019

Revision History

Date	Description	Author	Comments
21-08-2019	Presentation of the first version of the SRS.	Kushal Agarwal	Presentation upto Functional Requirements
05-11-2019	Presentation of the entire SRS document.	Kushal Agarwal	Final refining and presentation of entire SRS document.

Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Roll No	Member Name	Designation	Date	Signature
1706040	Himanshu	Project Designer	05-11-2019	
1706042	Ishani Aich	Software Tester/Maintainer		
1706044	Keshav Agarwal	Technical Analyst		
1706046	Krishnanya Swami	Systems Analyst		
1706048	Kushal Agarwal	Project Manager		
<Emp. Code>	Dr. H. K. Tripathy	Instructor, SE (IT 3003)		

Table of Contents

List of Figures

1. Introduction

There is an increasing demand for smart homes, where appliances react automatically to changing environmental conditions and can be easily controlled through one common device. Interconnected devices enable to intelligently monitor and control smart homes in future .Energy saving applications, for example, control indoor climate and electricity usage by employing information to switch off appliances (e.g., lights, fans), reduce room temperature, close windows, or stop warm water circulation. The main aim of the project is to focus on building a single GUI with which the user can communicate with all their home automation devices, in order to increase convenience for the users in established markets, and to reduce the taboo or fear of home automation in developing markets, by ensuring ease of use, and strictly secure communication lines.

1.1. Purpose

This project provides an overview of how a home automation system works. It will describe all the features of home automation and what is the purpose behind making it. It provides a description of all the functions, specifications, external behaviors, design constraints, requirements (function and non-functional) and other factors necessary to provide a complete and comprehensive description of the proposed document.
This document is written primarily for the stakeholders, such as the developers, programmers and maintainers of the system.

1.2. Scope of Project

The scope of the home automation project is to create an environment for the smart home concept and is for the developers of this project. This prototype shall be used to further develop the project, as well as intiate interest in smart homes. Specifically, this system is designed to create an environment to manage and communicate with the basic home appliances such as lights, fans, automatic curtains, garage doors, cameras, refrigerators, control room temperatures, among others.

1.3. References

- The invaluable teachings of Dr. Hrudaya K. Tripathy
- “Fundamentals Of Software Engineering” by Rajib Mall
- https://www.scribd.com/doc/76026766/Home-Security-SRS-documentation
- https://www.youtube.com/watch?v=fYQSOYslMi8&list=PL39u5ZEfYDEN3VeTLG-BHWYF0w_aBrNW6&index=5
- https://t4tutorials.com/srs-documentation-of-home-automation-and-security-system-android-project/
- https://www.youtube.com/watch?v=Ut1MNJYuajM&list=PLr3OnN7KdzejkcczKY8vcaSWN6clHds8e&index=7
- https://draw.io

2. Overall Description

This Home Automation Software uses NFC(Near Field Communication) to put all the smart home appliances in a network, making all of them compatible to each other. With the help of sensors and accurators our software will control the devices and recommend the actions required on it too. The automation software will also give recommendations what to use based on user location and interests with the recent usage data with the help of machine learning.

2.1 Product Perspective

- It solves the compatibility problem of the smart devices as we make them connect with each other by creating a platform for it.
- Users can view the history of the events and appliances. These features are also compatible for the CCTV cameras in case a bigger infrastructure is to be considered for automation.
- Users will be recommended the appliances and the actions to be performed on them based on their location and interests with the help of the sensors and machine learning data.
- Home automation will also help in the energy saving by recommending the actions required by the help of sensors, and by giving a very detailed analysis of the average power usage in terms of time of the day.
- Home security features will permit the user to be aware of the dangers.

2.2 Product Functions

- Basic home appliances such as lights, fans switching.
- Home appliances with many functions such as refrigerator, AC, television, geyser are also made compatible with smart device controllers such as Amazon Alexa.
- Temperature sensors and smoke detectors.
- Sensor based (IR)security system.
- Machine learning used for interest based recommendations.

2.3 User Characteristics

Our software is developed for the users for a specific place. Users must be familiar with the the use of smart appliances and also the smart device controllers i.e. the hardware host. Apart from that, the software will have a very intuitive design, so the user will not need any further familiarity with the software, and can get started very easily and quickly.

2.4 Constraints

- Slower connectivity of NFC.
- Limited range.
- Vulnerabilities due to security issues.
- Scope in the developing countries.
- Financial affordability of the smart devices.
- Security issues of the data saved of the recent uses.
- A smartphone or any device which can acces the Internet would be required.
- The device on which the software is loaded should be upto fairly modern standards of technology, in order to be able to access all the features of the software with ease.

3. Specific Requirements

There are many software, hardware and other system requirements that the user must ensure to achieve a fluidic access to all features of the software. These are listed out as follows.

3.1.1 User Interfaces

The project will be that of a software with a GUI(Graphical User Interface), hosted on both Android and Web platforms, such that the user will easily and instinctively be able to interact with the interface to get the desired functionality, as one of the chief purposes for this project is to improve user convenience.

3.1.2 Hardware Interfaces

Pre-existing hardware(in the form of various home automation devices already present in the home of the customers) will be required for consumers to be able to use this software will full efficiency. Moreover, as this software will be hosted on both mobile and web platforms, there will not be major restrictions in today's age, as more and more, we see people buying smartphones with specifications that would be more than enough to satisfy the demands of this software. To be able to make full use of the recommendation system offered by the software, an increasingly common NFC(Near Field Communication) chip is required in mobile devices in order for the software to detect the location of theuser within the house, to make more specific recommendations.
Moreover, a hub might be required for large scale full home automation purposes, so as to better be able to manage the entire smart home automation system.

3.1.3 Software Interfaces

Free software and interfaces might be required to enhance user experience. These software may include the Google Home app(if you are communicating to the software via a Google Home device), the free Alexa app(if you are interacting via the Amazon Alexa), and other software related to various smart devices in your home. But, the end user will not be required to interact with any of these apps/software, as this will be done by our software. These software are required only to make it possible for our software to communicate effectively with the various smart home devices, and to ensure that any proprietary pin associated with the user's account, if required for any form of verification, can be input by our software with ease.

3.1.4 Communications Interfaces

A secure 128-bit encrypted communication would take place via the Internet through WiFi which is available in every smart home. A secure communication line is very important because the software does not log user data, so the only weakness of the software in terms of the security aspect would be the communication between the various smart home devices, and between the devices and the software itself.

3.1.5 Memory Constraints

Focus groups have determined that our target market has a memory capacity between 2-16 GB of RAM, therefore the design footprint should not exceed 2 GB. A memory capacity of 2 GB is more than enough to run our software, so this should not be an actual valid constraint.

3.1.6 Operations

- All the user data is backed up on the cloud if this service is opted by the user; even then, it is much safer for the user to maintain a local back-up of their important data like account usernames and passwords, in case of problems with the cloud servers.

- All stored logs of the day, if any-depending on the user's choice, would be deleted at midnight of that day. So, if the user needs any information which could be found in or could be pertaining to those logs, they would be advised to either make a physical copy of those logs before midnight, or to peruse the logs online before midnight.

3.1.7 Site Adaptation Requirements

- Depending on the scale of the home automation of the user, a smart-Hub might be required to be used, in order for the software to effectively manage the requirements of the user in an efficient manner.
- If not present, a wireless Internet connection through a WiFi router would be needed for the software to function and communicate and intercommunicate between the various home automation devices.
- The wiring for the smart-Hub, if installed, can be done in such a way as to keep it hidden I=from view, in order to maintain an aesthetic appearance for the home.

3.2 Functional Requirements

The following will be the major functional requirements from the project, apart from which many minor functionalities might also be introduced either from the beginning or in future versions.

3.2.1 User Login & Sign Up

- Login and password: This is the criteria through which the authenticity of a client is checked. The system initially prompts the user to fill their login ID and respective password, and then proceeds further.
- New User Creation: In case the user has not registered for an ID, he/she can check the Sign-Up option and will be redirected to a new page to create a new account.

3.2.2 Input of Instruction & Details

- User Input: User will give an input based on their requirement, providing details of the instructions or activity that they want to be executed. For instance, the user might give an instruction to switch on the Air Conditioner at 30℃.
- Sending the signal to the corresponding smart home appliance: Based on the instruction given by the user, the software sends out the signal to the appropriate appliance to perform and execute the aforementioned activity properly.

3.2.3 Interaction with Sensors

- Some instructions might require a measure of the current sense, based on the appliance that instruction might relate to. In such cases, the software would be required to interact with various sensors such as temperature sensors, to get an accurate reading.
- For instance, the user could want get information about the current temperature condition in the room.
- The output from the sensor, i.e., the reading, is sent to the home automation software, so that it can successfull proceed with the execution of the instruction.

3.2.4 Implementation of Instruction by Appliance

- Input to appliance: The appliance finally receives an input from the home automation software, in the form of a signal conatining the instructions required to successfully execute the activity required by the user.
- Output from the appliance: After receiving the signal from the software, the appliance interprets the instruction and successully executes it to complete the activity as instructed by the user.

3.3 Use Case

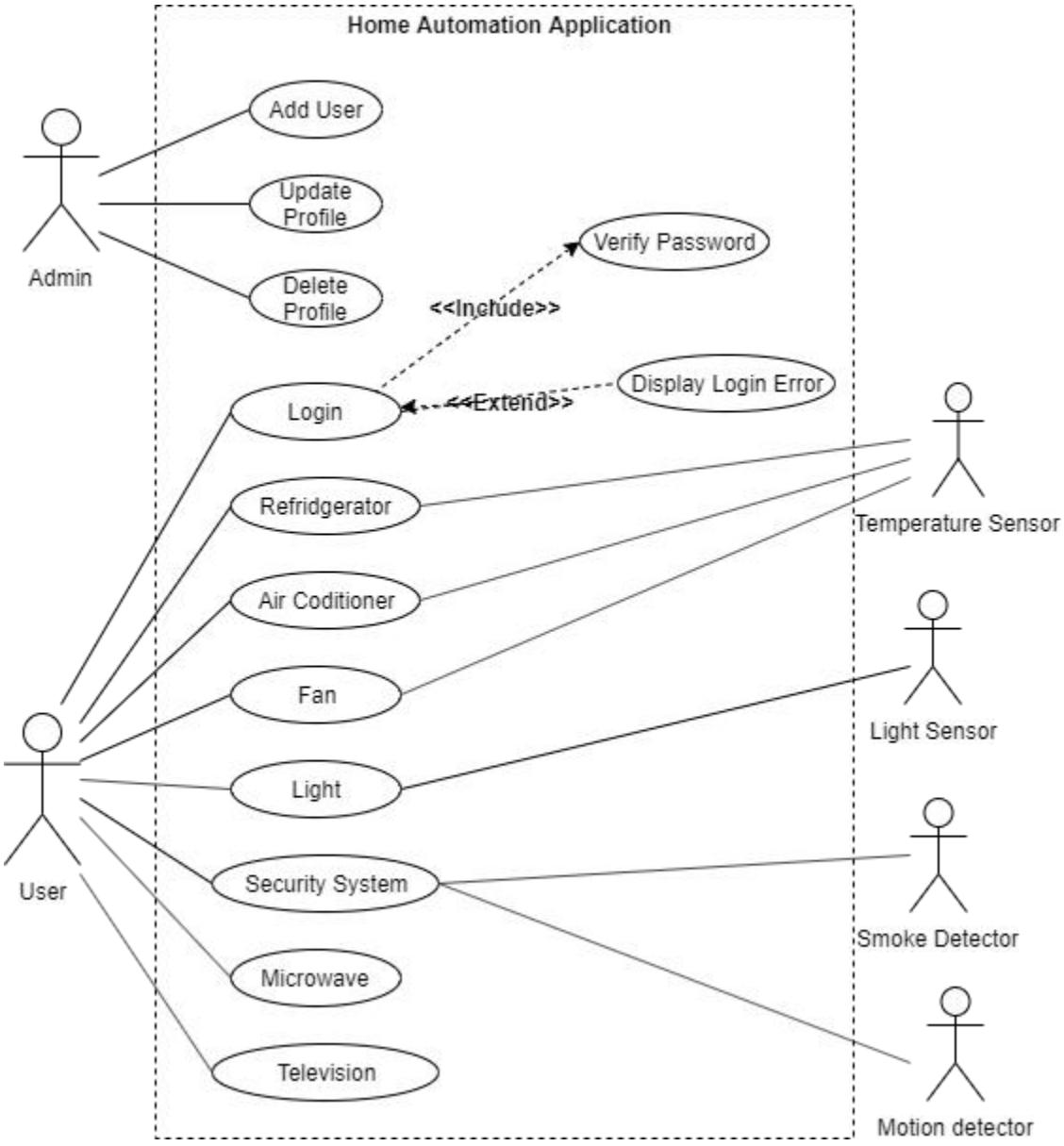

Figure 1: Use Case Diagram

3.4 Classes/Objects

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. The class diagram is the main building block of object-oriented modelling. It is used both for general conceptual modelling of the systematics of the application, and for detailed modelling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

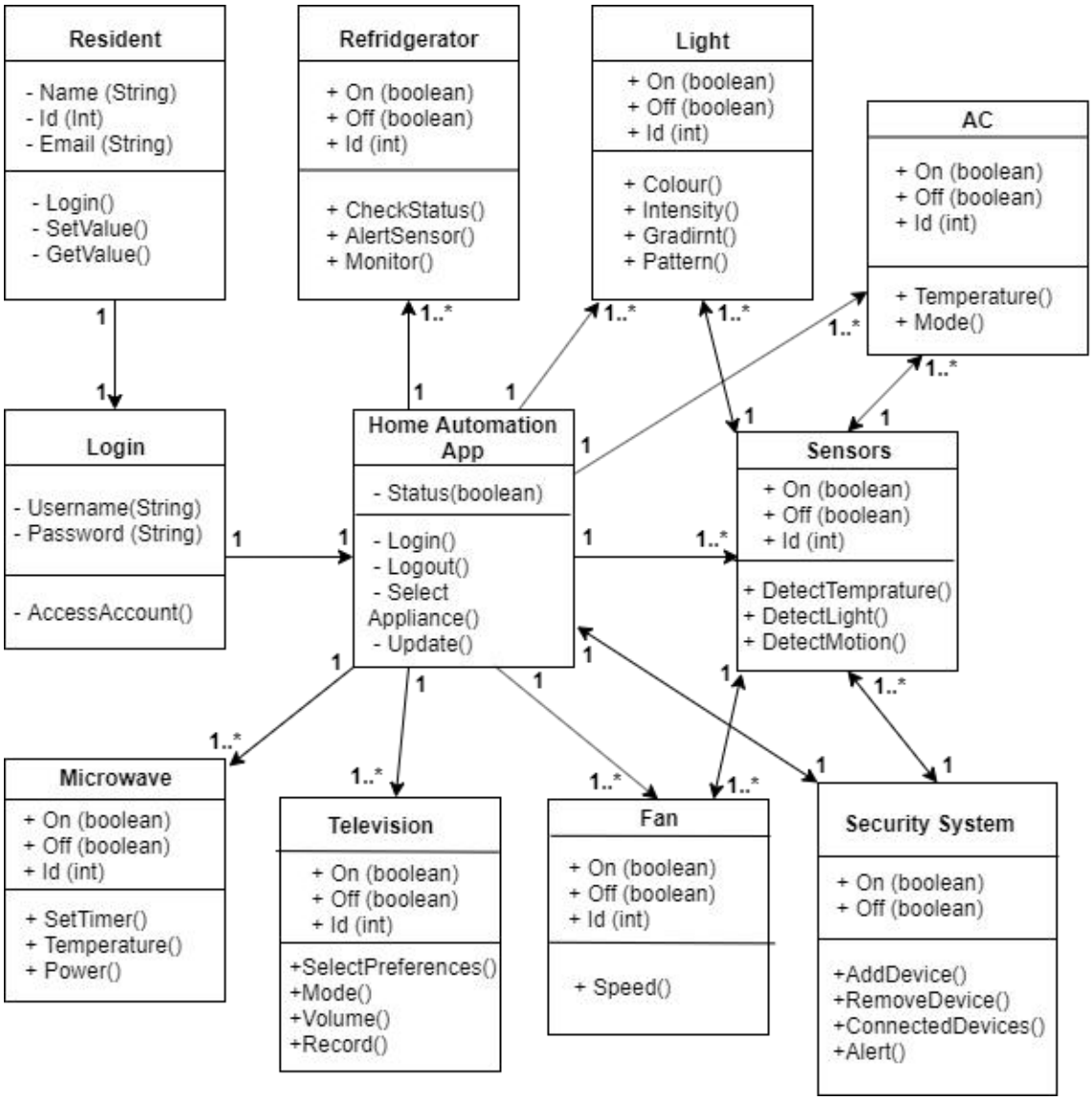

Figure 2: Class Object Diagram

3.5 Non-Functional Requirements

Non-functional requirements may exist for the following attributes. Often these requirements must be achieved at a system-wide level rather than at a unit level. The requirements are stated in the following sections in measurable terms.

3.5.1 Performance

- Speed of the Internet connection from the remote computer and the speed of the connection to the server will affect performance. The Home Automation System will be designed to be capable of operating even with slow Internet connections e.g. a standard 56K wireless connection to the network.
- The Home Automation System will be designed to operate on a standard personal computer or smartphone with a standard web browser or application installed. Response times to commands should be minimal from the browser.
- Hardware connected to the system may suffer from a latency factor between issuing of a command and when that command is carried out. This will be taken into account and may influence how fast the commands are processed within the system.

3.5.2 Reliability

The system has to be 100% reliable due to the importance of data and the damages that can be caused by incorrect or incomplete data. The system will run 7 days a week, 24 hours a day. Also, since it is a convenience driven product, the system has to be reliable such that all signals and instructions sent by the user are received by the appliances accurately, and are executed properly; incorrect signals should not be sent by the software. Moreover, since the software improves over time after learning from the habits of the user, it needs to be reliable in the recommendations it gives to the user based on what it has learned about the preferences of the user.

3.5.3 Availability

- Since the software is being developed in relation and connection with various “smart” IoT-based devices, it is a huge value service for the consumer, and so consumer convenience is key. To ensure consumer convenience, the system has to be available 100% for the user and is used 24 hours a day, 7 days a week and 365 days a year.
- For the system to function properly in a large-scale implementation, the availability and presence of a smart-Hub becomes important to ensure smooth instruction relay.
- Availability of a secure, stable and strong WiFi connection needs to be ensured for proper functioning of the software.

3.5.4 Security

There are security risks associated with using the Home Automation System as it is designed to operate on a network like the Internet. When accessing the system the user needs to be assured that intruders, such as hacker attempts and third party invasions, can not have access to the Home Automation System. The user therefore needs to be confident that the Home Automation System is secure. The ability to authenticate user connections to the system is required. To prevent unauthorised access the user has to log on to a website or application in order to have access to the Home Automation System. A secure system to validate the username and password information is required. The system will accept this information as proof of the identity and allow the user to access the Home Automation System. Also, the data of the user, if saved, should be safely encrypted and protected againstany kinds of security threats like hacks or other cyber or socio-engineering attacks.

3.5.5 Maintainability

The rapid progression of the technology means frequent software updates from manufacturers. These updates come in the form of both app updates and component firmware updates. Generally speaking, the app updates can be handled automatically by the user's smartphone or tablet, but in some cases, those automatic updates can lead to problems if the device firmware isn't updated at the same time. Updates to the devices can be tricky. There are other potential issues a maintenance visit can uncover. Heat is an enemy of electronic components. If conditions in the cabinet, closet, or room in which the equipment is installed change, it's possible that the components can reach temperatures that cause them to malfunction or to shut down in thermal protection mode. The research department should plan methods to know the states of the past implementation process. For that regular meeting should be arranged by the concerned officers about the implementation process and success.

3.5.6 Portability

Changes must be verified once per day at least .The system should automatically provide notification to patrons by email about items overdue, reservation results, availability of reserved item, etc. Also, the software should be re-installable and accessable from all smartphones and computers as such, given that the user is able to authenticate his/her login on any such device. So, none of the code should be host-dependent. The code is thus written in Java, a highly portable language, so as to provide code portability across all devices.

4. Analysis Models

4.1 Data Flow Diagram (DFD)

4.1.1 Level-0 Data Flow Diagram

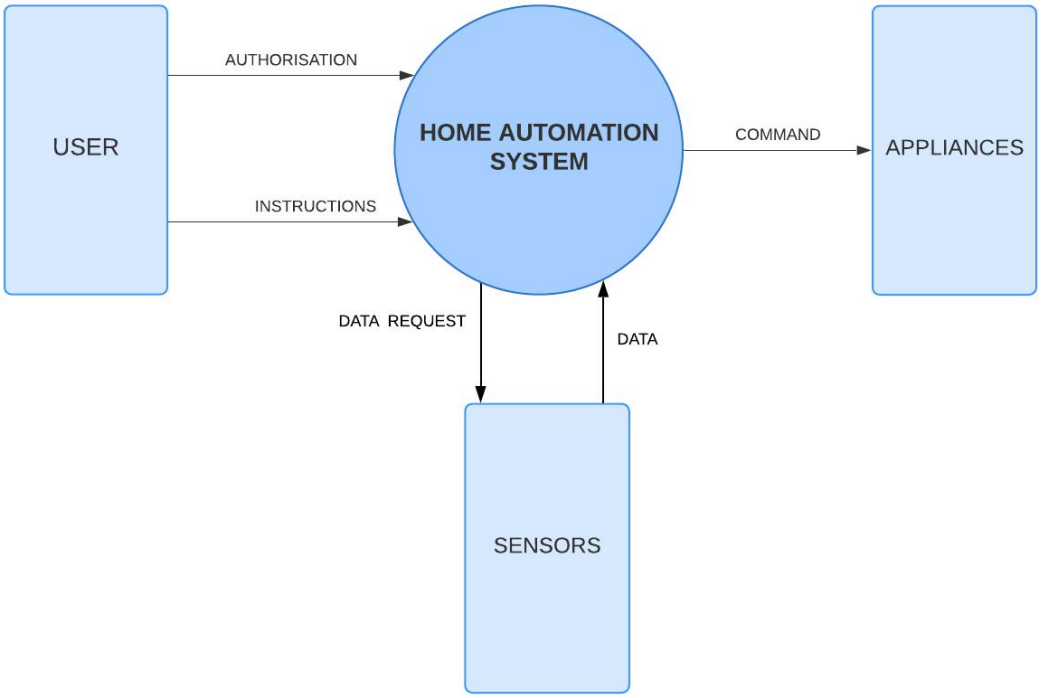

Figure 3: Level-0 Data Flow Diagram

4.1.2 Level-1 Data Flow Diagram

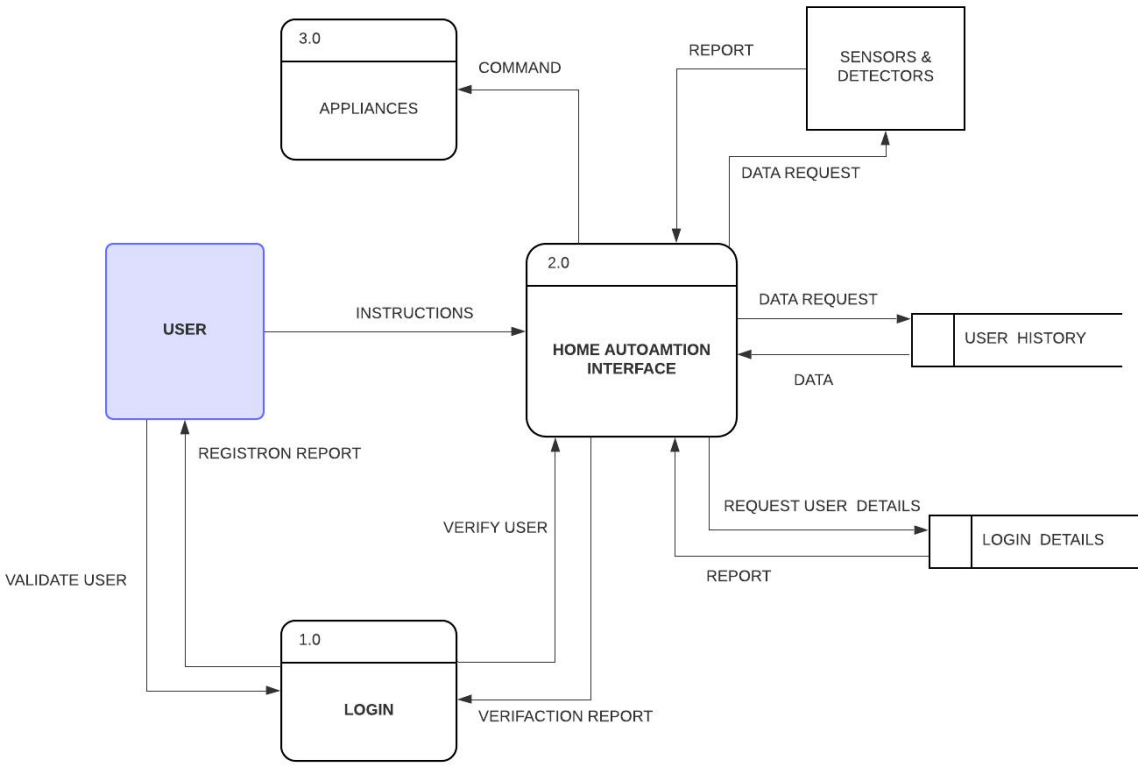

Figure 4: Level-1 Data Flow Diagram

4.2 Activity Diagram

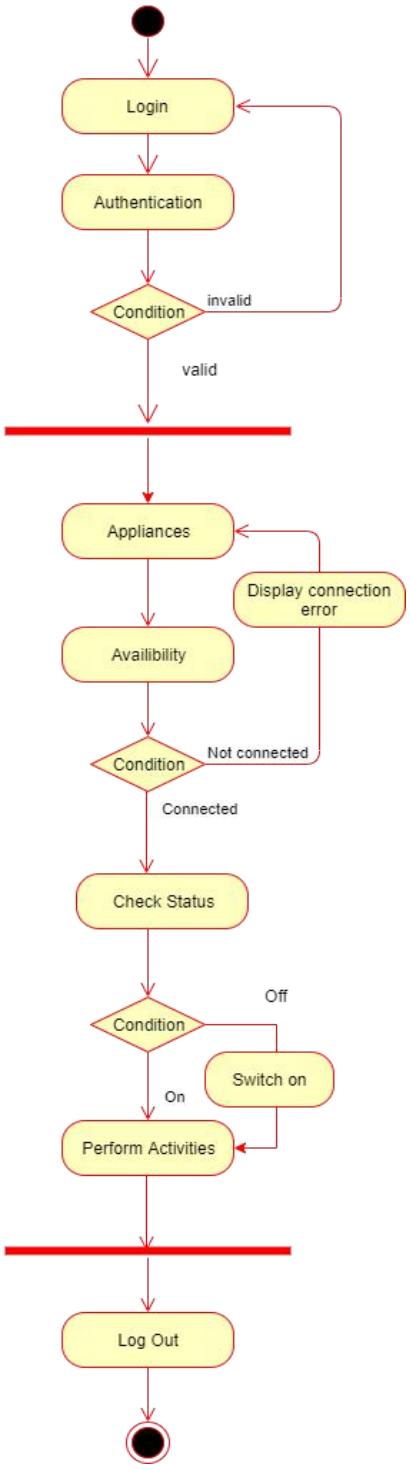

Figure 5: Activity Diagram

4.3 Sequence Diagram

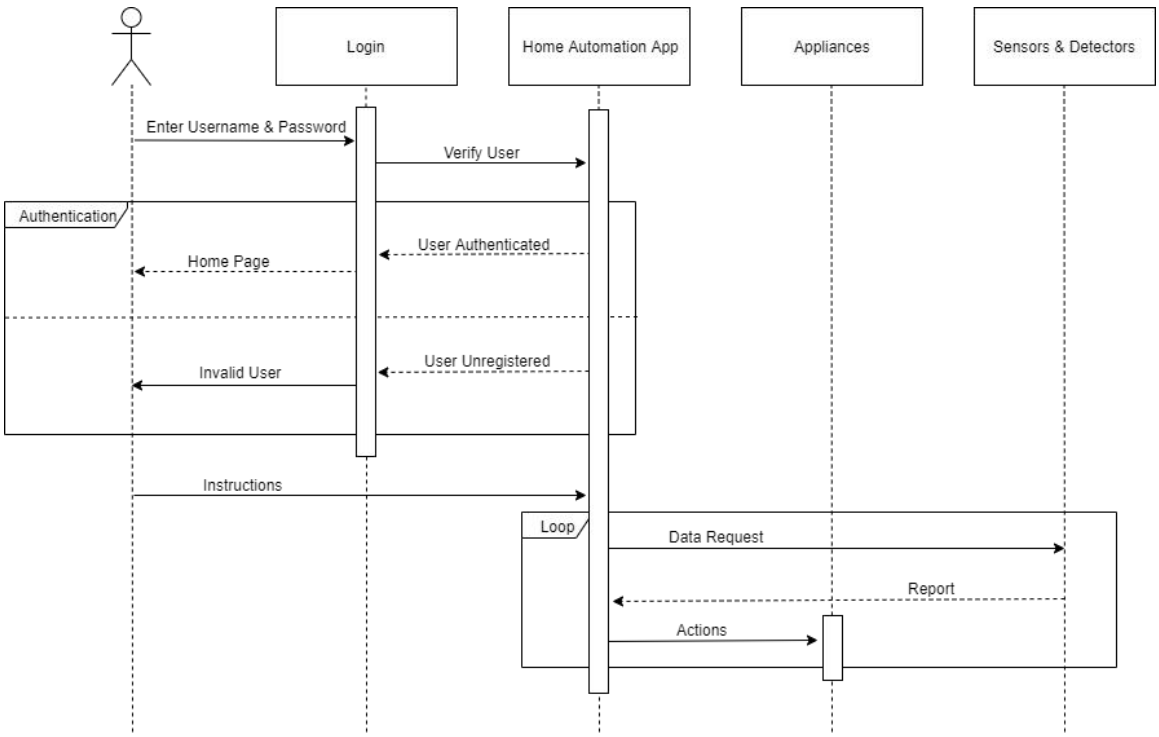

Figure 6: Sequence Diagram