Requirements Specification for Bicycle Garage Pro (Group 33, 2015)

Current version (commit hash): 0f04f35
Version history: `http://git.io/vvPe0`

Dennis Jin
`desuvader@gmail.com`
`https://github.com/Desuvader`
Alexander Skafte
`tfy13ask@student.lu.se`
Petter Berntsson
`dat14pbe@student.lu.se`
Emelie Löthman
`pol14elo@student.lu.se`
Adam Mzrozek
`dat14amr@student.lu.se`

Contents

1 Introduction

1.1 Purpose

This document describes the requirements and functionalities of the *Bicycle Garage Pro* software.

The intended audience of this document are mainly: software developers, in order to aid the development of the software; and clients, in order to provide an accurate overview of the project.

1.2 Glossary

Despite this being a software-only specification, relevant hardware-related terms are still explained for convenience's sake.

1. General terms
   (a) BGP - Bicycle Garage Pro (software)
   (b) ACME - The company which BGP is produced for
   (c) User - A cyclist who uses the Bicycle Garage Pro system
   (d) Operator - Subject responsible for managing BGP (on-site)
   (e) Entrance - Main entrance door of the garage. Electronic lock.
   (f) Exit 1 - Garage exit for users with bicycles. Electronic lock.
   (g) Exit 2 - Garage exit for users without bicycles. Can be opened from inside the garage.
2. Software-related terms
   (a) (The) System - Bicycle Garage Pro software
   (b) GUI - Graphical user interface
   (c) CRUD - Create, read, update and delete (operations)
   (d) PIN - 4-digit code, used to access the garage
   (e) API - Application programming interface
   (f) Database - System-relevant data is store in the database
3. Hardware-related terms
   (a) PC - Personal computer
   (b) LED - Light-emitting diode
   (c) Barcode reader - Device that can read/scan barcodes
   (d) Barcode printer - Device that can print barcodes
   (e) PIN terminal - Terminal where a PIN can be entered
   (f) Electronic lock - Electronically controlled lock on the entrance and exit 1

1.3 Scope

BGP is a software application that manages a bicycle garage. This requirement specification only covers the software part for the bicycle garage and the following assumptions are therefore made about the hardware:

1. The hardware handles recognition and reading (scanning) of barcodes
2. The hardware handles printing of barcodes
3. The hardware handles locking and unlocking of doors
4. The hardware handles reading of PINs

The hardware, as stated above, will be accessed by the system according to a predefined API. See appendix A for a more detailed description of the API.

1.4 Goals

1.4.1 Business goals

ACME is a company that works in the bicycle garage business in Sweden. ACME aims to provide low-cost, semi-automated bicycle garage solutions for public use. The goals for ACME with this software are mainly:

1. Customer retention through maintainability

1.4.2 Product goals

The goal of BGP is to assist in managing a bicycle garage by keeping track of users and their bicycles. The user interface should be simple to use and the software should be easily maintainable for future developers.

1.5 Overview

The structure of this specification is based on guidelines from the ETSA01 course website and the ETSA01 course compendium [1, 2].

2 Product description

For clarification purposes, see fig. 1 for an example on how the complete garage facility (physical) can look like. The PC runs BGP.

The interactions between the system and external entities, that may interact with the system, can be seen in fig. 2. The control unit is the BGP software. The arrows in fig. 2 mainly aim to specify software-related signals that are relevant to this specification.

1. The control unit sends out the following signals:
   (a) lightLED - Sent out depending on success/failure when entering PIN or scanning barcode

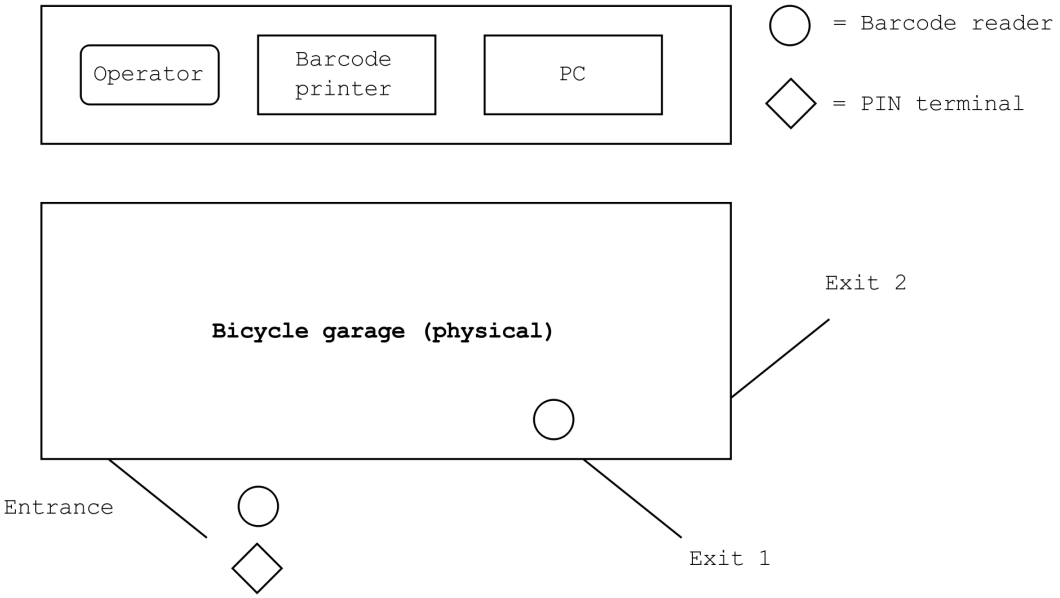

Figure 1: Simple example of how the complete garage system would look

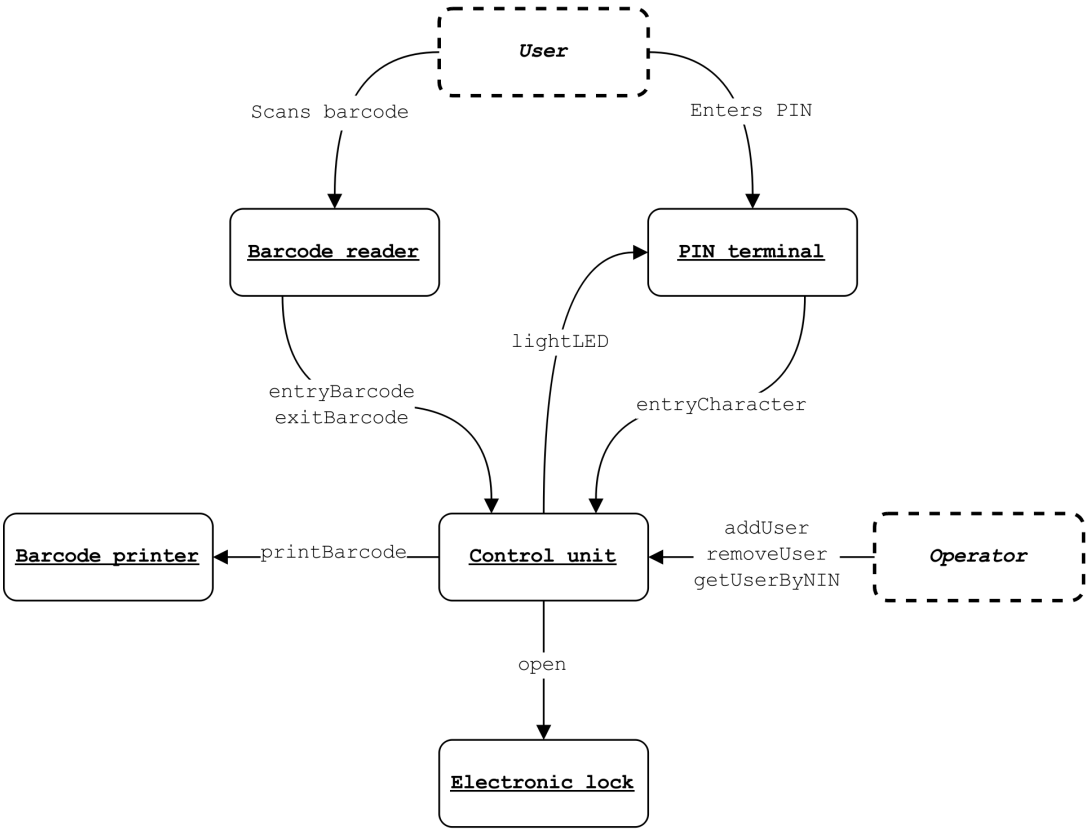

Figure 2: Context diagram for BGP

(b) printBarcode - Sent out when a barcode should be printed through the barcode printer

(c) open - Sent out to either the entrance or exit 1, and unlocks the door (electronic lock)

2. The control unit receives the following signals:

(a) entryCharacter - Received when a button is pressed on the PIN terminal

(b) entryBarcode - Received when a barcode is scanned at the entrance

(c) exitBarcode - Received when a barcode is scanned at the exit

(d) addUser - Received when the operator wants to add an user

(e) removeUser - Received when the operator wants to remove an user

(f) getUserByNIN - Received when the operator wants to search for an user by national identification number

3 Requirements

3.1 Interface

BGP should support the API defined in section 2.

3.2 Use cases

The use cases below are the most typical scenarios that one would run into when using the bicycle garage. The idea is that the system should at least be able to handle these use cases. Keep in mind that this isn't a complete description of the system and should therefore not be used as the main guidelines for developing the system.

Use case 1:	**User wants to store a bicycle**
Primary actor:	User
Preconditions:	User is registered.
Postconditions:	The bicycle of the user checked in.

Main success scenario:

1. User scans their barcode at the entrance.
2. The entrance door unlocks.

Exception scenarios:

1.a Barcode is not recognized:

1. Use case ends here. The remaining steps are skipped.

1.b Bicycle is already checked in:

1. Use case ends here. The remaining steps are skipped.

Use case 2:	**User wants to retrieve a bicycle**
Primary actor:	User
Preconditions:	User is registered. User has a registered bicycle, stored in the garage.
Postconditions:	The bicycle of the user is checked out.

Main success scenario:

1. User enters their PIN at the entrance.
2. The entrance door unlocks.
3. The user scans their barcode at exit 1.
4. Exit 1 unlocks.

Exception scenarios:

1.a PIN is incorrect:

1. Red LED blinks to indicate failure.
2. Use case ends here. The remaining steps are skipped.

3.a Barcode is not recognized:

1. Use case ends here. The remaining steps are skipped.

3.b Bicycle is already checked out:

1. Use case ends here. The remaining steps are skipped.

Use case 3:	**User wants to register**

Primary actor:	Operator
Preconditions:	User is unregistered.
Postconditions:	The user, coupled with their bicycle (indentified by barcode), has been added to the system. The user has access to the garage.

Main success scenario:

1. Operator adds the user to the system.
2. Operator provides the user with an unique barcode.
3. Operator provides the user with a PIN.

Use case 4:	**User wants to unregister**
Primary actor:	Operator
Preconditions:	User is registered.
Postconditions:	The user has been removed from the system.

Main success scenario:

1. The operator removes the user from the system.

Exception scenarios:

1.a User has a checked in bicycle:

1. User is removed anyways.
2. User will have to contact the operator for assistance.

Use case 5:	**User wants to remember their PIN**
Primary actor:	Operator
Preconditions:	User is registered.

Main success scenario:

1. The operator verifies that the user exists in the system.
2. The operator provides the user with their PIN.

Use case 6:	**User wants to store a bicycle in a full garage.**
Primary actor:	User
Preconditions:	User is registered. Garage is full.

Main success scenario:

1. User scans their barcode at the entrance.
2. Door stays locked because the garage is full.

3.3 Functional requirements

3.3.1 User-related requirements

1. The system shall keep track of following about an user:
   (a) First name and last name
   (b) National identification number (unique 10-digit string)
   (c) Unique PIN (unique 4-digit integer)
2. Each users shall have an unique 4-digit PIN that only consist of numbers ranging from 0 to 9 (inclusive).
3. The system shall store user data in a database.
4. The system shall be able to add users to the database.
5. The system shall be able to remove users from the database.
6. The system shall be able to search for users in the database, by national identification number.
7. The system shall not be able to add users once all unique PINs are in use.
8. When users are removed from the database, the system shall be able to reuse the unique PIN.

3.3.2 Bicycle-related requirements

1. The system shall keep track of following about a bicycle:
   (a) Bicycle ID (unique 5-digit integer)
   (b) Date of when the bicycles was last checked in
   (c) Date of when the bicycles was last checked out
   (d) Whether the bicycle is checked-in or not
2. The system shall store bicycle data in a database.
3. The system shall be able to add bicycles to the database.
4. The system shall be able to remove bicycles from the database.
5. The system shall not be able to add bicycles once all unique Bicycle IDs are in use.
6. The system shall not allow more than 500 checked in bicycles, in the garage, at a time.
7. When bicycles are removed from the database, the system shall be able to reuse the bicycle ID.

3.3.3 Interaction-related requirements

1. When a *correct* PIN is entered at the entrance:
   (a) The LED shall show green for 4 seconds.
   (b) The system shall unlock the entrance for 10 seconds.
2. When an *incorrect* PIN is entered at the entrance the LED shall show red for 4 seconds.
3. When a recognized barcode is scanned at the entrance:
   (a) The system shall unlock the entrance for 10 seconds.
   (b) A new check-in date should be set for the bicycle.
4. When a recognized barcode is scanned at exit 1, the system shall unlock the exit for 10 seconds.

3.3.4 Miscellaneous

1. The system shall provide a relevant graphical interface needed for an operator to manage the system. See use cases, where the operator is the primary actor, in section 3.2 for relevant scenarios.
2. If the delay between two user input signals from the PIN terminal exceeds 5 seconds, the system shall reset the PIN terminal.

3.4 Quality requirements

1. The database shall persist on disk (PC) despite power failure.
2. The user interface shall take at most 0.2 seconds to respond to each individual user input.

References

[1] ETSA01 project guidelines. `http://cs.lth.se/etsa01/projekt-2015/`. Accessed: 2015-04-16.

[2] ETSA01 VT 2015 Examples and Excercises in the Software Engineering Process. LTH, Lund, 2015.

A Hardware API

This API specification is directly copied from `http://cs.lth.se/etsa01/projekt-2015/haardvarugraenssnitt-och-drivrutiner/`. If there are any questions regarding the hardware API, please contact Markus Borg (`Markus.Borg@cs.lth.se`).

```
public interface BarcodePrinter {
        /* Print a bicycleID as a barcode.
         * Bicycle ID should be a string of 5 characters,
            where every
         * character can be '0', '1',... '9'. */

        public void printBarcode(String bicycleID);
}

public interface PinCodeTerminal {
        /* Register bicycle garage manager so
         * that the pin code terminal knows
         * which manager to call when a user has
         * pressed a key. */

        public void register(BicycleGarageManager manager);

        /* Turn on LED for lightTime seconds.
         * Colour:
         * colour = RED_LED = 0 => red
         * colour = GREEN_LED = 1 => green */

        public void lightLED(int colour, int lightTime);
        public static final int RED_LED = 0,
        GREEN_LED = 1;
}

public interface BarcodeReader {
        /* Register bicycle garage manager
         * so that the bar code reader knows
         * which manager to call when a user
         * has used the reader. */

        public void register(BicycleGarageManager manager);
}

public interface ElectronicLock {
        /* Open the lock for timeOpen seconds.  */

        public void open(int timeOpen);
}
```