Home Safe

Software Architecture Design

SAD Version 1.0

Team #1
20 September 2021

Mausam Shrestha [manager]
Safal Poudel
Saphal Karki
Karan Aryal
Rajesh Upadhayaya

CS 460 Software Engineering

TABLE OF CONTENTS

1 Introduction

This document provides a high-level software architecture design of the system. It describes the building components of the control software that controls the logic in the system.

This document is intended for the software development team, which they can reference to get an overview of how different components in the software communicate and interact with each other, and how the components relate to the physical interfaces.

This document is divided into four sections:
- Introduction: Describes the purpose and intended audience and provides the overview of the sections in the document.
- Design Overview: Section 2 consists of the design diagram which gives a high-level software architecture design of the system. It also provides a brief description of how the components are related to each other.
- Component Specifications: Section 3 consists of a detailed description of the objects and procedures involved in the design diagram along with the functionality it provides.
- Sample Use Cases: Section 4 consists of a walkthrough of the system to give an overview of how the system is used, and the options the user has.

2 Design Overview

This section includes a general description of all the physical components associated with the system along with a block diagram representing the physical interface. Section 2.1 represents the block diagram while Section 2.2 describes the features and functionalities of the physical components involved.

2.1 Design Diagram
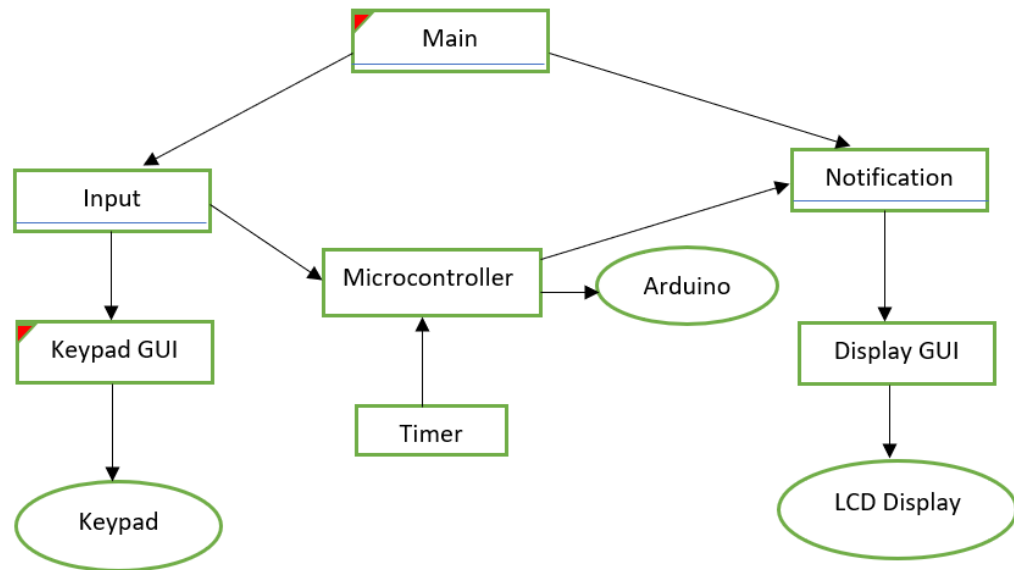
The program starts from Main class. Input and Notification are two major procedures. The keypad GUI is the active object in our diagram. It is the virtual implementation of the physical keypad of the safe. When the user presses the keypad, the key is sent to the microcontroller and a timer is started. Microcontroller is the virtual implementation of Arduino. For each digit pressed by the user, the microcontroller resets the timer and starts again. If the user fails to enter the pin within 10 seconds from the last digit pressed, the whole pin resets and we have to start from the beginning. If the pin entered matches the password stored, the microcontroller sends a notification that is displayed in Display GUI which is the virtual implementation of physical LCD display.

3   Component Specifications

This section describes the specifics of the component involved in the design diagram. It describes the classes and methods involved in the software design.

3.1   Input

This is a procedure in the program where the input is taken from the user. This procedure is invoked in the Main class. The input takes the integer parameter 'k' which is later used to check if it matches the password or not.

3.2   Notification

This is the procedure through which the program notifies the status of the safe. This procedure is also invoked in the Main class. It takes the parameter of String 's which is later displayed in the LCD screen.Initially the safe is locked, and when the pin matches the password the program notifies the user that the safe is unlocked.

3.3   KeypadGUI

This object acts as a virtual keypad. The user can enter the password through this object. It uses labels, buttons as well as actionlistener to communicate with the user and the program.

3.4   DisplayGUI

This object acts as a virtual LCD Display. It mainly uses labels to display the message about the status of the lock. It invokes the Notification procedure to get the corresponding status message.

3.5   MicroController

This object of the program handles the logic for checking the pin the user has entered with the password. If the password is correct or incorrect, it passes the corresponding message as a string parameter to the Notification object. If the

3.6   Timer

The timer object will implement a set function to set the timer once a pin is provided by the user, timeout function to indicate whether the 10 seconds time limit has passed or not and, the reset function to reset the timer once another pin is provided by the user within 10 seconds. If there is a timeout a boolean parameter is passed to a resetPassword function in the microcontroller.

4 Sample Use Cases

4.1 Regular entry

After the program is compiled and run, the user is prompted to enter 4 digit pin codes in the Display GUI which acts as a virtual LCD display. The timer is started once the user inputs one digit of the pin. If the pin matches the password, the lock gets unlocked and the user gets the message in the LCD display.

4.2 Reset pin

To reset the pin while the user knows the previous pin, the user needs to hold * for 2 seconds and it will prompt the user to display the previous pin first and then the new pin. After the new pin is entered the users can see the success message in the LCD display.

4.3 Override Mode

In the case where the user forgets the password, the user will need to unlock the safe with a mechanical key, which will automatically give the user an option to reset the pin in the LCD display with instructions. The user will need to hold # for 2 seconds and then enter the new pin to successfully override the forgotten pin.