CGK - Community Gate Keypad

Software Requirements Specification

SRS Version 1.2

Team 03
19 February 2021

Amun Kharel
Brandon Stringham (manager)
Cody Crane
Krista Conley
Shreeman Gautam
Tanner Hunt

CS 460 Software Engineering

Table of Contents

1 Introduction

“B-STACK” is a software design group consisting of Brandon Stringham, Shreeman Gautam, Tanner Hunt, Amun Kharel, Cody Crane, and Krista Conley. In this Software Requirements Specifications(SRS) document, B-STACK details the project plan for the development of the "Community Gate Keypad" software(CGK). The purpose of this project is to design a keypad control system for a vehicular access gate. This document is intended for developers, designers, and testers working on the "Community Gate Keypad". This document will include a summary of:

- General description of the system
- Specific Requirements of the system
- External Interfaces with input/output events with the physical devices
- Control logic with a OMT diagram, overview, and a technical approach to the system
- Design constraints of the system
- Definition of terms

2 General Description

The gate control keypad allows access into a gated facility via a code. It is a security software and hardware implementation that will communicate one-way with the gate software system to allow users access into the facility and to inform users of the keypad system messages via LCD display.

In the Community Gate Keypad, there are three main modules: the keypad hardware and software, the users (general access users and public service), and the admin. These are laid out in Figure 1.

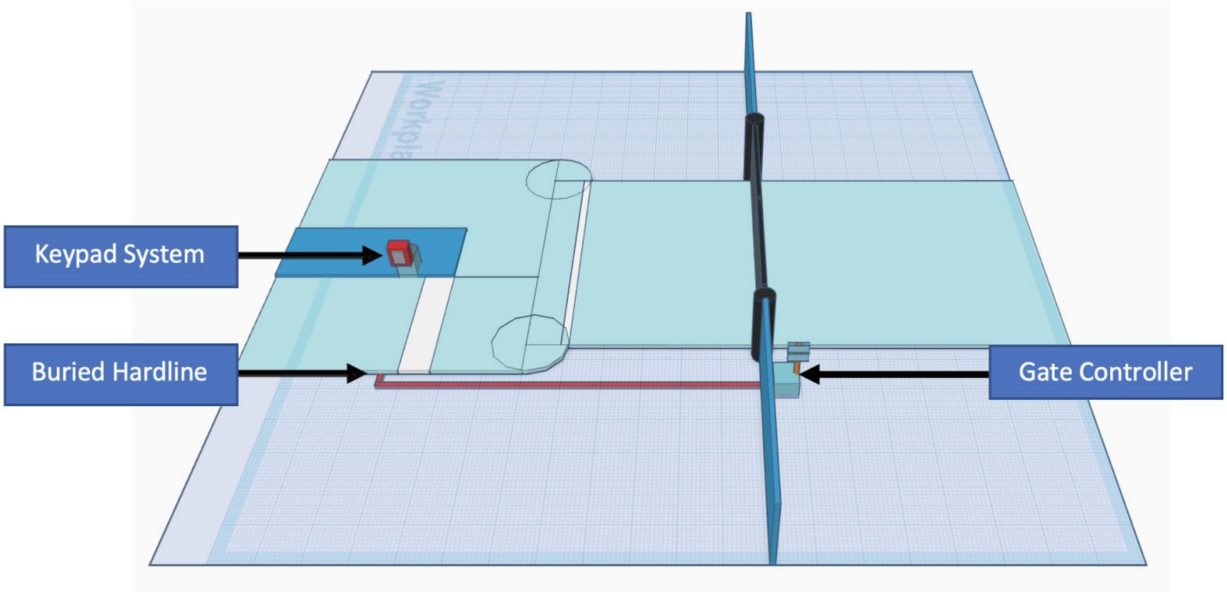

Figure 1 - Physical Design

The keypad is responsible for sending a signal to the gate controller system if the users enter the correct code. Otherwise, the keypad will send a message back to the users to inform them of any errors using a simple LCD display. The keypad is also responsible for sending an option menu to the admin upon the correct entry of the admin code to change the user codes. Otherwise, the keypad will send an invalid code message to the admin.

For the Community Gate Keypad to function correctly, the following criterias are needed. The Gate controller software must open when the keypad sends the open signal to it. The gate controller software and hardware are not implemented in this project, it is assumed it has already been established and is running correctly.

3 Specific Requirements

Figure 2 details the specific requirements for the system. These requirements include the external interfaces that the system needs to be able to communicate with to accomplish its goal, and the control logic that it needs to implement to function properly.

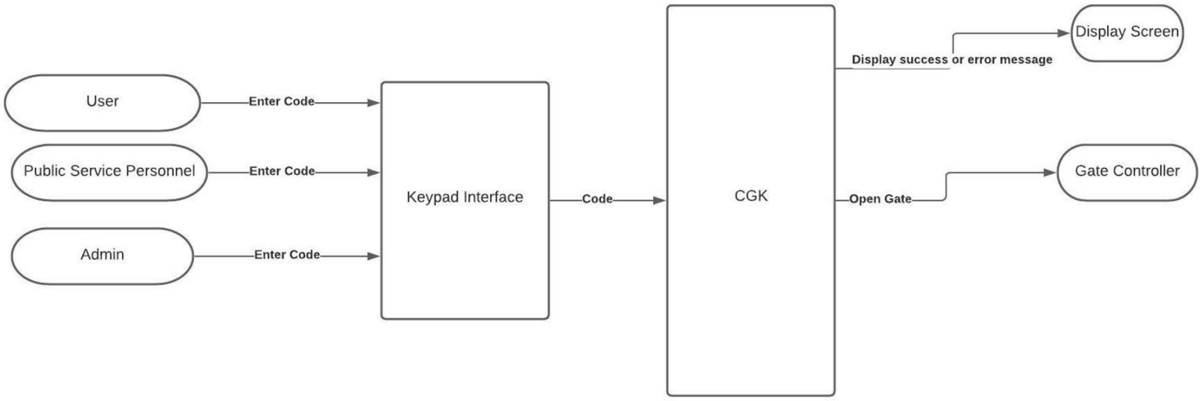

Figure 2 - Logical Diagram

3.1 External Interfaces

There are a couple external sources or interfaces that the CGK needs to communicate with, namely, the gate controller, and the users.

The CGK interfaces with users through the physical keypad. This interface has two-way communication between the user and the keypad, and has multiple events. A user will press the buttons on the keypad to send number combinations to the CGK. The CGK will respond to these inputs by displaying asterisks on the screen indicating that the input was registered. Once the user has entered the fourth number, a code verification event is triggered. This event causes the keypad system to check the inputted code against the codes in its data store. If the code is correct, the CGK sends the open event to the gate controller and alerts the user. If the code is incorrect the CGK will not trigger the gate event and will notify the user.

The CGK needs to have an external interface to communicate with the gate controller. This gate controller manages all facets of the gate's operations. The keypad system does not receive any events or communication from the gate controller and only sends messages one-direction. The CGK only sends one event to the gate controller, the message to open the gate. Once the user has entered a correct entry code, the CGK will send a simple signal to the gate controller telling the gate to open and allow the user in.

However, the interface for interaction with administrative users is different. When an administrator enters the admin code, it will trigger an event to put the keypad system in code management mode instead of opening the gate. The admin can then select which code they want to change by pressing one to change the general user access code or pressing two to change the public services code. After their selection, they will be prompted to enter the four digit code. They will then be prompted to enter the code a second time. If the entered codes match each other, then the keypad system will save the new code and notify the admin with a success message. If the codes do not match or the code matches one of the previously used codes, the admin will be notified and prompted to try again. If the keypad does not receive key inputs after some timeout interval, it will exit back to the main interface awaiting user interaction.

The keypad system also consists of a number of physical parts namely, the 10-digit keypad and display screen, a microcontroller to interpret user input, a data storage unit for the codes, and an output line to the gate controller. The keypad is the sole source of user input for the CGK, this is how the user inputs codes to open the gate. The display screen is how the CGK communicates with the user, providing them with short notifications and/or information, like "WELCOME" or "WRONG CODE". The microcontroller is the brains of the CGK, it interprets user input and determines output based on that input. The microcontroller validates codes and communicates with the user through the display screen and/or communicates with the gate controller through the line out. The data storage unit stores both the current valid codes for both the user and the public services as well as the previous two codes for each; it also stores the admin code. This is used by the microcontroller whenever it needs to validate a code. This information can also be updated by the microcontroller through the code change process. The final part is the line out to the gate controller. This is how the microcontroller communicates to the gate controller to open when a valid code is entered.

It is assumed that the gate controller system has a radio sensor that communicates with EMS personnel via their vehicle radio to allow for quick entry through the gate. Therefore, those types of users will not need a code to go into the facility and they will not interact with the keypad system in any way. This action will bypass the keypad system all together.

3.2 Control Logic

The Keypad systems functionality can be expressed with three distinct states. Idle, authentication, and code management, laid out in Figure 3. The control flow always begins and returns to the idle state.

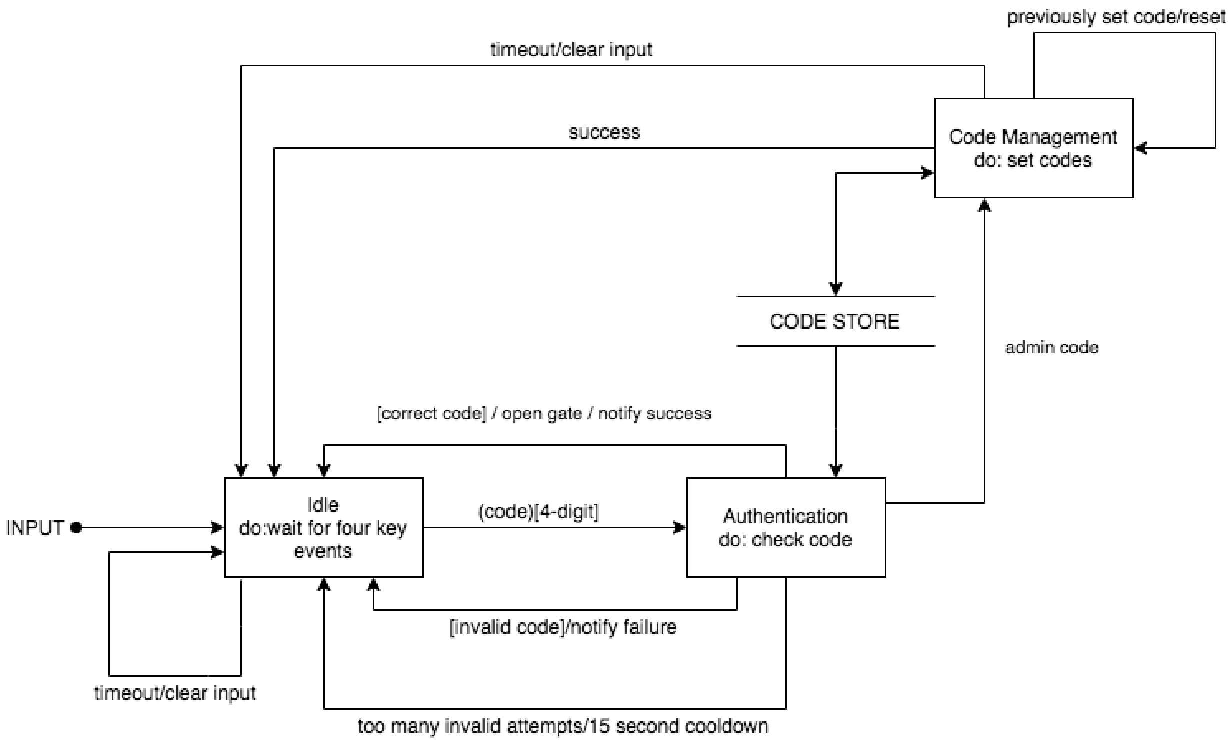
previously set code/reset
timeout/clear input
success
Code Management
do: set codes
CODE STORE
admin code
[correct code] / open gate / notify success
INPUT
Idle
do:wait for four key events
(code)[4-digit]
Authentication
do: check code
[invalid code]/notify failure
timeout/clear input
too many invalid attempts/15 second cooldown

Figure 3 - Control Logic

When in the idle state, the system waits for the user to input a four digit code. If the user enters a partial code, there will be a couple seconds of timeout and the entered code will be cleared from the screen, allowing the user to enter a 4 digit code again. Once a full code has been entered, the entered code is passed to the authentication state.

The authentication state interprets the code, and then if the entered code is successfully authenticated as the admin code, the system enters the code management state. If the entered code is authenticated as a user/public service personnel code, the gate is opened and the system returns to the idle state. If the entered 4-digit code is not valid, the display will show an error message and once again, the system returns to the idle state. If the user continuously attempts to enter the facility using invalid codes, they will have a 15 second cooldown and only after that, the system returns to the idle state.

When an admin code has been authenticated, the management state handles the process of creating new user/public service personnel codes. Now, the 2 previous user/public service personnel codes and the current user/public service personnel codes are stored in the code store. If the admin attempts to change one or both codes, they need to make the codes different from the two previous and the current codes. If they make it the same, the keypad screen will reset, entered input will be cleared and control is passed back to the code management state again. If the admin enters a partial code and no further key presses are detected for a certain amount of time, there will be a timeout, and the system returns to the idle state. If the admin makes new codes that are different from the two previous and the current codes, the new codes are stored in the code store and the system returns to the idle state.

4 Design Constraints

The keypad system will work with a gate controller that is already implemented. The communication is one-way from the keypad system to the gate controller via a buried hardwire. Our keypad system does not close the gate after a user enters nor does it open the gate from the inside; that task will be for the gate controller.

Due to the keypad system being constrained to be available 24/7, there will be a light attached to the keypad control box and the keypad system will have surge protection and waterproof sealing for inclement weather. The codes will be stored in non-volatile memory to protect the two previously used codes for both user and public service, the current user code for both user and public service, and the admin code in the event of a power outage.

The code combination will be constrained in that it will only allow input from the physical combination of the keypad numbers 0-9. The sequences are required to be 4 digits in length only.

The keypad system is also limited to simple message outputs to the user or admin because it only has a simple LCD display. The keypad system will not be connected to the property management wifi or phone service, therefore the system will not call security or the property manager. Due to the one way communication, in the event of a gate malfunction, the keypad system will not be aware of the malfunction, and it will be required of the user to call the property management.

All these system constraints ensure that the community gate keypad is simple, consistent, and convenient, and it also deters brute forcing into the facility for safety.

5 Definition of Terms

Term	**Description**
CGK	Community Gate Keypad
B-STACK	Our software company’s name
SRS	Software Requirements Specifications
RDD	Requirements Definition Document