Keypad Project

Software Requirement Specification

SRS Version 2.0

Team T02
12 February 2021

Tanner Evans (manager)
Thomas Bowidowicz (documents)
Robin Acosta
Marcos Lopez
Jared Bock
Jacob Varela

CS 460 Software Engineering

TABLE OF CONTENTS

Introduction

1.1 Purpose

The purpose of this document is to give a description of a gated community keypad that will unlock a gate when the correct password is entered. This document defines the specific software requirements, functional description, and design constraints with regards to this project. In this document we will also further define how the product will ultimately be seen and used by the client. This document is divided into four distinct sections: the introduction which defines the project, this document, and its intended audience, the general description which provides an overview of the system and gives background for later sections, the specific requirements which goes into detail concerning the external interfaces of the system in addition to the control logic that the system goes through in operation, and finally the design constraints section which outlines the challenges and limitations put onto the system and its design. There is also a section titled "*Definition of Terms*" located at the end of the document that defines many of the terms that are used throughout this document.

1.2 Intended Audience

The main audience for this document is the team manager and developers, as well as community developers and administrators. All parties associated with the project should be able to read and understand the various components of the keypad and it intended functions.

1.3 Product Scope

The community gate keypad product will act as the software that will enable a community to be gate-restricted to residents and public services. Our hardware assumption is described in this document. The gate will be able to be unlocked via access code which can be changed by those who have an admin code at any time. This software will only be responsible for opening/unlocking the gate. Timing and sensor mechanisms/software is provided by the gate itself and outside of the scope of this project.

General Description

The purpose of this section is to describe the keypad system as a whole. It will introduce all of the general interactions and functions of the system, but won't go in the same granularity as section three of this document. This section is intended to provide background on the specific details in section three so as to make them easier to understand.

This system is a keypad and accompanying software that manages whether or not a community gate is unlocked or not. To facilitate this main objective, it will contain the following sub components:

- There will be two separate states that the system can function in: a public community state and an administrative state. The public state will be used the majority of the time and will achieve the primary functionality of the system while the administrative state will allow for debugging and password changes to occur.

- At least two passwords stored in memory. There will be one password for the general community use and another password for administrative purposes.
- An input buffer that is active when the user begins to enter a password. This will store the incoming password and will check the final attempt against the valid password stored in memory.
- A timer of ten seconds that begins after each key is entered in the system. Once this timer reaches zero, the system will time out and clear the input buffer re-setting the input password.

2.1. Keypad Functions

The keypad will have the following features and functions:

- A standard set of ten digit keys [0, 1, 2, 3, 4, 5, 6, 7, 8, 9] arranged in a standard telephone keypad arrangement.
- A clear [clr] button located on top of the numerical buttons.
- An asterisk [*] button and a hash/pound [#] button.
- A speaker located below the numerical buttons.
- The ability to receive four digit passwords and compare them against the correct password.
- The ability to time out if a user is idle for ten seconds.
- The ability to change passwords through an administrative state. This state will also have a unique password.

Figure one below helps to illustrate the general physical layout and interactions of the system. The user will interact with the keypad via the buttons and will receive feedback through the speaker on the keypad. The keypad will process the input from the user and either unlock the gate or enter an administrative state depending on the input. The software will then interact with the gate to unlock it.

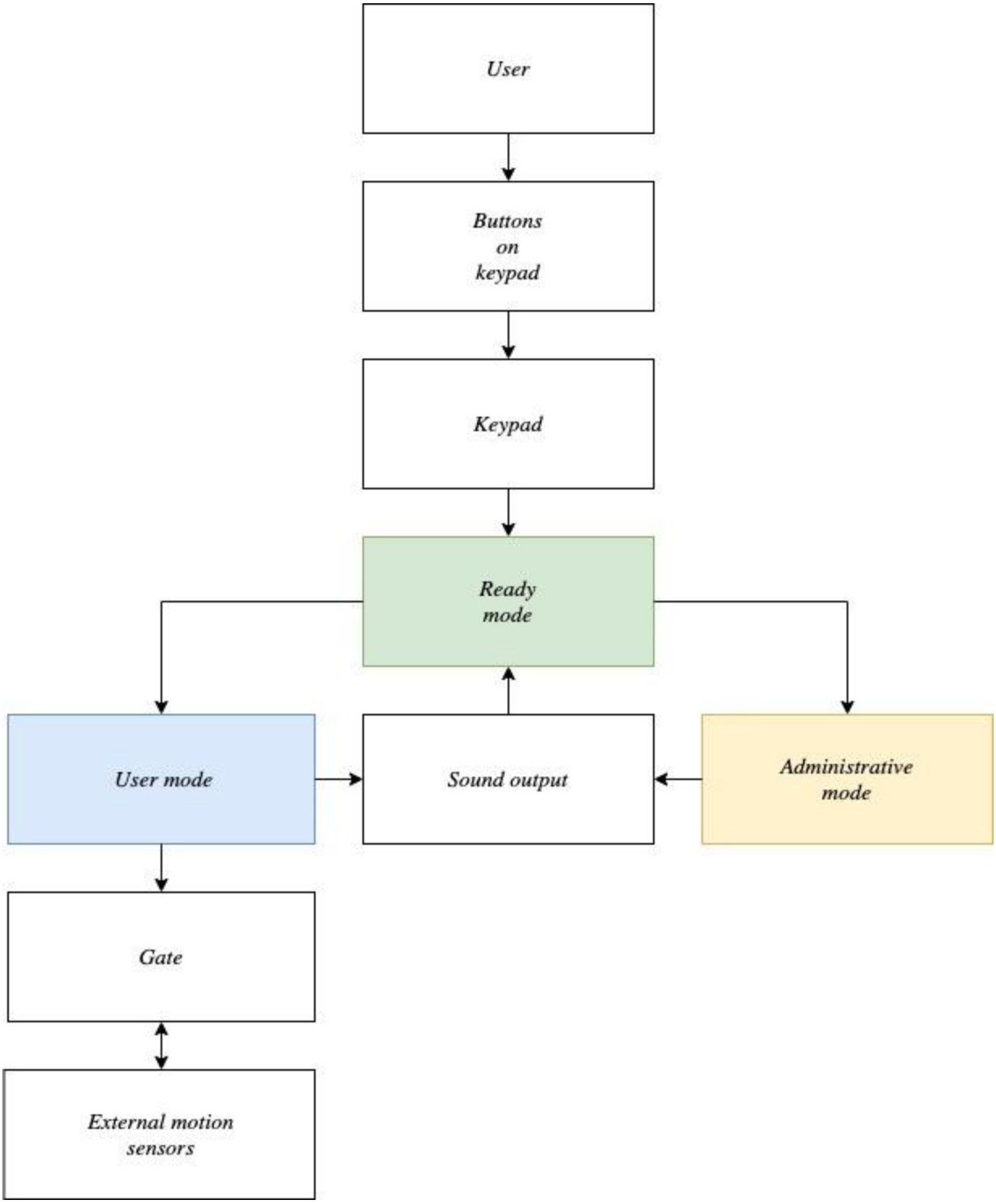

Figure 1 - Physical Block Diagram

2.2. User Characteristics

A general user of the keypad should be able to enter a four digit sequence into the keypad and have the gate either unlock or remain static depending on the password input. Encompassed in this, the user should be able to do the following functions:

- Enter a four digit password into the keypad.
- Receive audible feedback when the password is correct and the gate is unlocked.
- Be able to clear the entered password if they started to enter the password incorrectly.

2.3. Administrative Characteristics

An administrator of the system should be able to enter a special four digit password and enter the administrative state that allows them to change both the general user password and the administrative password. In addition to all of the general user characteristics, the administrator should be able to do the following functions:

- Gain access to the administrative state after correctly entering their password.
- Be able to change the general user password for the system.
- Be able to change the administrative password for the system.

2.4. Operating Environment

The internal operating environment for the keypad will be:

- A stand alone piece of software that does not interact with any existing operating system and additional software.

2.5. Design Constraints

This section will provide background on the design constraints which will be fleshed out in detail in section four. The main constraints are that the keypad must be accessible, resilient, and secure. This means that the keypad needs to be a specific height above the ground so a wide range of vehicles can access it, the software must be always functional and resistant to freezing or getting locked up, and that non community members cannot access the system or reset any passwords illicitly.

Figure two below illustrates the general physical layout of the keypad.

Specific Requirements

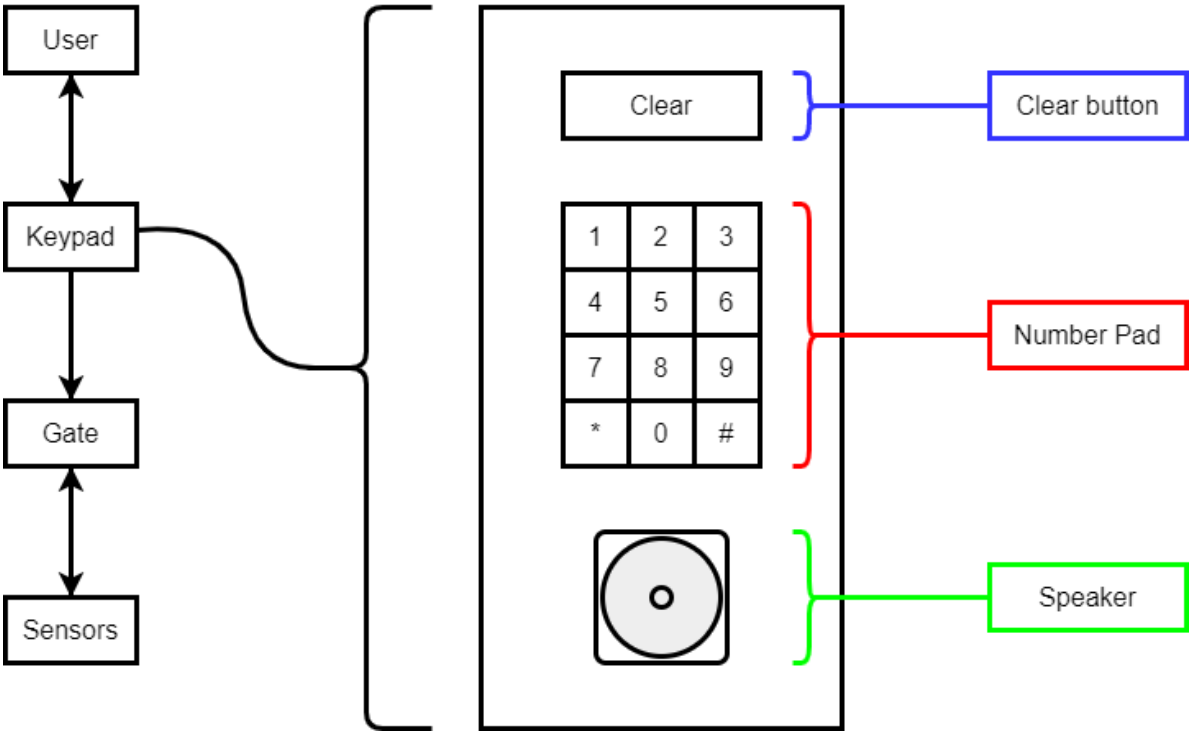

Figure 2 - System Organization Diagram

3.1 External Interfaces

This section provides an overview of the interfaces which govern communication with existing, external entities to the community gate keypad software. These interfaces allow changes to entities external to the software to require modification only to the interface. Interfaces transform either input from external entities into signals understandable by the software, or signals from the software into outputs to external entities. Specific protocols for communication with external entities are abstracted into interfaces. All input and output for the system passes through interfaces.

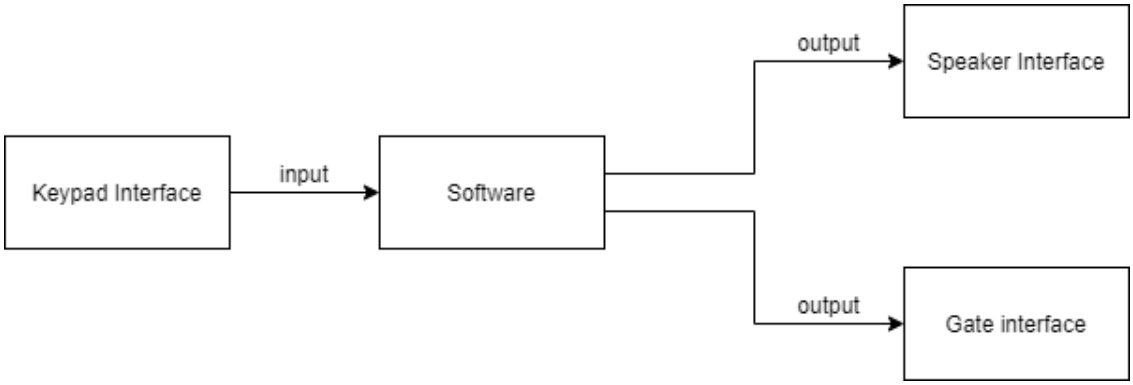

Figure 3 - Logical Diagram

Keypad Interface:
This facilitates communication from the physical keypad to the software and supports transformations for a specific set of key presses into the associated signals to the software.
Transformations:

- [0-9] Digit keys: When the digit keys are pressed the keypad interface will send a key signal to the software to enter the digit into a buffer.
- [#] Pound key: When the pound key is pressed the keypad interface will send a signal to the software to process the key buffer and send an appropriate signal to the gate interface accordingly.
- [*] Asterisk key: When the asterisk key is pressed the keypad interface will send a signal to the software to process the key buffer and and change modes if appropriate.
- [clr] Clear key: When the clear key is pressed on the keypad it will send a signal to the software that it should clear the buffer and reset to an idle state.

Speaker Interface:
This facilitates communication from the software to the output speaker and supports transformations for the software's specific set of output signals into appropriate, distinguishable sounds.
Transformations:

- Negative sound: When the speaker receives a signal from the software for a negative action, such as the wrong gate code being entered or the password timing out, it will output a “negative” sound. The specific sound will be determined later.
- Neutral sound: When the speaker receives a signal from the software for a neutral action such as a button is pressed, the speaker will then output a “neutral” sound.
- Positive sound: When the speaker receives a signal from the software for a positive action, such as when the gate code is entered correctly, the speaker will then output a “positive” sound.

Gate Interface:
This facilitates communication from the software to the gate and supports the transformation of the software's open gate signal into the appropriate message to the gate.
Transformation:

- When the gate interface receives a signal from the software it will then open the gate.

3.1.1 Input events

- Button presses: The keypad buttons are the sole input device to the software. These buttons will provide an individual input to the software.
- User: The user can input the community passcode to open the gate. The user will enter the passcode by pressing the various buttons on the keypad face.
- Admin: The admin can input the admin passcode to place the keypad into the admin state. The admin will enter the passcode by pressing the various buttons on the keypad face. This will then place it into the admin state or reset the software to an idle state.

3.1.2 Output Events

- Audio: The keypad will communicate between the users and admins using a series of audio beeps. These beeps will be played through the speaker on the face of the keypad. These beeps will have different tones based on what state the keypad is in. These different beeps will be determined as further testing occurs to make certain that positive and negative audio feedback are clear and unambiguous.
- Gate: The gate will open and provide a visual indicator of when the keypad receives a correct passcode.

3.1.3 Relation to Physical Devices

- Keypad: The keypad is a physical piece of hardware which will house the physical buttons and a speaker for input and output from the system.
- Buttons: The buttons will be the hardware pieces in charge of input events to the software.
- Speaker: The speaker will be the main piece of hardware in charge of output events from the software.
- Gate: The keypad will be installed to open a physical gate by way of physical connection. The keypad can send a signal to the gate controller telling it to open when the correct community code has been entered.
- Sensors : The sensors' communication is important to the gate. However the communication between the sensors and gate are not important to the function of the keypad.

3.2 Control Logic

This section describes the logical behavior of the keypad. The included diagrams capture the functionality as state transition diagrams, detailing the various states of the keypad structure, the inputs and events that govern transitions between states, and the outputs that occur as a result of specific states and/or transitions. A legend for the symbols and representations used for the state diagrams follows.

The system will have three modes of operation:

- Ready mode
- User mode
- Admin mode

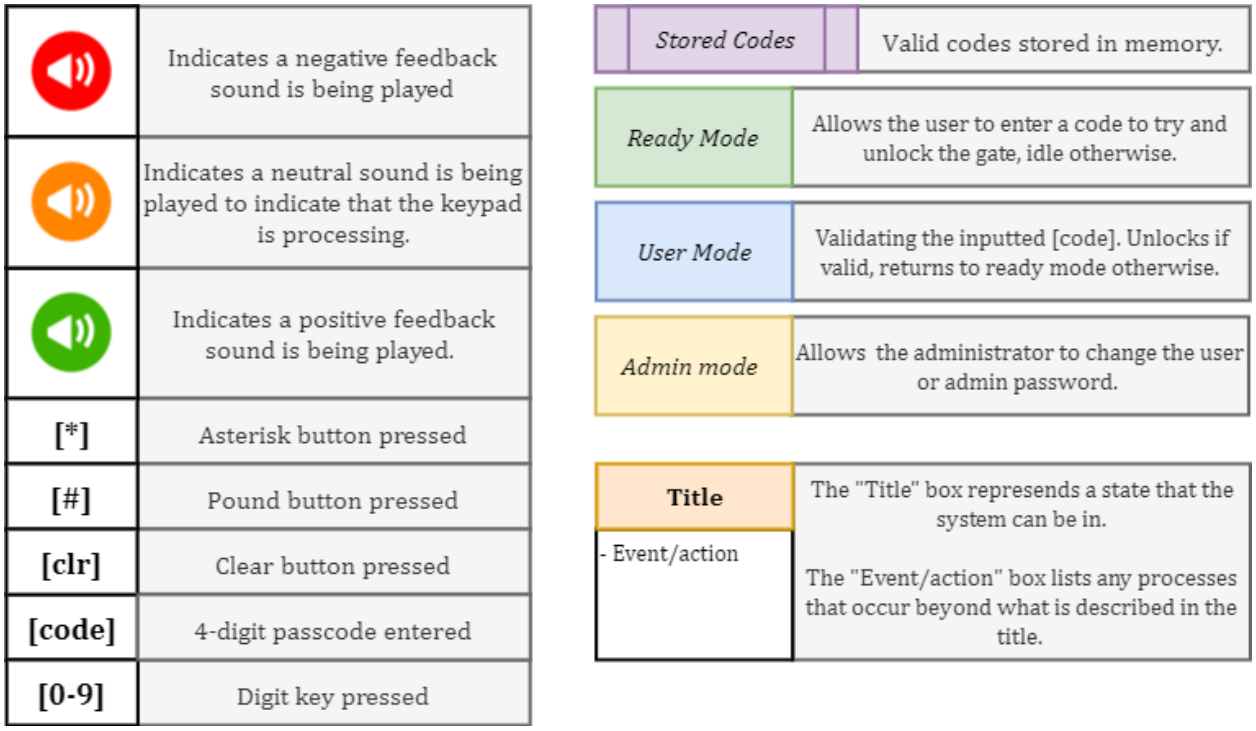

Figure 4 - State Diagram Legend

3.2.1 Ready Mode

Ready mode is the default mode of the system with a default state of idle. All other states will return to Ready mode's idle state by input or by timeout.

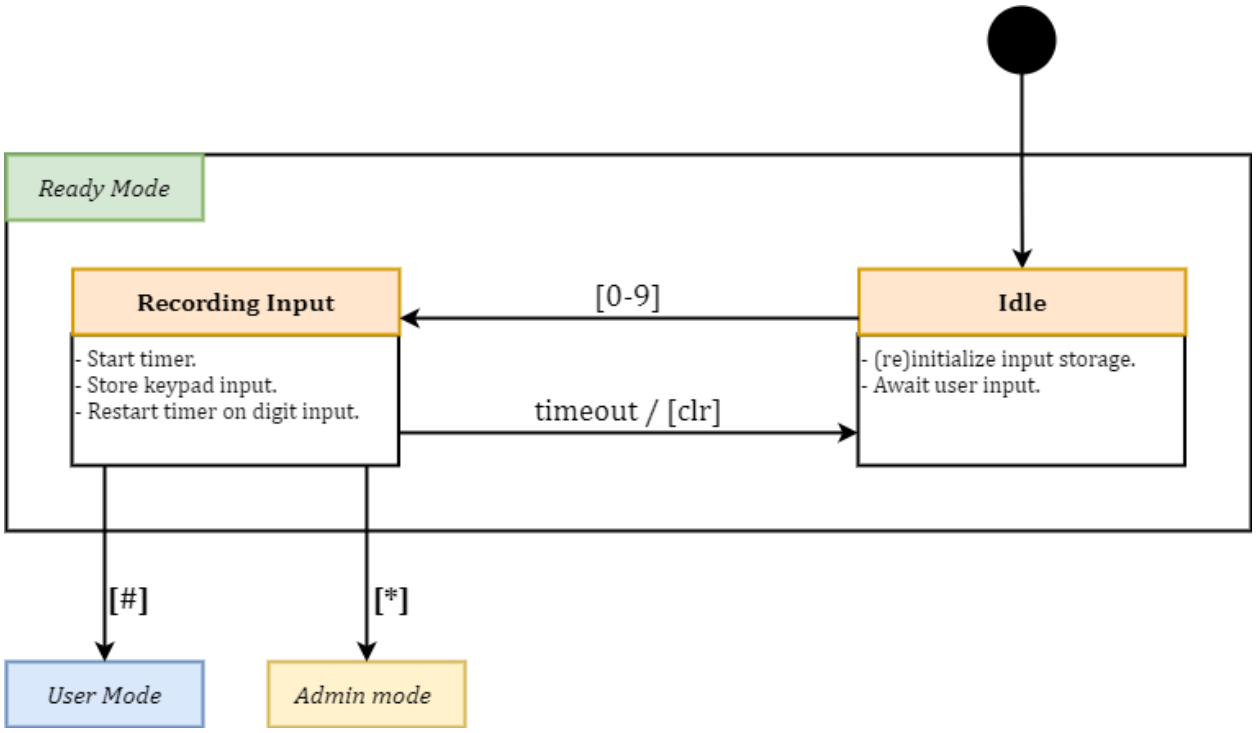

Figure 5 - State Diagram Ready Mode

Idle (Entry State)

The idle state is the entry point of the system, and the resting point at all times that users are not currently interacting with it.

Actions:

- Resets any recorded user input upon entry.
- Indefinitely awaits user input.

Transitions:

- When a number key is pressed, the system transitions to recording input.

Recording Input

Actions:

- Starts an input timer to return to idle if input ceases.
- Resets the timer upon a keypress.
- Stores keypad input.

Transitions:

- When [#] is pressed, the system transitions to User mode.
- When [*] is pressed, the system transitions to Admin mode.
- When the time runs out or [clr] is pressed, the system transitions to Idle.

3.2.2 User Mode

User mode validates the entered code and unlocks the gate if it is valid.

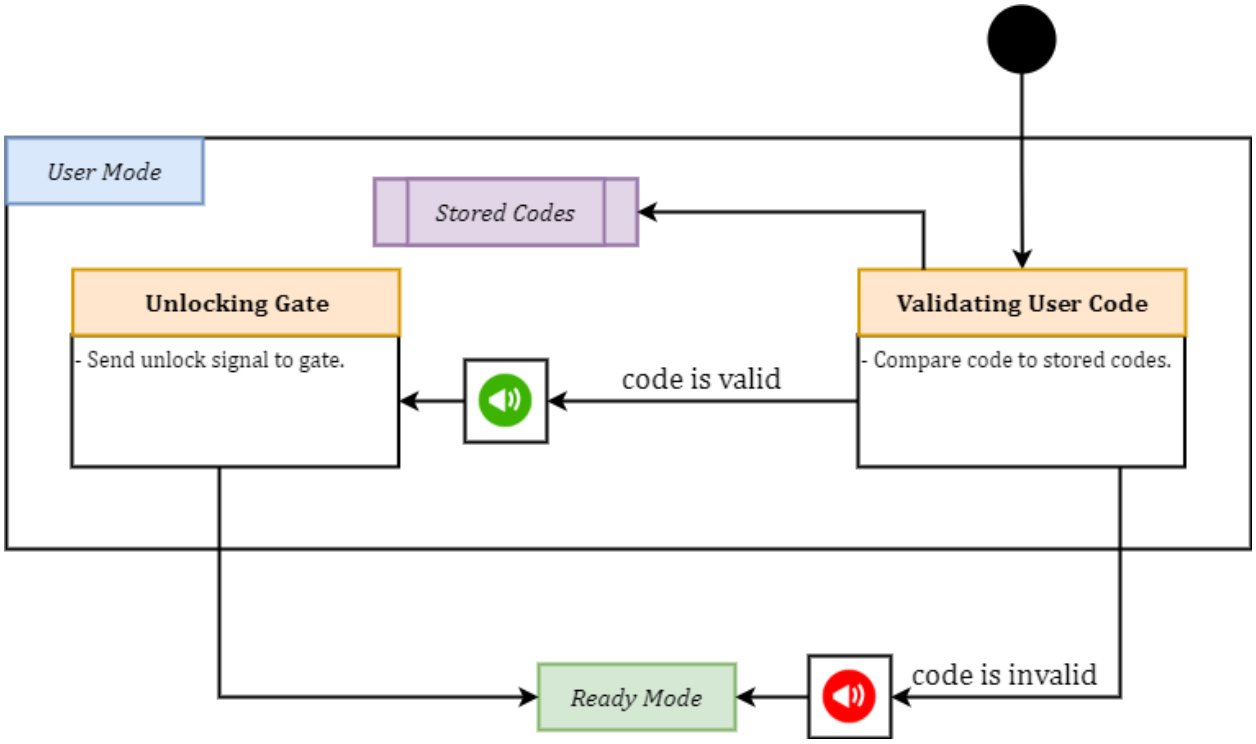

Figure 6 - State Diagram User Mode

Validating User Code (Entry State)

When ready mode transitions to user mode, it enters here.

Actions:

- Checks user input against the user code stored in memory.

Transitions:

- When the code is valid, the system outputs positive feedback and transitions to Unlocking ate.
- When the code is invalid, the system outputs negative feedback and transitions to Ready mode.

Unlocking Gate

Actions:

- Sends the unlock signal to the gate.

Transitions:

- The system transitions automatically back to Ready mode.

3.2.3 Admin Mode

Admin mode allows administrators to change user and admin passwords if they entered the correct admin code.

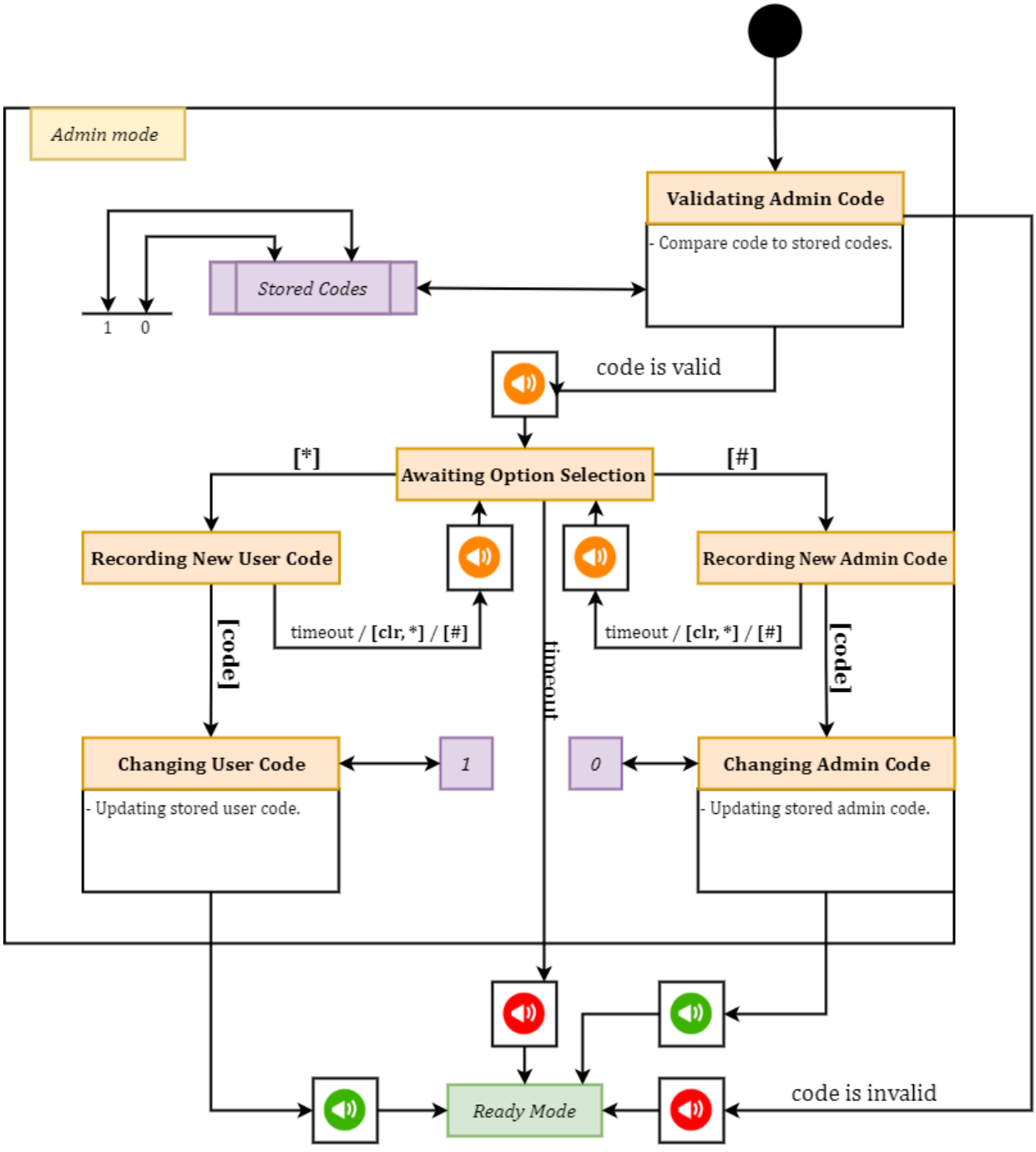

Figure 7 - State Diagram Admin Mode

Validating Admin Code (Entry State)
When Ready mode transitions to admin mode, it enters here.
Actions:

- Checks user input against the admin code stored in memory.

Transitions:

- When the code is valid, the system outputs neutral feedback to indicate that admin mode has been entered, and transitions to awaiting Option Selection.
- When the code is invalid, the system outputs negative feedback and transitions to Ready Mode.

Awaiting Option Selection
Actions:

- Starts an input timer to return to Ready Mode if input ceases.

Transitions:

- When [*] is pressed, the system transitions to Recording New User Code.
- When [#] is pressed, the system transitions to Recording New Admin Code.
- When the time runs out the system outputs negative feedback and transitions to Ready Mode.

Recording New User Code
Actions:

- Starts an input timer to return to Awaiting Option Selection if input ceases.

Transitions:

- When a 4-digit code has been entered, the system transitions to Changing User Code.
- When the time runs out or the [clr], [#], or [*] key is pressed, the system outputs neutral feedback and returns to Awaiting Option Selection.

Recording New Admin Code
Actions:

- Starts an input timer to return to Awaiting Option Selection if input ceases.

Transitions:

- When a 4-digit code has been entered, the system transitions to Changing Admin Code.
- When the time runs out or the [clr], [#], or [*] key is pressed, the system outputs neutral feedback and returns to Awaiting Option Selection.

Changing User Code
Actions:

- Updates the user code stored in memory to match the input provided.

Transitions:

- The system automatically outputs positive feedback and transitions to Ready mode.

Changing Admin Code

Actions:

- Updates the admin code stored in memory to match the input provided.

Transitions:

- The system automatically outputs positive feedback and transitions to Ready mode.

Design Constraints

4.1 Interface

- The keypad will be mounted five feet off the ground so that it is at a usable height for vehicles and an administrator.
- The keypad will have thirteen buttons to enter the password and debug (the buttons are listed above).
- The buttons must be large enough to be visible and easy to press from a car or truck.
- The passwords must be four digits in length.
- Users will be able to correct mistakes and try again by pressing the clear button anytime before they press the [#] button. This will erase their current input and they will be able to retype the password.
- There will be a clear audible indication if the code is valid or not valid.
- The keypad must be able to send a signal to open the gate. This signal is sent through a physical connection between the keypad and the gate.
- The administrator must have a unique code that when entered, will switch the keypad into admin mode, allowing the administrator to change the community code as well as the admin code.

4.2 Performance

Once a valid code is entered followed by the [#] button, the keypad immediately sends a signal to the gate through a physical connection between the keypad and the gate. If a correct code is not entered, the keypad must remain locked.

4.3 Security

- The keypad will be securely locked in a case that requires a physical key.
- There may be safety concerns regarding emergency personnel needing to gain access to the community. There should be a protocol in place for granting non-permanent members access.

4.4 Implementation

The keypad must be implemented using Java 8 or higher.

Definition of Terms

- **Admin/Administrator:**
  - A person or persons, who interact with the Gate Keypad with the intended purpose to access and modify the passcodes stored in the Keypads memory.
- **Asterisk/Star:**
  - One of the keys on the keypad, denoted by the following symbol: [*]
- **Gate:**
  - Any point of access to a walled community that the keypad will interface with.
- **Hashtag/Pound:**
  - One of the keys on the keypad, denoted by the following symbol: [#]
- **State:**
  - A representation of the system's position in a logical control flow. Specific states result from a series of inputs, and can be considered a representation of the data received up to that point.
- **System:**
  - A set of interconnected components, some of which are systems themselves.
- **Transition:**
  - A representation of the system's movement from one state to another. Transitions are caused by input events or the completion of internal actions. They result in a change in how the system will handle new input.
- **User:**
  - A person or persons, who interact with the Gate Keypad with the intended purpose of opening the gate to gain access into the community.
- **Passcode/password/code:**
  - A specific four digit string of numbers that is defined to unlock the gate when entered.