Software Design Specification

for

Smart Home Security System

Revision 0.1 draft

Prepared by Theyab Alsubaie, Abdullah Altuwayrsh and Muntathir Alsaleh

MX-222-555\_SWE-Course\_Project

2023-03-28

Table of Contents

Table of Contents ii

Revision History iii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

2. Software Design Specifications 1

2.1 Use Case Diagram 2

2.2 Activity Diagram 4

2.3 System Layout Diagram 6

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| **Smart Home Security System**’s Software Design Specification | 2023-03-28 | First Draft | 0.1 |
|  |  |  |  |

# Introduction

## Purpose

This document presents a detailed description of an automated home security alarm system to protect the home from intrusion. It will explain the design specification that in conjunction with the Software Requirements Specification should provide a well-rounded image about the System.

## Document Conventions

Main Section Title:

Font: Times New Roman. Face: Bold Size: 14

Sub Section Title:

Font: Times New Roman. Face: Bold Size: 12

Other Text Explanation:

Font: Times New Roman. Face: Normal Size: 12

## Intended Audience and Reading Suggestions

This document is intended for general use including the stakeholders, developers, clients, and customers of the system.

## Product Scope

The scope of this product is to build a small embedded systems in case of intrusion so the system will trigger of someone pass by with the help of using multiple sensors, actuator, and an NFC Card reader, any entry attempt will trigger the alarm and a notification will be sent to the homeowner.

# Software Design Specifications

All diagrams are generated through visual-paradigm.com, using UML Code (PlantUML).

## Use Case Diagram

Here, the document overviews the Use Case Diagram for the Smart home Security System.

### Diagram

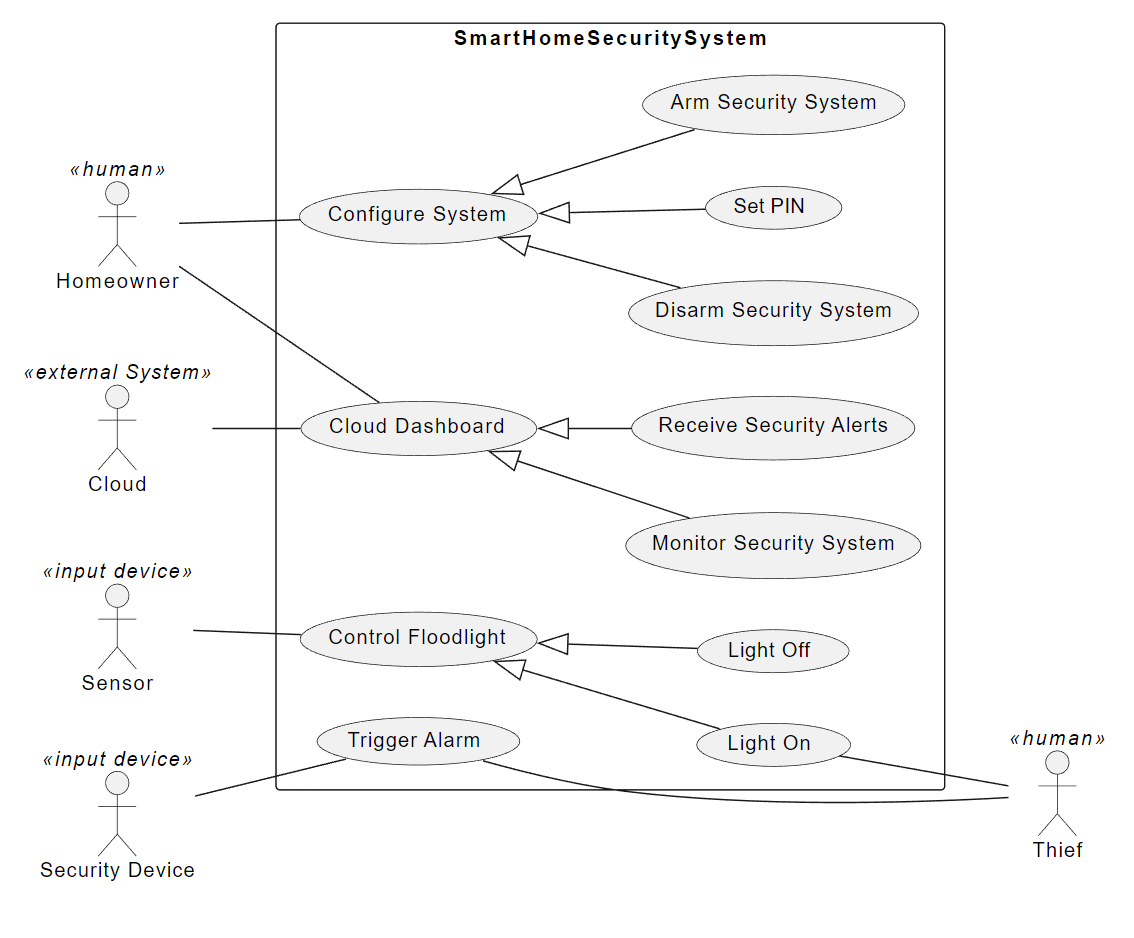


Figure :Use Case Diagram

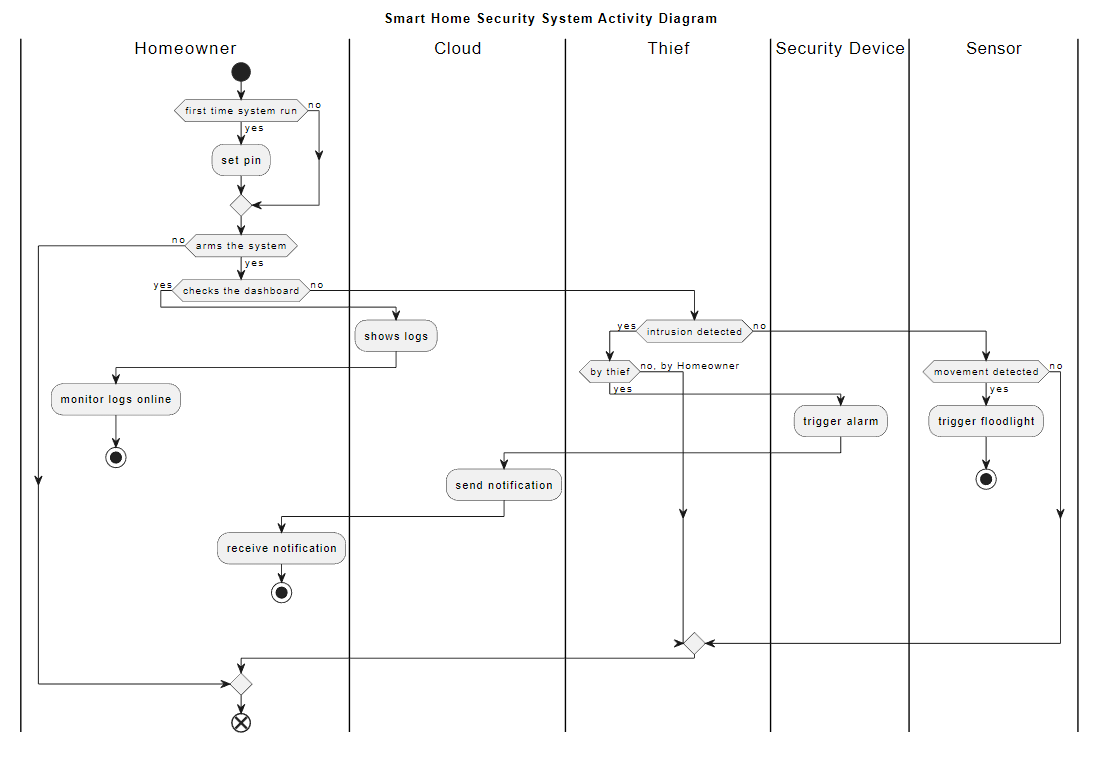
### UML Code

|  |
| --- |
| @startuml  left to right direction  actor Homeowner <<human>>  actor Thief <<human>>  actor Sensor <<input device>>  actor "Cloud" <<external System>>  actor "Security Device" <<input device>> as Security  rectangle SmartHomeSecuritySystem {  usecase "Arm Security System" as arm  usecase "Disarm Security System" as disarm  usecase "Monitor Security System" as monitor  usecase "Receive Security Alerts" as receive  usecase "Set PIN" as pin  usecase "Configure System" as config  usecase "Cloud Dashboard" as cloud  usecase "Control Floodlight" as lcontrol  usecase "Light On" as lon  usecase "Light Off" as loff  usecase "Trigger Alarm" as trigger  }  config <|-- pin  config <|-- arm  config <|-- disarm  lcontrol <|-- lon  lcontrol <|-- loff  cloud <|-- receive  cloud <|-- monitor  Thief -- trigger  Homeowner -- config  Homeowner -- cloud  Sensor -- lcontrol  Cloud -- cloud  lon -- Thief  Security -- trigger  @enduml |

## Activity Diagram

This section shows **Smart Home Security System** from the **Activity Diagram** prospective.

### Diagram



### UML Code

|  |
| --- |
| @startuml  title Smart Home Security System Activity Diagram  |Homeowner|  start  if (first time system run) then (yes)  :set pin;  else (no)  endif  if (arms the system) then (yes)  if (checks the dashboard) then (yes)  |Cloud|  :shows logs;  |Homeowner|  :monitor logs online;  stop  else (no)  |Thief|  if (intrusion detected) then (yes)  if (by thief) then (yes)  |Security Device|  :trigger alarm;  |Cloud|  :send notification;  |Homeowner|  :receive notification;  stop  else (no, by Homeowner)  endif  else (no)  |Sensor|  if (movement detected) then (yes)  :trigger floodlight;  stop  else (no)  endif  endif  endif  else (no)  endif  |Homeowner|  End  @enduml |

## System Layout Diagram

The system diagram and code are illustrated in this section for the Smart Home Security System.

### Diagram

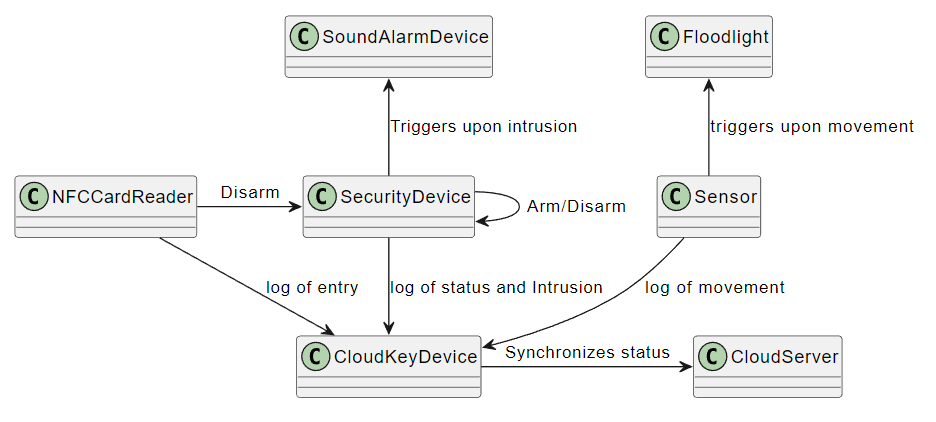


Figure :System Diagram

### UML Code

|  |
| --- |
| @startuml  class SecurityDevice {}  class Sensor {}  class Floodlight {}  class NFCCardReader {}  class CloudKeyDevice {}  class CloudServer {}  class SoundAlarmDevice {}  Floodlight <-- Sensor : triggers upon movement  NFCCardReader -> SecurityDevice : Disarm  NFCCardReader --> CloudKeyDevice : log of entry  Sensor --> CloudKeyDevice : log of movement  SecurityDevice -> SecurityDevice : Arm/Disarm  SecurityDevice --> CloudKeyDevice : log of status and Intrusion  SoundAlarmDevice <--SecurityDevice : Triggers upon intrusion  CloudKeyDevice -> CloudServer: Synchronizes status  @enduml |