|  |  |
| --- | --- |
| logo ngan.png | MINISTRY OF EDUCATION  AND TRAINING |
| **FPT UNIVERSITY** | |
| Capstone Project Document | |
| **Barcode as key door using in smart house or smart hotel** | |
|  | |
| |  |  | | --- | --- | | **Group 5** | | | **Group Members** | Vo DuyThuc - Leader - SE60901  Luong Van Doan - Dropped out  Tran Vo Hoang - Member - SE61104  Nguyen Le Canh - Dropped out  Nguyen Thanh Nam - Member - SE60858 | | **Supervisor** | Nguyen Duc Loi | | **Ext Supervisor** | N/A | | **Capstone Project code** | BAKDISHH | | |

-Ho Chi Minh City, January 2015-

*This page is intentionally left blank*

# Table of Contents

[Table of Contents 3](#_Toc417277598)

[List of Figures 6](#_Toc417277599)

[List of Tables 9](#_Toc417277600)

[Acknowledgements 10](#_Toc417277601)

[Definitions, Acronyms and Abbreviations 11](#_Toc417277602)

[Preface 12](#_Toc417277603)

[INTRODUCTION 13](#_Toc417277604)

[I. Project information 13](#_Toc417277605)

[II. Introduction 13](#_Toc417277606)

[III. Existing Solution 13](#_Toc417277607)

[IV. Problem Definition 14](#_Toc417277608)

[V. Proposed Solution 15](#_Toc417277609)

[1. Feature functions 15](#_Toc417277610)

[2. Advantages and disadvantages 15](#_Toc417277611)

[VI. Project Overview 16](#_Toc417277612)

[1. Hardware Overview 16](#_Toc417277613)

[2. System Overview 20](#_Toc417277614)

[VII. Project Scope 20](#_Toc417277615)

[VIII. Role and Responsibility 21](#_Toc417277616)

[PROJECT MANAGEMENT PLAN (PMP) 21](#_Toc417277617)

[I.Problem Definition 21](#_Toc417277618)

[1. Name of this Capstone Project 21](#_Toc417277619)

[2. Problem Abstract 21](#_Toc417277620)

[3. Project Overview 22](#_Toc417277621)

[II. Project organization 23](#_Toc417277622)

[1.Software Process Model 23](#_Toc417277623)

[2. Roles and responsibilities 24](#_Toc417277624)

[3. Tools and Techniques 25](#_Toc417277625)

[III. Project Management Plan 26](#_Toc417277626)

[1. Task plan 26](#_Toc417277627)

[2. Task sheet 29](#_Toc417277628)

[IV. Coding Convention 30](#_Toc417277629)

[SYSTEM REQUIREMENTS SPECIFICATIONS (SRS) 31](#_Toc417277630)

[I.User Requirement Specification 31](#_Toc417277631)

[II. System Requirement Specification – Home Use 31](#_Toc417277632)

[A. Hardware Requirement 31](#_Toc417277633)

[B. Software Requirement 31](#_Toc417277634)

[C. Controller Requirement 49](#_Toc417277635)

[III. System Requirement Specification – Hotel Use 65](#_Toc417277636)

[1. User Interface 65](#_Toc417277637)

[2. Functional Requirement 65](#_Toc417277638)

[3. Non-Functional Requirement 80](#_Toc417277639)

[SYSTEM DESIGN DESCRIPTION (SDD) 82](#_Toc417277640)

[I.Design Overview 82](#_Toc417277641)

[II. System Architectural Design 82](#_Toc417277642)

[III. Component Diagram 83](#_Toc417277643)

[IV. State Diagram 83](#_Toc417277644)

[V. Detail Description of Components 83](#_Toc417277645)

[1. Hardware 83](#_Toc417277646)

[2.Software 97](#_Toc417277647)

[3. Controller 150](#_Toc417277648)

[VI. Algorithm and Library 151](#_Toc417277649)

[1. Introduce QR code 151](#_Toc417277650)

[2. Zbar library 154](#_Toc417277651)

[SYSTEM IMPLEMENTATION & TEST (SIT) 156](#_Toc417277652)

[I. Introduction 156](#_Toc417277653)

[1. System overview 156](#_Toc417277654)

[2. Test Approach 156](#_Toc417277655)

[II. Test plan 156](#_Toc417277656)

[III. Test Case 156](#_Toc417277657)

[1. Mobile app test case 158](#_Toc417277658)

[2. Hardware test case 171](#_Toc417277659)

[SYSTEM USER’S MANUAL 174](#_Toc417277660)

[I.Installation Guide 174](#_Toc417277661)

[II. User’s Guide 174](#_Toc417277662)

[1. Mobile application 174](#_Toc417277663)

[2. Lock Control module 188](#_Toc417277664)

[3. Web application (for use in hotel) 188](#_Toc417277665)

[REFERENCES 194](#_Toc417277666)

# List of Figures

[Figure 1: Existing lock type 13](#_Toc417277354)

[*Figure 2: beaglebone black rev C* 15](#_Toc417277355)

[*Figure 3: BBB back side* 16](#_Toc417277356)

[*Figure 4: Solenoid lock* 18](#_Toc417277357)

[*Figure 5: system overview* 19](#_Toc417277358)

[*Figure 6: Finger lock* 21](#_Toc417277359)

[*Figure 7: RF smart lock* 22](#_Toc417277360)

[*Figure 8: Agile Process Model* 23](#_Toc417277361)

[*Figure 9: "Login Use Case Diagram"* 31](#_Toc417277362)

[*Figure 10: "Change Password" Use case Diagram* 33](#_Toc417277363)

[*Figure 11: Controller Use Case Diagram* 48](#_Toc417277364)

[*Figure 12: Overall system architecture* 81](#_Toc417277365)

[*Figure 13: Beaglebone black rev C top view* 82](#_Toc417277366)

[*Figure 14: beaglebone black rev C bottom view* 83](#_Toc417277367)

[*Figure 15: beaglebone black rev C block diagram* 84](#_Toc417277368)

[*Figure 16: Expansion header Pin 8 out* 85](#_Toc417277369)

[*Figure 17: Expansion header Pin 9 out* 85](#_Toc417277370)

[*Figure 18: power supplier circuit* 86](#_Toc417277371)

[*Figure 19: Power supplierschematic diagram* 87](#_Toc417277372)

[*Figure 20: Power supplier and lock PCB* 88](#_Toc417277373)

[*Figure 21: Adapter DC 12V* 88](#_Toc417277374)

[*Figure 22: Replay 12V* 89](file:///E:\Capstone\Document\Report%20Main\Final-report.docx#_Toc417277375)

[*Figure 23: IC 7805* 89](file:///E:\Capstone\Document\Report%20Main\Final-report.docx#_Toc417277376)

[*Figure 25: TX and RX led* 90](#_Toc417277377)

[*Figure 24: TIP41C* 90](file:///E:\Capstone\Document\Report%20Main\Final-report.docx#_Toc417277378)

[*Figure 26: Infrared sensor board* 91](#_Toc417277379)

[*Figure 27: Infrared sensor schematic* 92](#_Toc417277380)

[*Figure 28: Infrared sensor PCB layout* 92](#_Toc417277381)

[*Figure 29: Photodiode and infrared emitting diode* 93](#_Toc417277382)

[*Figure 30: IC LM393P* 93](#_Toc417277383)

[*Figure 31: Solenoid lock* 94](#_Toc417277384)

[*Figure 32: Solenoid lock constitute* 95](#_Toc417277385)

[*Figure 33: Circuit diagram of components* 96](#_Toc417277386)

[*Figure 34: Class diagram* 99](#_Toc417277387)

[*Figure 35: Login sequence* 112](#_Toc417277388)

[*Figure 36 : Logout sequence* 113](#_Toc417277389)

[*Figure 37: Create User sequence* 114](#_Toc417277390)

[*Figure 38: Change Password sequence* 115](#_Toc417277391)

[*Figure 39: Change Key sequence* 116](#_Toc417277392)

[*Figure 40: Display QR Code sequence* 117](#_Toc417277393)

[*Figure 41: Reset User Password sequence* 118](#_Toc417277394)

[*Figure 42: View Log sequence* 119](#_Toc417277395)

[*Figure 43: Setting Log sequence* 120](#_Toc417277396)

[*Figure 44: Login Page* 121](#_Toc417277397)

[*Figure 45: Admin Page* 122](#_Toc417277398)

[*Figure 46: User Page* 124](#_Toc417277399)

[*Figure 47: Change Key Page* 125](#_Toc417277400)

[*Figure 48: Change Password Page* 126](#_Toc417277401)

[*Figure 49: View Log Page* 128](#_Toc417277402)

[*Figure 50: Open Door Now Page* 129](#_Toc417277403)

[*Figure 51: Create New User Page* 130](#_Toc417277404)

[*Figure 52: Manager User Page* 131](#_Toc417277405)

[*Figure 53: Reset Password Page* 132](#_Toc417277406)

[*Figure 54: Setting Log Page* 133](#_Toc417277407)

[*Figure 55: Booking room sequence* 135](#_Toc417277408)

[*Figure 56: change static key sequence* 136](#_Toc417277409)

[*Figure 57: Show key and send email sequence* 137](#_Toc417277410)

[*Figure 58: Get static key on schedule sequence* 138](#_Toc417277411)

[*Figure 59: Unlock sequence* 139](#_Toc417277412)

[*Figure 60: Login* 140](#_Toc417277413)

[*Figure 61: Magage page* 141](#_Toc417277414)

[*Figure 62: Manage Key* 142](#_Toc417277415)

[*Figure 63: Manage information* 143](#_Toc417277416)

[*Figure 64: Staff page* 144](#_Toc417277417)

[*Figure 65: Manage key- staff* 144](#_Toc417277418)

[*Figure 66: Guest Page* 145](#_Toc417277419)

[*Figure 67: Reservation page* 146](#_Toc417277420)

[*Figure 68: Book room flowchart* 147](#_Toc417277421)

[*Figure 69: Unlock flowchart* 148](#_Toc417277422)

[*Figure 70: A QR code* 150](#_Toc417277423)

[*Figure 71: Read information on QR by mobile* 151](#_Toc417277424)

[*Figure 72: Structure of QR code* 152](#_Toc417277425)

[*Figure 73: Format information* 152](#_Toc417277426)

[*Figure 74: Character Placement* 153](#_Toc417277427)

[*Figure 75: Zbar* 153](#_Toc417277428)

[*Figure 76: Login* 173](#_Toc417277429)

[*Figure 77: Re input information* 174](#_Toc417277430)

[*Figure 78: User page* 174](#_Toc417277431)

[*Figure 79: Admin Page* 175](#_Toc417277432)

[*Figure 80: Change Password* 176](#_Toc417277433)

[*Figure 81: Reinput information change password* 176](#_Toc417277434)

[*Figure 82: Wrong confirm new password* 177](#_Toc417277435)

[*Figure 83: Come back User page after change password* 177](#_Toc417277436)

[*Figure 84: View Log* 178](#_Toc417277437)

[*Figure 85: Change QR code* 179](#_Toc417277438)

[*Figure 86: Open Door Now* 180](#_Toc417277439)

[*Figure 87: Reinput QR code to open door* 180](#_Toc417277440)

[*Figure 88: Display QR code* 181](#_Toc417277441)

[*Figure 89: Confirm Logout* 182](#_Toc417277442)

[*Figure 90: After Logout will come back Login page* 182](#_Toc417277443)

[*Figure 91: Admin page* 183](#_Toc417277444)

[Figure 92: *Setting Log* 183](#_Toc417277445)

[*Figure 93:Reset User Password* 184](#_Toc417277446)

[*Figure 94: Create User* 185](#_Toc417277447)

[*Figure 95: Wrong format information* 185](#_Toc417277448)

[*Figure 96: Lack information* 186](#_Toc417277449)

[*Figure 97: Manager User* 187](#_Toc417277450)

[*Figure 98: Access* 187](file:///E:\Capstone\Document\Report%20Main\Final-report.docx#_Toc417277451)

[*Figure 100: Manage Key* 188](#_Toc417277452)

[*Figure 99: Manage room* 188](file:///E:\Capstone\Document\Report%20Main\Final-report.docx#_Toc417277453)

[*Figure 101: Comeback Manage Page* 189](file:///E:\Capstone\Document\Report%20Main\Final-report.docx#_Toc417277454)

[*Figure 102: Manage reservations* 189](file:///E:\Capstone\Document\Report%20Main\Final-report.docx#_Toc417277455)

[*Figure 104: Send key to email staff* 190](#_Toc417277456)

[*Figure 103: Display Key* 190](file:///E:\Capstone\Document\Report%20Main\Final-report.docx#_Toc417277457)

[*Figure 105: View room* 191](#_Toc417277458)

[*Figure 106: Booking room* 191](file:///E:\Capstone\Document\Report%20Main\Final-report.docx#_Toc417277459)

# List of Tables

[*Table 1 List of Acronyms and Abbreviations 8*](#_Toc417175311)

[*Table 2: describe some characteristic of existing devices* 11](#_Toc417175312)

[*Table 3: Role and responsibility* 17](#_Toc417175313)

[*Table 4: Roles and Responsibilities* 21](#_Toc417175314)

[*Table 5: Task plan* 24](#_Toc417175315)

[*Table 6: Task sheet* 26](#_Toc417175316)

[*Table 7: All meeting minutes* 26](#_Toc417175317)

[*Table 8: Replay 12V technical details* 47](#_Toc417175318)

[*Table 9: IC 7805 technical details* 47](#_Toc417175319)

# Acknowledgements

This project consumed huge amount of research and work. Still, implementation would not have been possible, if we did not have a support of many instructors, families and friends. Therefore, we would like to extent our sincere gratitude to all of them.

In addition, we had to take the help and guideline of some respected persons, who deserve our greatest gratitude, especially, we would like toshow our gratitude Mr. Bui (Nguyen Duc Loi), ES supervisor, FPT University, for giving us provision of expertise for the project throughout numerous consultations in the implementation.

Nevertheless, we express our gratitude toward classmates and team members itself who have made valuable comment suggestions and helped ours a lot in finalizing this project within the limited time frame. We also would like to send special thanks to our family who always encourages and helped us during the time.

Finally, we want to give a special thank all the people for their help directly and indirectly to complete the project.

# Definitions, Acronyms and Abbreviations

|  |  |  |
| --- | --- | --- |
| No. | Abbreviations/Acronyms | Definitions |
| 1 | BBB | Beaglebone Black |
| 2 | GPIO | General Purpose Input Output |
| 3 | OPAMP | Operational amplifier |

*Table 1 List of Acronyms and Abbreviations*

# Preface

This document …

# INTRODUCTION

## I. Project information

* Project Name: Barcode as key door using in smart house or smart hotel
* Project Code:
* Product Type: Desktop app, mobile app, embedded software.
* Start Date: January 6 2015
* End Date: April 20 2015

## II. Introduction

By high technologies, today, the way of living becomes more modern and more easies. Developments of smart devices are topics of universal interest, especially smart house and smart hotel. In addition, one of the most importance of smart house or hotel is security.

Barcode…

The Barcode as key door is the safe, simple, and social way to manage your home’s lock. Now you can control who can enter and who can’t—without the need for keys. And you can manage your house or hotel all from your smartphone or computer.

## III. Existing Solution

To day people often use traditional lock, fingerprint lock and electromagnetic lock is popular in many place but it have some place to agree with them because fingerprint lock and electromagnetic lock so expensive with some body have little money and they can’t buy it. About traditional lock is so simple it can be destroy and not security because it not give safe impression for user. Using barcode as key door in smart house or smart hotel is still new which many user.



Figure 1: Existing lock type

## IV. Problem Definition

|  |  |  |
| --- | --- | --- |
|  | **Advantages** | **Disadvantages** |
| **Traditional lock** | -High capacity  -Easy to use  -Cheap price | -Bad security  -Very compound when lost a key |
| **Fingerprint lock** | -High flexible  -High security  -High quanlity  -Easy to use  -Can change a key | -Expensive price  -Hard to change a fingerprint |
| **Electromagnetic lock** | -High flexible  -High security  -Easy to use  -Can change a key  -High quality | -Expensive price  -Hard to change a key  -Have a key card key like traditional key |
| **QR code lock** | -High flexible  -High security  -High quality  -Easy to use  -Easy change a key | -High price  -Have smart device  -Must connect with one wifi and internet. |

*Table 2: describe some characteristic of existing devices*

## V. Proposed Solution

Beagle Bone kit running embedded Linux gets qrcode data from a webcam. When it detects a barcode, it compares it with a database stored approved keys and will unlock the door for users.

### Feature functions

* Client:

+ Provide an access point for wifi connection.

+ Receive signal from main board and verify with key user give through camera.

+ Provide a key can open the door with key have been set before.

* Server:

+ Provide a web service and mobile application in order control and interact with mainboard and database.

+ Connect with wifi and synchronize with mobile application.

### Advantages and disadvantages

* Advantages:

+ Low cost in comparison with the function of the lock can do.

+ Increase usability.

+ Easy to use and easy to change the key.

+ Increase security.

* Disadvantages:

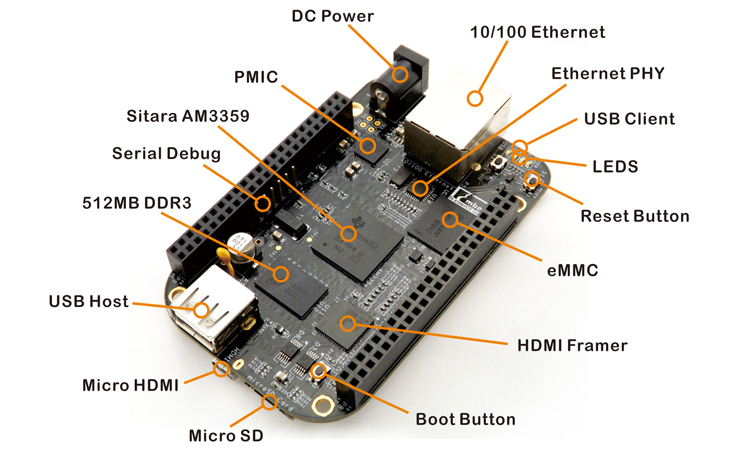
+ User must have a smart phone or smart device and can know use it.

+ Must connect with one wifi and internet because if user want change password user must connect with wifi or internet.

## VI. Project Overview

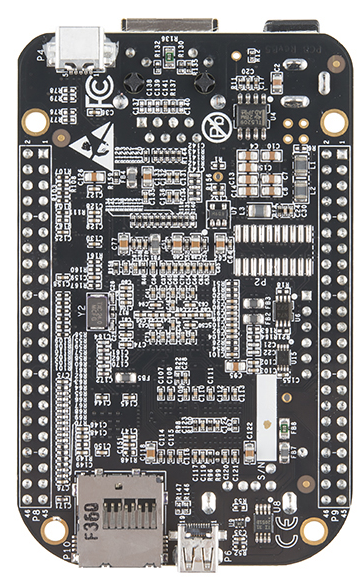
### Hardware Overview

#### Beablebone black rev C



*Figure 2: beaglebone black rev C*

BeagleBone Black is a low-cost, community-supported development platform for developers and hobbyists. Boot Linux in under 10 seconds and get started on development in less than 5 minutes with just a single USB cable.



*Figure 3: BBB back side*

Technical details:

* Processor: Sitara AM3358BZCZ100 1GHz, 2000 MIPS
* Graphics Engine: SGX530 3D, 20M Polygons/S
* SDRAM Memory: 512MB DDR3L 606MHZ
* Onboard Flash: **4GB**, 8bit Embedded MMC
* PMIC: TPS65217C PMIC regulator and one additional LDO.
* Debug Support: Optional Onboard 20-pin CTI JTAG, Serial Header
* Power Source: miniUSB USB or DC Jack, 5VDC External Via Expansion Header
* PCB: 86.44mm x 54.54mm / 3.4” x 2.15” 6 layers
* 88.98mm x 54.54mm x 15.41mm / 3.5" x 2.15" x 0.6"
* Indicators 1-Power, 2-Ethernet, 4-User Controllable LEDs
* HiSpeed USB 2.0 Client Port: Access to USB0, Client mode via miniUSB
* HiSpeed USB 2.0 Host Port Access to USB1, Type A Socket, 500mA LS/FS/HS
* Serial Port UART0 access via 6 pin 3.3V TTL Header. Header is populated
* Ethernet 10/100, RJ45
* SD/MMC Connector microSD , 3.3V
* User Input: Reset Button, Boot Button, Power Button
* Video Out: 16b HDMI, 1280x1024 (MAX), 1024x768,1280x720,1440x900 w/EDID Support
* Audio Via HDMI Interface, Stereo
* Expansion Connectors:
  + Power 5V, 3.3V , VDD\_ADC(1.8V)
  + 3.3V I/O on all signals
  + McASP0, SPI1, I2C, GPIO(65), LCD, GPMC, MMC1, MMC2, 7 AIN(1.8V MAX), 4 Timers, 3 Serial Ports, CAN0, EHRPWM(0,2),XDMA Interrupt, Power button, Expansion Board ID (Up to 4 can be stacked)
* Weight 40.55g / 1.43oz

#### Solenoid lock



*Figure 4: Solenoid lock*

Solenoids are basically electromagnets: they are made of a big coil of copper wire with an armature (a slug of metal) in the middle. When the coil is energized, the slug is pulled into the center of the coil. This makes the solenoid able to pull from one end.

This solenoid in particular is nice and strong, and has a slug with a slanted cut and a good mounting bracket. It's basically an electronic lock, designed for a basic cabinet or safe or door. Normally the lock is active so you can't open the door because the solenoid slug is in the way. It does not use any power in this state. When 9-12VDC is applied, the slug pulls in so it doesn't stick out anymore and the door can be opened

Technical details:

* 12VDC (you can use 9-12 DC volts, but lower voltage results in weaker/slower operation)
* Draws 650mA at 12V, 500 mA at 9V when activated
* Designed for 1-10 seconds long activation time
* Max Dimensions: 41.85mm / 1.64" x 53.57mm / 2.1" x 27.59mm / 11.08"
* Dimensions: 23.57mm / 0.92" x 67.47mm / 2.65" x 27.59mm / 11.08"
* Wire length: 222.25mm / 8.75"
* Weight: 147.71g

### System Overview



*Figure 5: system overview*

## VII. Project Scope

**Use at home:**

* Create a training database with qrcodethat are and are not allowed to open the door.
* Use ZBar libraries to read barcodes.
* Build electrical circuits to drive and control peripheral devices such as solenoid lock, led indicator.
* Send to users an alarm message when the users with wrong barcode attempting to open the door.

**Use at hotel:**

* Perform the same tasks as above. Additional the following functions were appended to the program.
* Allow Guest to books for the hotel on mobile application/Web application.
* The system allows user creating representative barcode and setting the period of stay, then send it to the guest.

## VIII. Role and Responsibility

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Full Name | Role | Position | Contact |
| 1 | Nguyen Duc Loi | Project manager | Supervisor |  |
| 2 | Vo DuyThuc | Developer | Leader |  |
| 3 | Luong Van Doan | Developer | Member |  |
| 4 | Tran Vo Hoang | Developer | Member |  |
| 5 | Nguyen Thanh Nam | Developer | Member |  |
| 6 | Nguyen Le Canh | Dropped out |  |  |

*Table 3: Role and responsibility*

# PROJECT MANAGEMENT PLAN (PMP)

## I.Problem Definition

### 1. Name of this Capstone Project

Project name: Barcode as key door using in smart house or smart hotel

Project code: BAKDISHH

### 2. Problem Abstract

Barcode is very popular in our life. It’s easy and cheap to have a barcode by generate it in a smartphone or print them on the paper from the computer or laptop. Barcode can include information and we can use them as a password to open our door.

We will use barcode as a key door, the door have a camera will scan the barcode, compare the data they receive in barcode with the data in database and decide to open the door or not. We also need to build a driver circuit to control a lock style solenoid. We can add some feature to system like some indicator leds, play sound to notice the user. One importance thing that our door can work by both electric or battery power, so we should have a replay to switch between them.

We can use this system in hotel. The guests book the hotel room online, then the representative barcode was generated on hotel Web application and was sent to guest (via email, message, etc). The guests use this barcode to open the hotel door when staying at hotel.

### 3. Project Overview

#### 3.1 Current Situation and Disadvantages

Nowadays, with the development of technology, there are many kinds of lock in the market like finger lock, RF smart lock… and of course they are expensive. With RF smart lock, what happen if we lost RF card? How can we open the door in this case?



*Figure 6: Finger lock*



*Figure 7: RF smart lock*

#### 3.2 The Proposed System

Our system have a Beagle Bone kit running embedded Linux gets barcode data from a webcam. When it detects a barcode, it compares it with a database stored approved keys and will unlock the door for users.

Our system can be use in hotel : the guests book the hotel room online, then the representative barcode was generated on hotel Web application and was sent to guest (via email, message, etc). The guests use this barcode to open the hotel door when staying at hotel.

#### 3.3 Boundaries of the System

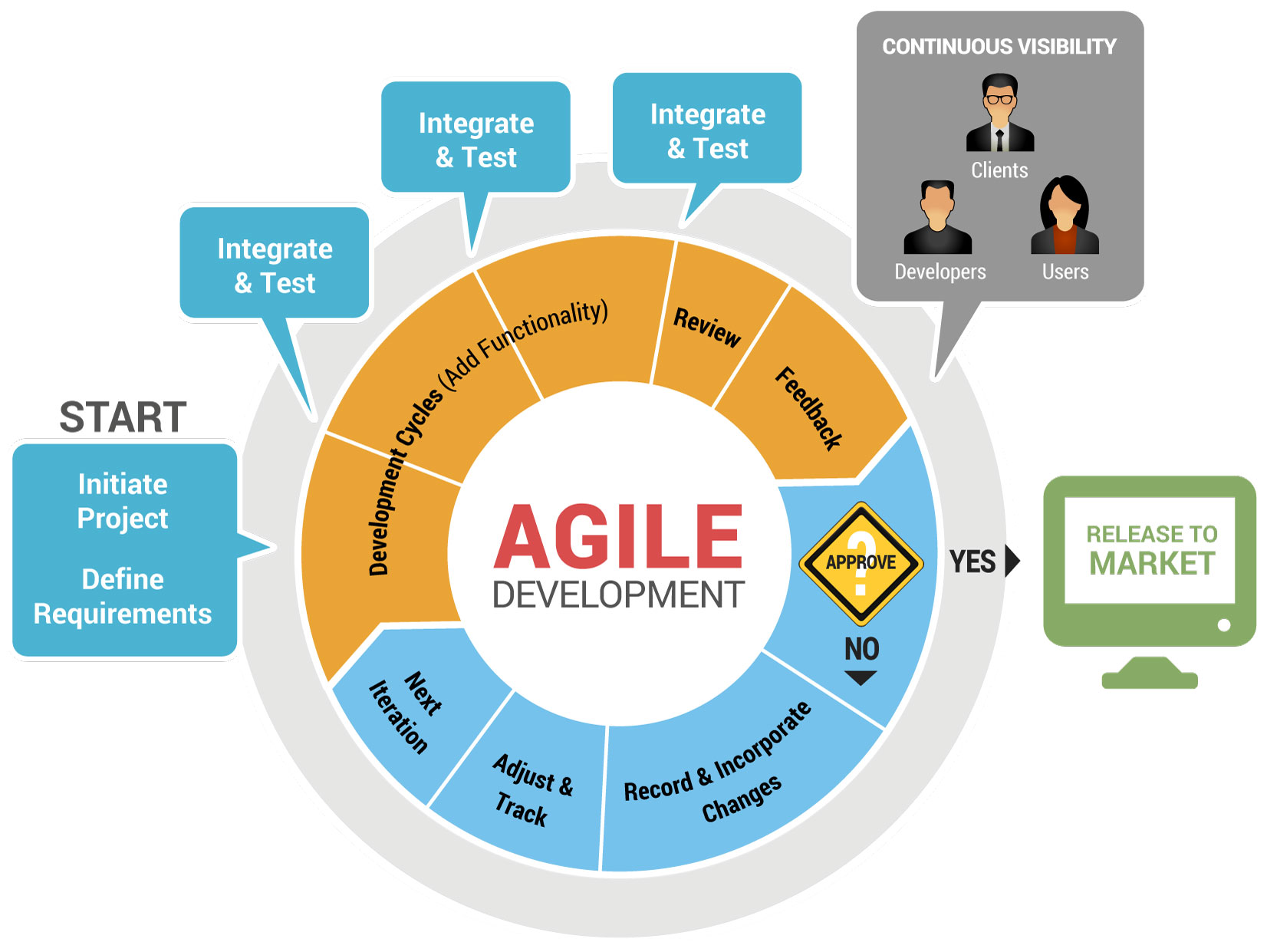
#### 3.4 Development Environment

## II. Project organization

### 1.Software Process Model

Agile software development is a group of software development methods in which requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. It promotes adaptive planning, evolutionary development, early delivery, continuous improvement, and encourages rapid and flexible response to change.

Agile software development is a flexible process that rapidly builds each piece of a software project, adding functionality every few weeks until the entire project is complete. This kind of rapid development has the flexibility to meet the demands of evolving business requirements. In this case, embedded system development.



*Figure 8: Agile Process Model*

### 2. Roles and responsibilities

|  |  |  |
| --- | --- | --- |
| Full name | Role | Responsibilities |
| Nguyen Duc Loi | Supervisor | -Specify user requirements  -Advisor for ideas & solutions  -Control the developing process |
| Vo DuyThuc | Team leader,  Developer,  Tester | -Managing process  -Assign task for team members.  -Complete all individual work.  -Support team members |
| Luong Van Doan | Team leader,  Developer,  Tester | -Complete individual task on time.  -Provide opinions and feedbacks.  -Support team members. |
| Tran Vo Hoang | Team leader,  Developer,  Tester | -Complete individual task on time.  -Provide opinions and feedbacks.  -Support team members. |
| Nguyen Thanh Nam | Team leader,  Developer,  Tester | -Complete individual task on time.  -Provide opinions and feedbacks.  -Support team members. |
| Nguyen Le Canh | Dropped out |  |

*Table 4: Roles and Responsibilities*

### 3. Tools and Techniques

#### 3.1 Development environment

Hardware requirements

* Beaglebone Bone Black Rev C kit
* Solenoid Lock
* Driver Circuit
* Webcam or camera
* Android phone
* Personal computer with minimum configuration: 2Gb of Ram, 50Gb of Hard disk, core 2 Duo 2.0Ghz, USB 2.0
* Internet connection

Software requirements

* Windows 7 or above
* Ubuntu 12 or above
* Eclipse for Android development
* Eclipse CDT
* Visual Studio 2013
* Notepad++
* Start UML
* Microsoft Office 2010 or above
* Virtual box

#### 3.2 Management environment

* Google Code to store all document and source code
* Source Version Control : to synchronize /control source code and document

#### 3.3 Communication environment

* Skype
* Google mail, Fpt mail
* Facebook

## III. Project Management Plan

### 1. Task plan

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| N  o | Task /Phase | Description | Deliverables  /Output | Resources Needed (man-day) | Dependencies & Constraints | Risks |
| 1 | Initiating and planning | -Project register, kick-off meeting and planning. | -Project introduction  -Project plan | 5 | At least 3 members | -Conflict project with other team.  -Not enough members. |
| 2 | Requirements Specification | -Study and make the requirements clear.  -Hardware selecting | - Software requirements Specification document.  -Hardware selected | 5 | Project registered | -Miss understand requirement. |
| 3 | Hardware and device purchasing and studying | -Purchase selected hardware : Beaglebone Black Rev C  -Purchase device : Solenoid lock, Webcam,  - Study Beagle Bone datasheet  -Study Soleonid lock usage | - Hardware and device  -Hardware related documents. |  | N/A | -Out of stock |
| 4 | System Design | -Design modules, communicate method between modules and the whole system. | -System Detail Design document |  |  |  |
| 5 | Zbar library and QrCode Studying |  |  |  |  |  |
| 6 | Qrcode encode |  |  |  |  |  |
| 7 | QrCode decode |  |  |  |  |  |
| 8 | Driver circuit |  |  |  |  |  |
| 9 | Indicator led controlling |  |  |  |  |  |
| 10 | Sound Controlling |  |  |  |  |  |
| 11 | Electric power and battery switching |  |  |  |  |  |
| 12 | Sensor to detect people |  |  |  |  |  |
| 13 | Develop Android Program |  |  |  |  |  |
| 14 | Develop a Server in beaglebone kit |  |  |  |  |  |
| 15 | Develop a Website for Use in hotel |  |  |  |  |  |
| 16 | Code Combination |  |  |  |  |  |
| 17 | Package the product |  |  |  |  |  |
| 18 | Integration and system test |  |  |  |  |  |
| 19 | Monitoring and controlling |  |  |  |  |  |

*Table 5: Task plan*

### 2. Task sheet

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Parent Module** | **Task** | **ThucVD** | **DoanLV** | **HoangTV** | **CanhNL** | **NamNT** | **Unit** | **Size** |
| 1 | Initiating and Planning | **Report 1 - Introduction** |  |  |  |  |  |  |  |
| Project informantion |  |  |  |  |  |  |  |
| Introduction |  |  |  |  |  |  |  |
| Curent Situation |  |  |  |  |  |  |  |
| Problem Definition |  |  |  |  |  |  |  |
| Proposed Solution |  |  |  |  |  |  |  |
| Functional requirement |  |  |  |  |  |  |  |
| Review |  |  |  |  |  |  |  |
| **Report 2 - Project Management Plan** |  |  |  |  |  |  |  |
| Problem Definition |  |  |  |  |  |  |  |
| Project Organization |  |  |  |  |  |  |  |
| Project Management Plan |  |  |  |  |  |  |  |
| Review |  |  |  |  |  |  |  |
| 2 | On board program | **Report 3 -System Requirements  Specifications** |  |  |  |  |  |  |  |
| **Implementation** |  |  |  |  |  |  |  |
| QrCode encode |  |  |  |  |  |  |  |
| GPIO controlling |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| **Testing** |  |  |  |  |  |  |  |
| 3 | Web Application | **System Requirements Specifications** |  |  |  |  |  |  |  |
| **Implementation** |  |  |  |  |  |  |  |
| Business Logic |  |  |  |  |  |  |  |
| Main Flow |  |  |  |  |  |  |  |
| **Testing** |  |  |  |  |  |  |  |
| 4 | Android Application | **System Requirements Specifications** |  |  |  |  |  |  |  |
| **Implementation** |  |  |  |  |  |  |  |
| Business Logic |  |  |  |  |  |  |  |
| Main Flow |  |  |  |  |  |  |  |
| **Testing** |  |  |  |  |  |  |  |
| 5 | Report 6 - Software User's Manual | Installation Guide |  |  |  |  |  |  |  |
| User’s Guide |  |  |  |  |  |  |  |

*Table 6: Task sheet*

## IV. Coding Convention

# SYSTEM REQUIREMENTS SPECIFICATIONS (SRS)

## I.User Requirement Specification

## II. System Requirement Specification – Home Use

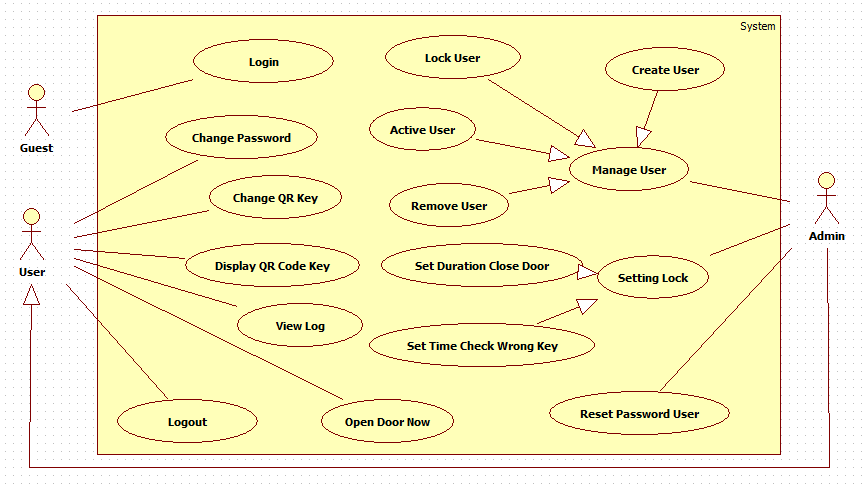
### A. Hardware Requirement

### B. Software Requirement

#### 1. User Interfaces

#### 2. Functional Requirements

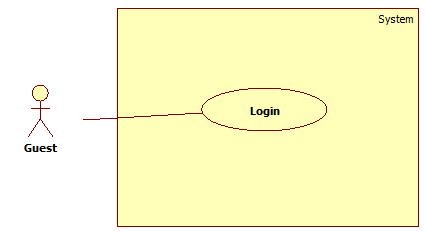
##### 2.1 Use Case Diagram



##### 2.2 Use Case Description

###### 2.2.1 “Login” Use Case

**Diagram:**



*Figure 9: "Login Use Case Diagram"*

**Specification:**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE No.1** | | | |
| **Use Case No** | AP02 | **Use Case version** | 1.0 |
| **Use Case Name** | Login | | |
| **Author** | HoangTV | | |
| **Date** | 7/02/2015 | **Priority** | High |
| **Actor:**   * Guest   **Summary:**   * This use case allows guest login to android application.   **Goal:**   * Guest can login to android application to get QR code key, change password, view log, change QR code key.   **Triggers:**   * Guest access to login layout and input username, password to login.   **Preconditions:**   * Username and password must be create by admin.   **Post Conditions:**   * Success: go to user page or admin page layout. * Fail: Show login layout and show error.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | User will input username and password in EditText and click button “Login”. | Show Layout login user input following information:  -“Username”: EditText.  -“Password”: EditText .  User click button “Login” to login user to application. | | **2** | User input following information:  -“Username”: EditText.  -“Password”: EditText .  User click button “Login” to login user to application. | Show layout of user or admin. |   **Exception:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** | Guest input wrong username or password. | Show Dialog and help user know what wrong ”Login Fail! Please input right username, password”. | | **2** | Guest don’t input username or password. | Show dialog “Please input your username and password”. |   **Relationships:**   * N/A   **Business Rules:**   * Username and password will to compare with database in server and accept guest login or not. * The role will be to accept some function of user or admin. | | | |

###### 2.2.2 “Change Password” Use Case

**Diagram:**



*Figure 10: "Change Password" Use case Diagram*

**Specification:**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE No.1** | | | |
| **Use Case No** | AP03 | **Use Case version** | 1.0 |
| **Use Case Name** | Change Password | | |
| **Author** | HoangTV | | |
| **Date** | 7/02/2015 | **Priority** | High |
| **Actor:**   * User, Admin   **Summary:**   * This use case allows user, admin can change password.   **Goal:**   * User, admin can change password of user, admin.   **Triggers:**   * User, admin access to login layout and input username, password to login.   **Preconditions:**   * User, admin must login in application.   **Post Conditions:**   * Success: go to layout of user, admin and change password. * Fail : Show dialog and show error toast.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | User click to button “Change Password” in layout. | Show layout Change password | | **2** | User click button “Change Password” to change password. | Show user or admin layout. |   **Exception:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** | User don’t input information and click “Change Password” button. | Show Dialog “Please input enough information!!!”. | | **2** | User input wrong old password. | Show Toast “FAIL\_USER\_NOT\_EXIST ”. | | **3** | User input wrong confirm new password. | Show Dialog “FAIL\_INCORRECT\_PASSWORD ”. |   **Relationships:**   * N.A   **Business Rules:**   * Username, password will to compare with database in server when user click “Change Password” button. | | | |

###### 2.2.3 “Display QR code” Use Case

**Diagram:**



**Specification:**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE No.1** | | | |
| **Use Case No** | AP03 | **Use Case version** | 1.0 |
| **Use Case Name** | Display QR Code | | |
| **Author** | HoangTV | | |
| **Date** | 7/02/2015 | **Priority** | High |
| **Actor:**   * User, Admin.   **Summary:**   * This use case allows user, admin can display QR code key.   **Goal:**   * User, admin can display QR code key to unlock the door.   **Triggers:**   * User, admin access to login layout and login.   **Preconditions:**   * User, admin must login in application one time and display QR code key.   **Post Conditions:**   * Success: Go to layout show QR code key. * Fail : Show dialog and show “Please try again!”.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | User click to label “Get QR Code Key” in layout. | Show layout to show QR code .  User click button “Ok” to return user layout. |   **Exception:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | |  |  |  |   **Relationships:**   * N.A   **Business Rules:**   * Username, password will to compare with database in server and accept user, admin login and display QR code or not. * When user change QR code when user click button “Display QR Code” application will synchronize with database to show new QR code. | | | |

###### 2.2.4 “View Log” Use Case

**Diagram:**



**Specification:**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE No.1** | | | |
| **Use Case No** | AP03 | **Use Case version** | 1.0 |
| **Use Case Name** | View Log | | |
| **Author** | HoangTV | | |
| **Date** | 7/02/2015 | **Priority** | High |
| **Actor:**   * User, Admin.   **Summary:**   * This use case allows user, admin View Log of system.   **Goal:**   * User, admin can view log about who open door with the date user want.   **Triggers:**   * User, admin access to login layout and login.   **Preconditions:**   * User, admin must login in application.   **Post Conditions:**   * Success: Go to layout View Log and show log. * Fail : Show dialog and show “Please try again!”.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | User click to button “View Log” in layout. | Show layout to view log. | | **2** | User click EditText date show DatePicker or click button back or button forward . | Show Log. |   **Exception:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | |  |  |  |   **Relationships:**   * N.A   **Business Rules:**   * Username, password will to compare with database in server and accept user login and view log or not. | | | |

###### 2.2.5 “Change QR code Key” Use Case

**Diagram:**



**Specification:**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE No.1** | | | |
| **Use Case No** | AP03 | **Use Case version** | 1.0 |
| **Use Case Name** | Change QR Key | | |
| **Author** | HoangTV | | |
| **Date** | 7/02/2015 | **Priority** | High |
| **Actor:**   * User, Admin.   **Summary:**   * This use case allows user, admin change QR key.   **Goal:**   * User, admin can change QR key to open door.   **Triggers:**   * User, admin access to login layout and login.   **Preconditions:**   * User, admin must login in application.   **Post Conditions:**   * Success: Go to layout user, admin and change QR key. * Fail : Show dialog and show “Please try again!”.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | User, admin click to button “Change QR Key” in user or admin layout. | Show change QR key layout. | | **2** | User, admin input information and click “Change Key” button . | Go to layout of user or admin. |   **Exception:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** | User, admin don’t input information and click “Change Key” button. | Show Dialog “Please input enough information!!!”. | | **2** | User, admin input wrong password | Show Toast “FAIL\_INCORRECT\_PASSWORD” |   **Relationships:**   * N.A   **Business Rules:**   * Username, password will to compare with database in server and accept user login and view log or not. | | | |

###### 2.2.6 “Open Door” Use Case

**Diagram:**



**Specification:**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE No.1** | | | |
| **Use Case No** | AP03 | **Use Case version** | 1.0 |
| **Use Case Name** | Open Door Now | | |
| **Author** | HoangTV | | |
| **Date** | 7/02/2015 | **Priority** | High |
| **Actor:**   * User, Admin.   **Summary:**   * This use case allows user, admin open door now.   **Goal:**   * User, admin can open door not must user QR code.   **Triggers:**   * User, admin access to login layout and login.   **Preconditions:**   * User, admin must login in application.   **Post Conditions:**   * Success: Go to layout user, admin and open door. * Fail : Show dialog and show error.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | User, admin click to button “Open Door Now” in user or admin layout. | Show open door now layout. | | **2** | User, admin input information and click “Open Door” button . | Go to layout of user or admin. |   **Exception:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** | User, admin don’t input information and click “Change Key” button. | Show Dialog “Please input enough information!!!”. | | **2** | User, admin input wrong key | Show Toast “FAIL\_INVALID\_KEY” |   **Relationships:**   * N.A   **Business Rules:**   * Username, password will to compare with database in server and accept user, admin login and open door. | | | |

###### 2.2.7 “Logout” Use Case

**Diagram:**



**Specification:**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE No.1** | | | |
| **Use Case No** | AP02 | **Use Case version** | 1.0 |
| **Use Case Name** | Logout | | |
| **Author** | HoangTV | | |
| **Date** | 7/02/2015 | **Priority** | High |
| **Actor:**   * User, Admin   **Summary:**   * This use case allows user, admin logout to android application.   **Goal:**   * User, admin can logout to android application.   **Triggers:**   * User, admin click logout in android application.   **Preconditions:**   * User, admin must login in android application.   **Post Conditions:**   * Success: go to login layout.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | User click “Logout” in android application. | Show layout of user or admin. |   **Exception:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | |  |  |  |   **Relationships:**   * N/A   **Business Rules:**   * Delete username in database. | | | |

###### 2.2.8 “Reset password” Use Case

**Diagram:**

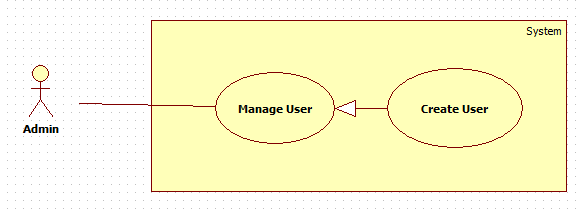


**Specification:**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE No.1** | | | |
| **Use Case No** | AP03 | **Use Case version** | 1.0 |
| **Use Case Name** | Reset User Password | | |
| **Author** | HoangTV | | |
| **Date** | 7/02/2015 | **Priority** | High |
| **Actor:**   * Admin.   **Summary:**   * This use case allows admin can reset user password.   **Goal:**   * Admin reset password user when user forgot password to login in android application.   **Triggers:**   * Admin click reset password user button.   **Preconditions:**   * Admin must login in application.   **Post Conditions:**   * Success: Go to layout admin and reset password user. * Fail : Show dialog and show error.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | Admin click to button “Reset Password User” in admin layout. | Show reset password user layout. | | **2** | Admin input information and click “Reset” button . | Go to layout of admin. |   **Exception:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** | Admin don’t input information and click “Reset” button. | Show Dialog “Please input enough information!!!”. | | **2** | Admin input wrong username of user | Show Dialog “FAIL\_USER\_NOT\_EXIST”. |   **Relationships:**   * N.A   **Business Rules:**   * Username and new password will save in database in server. | | | |

###### 2.2.9 “Create User” Use Case

**Diagram:**

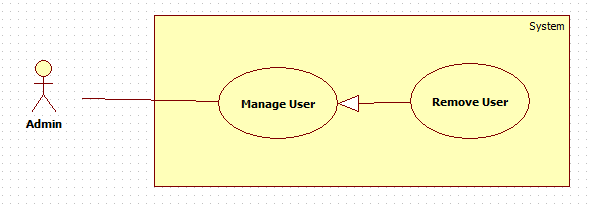


**Specification:**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE No.1** | | | |
| **Use Case No** | AP03 | **Use Case version** | 1.0 |
| **Use Case Name** | Create User | | |
| **Author** | HoangTV | | |
| **Date** | 7/02/2015 | **Priority** | High |
| **Actor:**   * Admin.   **Summary:**   * This use case allows admin can create new user.   **Goal:**   * Admin can create new user to login in android application.   **Triggers:**   * Admin access to login layout and login.   **Preconditions:**   * Admin must login in application.   **Post Conditions:**   * Success: Go to layout admin and create new user. * Fail : Show dialog and show error.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | Admin click to button “Create User” in admin layout. | Show create user layout. | | **2** | Admin input information and click “Create New User” button . | Go to layout of admin. |   **Exception:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** | Admin don’t input information and click “Create New User” button. | Show Dialog “Please input enough information!!!”. | | **2** | Admin input special character in FullName . | Show Dialog “InvaLid!!! Don’t input special character in name”. | | **3** | Admin input wrong confirm password. | Show Dialog “Please input right confirm password!!!”. | | **4** | Admin input with username exist. | Show Toasr “FAIL\_USER\_EXIST”. |   **Relationships:**   * N.A   **Business Rules:**   * Username andpassword will save in database in server. | | | |

###### 2.2.10 “Remove User” Use Case

**Diagram:**

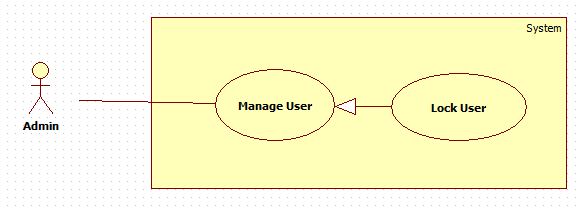


**Specification:**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE No.1** | | | |
| **Use Case No** | AP03 | **Use Case version** | 1.0 |
| **Use Case Name** | Remove User | | |
| **Author** | ThucVD | | |
| **Date** | 7/02/2015 | **Priority** | High |
| **Actor:**   * Admin.   **Summary:**   * This use case allows admin can create new user.   **Goal:**   * Admin can create new user to login in android application.   **Triggers:**   * Admin access to login layout and login.   **Preconditions:**   * Admin must login in application.   **Post Conditions:**   * Success: Go to layout admin and create new user. * Fail : Show dialog and show error.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | Admin click to button “Create User” in admin layout. | Show create user layout. | | **2** | Admin input information and click “Create New User” button . | Go to layout of admin. |   **Exception:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** | Admin don’t input information and click “Create New User” button. | Show Dialog “Please input enough information!!!”. | | **2** | Admin input special character in Full Name . | Show Dialog “Invalid!!! Don’t input special character in name”. | | **3** | Admin input wrong confirm password. | Show Dialog “Please input right confirm password!!!”. | | **4** | Admin input with username exist. | Show Toast “FAIL\_USER\_EXIST”. |   **Relationships:**   * N.A   **Business Rules:**   * Username andpassword will save in database in server. | | | |

###### 2.2.11 “Lock User” Use case

**Diagram:**

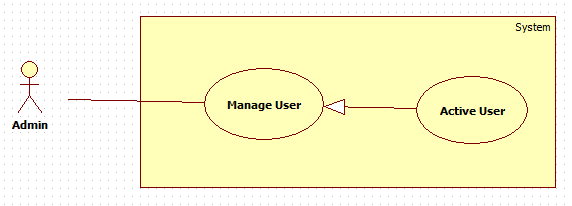


**Specification:**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE No.1** | | | |
| **Use Case No** | AP03 | **Use Case version** | 1.0 |
| **Use Case Name** | LockUser | | |
| **Author** | HoangTV | | |
| **Date** | 7/02/2015 | **Priority** | High |
| **Actor:**   * Admin.   **Summary:**   * This use case allows admin can lock user and this user can’t login to android application.   **Goal:**   * Admin can lock user.   **Triggers:**   * Admin access to login layout and login.   **Preconditions:**   * Admin must login in application.   **Post Conditions:**   * Success: Go to layout admin and lock user. * Fail : Show dialog and show error.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | Admin click to button “Lock User” in admin layout. | Show Lock User layout. | | **2** | Admin click to button “Lock” | Go to LockUserLayout and delete user is lock in layout. |   **Exception:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | |  |  |  |   **Relationships:**   * N.A   **Business Rules:**   * User is lock will be delete in database in server. | | | |

###### 2.2.12 “Active User” Use case

**Diagram:**

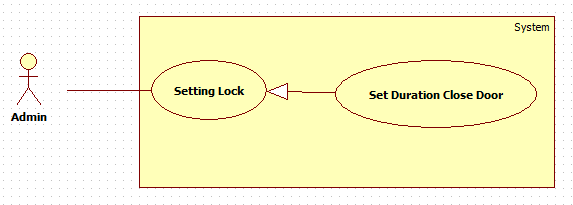


**Specification:**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE No.1** | | | |
| **Use Case No** | AP03 | **Use Case version** | 1.0 |
| **Use Case Name** | Activate User | | |
| **Author** | HoangTV | | |
| **Date** | 7/02/2015 | **Priority** | High |
| **Actor:**   * Admin.   **Summary:**   * This use case allows admin can lock user and this user can’t login to android application.   **Goal:**   * Admin can lock user.   **Triggers:**   * Admin access to login layout and login.   **Preconditions:**   * Admin must login in application.   **Post Conditions:**   * Success: Go to layout admin and lock user. * Fail : Show dialog and show error.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | Admin click to button “Lock User” in admin layout. | Show Lock User layout. | | **2** | Admin click to button “Lock” | Go to LockUserLayout and delete user is lock in layout. |   **Exception:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | |  |  |  |   **Relationships:**   * N.A   **Business Rules:**   * User is lock will be delete in database in server. | | | |

###### 2.2.13 “Set duration close door” Use Case

**Diagram:**

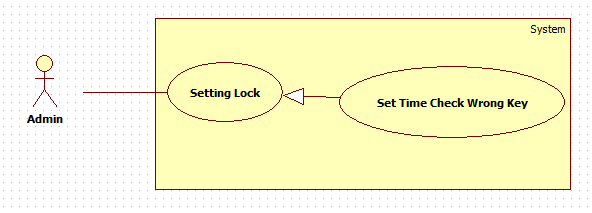


**Specification:**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE No.1** | | | |
| **Use Case No** | AP03 | **Use Case version** | 1.0 |
| **Use Case Name** | SettingLock | | |
| **Author** | HoangTV | | |
| **Date** | 7/02/2015 | **Priority** | High |
| **Actor:**   * Admin.   **Summary:**   * This use case allows admin can setting lock in system.   **Goal:**   * Admin can setting time close door and time give wrong key.   **Triggers:**   * Admin access to login layout and login.   **Preconditions:**   * Admin must login in application.   **Post Conditions:**   * Success: Go to layout admin. * Fail : Show dialog and show error.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | Admin click to button “SettingLock” in admin layout. | Show Setting Lock layout. | | **2** | Admin input information and click to button “Setting” | Go to admin page and success setting lock. |   **Exception:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** | When admin don’t input in formation and click “Setting” button. | Show error dialog “Please input enough information!!!”. | | **2** | When Admin input wrong format. | Show error dialog format. |   **Relationships:**   * N.A   **Business Rules:**   * All information of lock will save in database in server. | | | |

###### 2.2.14 “Set time check wrong key ” Use case

**Diagram:**



**Specification:**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE No.1** | | | |
| **Use Case No** | AP03 | **Use Case version** | 1.0 |
| **Use Case Name** | SettingLock | | |
| **Author** | HoangTV | | |
| **Date** | 7/02/2015 | **Priority** | High |
| **Actor:**   * Admin.   **Summary:**   * This use case allows admin can setting lock in system.   **Goal:**   * Admin can setting time close door and time give wrong key.   **Triggers:**   * Admin access to login layout and login.   **Preconditions:**   * Admin must login in application.   **Post Conditions:**   * Success: Go to layout admin. * Fail : Show dialog and show error.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | Admin click to button “SettingLock” in admin layout. | Show Setting Lock layout. | | **2** | Admin input information and click to button “Setting” | Go to admin page and success setting lock. |   **Exception:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** | When admin don’t input in formation and click “Setting” button. | Show error dialog “Please input enough information!!!”. | | **2** | When Admin input wrong format. | Show error dialog format. |   **Relationships:**   * N.A   **Business Rules:**   * All information of lock will save in database in server. | | | |

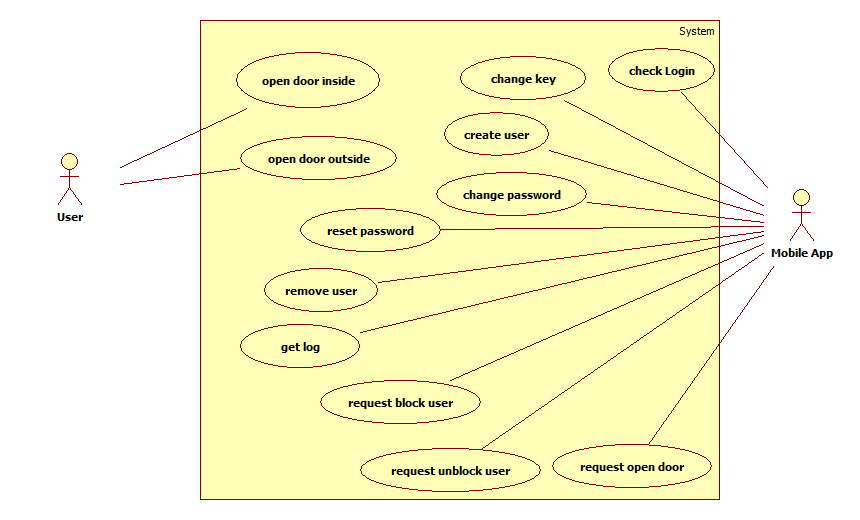
#### 3. Non-Functional Requirement

### C. Controller Requirement

#### 1. User Interfaces

#### 2. Functional Requirement

##### 2.1 Use Case Diagram

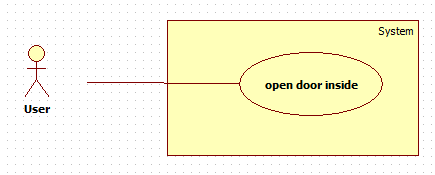


*Figure 11: Controller Use Case Diagram*

##### 2.2 Use Case Description

###### 2.2.1 “Open door inside” Use case

**Diagram:**

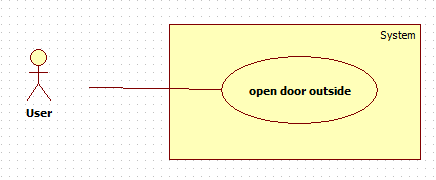


**Specification:**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE No.1** | | | |
| **Use Case No** | AP01 | **Use Case version** | 1.0 |
| **Use Case Name** | Open door inside | | |
| **Author** | ThucVD | | |
| **Date** | 7/02/2015 | **Priority** | High |
| **Actor:**   * User (Guest, member)   **Summary:**   * This use case allows user open door inside room or house.   **Goal:**   * User can open door.   **Triggers:**   * User presses the open-door button. * User hold the door-handle.   **Preconditions:**   * User must be inside of room or house.   **Post Conditions:**   * Success: The door lock is unlocked, then user can go out. * Fail: The door lock is still locked.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | User goes to front of the door.  [Alternative 1]  [Alternative 2] | Unlock the lock, door will be opened and write log  [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | User presses the openning button. | The door lock is unlocked.  [Exception1 ] | | **2** | User hold the door-handle | Sensor detect human in front of the door, and unlock the door lock  [Exception 2] |   **Exception:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** |  | The lock is spoiled | | **2** |  | Sensor is spoiled or absence of voltage |   **Relationships:**   * N/A   **Business Rules:**   * N/A | | | |

###### 2.2.2 “Open door outside”

**Diagram:**

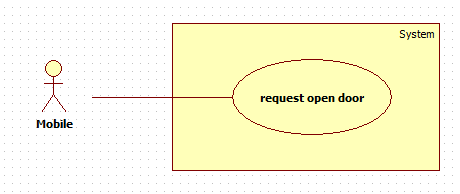


**Specification:**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE No.1** | | | |
| **Use Case No** | AP01 | **Use Case version** | 1.0 |
| **Use Case Name** | Open door outside | | |
| **Author** | ThucVD | | |
| **Date** | 7/02/2015 | **Priority** | High |
| **Actor:**   * User (member)   **Summary:**   * This usecase allows user open door outside house.   **Goal:**   * User can open door.   **Triggers:**   * User show barcode in front of camera.   **Preconditions:**   * User must be outside the room * User have correct barcode.   **Post Conditions:**   * Success: The door lock is unlocked, then user can come into the house. * Fail: The door lock is still locked.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | Use show barcode in front of senor.  [Alternative 1] | - Scan barcode  - Check barcode that is correct or not.  - If it is correct password, the lock will be unlock and the system will write log.  [Alternative 2]  [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | User use mobile app to open the door. |  | | **2** | User show incorrect password. | **Show message “Xin thửlạinếu 3 lầnsaihệthốngsẽkhóa”.** |   **Exception:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** |  | Connect to server fail. |   **Relationships:**   * N/A   **Business Rules:**   * N/A | | | |

###### 2.2.3 “Request open door” Use case

**Diagram:**

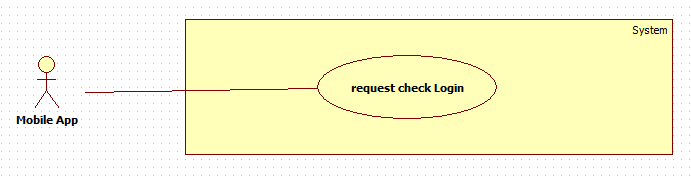


**Specification:**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE No.1** | | | |
| **Use Case No** | AP01 | **Use Case version** | 1.0 |
| **Use Case Name** | Request open door | | |
| **Author** | ThucVD | | |
| **Date** | 7/02/2015 | **Priority** | High |
| **Actor:**   * Mobile app   **Summary:**   * This use case allows mobile sent request to system to open the door.   **Goal:**   * User can open door by mobile app.   **Triggers:**   * Mobile application sent request to the controller.   **Preconditions:**   * The system is activated.   **Post Conditions:**   * Success: The door lock is unlocked * Fail: The door lock is still locked.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | Send a request to server  [Exception 1] | -Receive a request  -Check key from the request  -If the key is valid, the system will send a message to unlock the lock.  [Alternative 1]  [Exception 2] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | Send a invalid key. | Server will return fail and still unlock the door. |   **Exception:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** | Connect to server fail. |  | | **2** |  | Request time out. |   **Relationships:**   * N/A   **Business Rules:**   * N/A | | | |

###### 2.2.4 “Request check login” Use case

**Diagram:**

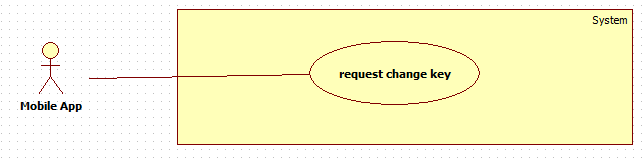


**Specification:**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE No.1** | | | |
| **Use Case No** | AP01 | **Use Case version** | 1.0 |
| **Use Case Name** | Request check login | | |
| **Author** | ThucVD | | |
| **Date** | 7/02/2015 | **Priority** | High |
| **Actor:**   * Mobile app   **Summary:**   * This use case allows mobile sent request to login into the system.   **Goal:**   * Mobile app get a permission message to login into system.   **Triggers:**   * Button login was pressed in mobile app.   **Preconditions:**   * The system is activated. * Mobile app is connecting in the same LAN network with system   **Post Conditions:**   * Success: The system return json of the user login. * Fail: The system return fail message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | Send a check login request to server  [Exception 1] | -Receive a request  -Check username and password.  -If the key is valid, the system will send json to allow user access into the system.  [Alternative 1]  [Exception 2] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | Send a invalid key. | Server will return fail message json. |   **Exception:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** | Connect to server fail. |  | | **2** |  | Request time out. |   **Relationships:**   * N/A   **Business Rules:**   * N/A | | | |

###### 2.2.5 “Request change key” Use case

**Diagram:**

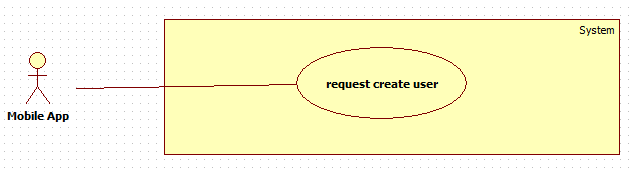


**Specification:**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE No.1** | | | |
| **Use Case No** | AP01 | **Use Case version** | 1.0 |
| **Use Case Name** | Request change key | | |
| **Author** | ThucVD | | |
| **Date** | 7/02/2015 | **Priority** | High |
| **Actor:**   * Mobile app   **Summary:**   * This use case allows mobile sent request to change key   **Goal:**   * User can change key by mobile app.   **Triggers:**   * Input new key. * Button change key was pressed in mobile app.   **Preconditions:**   * The system is activated. * Mobile app is connecting in the same LAN network with system and is into system.   **Post Conditions:**   * Success: The system change current key to new key. * Fail: The current key is not changed.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | - Send a change request to server  [Exception 1] | -Receive a request  -Check authentication  -Update new key into server.  -The system will send json to allow user access into the system.  [Exception 2] |   **Alternative Scenario:**   * N/A   **Exception:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** | Connect to server fail. |  | | **2** |  | Request time out. |   **Relationships:**   * N/A   **Business Rules:**   * N/A | | | |

###### 2.2.6 “Request create user” Use case

**Diagram:**

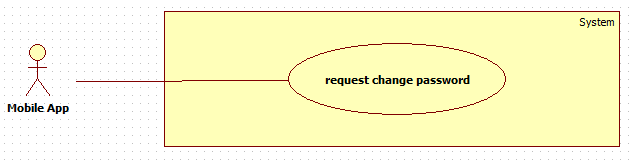


**Specification:**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE No.1** | | | |
| **Use Case No** | AP01 | **Use Case version** | 1.0 |
| **Use Case Name** | Request create user | | |
| **Author** | ThucVD | | |
| **Date** | 7/02/2015 | **Priority** | High |
| **Actor:**   * Mobile app   **Summary:**   * This use case allows mobile sent request to create account   **Goal:**   * User can use mobile app to create a account   **Triggers:**   * Button login was pressed in mobile app.   **Preconditions:**   * The system is activated. * Mobile app is connecting into system   **Post Conditions:**   * Success: The system return success json * Fail: The system return fail message json.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | Send a create account request to server  [Exception 1] | -Receive a request  -Check authentication.  -Create a new account in database.  -the system will send json to allow user access into the system.  [Exception 2] |   **Alternative Scenario:**   * N/A   **Exception:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** | Connect to server fail. |  | | **2** |  | Request time out. |   **Relationships:**   * N/A   **Business Rules:**   * N/A | | | |

###### 2.2.7 “Request change password” Use Case

**Diagram:**

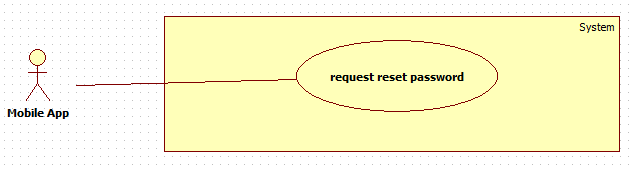


**Specification:**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE No.1** | | | |
| **Use Case No** | AP01 | **Use Case version** | 1.0 |
| **Use Case Name** | Request change password | | |
| **Author** | ThucVD | | |
| **Date** | 7/02/2015 | **Priority** | High |
| **Actor:**   * Mobile app   **Summary:**   * This use case allows mobile sent request to change password of this account.   **Goal:**   * User can change password themselves.   **Triggers:**   * New password and old password is filled. * Button change password was pressed in mobile app.   **Preconditions:**   * The system is activated. * Mobile app is connecting in the same LAN network with system and is into system.   **Post Conditions:**   * Success: The system change current key to new key. * Fail: The current key is not changed.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | - Send a change password to server  [Exception 1] | -Receive a request  -Check authentication  -Update new password into server.  -The system will send success json  [Exception 2] |   **Alternative Scenario:**   * N/A   **Exception:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** | Connect to server fail. |  | | **2** |  | Request time out. |   **Relationships:**   * N/A   **Business Rules:**   * N/A | | | |

###### 2.2.8 “Request reset password” Use Case

**Diagram:**

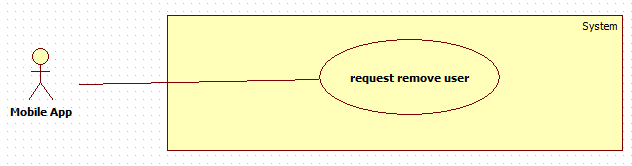


**Specification:**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE No.1** | | | |
| **Use Case No** | AP01 | **Use Case version** | 1.0 |
| **Use Case Name** | Request reset password | | |
| **Author** | ThucVD | | |
| **Date** | 7/02/2015 | **Priority** | High |
| **Actor:**   * Mobile app   **Summary:**   * This use case allows mobile sent request to reset password of a certain account.   **Goal:**   * Admin can reset password for an certain account.   **Triggers:**   * Button reset password is pressed in mobile app.   **Preconditions:**   * The system is activated. * Mobile app is connecting in the same LAN network with system and is into system.   **Post Conditions:**   * Success: The system reset password for certain account an d send success json to mobile app. * Fail: The system send fail json.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | - Send a reset password to server  [Exception 1] | -Receive a request  -Check admin authentication  -Get default password of certain account and update the account.  -The system will send success json  [Exception 2] |   **Alternative Scenario:**   * N/A   **Exception:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** | Connect to server fail. |  | | **2** |  | Request time out. |   **Relationships:**   * N/A   **Business Rules:**   * N/A | | | |

###### 2.2.9 “Request remove user” Use Case

**Diagram:**

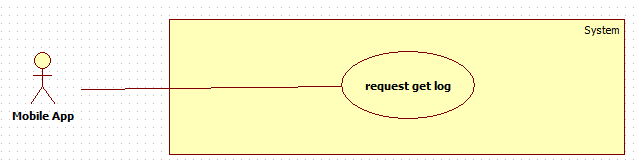


**Specification:**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE No.1** | | | |
| **Use Case No** | AP01 | **Use Case version** | 1.0 |
| **Use Case Name** | Request remove user | | |
| **Author** | ThucVD | | |
| **Date** | 7/02/2015 | **Priority** | High |
| **Actor:**   * Mobile app   **Summary:**   * This use case allows mobile sent request to delete an account.   **Goal:**   * Admin can delete an account in system by mobile app.   **Triggers:**   * Button delete account is pressed in mobile app.   **Preconditions:**   * The system is activated. * Mobile app is connecting to the same network with the system.   **Post Conditions:**   * Success: The system remove a account and send success json to mobile app. * Fail: The system send fail json.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | - Send a removing account message to server  [Exception 1] | -Receive a request  -Check admin authentication  -Find the account and delete in database.  -The system will send success json  [Exception 2] |   **Alternative Scenario:**   * N/A   **Exception:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** | Connect to server fail. |  | | **2** |  | Request time out. |   **Relationships:**   * N/A   **Business Rules:**   * N/A | | | |

###### 2.2.10 “Request get log” Use Case

**Diagram:**

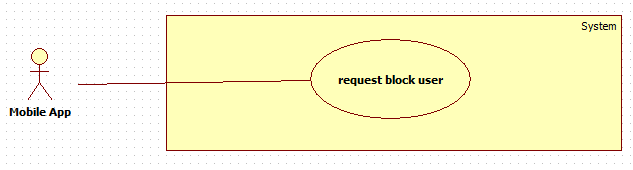


**Specification:**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE No.1** | | | |
| **Use Case No** | AP01 | **Use Case version** | 1.0 |
| **Use Case Name** | Request get log | | |
| **Author** | ThucVD | | |
| **Date** | 7/02/2015 | **Priority** | High |
| **Actor:**   * Mobile app   **Summary:**   * This use case allows mobile sent request to get information in log file of system.   **Goal:**   * User can read log file by mobile app.   **Triggers:**   * Getting log button is pressed in mobile app.   **Preconditions:**   * The system is activated. * Mobile app is connecting in the same LAN network with system and is into system.   **Post Conditions:**   * Success: The system return json which contains information of log file * Fail: The system send fail json.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | - Send a get log to server  [Exception 1] | -Receive a request  -Check authentication.  -The system will return json which contains information of log file  [Exception 2] |   **Alternative Scenario:**   * N/A   **Exception:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** | Connect to server fail. |  | | **2** |  | Request time out. |   **Relationships:**   * N/A   **Business Rules:**   * N/A | | | |

###### 2.2.11 “Request block user” Use Case

**Diagram:**

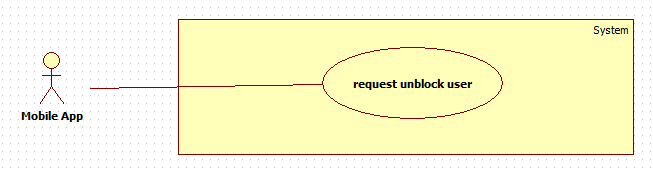


**Specification:**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE No.1** | | | |
| **Use Case No** | AP01 | **Use Case version** | 1.0 |
| **Use Case Name** | Request block user | | |
| **Author** | ThucVD | | |
| **Date** | 7/02/2015 | **Priority** | High |
| **Actor:**   * Mobile app   **Summary:**   * This use case allows mobile sent request to block an account.   **Goal:**   * Admin can block an account in system by mobile app.   **Triggers:**   * Button block account is pressed in mobile app.   **Preconditions:**   * The system is activated. * Mobile app is connecting in the same LAN network with system and is into system. * The account which will be block is activated account.   **Post Conditions:**   * Success: The system block a account and send success json to mobile app. * Fail: The system send fail json.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | - Send a blocking account message to server  [Exception 1] | -Receive a request  -Check admin authentication  -Find the account and update blocking status in database.  -The system will send success json  [Exception 2] |   **Alternative Scenario:**   * N/A   **Exception:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** | Connect to server fail. |  | | **2** |  | Request time out. |   **Relationships:**   * N/A   **Business Rules:**   * N/A | | | |

###### 2.2.12 “Request unblock user” Use Case

**Diagram:**



**Specification:**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE No.1** | | | |
| **Use Case No** | AP01 | **Use Case version** | 1.0 |
| **Use Case Name** | Request unblock user | | |
| **Author** | ThucVD | | |
| **Date** | 7/02/2015 | **Priority** | High |
| **Actor:**   * Mobile app   **Summary:**   * This use case allows mobile sent request to unblock an account.   **Goal:**   * Admin can unblock an account in system by mobile app.   **Triggers:**   * Button unblock account is pressed in mobile app.   **Preconditions:**   * The system is activated. * Mobile app is connecting in the same LAN network with system and is into system. * The account which will be unblock is blocking status.   **Post Conditions:**   * Success: The system unblock the account and send success json to mobile app. * Fail: The system send fail json.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | - Send a unblocking account message to server  [Exception 1] | -Receive a request  -Check admin authentication  -Find the account and update unblocking status in database.  -The system will send success json  [Exception 2] |   **Alternative Scenario:**   * N/A   **Exception:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** | Connect to server fail. |  | | **2** |  | Request time out. |   **Relationships:**   * N/A   **Business Rules:**   * N/A | | | |

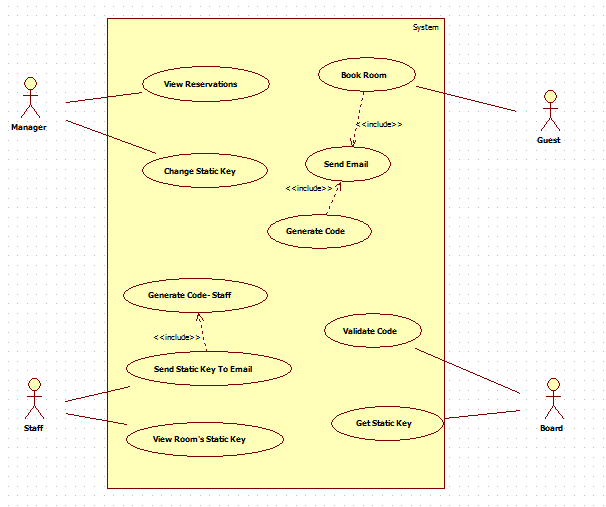
#### 3. Non-Functional Requirement

## III. System Requirement Specification – Hotel Use

### 1. User Interface

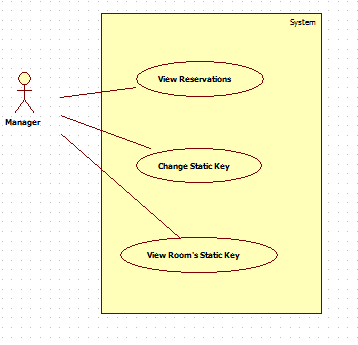
### 2. Functional Requirement

#### 2.1 Use case Diagram



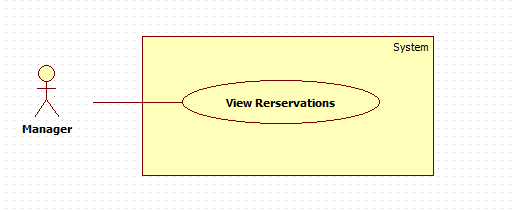
#### 2.2 Use case Description

##### 2.1.1 Manager



###### 2.1.1.1 View Reservations

**Diagram:**

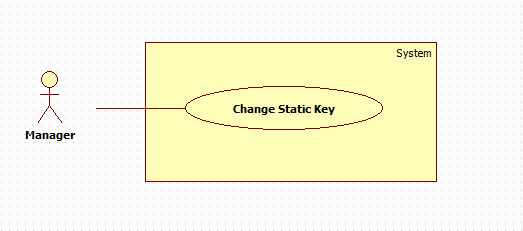
****

**Specification:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE** | | | | |
| **Use-case No.** | UC001 | **Use-case Version** | | 1.1 |
| **Use-case Name** | View Reservations | | | |
| **Author** | DoanLV | | | |
| **Date** | 04/13/2015 | **Priority** | Normal | |
| **Actor:**   * Manager   **Summary:**   * Get information about reservations   **Goal:**   * Reservations is shown   **Triggers:**   * Manager click [Info] button   **Preconditions:**   * User login with role Manager. * There is at least one room.   **PostConditions:**   * Success: Reservations are show on table on modal. * Fail: Nothing is shown.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | |  | Click on [Info] button | Receive request  Query to get reservations from database  Return data  Show modal with information |   **Alternative Scenario:**   * N/A   **Exceptions:**   * N/A   **Relationships:**   * N/A | | | | |

###### 2.1.1.2 Change Static Key

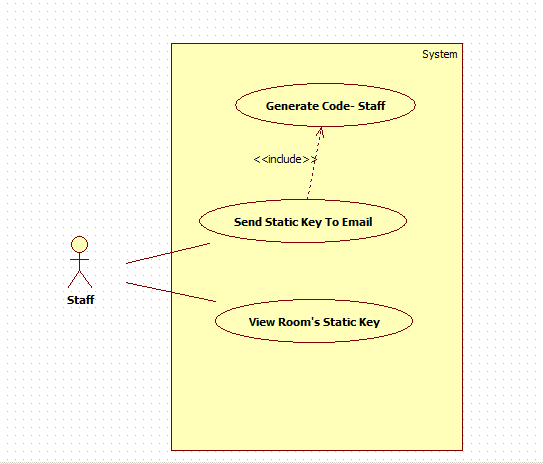
**Diagram:**

****

**Specification:**

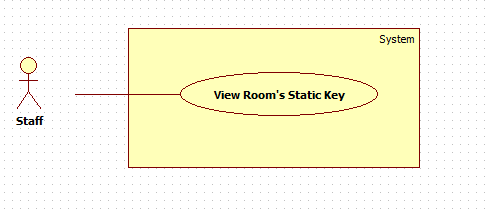
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE-CASE** | | | | |
| **Use-case No.** | UC002 | **Use-case Version** | | 1.2 |
| **Use-case Name** | Change Static Key | | | |
| **Author** | DoanLV | | | |
| **Date** | 04/13/2015 | **Priority** | Normal | |
| **Actor:**   * Manager.   **Summary:**   * Change room’s static key.   **Goal:**   * Room’s static key is changed.   **Triggers:**   * Manager click [Change Key] button   **Preconditions:**   * User login with role Manager. * There is at least one room. * New key field is fulfilled.   **PostConditions:**   * N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | |  | Click on [Change Key] button | Receive request to change key  Update database with new information  Show message |   **Alternative Scenario:**   * N/A   **Exceptions:**   * N/A   **Relationships:**   * N/A | | | | |

##### 2.1.2 Staff



###### 2.1.2.1 View Static Key

**Diagram:**

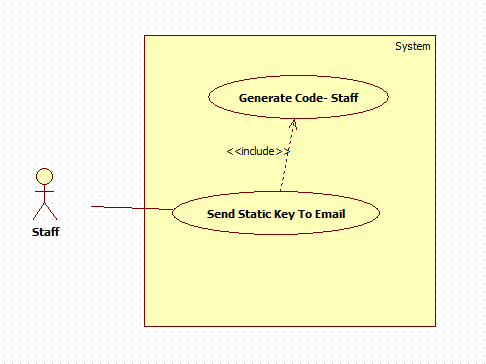
****

**Specification:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE-CASE** | | | | |
| **Use-case No.** | UC003 | **Use-case Version** | | 1.1 |
| **Use-case Name** | View Static Key | | | |
| **Author** | DoanLV | | | |
| **Date** | 04/13/2015 | **Priority** | Normal | |
| **Actor:**   * Staff.   **Summary:**   * View room’s static key.   **Goal:**   * Room’s static key is shown.   **Triggers:**   * Staff clicks [Key] button   **Preconditions:**   * User login with role Staff. * There is at least one room.   **PostConditions:**   * N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | |  | Click on [Key] button | Receive request  Query to get static key from database  Show static key on modal |   **Alternative Scenario:**   * N/A   **Exceptions:**   * N/A   **Relationships:**   * N/A | | | | |

###### 2.1.2.2 Send Static Key to Email

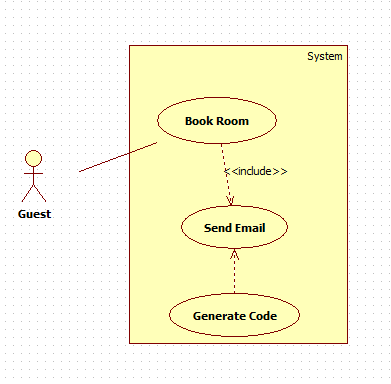
**Diagram:**

****

**Specification:**

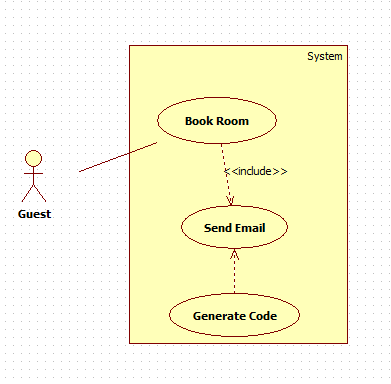
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE-CASE** | | | | |
| **Use-case No.** | UC004 | **Use-case Version** | | 1.1 |
| **Use-case Name** | Send Static Key To Email | | | |
| **Author** | DoanLV | | | |
| **Date** | 04/13/2015 | **Priority** | Normal | |
| **Actor:**   * Staff.   **Summary:**   * Send an email containing static key.   **Goal:**   * QR code is sent to staff’s email.   **Triggers:**   * Staff clicks [Send mail] button   **Preconditions:**   * User login with role Staff. * There is at least one room.   **PostConditions:**   * N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | |  | Click on [Send mail] button | Receive request  Generate code  Send static key to staff’s email |   **Alternative Scenario:**   * N/A   **Exceptions:**   * N/A   **Relationships:**   * N/A | | | | |

##### 2.1.3 Guest



###### 2.1.3.1 Book Room

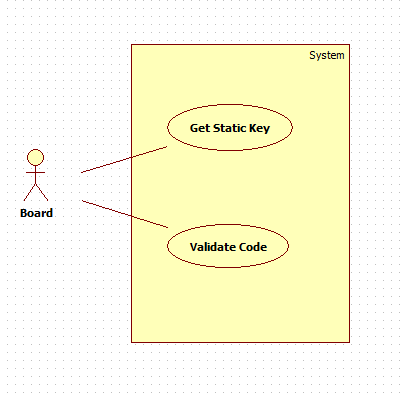
**Diagram:**



**Specification:**

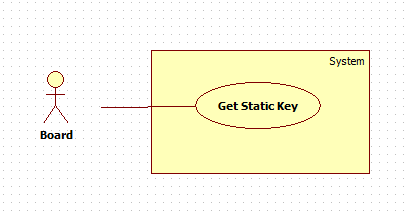
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE-CASE** | | | | |
| **Use-case No.** | UC004 | **Use-case Version** | | 1.1 |
| **Use-case Name** | Book Room | | | |
| **Author** | DoanLV | | | |
| **Date** | 04/13/2015 | **Priority** | Normal | |
| **Actor:**   * Guest.   **Summary:**   * Book room and receive code via email.   **Goal:**   * Email containing code is sent to guest.   **Triggers:**   * Guest clicks [Reserve] button   **Preconditions:**   * User login with role Guest. * There is at least one room. * Check in and check out date fields are not empty.   **PostConditions:**   * Success: reservation information is inserted to database. Email is sent to guest’s email. * Failure: Nothing happens   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | |  | Click on [Key] button  Choose check in and check out date by fulfilling fields and click Reserve button | Show modal  Receive request  Check input data  Generate key  Insert information to database  Send booking information to guest’s email |   **Alternative Scenario:**   * N/A   **Exceptions:**   * N/A   **Relationships:**   * N/A | | | | |

##### 2.1.4 Board



###### 2.1.4.1 Get static key

**Diagram:**

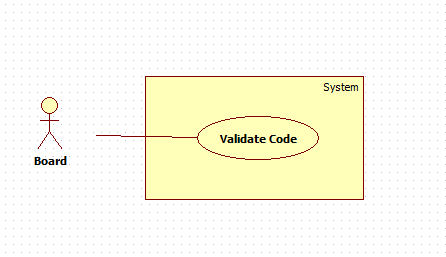
****

**Specification:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE-CASE** | | | | |
| **Use-case No.** | UC004 | **Use-case Version** | | 1.1 |
| **Use-case Name** | Get static key | | | |
| **Author** | DoanLV | | | |
| **Date** | 04/13/2015 | **Priority** | Normal | |
| **Actor:**   * Board.   **Summary:**   * Get static key on the schedule.   **Goal:**   * Get static key.   **Triggers:**   * Board call function to get key on the schedule.   **Preconditions:**   * Room is available on the web app.   **PostConditions:**   * Success: Key is sent back. * Failure: Empty data is sent back.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | |  | Trigger function on schedule to send request  Get data from API  Check whether data is empty or not  Write to file | Receive request  Check request data  Query database to get information  Return data |   **Alternative Scenario:**   * N/A   **Exceptions:**   * N/A   **Relationships:**   * N/A | | | | |

###### 2.1.4.2 Validate Key

**Diagram:**

****

**Specification:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE-CASE** | | | | |
| **Use-case No.** | UC004 | **Use-case Version** | | 1.1 |
| **Use-case Name** | Validate Key | | | |
| **Author** | DoanLV | | | |
| **Date** | 04/13/2015 | **Priority** | Normal | |
| **Actor:**   * Board.   **Summary:**   * Check whether input code is valid or not to decide following task.   **Goal:**   * Validate certain key.   **Triggers:**   * Board call API.   **Preconditions:**   * Board get data from reading QR code.   **PostConditions:**   * Success: Key is checked properly. Additionally, door is unlock when key is valid. * Failure: Nothing happens   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | |  | Get information from QR code  Send information to server (API) to check  Get return data  Do task base on result (unlock or keep lock). | Receive request  Validate request  Get data from request  Query database to get information  Check received data  Return status |   **Alternative Scenario:**   * N/A   **Exceptions:**   * N/A   **Relationships:**   * N/A | | | | |

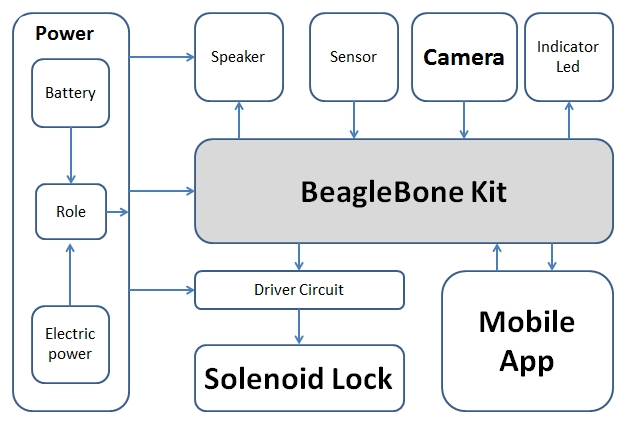
### 3. Non-Functional Requirement

# SYSTEM DESIGN DESCRIPTION (SDD)

## I.Design Overview

This document describes the design of this system. Information about hardware, device, software, component and architecture will be provided.

## II. System Architectural Design



*Figure 12: Overall system architecture*

This system contains those parts:

* + Main board BBB
  + Solenoid lock
  + Power supplier
  + A portable device (android smart phone)
  + Webcam
  + Indicator led
  + Infrared sensor

BBB connects to portable device via local LAN network, to webcam by USB port, to solenoid lock and indicator led by GPIO.

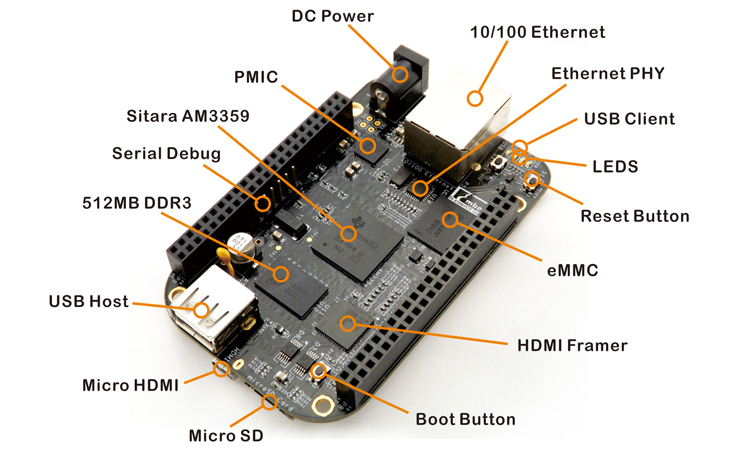
## III. Component Diagram

## IV. State Diagram

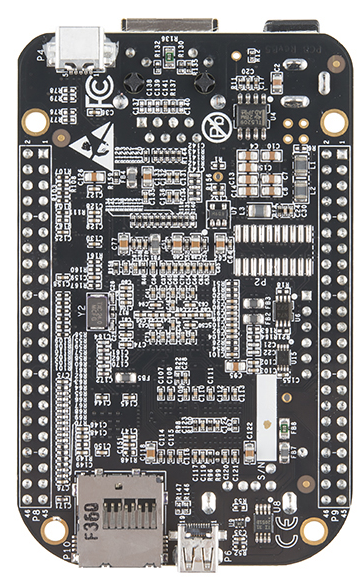
## V. Detail Description of Components

### 1. Hardware

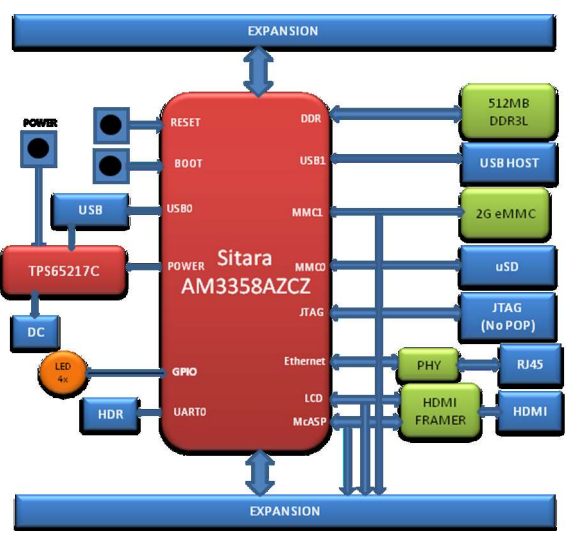
#### 1.1 Beaglebone black rev C



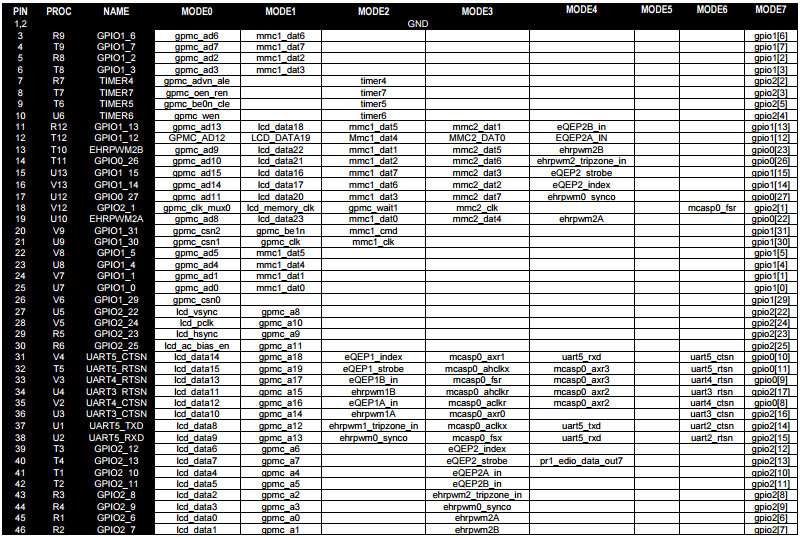
*Figure 13: Beaglebone black rev C top view*



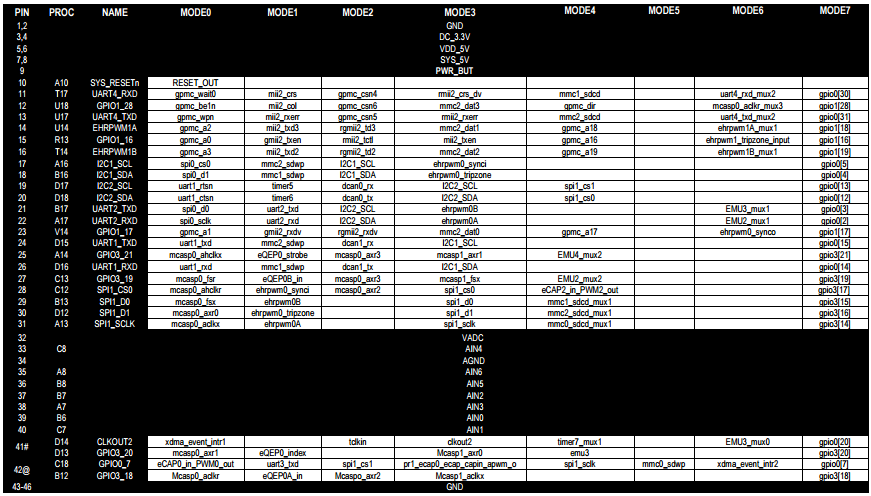
*Figure 14: beaglebone black rev C bottom view*



*Figure 15: beaglebone black rev C block diagram*



*Figure 16: Expansion header Pin 8 out*



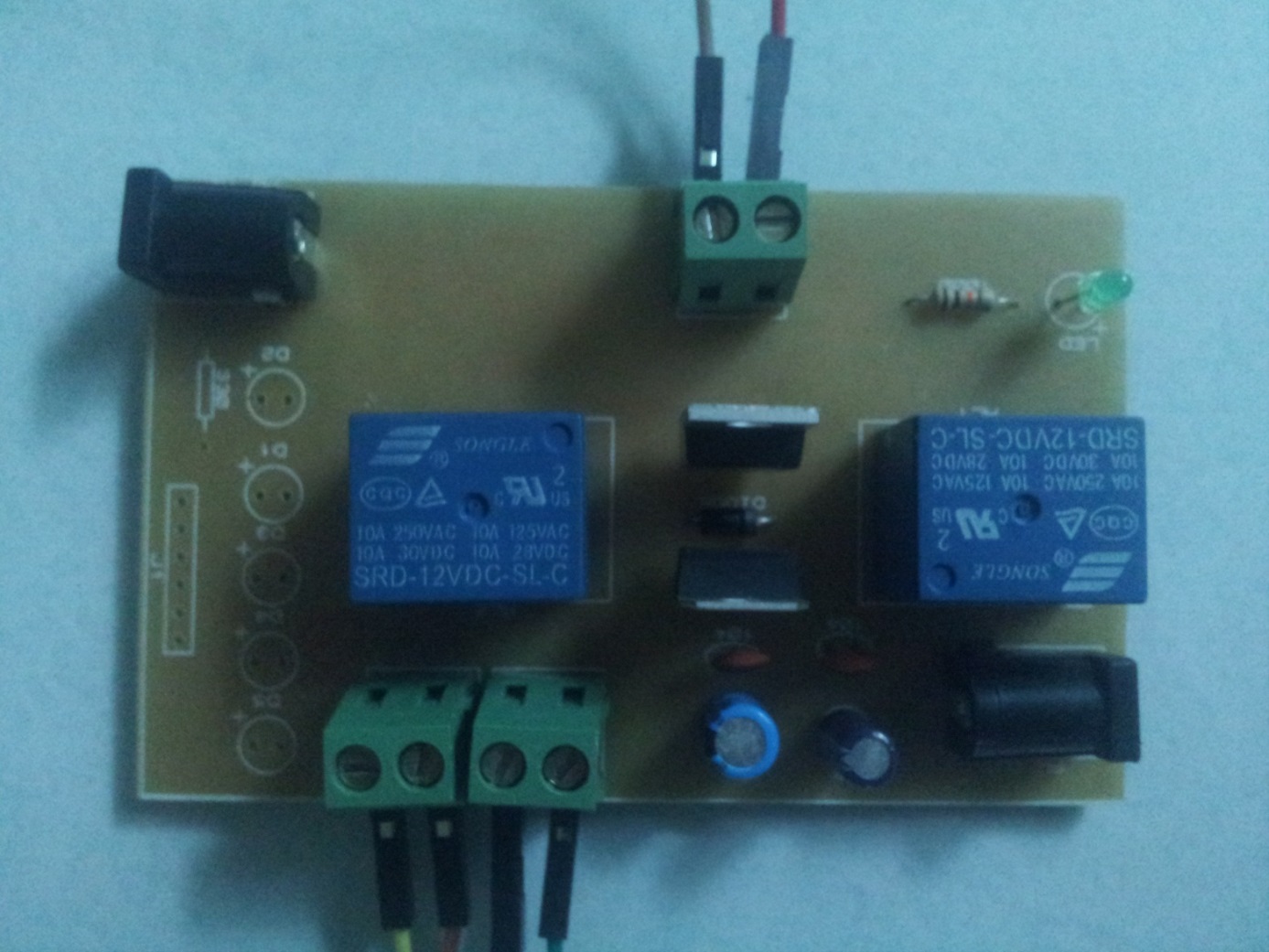
*Figure 17: Expansion header Pin 9 out*

#### 1.2 Power supplier circuit

##### Function:

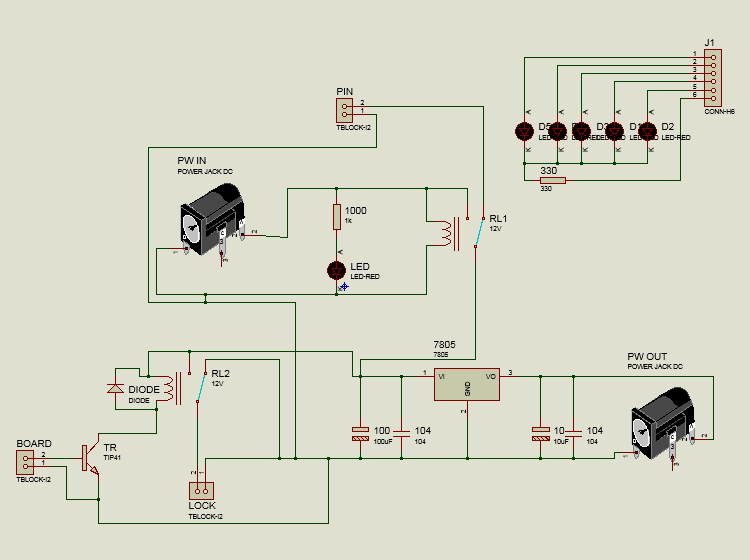
This circuit is used to switch from electric power to pin power when the electric power is cut off. Other way, this circuit also used to control solenoid lock and indicator led. It input 12V DC power, output 5V for beaglebone.

##### Interface:



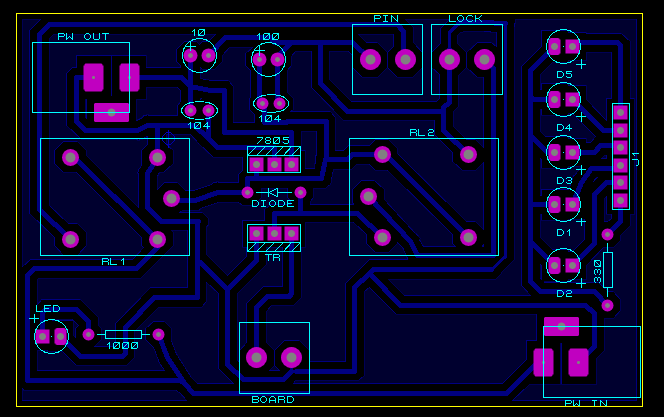
*Figure 18: power supplier circuit*

##### Schematic:



*Figure 19: Power supplierschematic diagram*

##### PCB layout:



*Figure 20: Power supplier and lock PCB*

##### Component parts

Use an Adapter DC 12V to supply power for the whole board



*Figure 21: Adapter DC 12V*

Use a Relay 12V to change state between power supplier adapter and battery for sure system still work while power outage.

* Contact Arrangement: 1 FORM C
* Coil Voltage:12V @ 0.36W
* Contact Rating: 10A 250VAC/30VDC,10A 125VAC/28VDC
* Recognized Safety: UL,CUL,TUV
* Outline L x W x H: 19.1 x 15.5 x 15.3 mm (Max)
* Weight: 8.45g

|  |  |
| --- | --- |
| Product Category | C:\Users\nam\Desktop\New folder\s-l1000.jpg[Fixed Voltage Regulators](http://www.tme.vn/Products.aspx?cateId=204)  *Figure 22: Replay 12V* |
| Output Type | Fixed |
| Regulator Topology | Positive Fixed |
| Voltage - Output | 5V |
| Voltage - Input | Up to 35V |
| Voltage - Dropout (Typical) | 2V @ 1A |
| Number of Regulators | 1 |
| Current - Output | 1.5A |
| Line Regulation | 100 mV |
| Load Regulation | 100 mV |
| PSRR / Ripple Rejection (Typ) | 62 dB |
| Input Voltage Min. | 7 V |
| Operating Temperature | 0°C ~ 125°C |
| Mounting Type | Through Hole |
| Package / Case | TO-220 |
| Packaging | Tube |
| Standard Package | 50 |

*Table 8: Replay 12V technical details*

An IC 7805 use 12v from Relay to supply power 5V for Board BeagleBone.

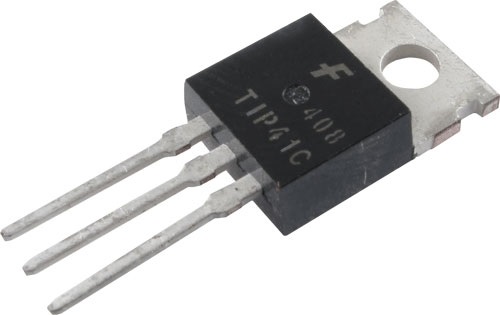
|  |  |
| --- | --- |
| Product Category | C:\Users\nam\Desktop\New folder\5pcs-LM7805-L7805-font-b-7805-b-font-Voltage-Regulator-IC-font-b-5V-b-font.jpg[Fixed Voltage Regulators](http://www.tme.vn/Products.aspx?cateId=204)  *Figure 23: IC 7805* |
| Output Type | Fixed |
| Regulator Topology | Positive Fixed |
| Voltage - Output | 5V |
| Voltage - Input | Up to 35V |
| Voltage - Dropout (Typical) | 2V @ 1A |
| Number of Regulators | 1 |
| Current - Output | 1.5A |
| Line Regulation | 100 mV |
| Load Regulation | 100 mV |
| PSRR / Ripple Rejection (Type) | 62 dB |
| Input Voltage Min. | 7 V |
| Operating Temperature | 0°C ~ 125°C |
| Mounting Type | Through Hole |
| Package / Case | TO-220 |
| Packaging | Tube |
| Standard Package | 50 |

*Table 9: IC 7805 technical details*

Driver Circuit:

Use Relay 12V to control Solenoids, with a **TIP41C** - NPN Epitaxial Silicon Transistor to active mass coil pin.

**Features**

* Medium Power Linear Switching Applications

*Figure 24: TIP41C*

* Complement to TIP42C

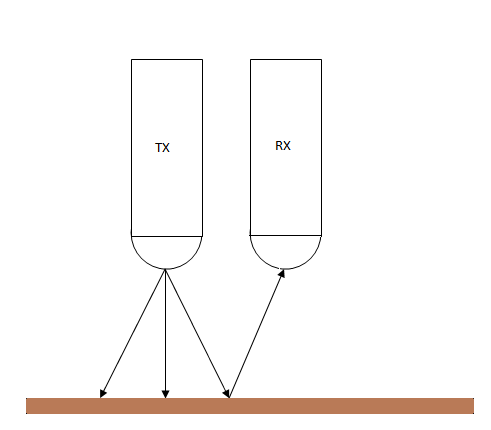
**Applications**

* GENERAL PURPOSE CIRCUITS
* AUDIO AMPLIFIER
* POWER LINEAR AND SWITCHING

#### 1.3 Infrared sensor

##### Function:

Infrared sensor is used to detect human.

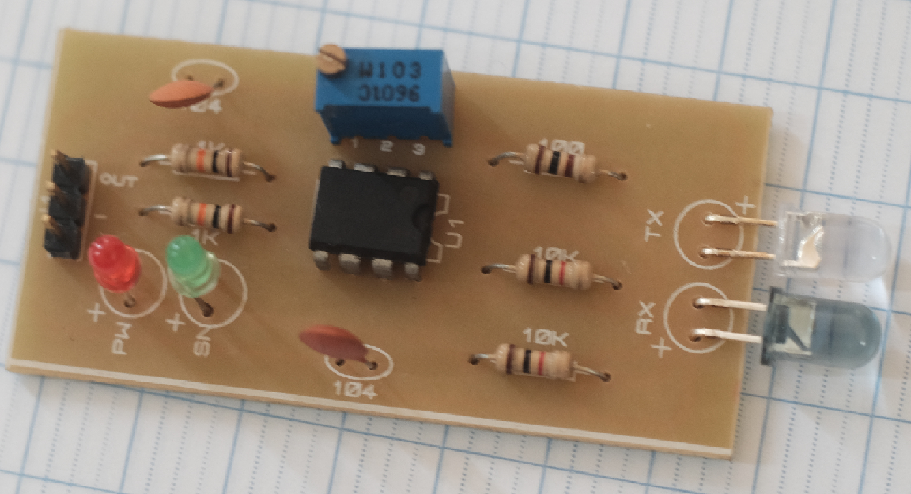


*Figure 25: TX and RX led*

IR LED emits infrared radiation. This radiation illuminates the surface in front of LED. Surface reflects the infrared light. Depending on reflectivity of the surface, amount of light reflected varies. This reflected light is made incident on reverse biased IR sensor. When photons are incident on reverse biased junction of this diode, electron-hole pairs are generated, which results in reverse leakage current. Amount of electron-hole pairs generated depends on intensity of incident IR radiation. More intense radiation results in more reverse leakage current. This current can be passed through a resistor so as to get proportional voltage. Thus as intensity of incident rays varies, voltage across resistor will vary accordingly.

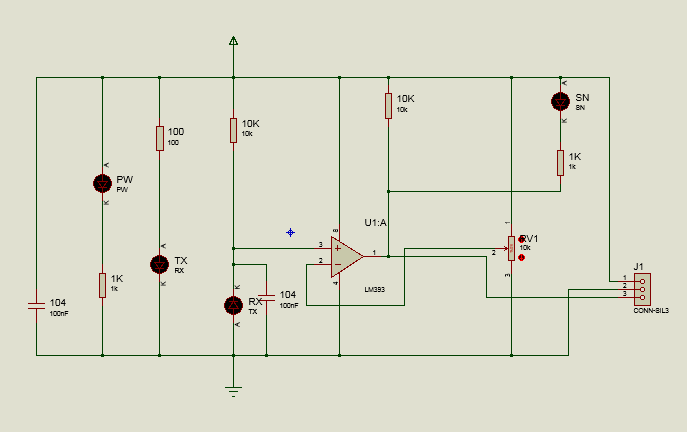
This voltage can then be given to OPAMP based comparator to make output 1, 0.

##### Interface:



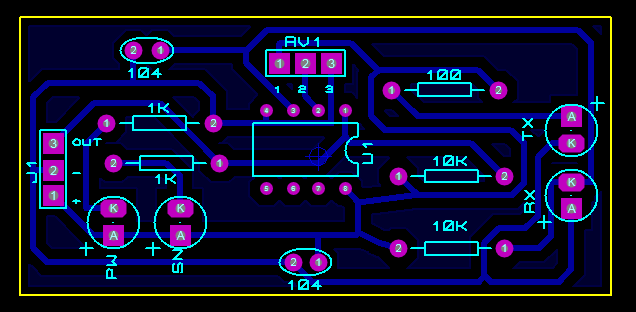
*Figure 26: Infrared sensor board*

##### Schematic:



*Figure 27: Infrared sensor schematic*

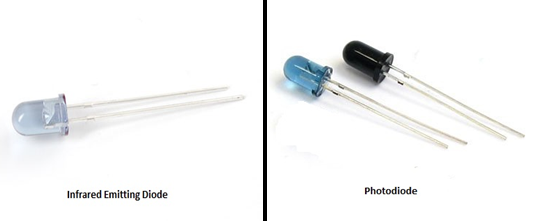
##### PCB layout:



*Figure 28: Infrared sensor PCB layout*

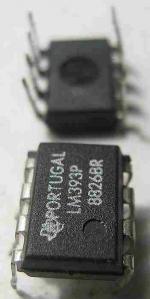
##### Component:

Infrared Sensor detector: use a couple Infrared Emitting Diode and Photodiode to make an infrared sensor module. Infrared Emitting Diode is a led used as a source of infrared rays. Photodiode is a diode, which is sensitive for infrared radiation.



*Figure 29: Photodiode and infrared emitting diode*

**LM393P- Low power dual voltage comparators**



*Figure 30: IC LM393P*

These devices consist of two independent voltage comparators that are designed to operate from a single power supply over a wide range of voltages. Operation from dual supplies also is possible as long as the difference between the two supplies is 2 V to 36 V, and VCC is at least 1.5 V more positive than the input common-mode voltage. Current drain is independent of the supply voltage. The outputs can be connected to other open-collector outputs to achieve wired-AND relationships. The LM193 is characterized for operation from –55°C to 125°C. The LM293 and LM293A are characterized for operation from –25°C to 85°C. The LM393 and LM393A are characterized for operation from 0°C to 70°C.  
  
**Technical details:**

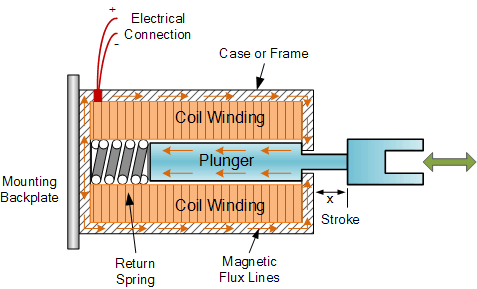
* Single Supply or Dual Supplies
* Wide Range of Supply Voltage...2 V to 36 V
  + Max Rating . . . 2 V to 36 V
  + Tested to 30 V… Non-V Devices
  + Tested to 32 V… V-Suffix Devices
* Low Supply-Current Drain Independent of Supply Voltage . . . 0.4 mA Typ Per Comparator
* Low Input Bias Current . . . 25 nATyp
* Low Input Offset Current ...3 nATyp (LM193)
* Low Input Offset Voltage ...2 mV Typ
* Common-Mode Input Voltage Range Includes Ground
* Differential Input Voltage Range Equal to Maximum-Rated Supply Voltage . . . ±36 V
* Low Output Saturation Voltage
* Output Compatible With TTL, MOS, and CMOS

#### 1.4 Solenoid lock



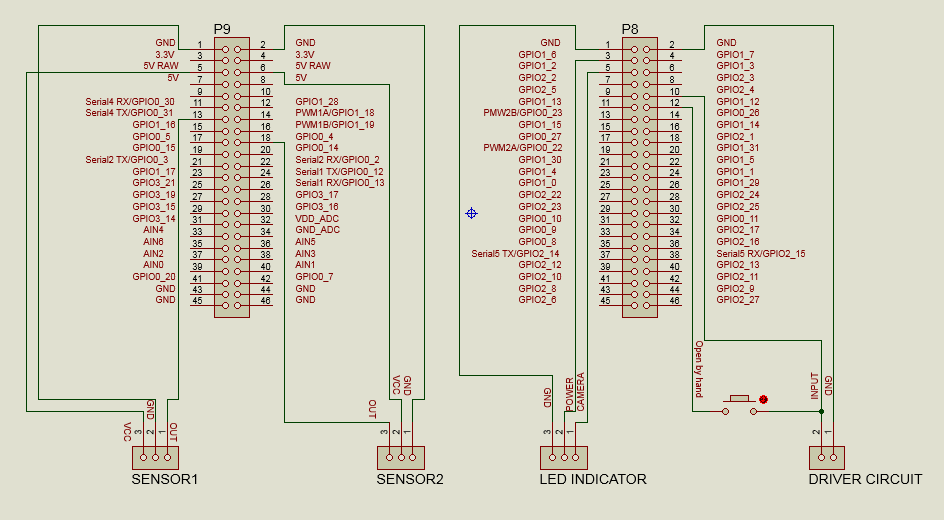
*Figure 31: Solenoid lock*

A **solenoid** is simply a specially designed electromagnet. A solenoid usually consists of a coil and a movable iron core called the *armature*. Here's how it works. When current flows through a wire, a magnetic field is set up around the wire. If we make a coil of many turns of wire, this magnetic field becomes many times stronger, flowing around the coil and through its center in a doughnut shape. When the coil of the solenoid is energized with current, the core moves to increase the flux linkage by closing the air gap between the cores. The movable core is usually spring-loaded to allow the core to retract when the current is switched off. The force generated is approximately proportional to the square of the current and inversely proportional to the square of the length of the air gap.



*Figure 32: Solenoid lock constitute*

#### 1.5 Circuit diagram



*Figure 33: Circuit diagram of components*

### 2.Software

#### 2.1 Android application

##### 2.2.1 Class diagram







*Figure 34: Class diagram*

##### 2.2.2 Class diagram explanation

###### a)UserPageActivity

**Attribute:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| btnChangePass | Button | private | Button go to Change Password layout |
| btnChangrKey: | Button | private | Button go to Change Key layout |
| btnOpenDoor | Button | private | Button go to Open Door layout |
| btnDisPlayQRCode | Button | private | Button Display QR Code layout |

**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| onCreate | void | protected | Create activity |
| onClick | void | public | Click listener |
| onCreateOptionsMenu | boolean | public | Create menu action bar |
| onOptionsItemSelected | boolean | public | Set event selected item action bar |
| onBackPressed() | void | public | Handle back button |

###### b)AdminPageActivity

**Attribute:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| btnCreateUser | Button | private | Button go to Create User layout |
| btnViewLogAd | Button | private | Button go to View Log layout |
| btnChangeKeyAd | Button | private | Button go to Change Key layout |
| btnChangePassAd | Button | private | Button go to Change Password layout |
| btnSettingLockAd | Button | private | Button go to Setting Lock layout |
| btnLockUse | Button | private | Button go to Lock User layout |

**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| onCreate | void | protected | Create activity |
| onClick | void | public | Click listener |
| onCreateOptionsMenu | boolean | public | Create menu action bar |
| onOptionsItemSelected | boolean | public | Set event selected item action bar |
| onBackPressed() | void | public | Handle back button |

###### c) LoginActivity

**Attribute:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| ADMIN | String | private | Variable role admin |
| USER | String | private | Button go to View Log layout |
| txtusername | Edit Text | private | Input username of user |
| txtpassword | Edit Text | private | Input password of user |
| login | Button | private | Button do login function |

**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| onCreate | void | protected | Create activity |
| onClick | void | public | Click listener |

###### d) CheckLogin

**Attribute:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| username | String | private |  |
| password | String | private | Button go to View Log layout |
| response | String | private | Input username of user |

**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| doInBackground | void | protected | Perform compute on a background thread |
| onPostExecute | void | protected | Receive result of doInBackGround method |

###### e) ChangePassWordActivity

**Attribute:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| txtOldPasswords | Edit Text | private | Input Old Password of user |
| txtNewPasswords | Edit Text | private | Input New Password of user |
| txtConfirmNewPasswords | Edit Text | private | Button go to Change Key layout |
| btnChangPasswords | Button | private | Button go to Change Password layout |

**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| onCreate | void | protected | Create activity |
| onClick | void | public | Click listener |
| onCreateOptionsMenu | boolean | public | Create menu action bar |
| onOptionsItemSelected | boolean | public | Set event selected item action bar |

###### f) ChangePassword

**Attribute:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| username | String | public | Save value of username |
| oldPassword | String | public | Save value of oldPassword |
| newPassword | String | public | Save value of newPassword |
| response | String | public | Get result of request |

**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| doInBackground | void | protected | Perform compute on a background thread |
| onPostExecute | void | public | Receive result of doInBackGround method |

###### g) ChangeKeyActivity

**Attribute:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| txtPassword | Edit Text | private | Input Password |
| txtNewKey | Edit Text | private | Input New Key |
| btnChangkey | Button | private | To do change key function |

**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| onCreate | void | protected | Create activity |
| onClick | void | public | Click listener |
| onCreateOptionsMenu | boolean | public | Create menu action bar |
| onOptionsItemSelected | boolean | public | Set event selected item action bar |

###### h) ChangeKey

**Attribute:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| username | String | public | Save value of username |
| password | String | public | Save value of password |
| newKey | String | public | Save value of newKey |
| response | String | public | Get result of request |

**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| doInBackground | void | protected | Perform compute on a background thread |
| onPostExecute | void | protected | Receive result of doInBackGround method |

###### i) ViewLogActivity

**Attribute:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| txtDate | Edit Text | private | Show/Get date in edit text |
| txtLog | Edit Text | private | Show log |
| cal | Calendar | public | Save value of newKey |
| dateFinish | Date | public | Get result of request |

**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| onCreate | void | protected | Create activity |
| onClick | void | public | Click listener |
| onCreateOptionsMenu | boolean | public | Create menu action bar |
| onOptionsItemSelected | boolean | public | Set event selected item action bar |
| getNextDate | String | public | Show and get next date |
| getPreviousDate | String | public | Show and get previous date |
| executeGetLog | void | public | Format day to show |

###### j) GetLog

**Attribute:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| response | String | private | Get result of request |
| dateGetLog | String | private | Save value of dateGetLog |

**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| doInBackground | void | protected | Perform compute on a background thread |
| onPostExecute | void | protected | Receive result of doInBackGround method |

###### k) ManageUserActivity

**Attribute:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| lisUser | ListView | private | Show list user |
| btnLock | Button | private | To do lock function |
| btnActive | Button | private | To do active function |
| btnRemove | Button | private | To do remove function |

**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| onCreate | void | protected | Create activity |
| onClick | void | protected | Click listener |
| onCreateOptionsMenu | boolean | public | Create menu action bar |
| onOptionsItemSelected | boolean | public | Set event selected item action bar |

###### l) LockUser

**Attribute:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| username | String | public | Save value of username |
| status | String | public | Save status |
| response | String | public | Get result of request |

**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| doInBackground | void | protected | Perform compute on a background thread |
| onPostExecute | void | protected | Receive result of doInBackGround method |

###### m) ActiveUser

**Attribute:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| username | String | public | Save value of username |
| status | String | public | Save status |
| response | String | public | Get result of request |

**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| doInBackground | void | protected | Perform compute on a background thread |
| onPostExecute | void | protected | Receive result of doInBackGround method |

###### n) RemoveUser

**Attribute:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| username | String | public | Save value of username |
| response | String | public | Get result of request |

**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| doInBackground | void | protected | Perform compute on a background thread |
| onPostExecute | void | protected | Receive result of doInBackGround method |

###### o) DisplayQRCodeActivity

**Attribute:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| qrInputText | String | public | Save value of qrInput Text |

**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| onCreate | void | protected | Create activity |

###### p) SettingLockActivity

**Attribute:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| txtTimelock | EditText | private | Input time lock |
| txtTimeOpenWrong | EditText | private | Input time open wrong |
| btnSetting | Button | private | To do function setting lock |

**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| onCreate | Void | public | Create Activity |
| onClick | Void | public | Click listener |
| onCreateOptionsMenu | Void | public | Create Action Bar |
| onOptionsItemSelected | Void | public | Set event select action bar |

###### q) SettingLocks

**Attribute:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| timeLock | String | public | Save value of time lock |
| timeOpenWrong | String | public | Save value of time open wrong |
| response | String | public | Get result of network request |

**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| doInBackGround | Void | protected | Perform compute on a background thread |
| onPostExecute | Void | protected | Receive result of doInBackGround method |

###### r) ResetPassowrdActivity

**Attribute:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| txtUsernameUser | EditText | private | Input username of User |
| btnReset | Button | private | To do function Reset |

**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| onCreate | Void | protected | Create Activity |
| onClick | Void | public | Click listener |
| onCreateOptionsMenu | Void | public | Create Action Bar |
| onOptionsItemSelected | Void | public | Set event select action bar |

###### s) ResetPassowrdUser

**Attribute:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| Username | String | Public | Save value of username |
| Response | String | Public | Get result of network request |

**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| doInBackGround | Void | protected | Perform setting lock on a background thread |
| onPostExecute | Void | protected | Receive result of doInBackGround method |

###### t) User

**Attribute:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| mInternalJSONObject | JSONObject | public |  |
| username | String | private | Username of user |
| password | String | private | Password of user |
| role | String | private | Role of user |
| fullname | String | private | Fullname of user |
| status | String | private | Status of user (lock or unlock) |
| key | String | private | Use to open door |

**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
|  |  |  |  |

###### u) OpenDoorActivity

**Attribute:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| txtUsernameOpenDoor | EditText | Private | Input username open door |
| txtKeyOpenDoor | EditText | Private | Input key open door |
| btnOpenDoor | Button | Private | To do function open door |

**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| onCreate | Void | Protected | Create Activity |
| onClick | Void | Public | Click listener |
| onCreateOptionsMenu | Void | Public | Create Action Bar |
| onOptionsItemSelected | Void | Public | Set event select action bar |

###### v) OpenDoor

**Attribute:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| username | String | Public | Save value of username |
| keyOpen | String | Public | Save value of key open |
| response | String | Public | Get result of network request |

**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| doInBackGround | Void | Protected | Perform create user on a background thread |
| onPostExecute | Void | Protected | Receive result of doInBackGround method |

###### w) CreateUserActivity

**Attribute:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| txtFullnameUser | EditText | Private | Input fullname of User |
| txtUsernameUser | EditText | Private | Input username of User |
| txtPasswordUser | EditText | Private | Input password of User |
| txtConfirmPasswordUser | EditText | Private | Input confirm password of User |
| btnCreate | Button | Private | To do function Create User |

**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| onCreate | Void | Protected | Create Activity |
| onClick | Void | Public | Click listener |
| onCreateOptionsMenu | Void | Public | Create Action Bar |
| onOptionsItemSelected | Void | Public | Set event select action bar |

###### x) CreateUser

**Attribute:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| username | String | Public | Save value of username |
| password | String | Public | Save value of password |
| fullname | String | Public | Save value of fullname |
| response | String | Public | Get result of network request |

**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| doInBackGround | Void | Protected | Perform create user on a background thread |
| onPostExecute | Void | Protected | Receive result of doInBackGround method |

###### y) QRCodeEncoder

**Attribute:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| WHITE | Int | Private | White status of QRCode |
| BLACK | Int | Private | Black status of QRCode |
| dimension | Int | Private | Dimension of QRCode |
| contents | String | Private | Content of QRCode |
| displayContents | String | Private | Display content of QRCode |
| title | String | Private | Title of QRCode |
| format | barcodeFormat | Private | Format of QRCode |
| encoded | Boolean | Private | Status of QRCode |

**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| encodeContents | Void | Private | Encode content |
| encodeQRCodeContents | Void | Private | Encode QR Code |
| encodeAsBitmap | Void | Public | Encode As Bitmap |
| guessAppropriateEncoding | Void | Private | Guess appropriate encoding |
| trim | String | Private | Compare result |
| escapeMECARD | String | Private | Escape MECARD |

###### z) LogView

**Attribute:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| mInternalJSONObject | JSONObject | Public | Save JSON Object value |

**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
|  |  |  |  |

##### 2.2.3 Sequence diagram

###### a) Login



*Figure 35: Login sequence*

###### b)Logout



*Figure 36 : Logout sequence*

###### c)Create User



*Figure 37: Create User sequence*

###### d)Change Password



*Figure 38: Change Password sequence*

###### e)Change Key



*Figure 39: Change Key sequence*

###### f)Display QR Code Key



*Figure 40: Display QR Code sequence*

###### g)Reset User Password



*Figure 41: Reset User Password sequence*

###### h)View Log



*Figure 42: View Log sequence*

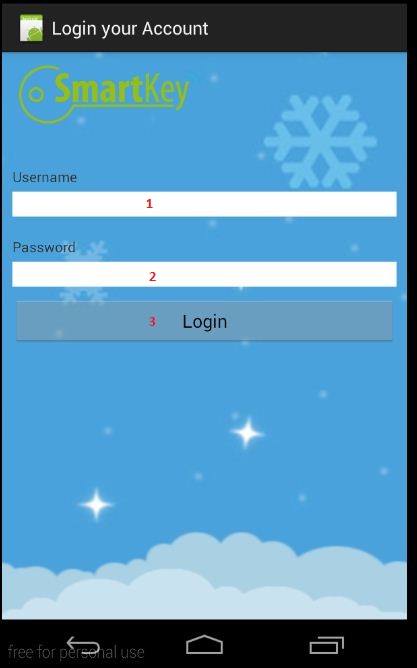
###### i)Setting Lock



*Figure 43: Setting Log sequence*

##### 2.2.4 User Interface

###### a) Login



*Figure 44: Login Page*

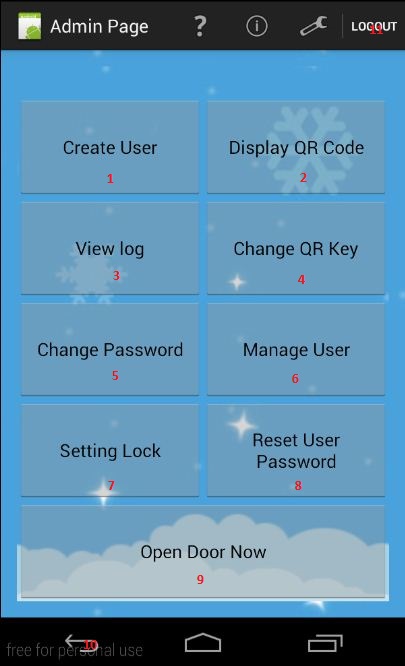
**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** | **Length** |
| 1 | Username | Fill username | No | Yes | Text Box | String | N/A |
| 1 | Password | Fill password | No | Yes | Password | String | N/A |

**Button/Hyperlink**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 3 | Login | Log-in to the system | N/A | Transfer to user page |

###### b) Admin page

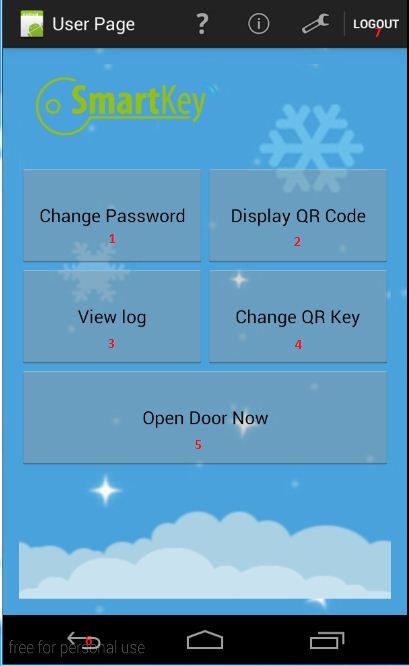


*Figure 45: Admin Page*

**Button/Hyperlink**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Create User | Log-in to the system | N/A | Transfer to user page |
| 2 | Display QR Code | Log-in to the system | N/A | Transfer to user page |
| 3 | View Log | Access to View Log page | N/A | Transfer to View Log page |
| 4 | Change QR Key | Access to Change QR Key page | N/A | Transfer to Change QR Key page |
| 5 | Change Password | Access to Change Password page | N/A | Transfer to Change Password page |
| 6 | Manager User | Access to Manager User page | N/A | Transfer to Manager User page |
| 7 | Setting Lock | Access to Setting Lock page | N/A | Transfer to Setting Log page |
| 8 | Reset Password | Access to Reset Password page | N/A | Transfer to Reset Password page |
| 9 | Open Door Now | Access to Open Door page | N/A | Transfer to Open Door page |
| 10 | Exit | Exit app without log account | N/A | Exit android app |
| 11 | Logout | Logout the system | N/A | Transfer to Login page |

###### c)User page

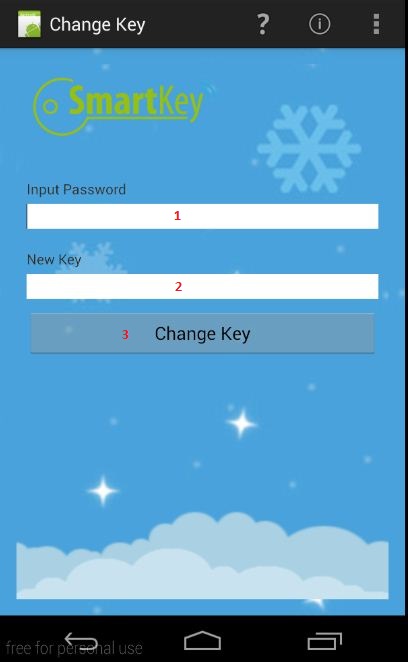


*Figure 46: User Page*

**Button/Hyperlink**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Change Password | Access to Change Password page | N/A | Transfer to Change Password page |
| 2 | Display QR Code | Log-in to the system | N/A | Transfer to user page |
| 3 | View Log | Access to View Log page | N/A | Transfer to View Log page |
| 4 | Change QR Key | Access to Change QR Key page | N/A | Transfer to Change QR Key page |
| 5 | Open Door Now | Access to Open Door page | N/A | Transfer to Open Door page |
| 6 | Exit | Exit app without log account | N/A | Exit android app |
| 7 | Logout | Logout the system | N/A | Transfer to Login page |

###### d) Change key



*Figure 47: Change Key Page*

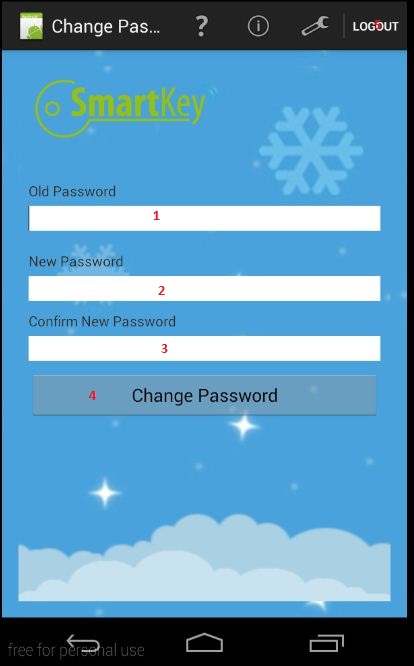
**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** | **Length** |
| 1 | Password | Fill password | No | Yes | Password | String | N/A |
| 2 | New Key | Fill New Key | No | Yes | Text Box | String | N/A |

**Button/Hyperlink**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 3 | Change Key | Change QR key | N/A | Transfer to user page |

###### e) Change password



*Figure 48: Change Password Page*

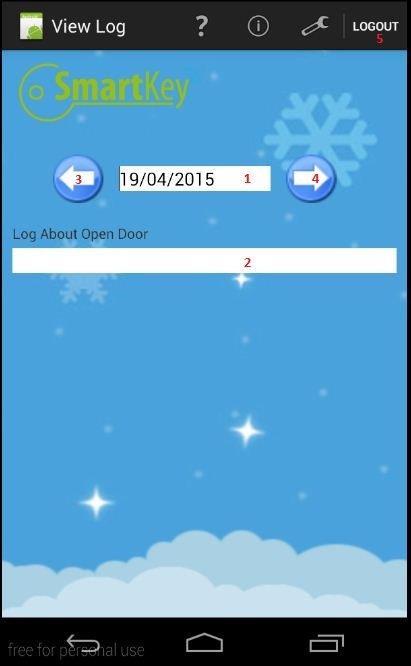
**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** | **Length** |
| 1 | Old Password | Fill old password | No | Yes | Password | String | N/A |
| 2 | New Password | Fill new password | No | Yes | Password | String | N/A |
| 3 | Confirm New Password | Fill confirm new password | No | Yes | Password | String | N/A |

**Button/Hyperlink**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 4 | Change Password | Change Password of user | N/A | Transfer to user page |
| 5 | Logout | Logout the system | N/A | Transfer to Login page |

###### f)View Log



*Figure 49: View Log Page*

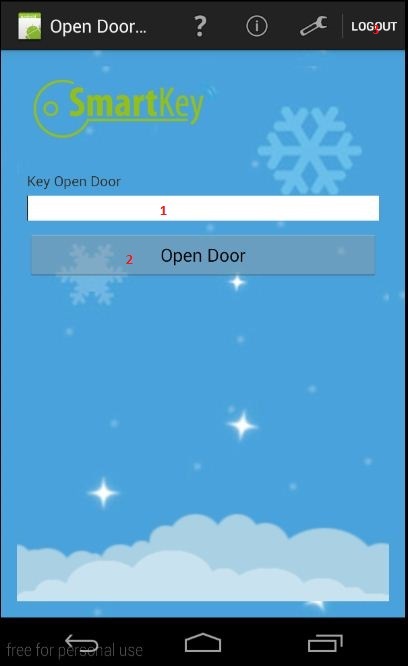
**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** | **Length** |
| 1 | Date time picker | Pick date time | No | Yes | Datetimepicker | String | N/A |
| 2 | Log information | Log information | No | Yes | Text field | String | N/A |

**Button/Hyperlink**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 3 | Arrow backward | Change 1 day before | N/A | No1 – 1 day |
| 4 | Arrow forward | Change 1 day after | N/A | No1+ 1 day |
| 5 | Logout | Logout the system | N/A | Transfer to Login page |

###### g)Open door



*Figure 50: Open Door Now Page*

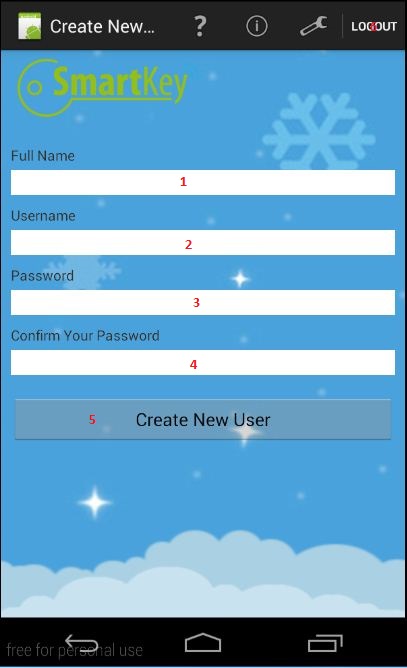
**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** | **Length** |
| 1 | QR Code | Fill QR code | No | Yes | Text Box | String | N/A |

**Button/Hyperlink**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 3 | Open Door | Open the door | N/A | Transfer to user page |

###### h)Create User



*Figure 51: Create New User Page*

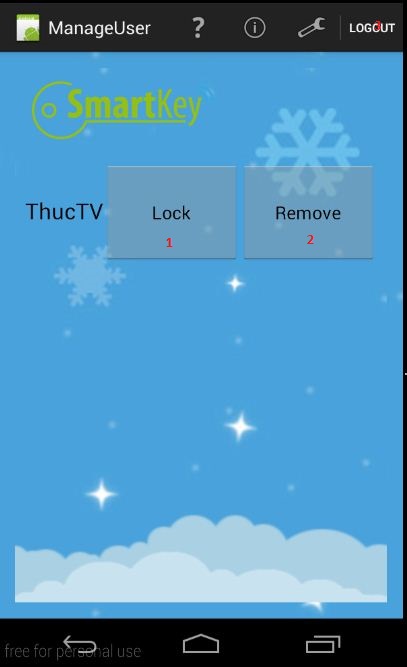
**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** | **Length** |
| 1 | Full Name | Fill full name of user | No | no | Text Box | String | N/A |
| 2 | Username | Fill Username | No | Yes | Text Box | String | N/A |
| 3 | Password | Fill password | No | Yes | Password | String | N/A |
| 4 | Confirm Password | Fill confirm password | No | Yes | Password | String | N/A |

**Button/Hyperlink**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 5 | Create New User | Create new user | N/A | Transfer to user page |
| 6 | Logout | Logout the system | N/A | Transfer to Login page |

###### i)Mange User

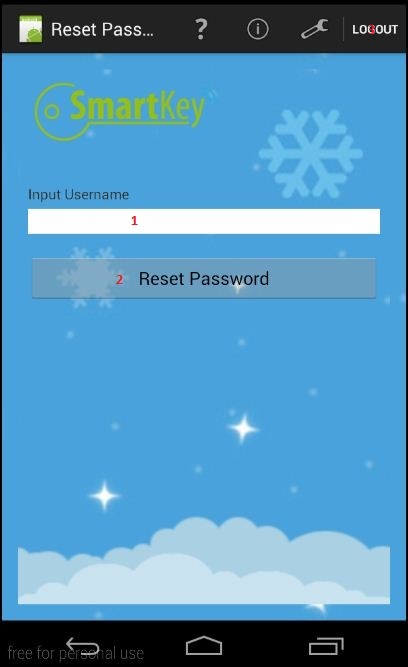


*Figure 52: Manager User Page*

**Button/Hyperlink**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | | **Validation** | **Outcome** |
| 1 | Lock User | Lock User | N/A | | Transfer to user page |
| 2 | Delete User | Delete User on database | N/A | | Transfer to user page |
| 3 | Logout | Logout the system | N/A | | Transfer to Login page |

###### j)Reset password



*Figure 53: Reset Password Page*

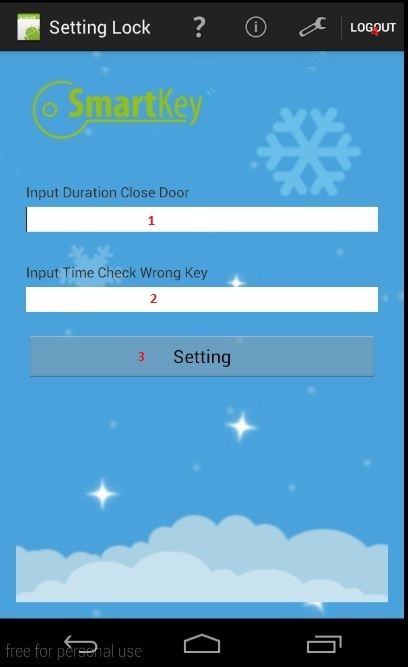
**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** | **Length** |
| 1 | Input Username | Input username of account need to reset password | No | Yes | Text Box | String | N/A |

**Button/Hyperlink**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 2 | Reset Password | Reset password of an user to default | N/A | Transfer to user page |
| 3 | Logout | Logout the system | N/A | Transfer to Login page |

###### k)Setting lock



*Figure 54: Setting Log Page*

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** | **Length** |
| 1 | Input Duration Close Door | Fill time to lock the Lock after unlock | No | Yes | Text Box | Int | N/A |
| 2 | Input Time Check Wrong Key | Fill times can check wrong key, if wrong time over setting, lock will invalid in 5 minutes | No | Yes | Text Box | Int | N/A |

**Button/Hyperlink**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 3 | Setting | Apply setting lock | N/A | Transfer to user page |
| 4 | Logout | Logout the system | N/A | Transfer to Login page |

#### 2.2 Web application (For use in hotel)

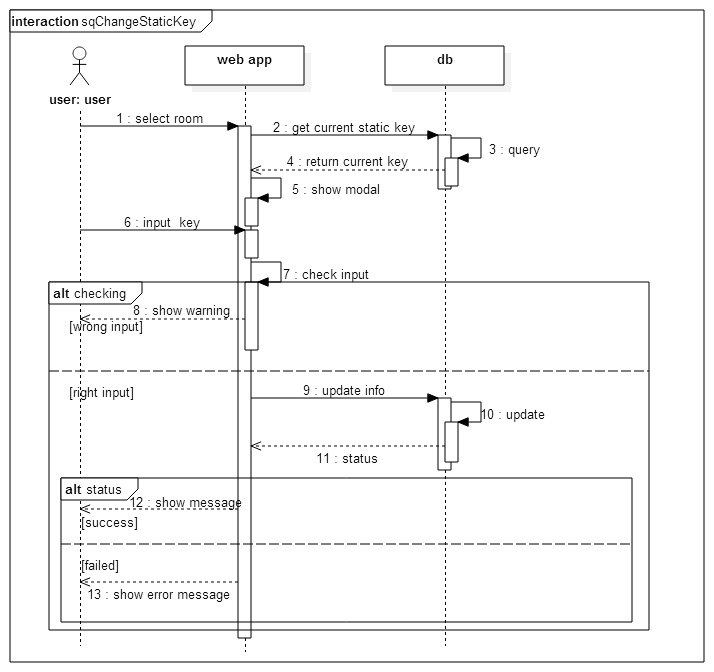
##### 2.2.1 Sequence diagram

###### a) Booking room



*Figure 55: Booking room sequence*

###### b) Change static key



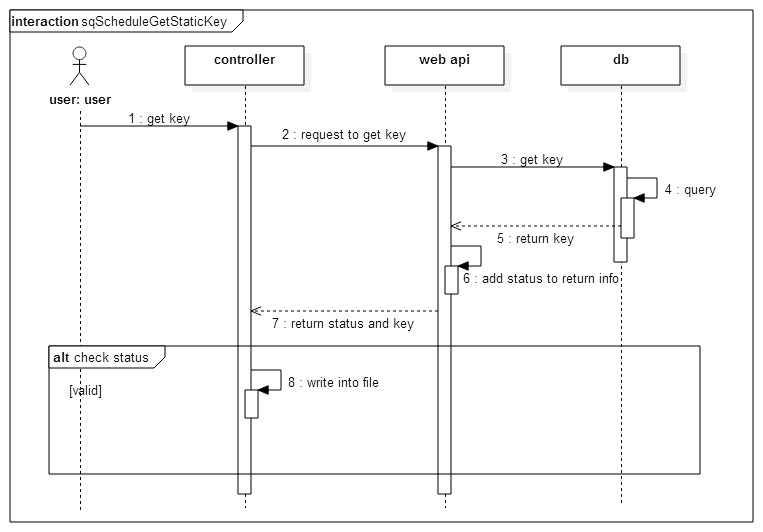
*Figure 56: change static key sequence*

###### c) Show key and send email (Staff)



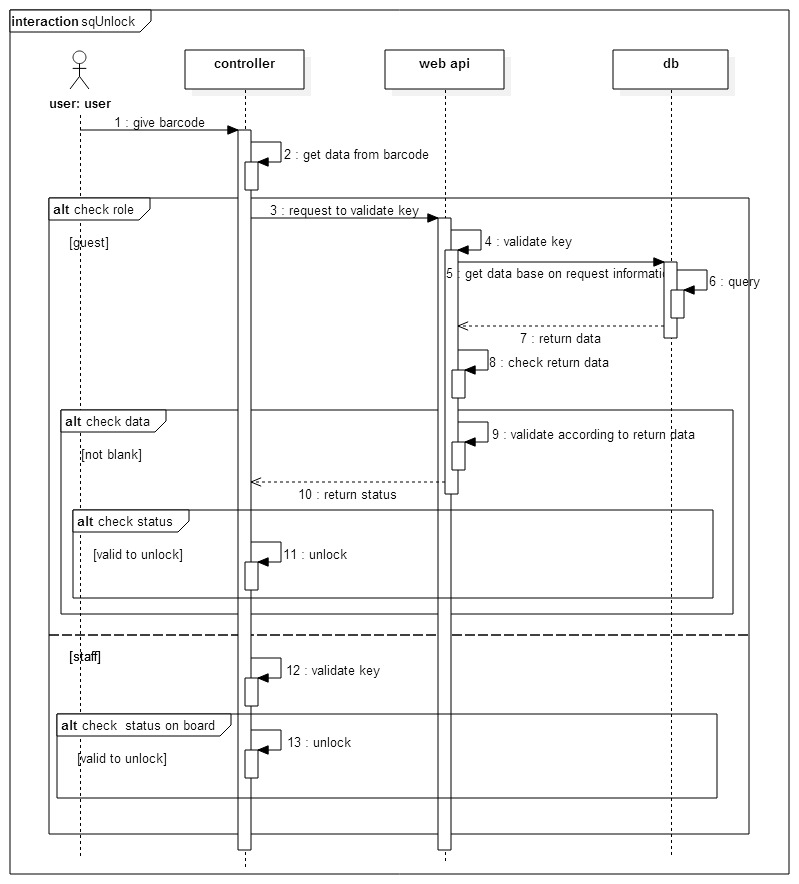
*Figure 57: Show key and send email sequence*

###### d)Get static key on schedule



*Figure 58: Get static key on schedule sequence*

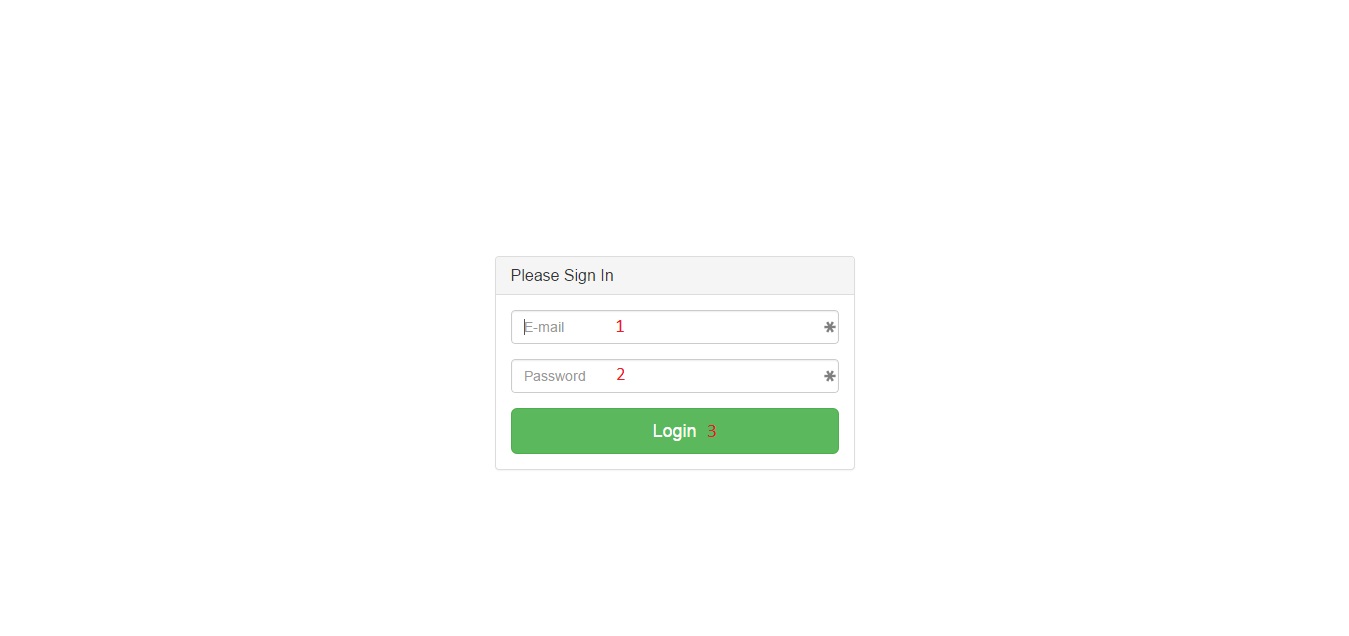
###### e) Unlock



*Figure 59: Unlock sequence*

##### 2.2.2 User Interface

###### a) Login



*Figure 60: Login*

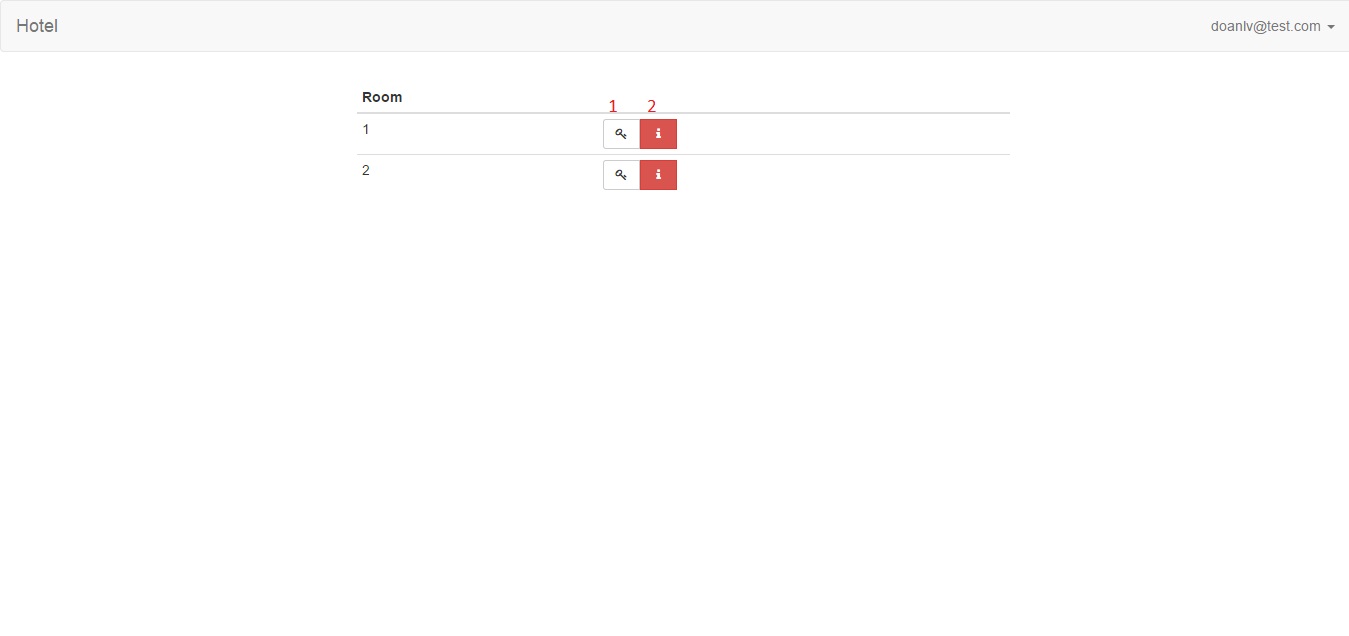
**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** | **Length** |
| 1 | E-mail | Fill e-mail | No | Yes | Textbox | String | N/A |
| 2 | Password | Fill password | No | Yes | Password | String | N/A |

**Buttons/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 3 | Log in | Log in to the system | N/A | Redirect to specific page base on role |

###### b)Manager Page

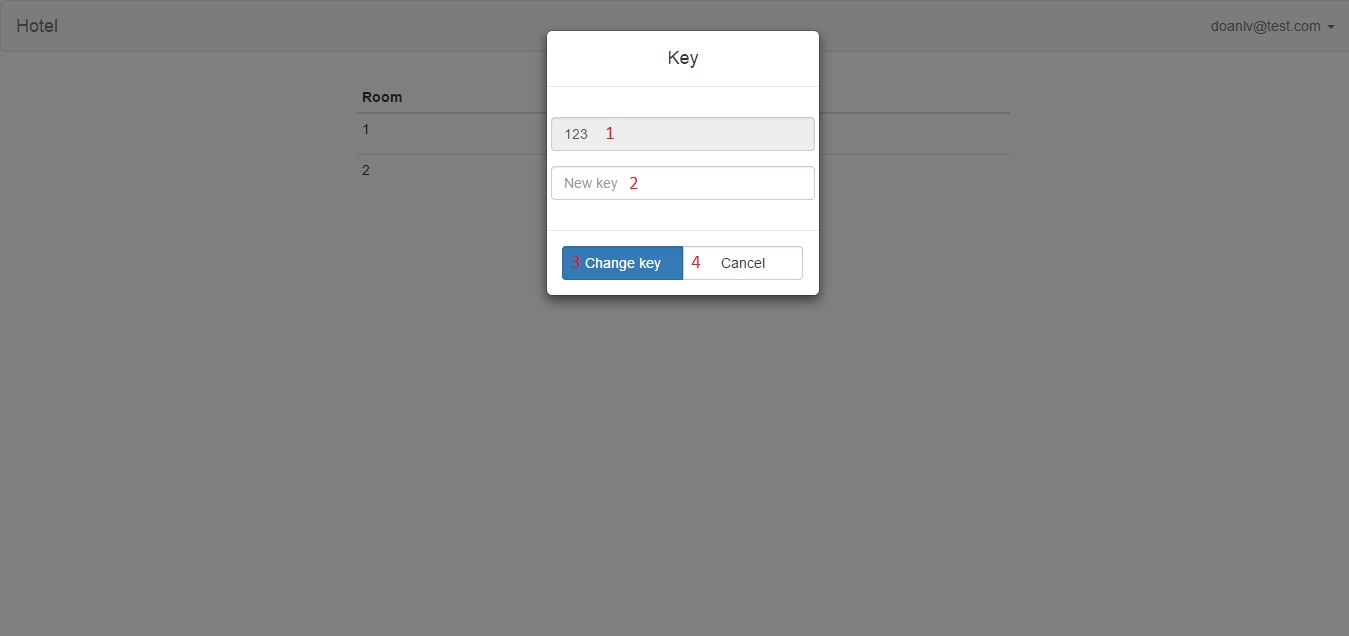


*Figure 61: Magage page*

**Buttons/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Show Key modal | Show Key modal that allows manager views and change static key | N/A | Key Modal |
| 2 | Show Info modal | Show Info modal that allows manager views 10 latest | N/A | Info Modal |

###### c)Key Modal – Manager Page



*Figure 62: Manage Key*

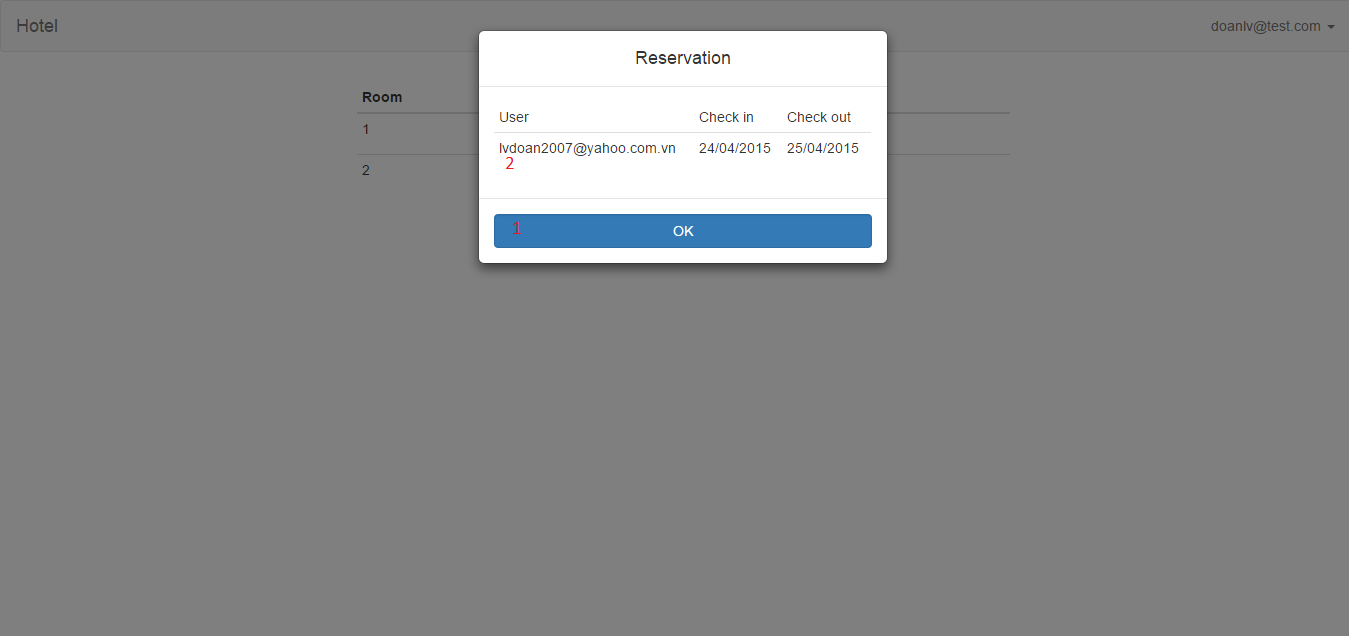
**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** | **Length** |
| 1 | Static Key | Static key of room | Yes | No | Textbox | String | N/A |
| 2 | New key | Fill password | No | No | Textbox | String | N/A |

**Buttons/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 3 | Change key | Change static key of room | N/A | Manager Page |
| 4 | Close modal | Close modal | N/A | Manager Page |

###### d)Info Model- Manager Page



*Figure 63: Manage information*

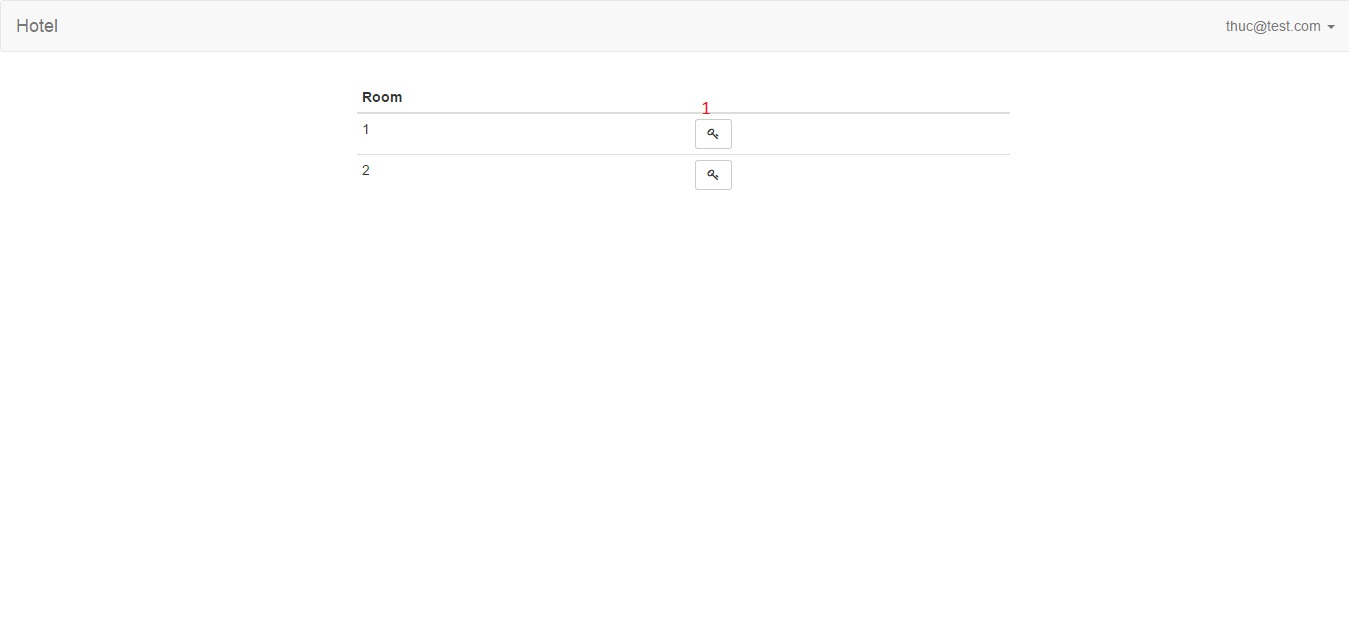
**Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Field** | **Description** | **Read only** | **Mandatory** | **Control Type** |
| 2 | Reservations Table | Show reservations | Yes | No | Table |

**Buttons/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Close modal | Close modal | N/A | Manager Page |

###### e)Staff Page

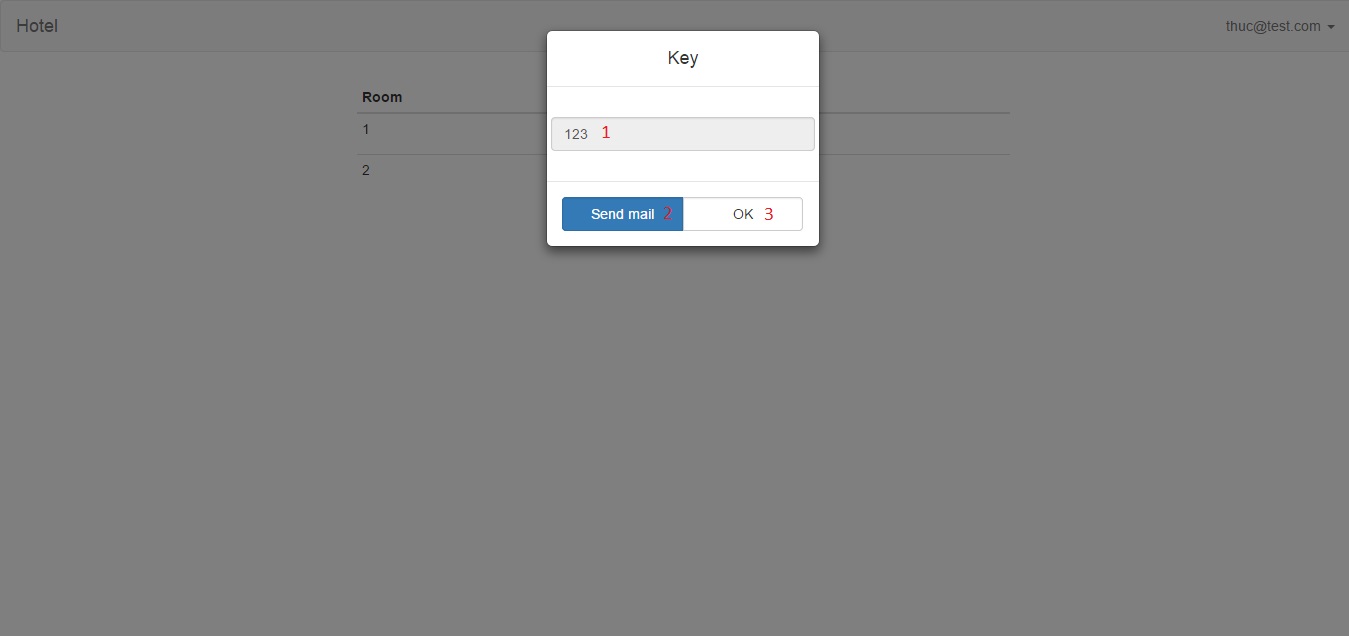


*Figure 64: Staff page*

**Buttons/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Show Key modal | Show Key modal that allows staff view static key and sends code to email. | N/A | Key modal |

###### f)Key Modal – Staff Page



*Figure 65: Manage key- staff*

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** | **Length** |
| 1 | Static Key | Static key of room | Yes | No | Textbox | String | N/A |

**Buttons/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 2 | Send mail | Send code to email | N/A | Staff Page |
| 3 | Close modal | Close modal | N/A | Staff Page |

###### g)Guest Page



*Figure 66: Guest Page*

**Buttons/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Show Key modal | Show Key modal that allows guest books room | N/A | Book Room modal |

###### h)Reservation Modal – Staff Page



*Figure 67: Reservation page*

**Fields**

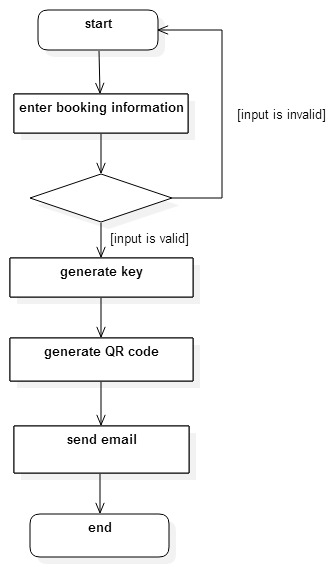
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** | **Length** |
| 1 | Check in | Check in date | No | Yes | Textbox | String | N/A |
| 2 | Check out | Check out date | No | Yes | Textbox | String | N/A |

**Buttons/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 3 | Book room | Reserve room | N/A | Guest Page |
| 4 | Close modal | Close modal | N/A | Guest Page |

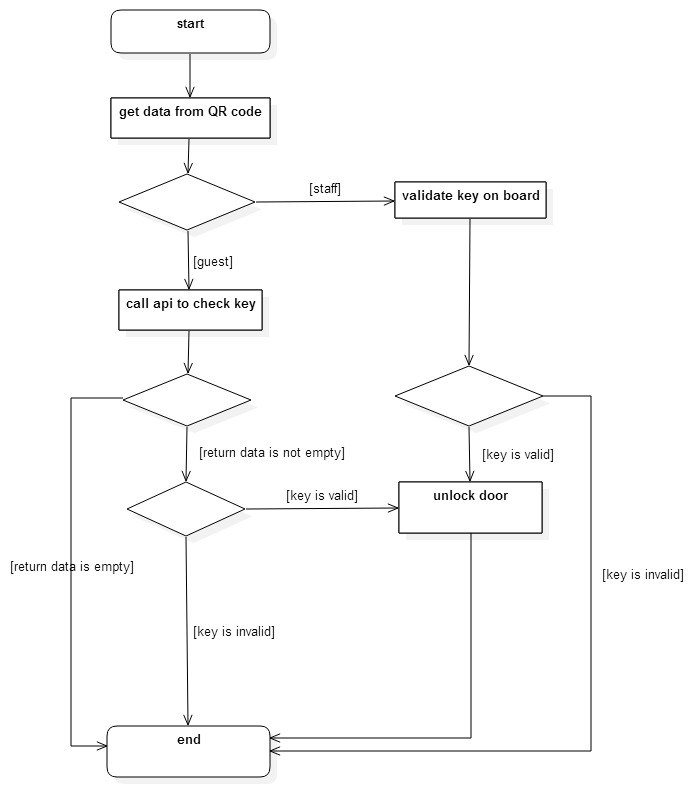
##### 2.2.3 Main flow

###### a)Book room



*Figure 68: Book room flowchart*

###### b) Unlock



*Figure 69: Unlock flowchart*

##### 2.2.4 Database Design

|  |  |
| --- | --- |
| **Entity Data dictionary: describe content of all entities** | |
| **Entity Name** | **Description** |
| User | Information about user that uses the system. |
| Room | Room’s specific information using to identify |
| Reservation | Reservation’s information |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entity Name | Attributes | Description | Domain | Empty |
| User | email | User’s email | String | No |
| password | Password | String | No |
| name | User’s name | String | Yes |
| phone | User’s phone number | String | Yes |
| role | Role of user | String | Yes |
| Room | room\_id | Room identifier | String | No |
| static\_key | Fixed unlock key for room | String | Yes |
| Reservation | email | Email of user who booked room | String | No |
| room\_id | Room identifier | String | No |
| check\_in | Check in day | Date | No |
| check\_out | Check out day | Date | No |
| key | Unique key used to confirm reservation | String | No |

### 3. Controller

## VI. Algorithm and Library

### 1. Introduce QR code

QR code (abbreviated from **Quick Response Code**) is a matrix barcode ( or two-dimensional barcode). First version for the automotive industry in Japan by Denso Wave. A barcode is a machine-readable optical label that contains information about the item to which it is attached. A QR code uses four standardized encoding modes (numeric, alphanumeric, byte / binary, and kanji) to efficiently store data; extensions may also be used.

The QR Code system became popular outside the automotive industry due to its fast readability and greater storage capacity compared to standard UPC barcodes. Applications include product tracking, item identification, time tracking, document management, and general marketing.



*Figure 70: A QR code*

QR code can be used on various mobile device operating systems. Use smartphone can read information quickly.

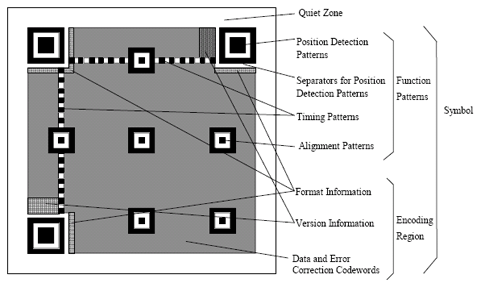
**

*Figure 71: Read information on QR by mobile*

**Design:**

A QR code consists of black modules (square dots) arranged in a square grid on a white background, which can be read by an imaging device (such as a camera) and processed using Reed–Solomon error correction until the image can be appropriately interpreted. The required data are then extracted from patterns present in both horizontal and vertical components of the image

The processor locates the three distinctive squares at the corners of the QR code image, using a smaller square (or multiple squares) near the fourth corner to normalize the image for size, orientation, and angle of viewing. The small dots throughout the QR code are then converted to binary numbers and validated with an error-correcting code



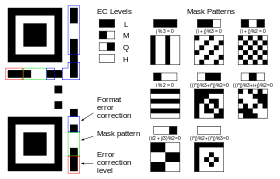
*Figure 72: Structure of QR code*

.

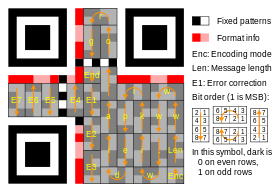
**Encode:**

The format information records two things: the error correction level and the mask pattern used for the symbol. Masking is used to break up patterns in the data area that might confuse a scanner, such as large blank areas or misleading features that look like the locator marks. The mask patterns are defined on a grid that is repeated as necessary to cover the whole symbol. Modules corresponding to the dark areas of the mask are inverted. The format information is protected from errors with a BCH code, and two complete copies are included in each QR symbol.

The message dataset is placed from right to left in a zigzag pattern, as shown below. In larger symbols, this is complicated by the presence of the alignment patterns and the use of multiple interleaved error-correction blocks.

****

*Figure 73: Format information*

****

*Figure 74: Character Placement*

### 2. Zbar library



*Figure 75: Zbar*

ZBar is an open source software suite for reading bar codes from various sources, such as video streams, image files and raw intensity sensors. It supports many popular symbologies (types of bar codes) including EAN-13/UPC-A, UPC-E, EAN-8, Code 128, Code 39, Interleaved 2 of 5 and QR Code.

The flexible, layered implementation facilitates bar code scanning and decoding for any application: use it stand-alone with the included GUI and command line programs, easily integrate a bar code scanning widget into your Qt, GTK+ or PyGTK GUI application, leverage one of the script or programming interfaces (Python, Perl, C++) ...all the way down to a streamlined C library suitable for embedded use.

**Application**

* Retail
* Automated document processing
* Inventory tracking
* Mobile applications

**Features**

* Cross platform - Linux/Unix, Windows, iPhone®, embedded...
* High speed - real time scanning from video streams
* Small memory footprint
* Small code size - the core scanner and EAN decoder represent under 1K lines of C code
* Not limited to images
* No floating point operations
* Suitable for embedded applications using inexpensive processors/hardware
* Modular components can be used together or separately

# SYSTEM IMPLEMENTATION & TEST (SIT)

## Introduction

### System overview

In this section, there has all necessary information about test plan, test cases, test result, the environment for testing, test pass/fail criteria, risks estimations and a checklist to check when test this system.

### Test Approach

* + Method : Black- box Testing

## Test plan

1. Features to be tested:

- Hardware:Sensor, Power Supplier, Camera.

- Software: Android app, Web module

1. Features not to be test:
   * Board BeagleBone, Lock.

## Test Case

### 1. Mobile app test case

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Function | | | Precondition | Step of testing | Expected Result | Created by | Executed by | Result |
| Large Function | Medium Function | Small Function |
| 1 | Login | Login layout | Text box “User Name” | Open Android application | Enter “ 111111” in textbox “Username” | Textbox show”111111” | NamNT | NamNT | Pass |
| 2 | Textbox ”Password” | Open Android application | Enter password “111111” | Password will hide with character “●●●●●●” | NamNT | NamNT | Pass |
| 3 | Login button | Open Android application | Enter Username and Password | System connect and login | NamNT | NamNT | Pass |
| 4 |  | Login without enter Username and Password | Open Android application | Click on button “Login” without Username and Password | System show dialog “Please input your username and password”. | NamNT | NamNT | Pass |
| 5 | Login Fail | Open Android application | Enter wrong Username or Password then click on button Login | System show dialog “Login Fail! Please input right username, password”. | NamNT | NamNT | Pass |
| 6 |  | Login Successful | Open Android application | Enter right Username and Password then click on button Login | Show interface of user or admin, depend on kind of account. | NamNT | NamNT | Pass |
| 7 | User layout | Change password | Go to change password layout | Login successful | Click on Change password | Go to Change password layout | NamNT | NamNT | Pass |
| 8 | Enter old password | Change password layout | Enter old password on textbox Old Password | Old Password will show character “●●●●●●” | NamNT | NamNT | Pass |
| 9 | Enter new password | Change password layout | Enter new password on textbox New Password | New Password will show character “●●●●●●” | NamNT | NamNT | Pass |
| 10 | Change password without enter old password or new password | Change password layout | Click on button Change Password | System show dialog” Please input enough information!!!”. | NamNT | NamNT | Pass |
| 11 | Enter wrong Old Password | Change password layout | Enter wrong old pass word  Enter new password and confirm new password  Click on button Change Password | System show dialog “FAIL\_USER\_NOT\_EXIST”. | NamNT | NamNT | Pass |
| 12 |  |  | Enter wrong confirm new password | Change password layout | Enter Old Password  Enter new password and wrong confirm new password  Click on button Change Password | System show dialog “FAIL\_INCORRECT\_PASSWORD”. | NamNT | NamNT | Pass |
| 13 | Change password successful | Change password layout | Enter Old Password, New Password and Confirm New Password  Click on button Change Password | Screen come back user or admin layout | NamNT | NamNT | Pass |
| 14 | Display QR code | Display QR code | Login successful | At user or admin layout click on “Get QR Code Key” | System show QR code | NamNT | NamNT | Pass |
| 15 |  | Show log | Show log | Login successful | At user or admin layout click on “ View Log” | Show information log of user | NamNT | NamNT | Pass |
| 16 | Filter log | Log information had showed | User click EditText date show DatePicker or click button back or button forward . | View log arranged by time | NamNT | NamNT | Pass |
| 17 | Change QR Key | Go to Change QR Key layout | Login successful | Click on button “Change QR Key” in user or admin layout. | Show change QR key layout. | NamNT | NamNT | Pass |
| 18 |  |  | Change key without information new key | At the Change QR Key layout | Click on button “ Change Key” without enter new QR code | Show dialog “Please input enough information!!!”. | NamNT | NamNT | Pass |
| 19 | Change key with wrong password | At the Change QR Key layout | Click on button “ Change Key” with wrong password | Show Toast “FAIL\_INCORRECT\_PASSWORD” | NamNT | NamNT | Pass |
| 20 | Change key successful | At the Change QR Key layout | Enter new QR code and click on button “Change Key” | Come back user or admin layout. | NamNT | NamNT | Pass |
| 21 |  | Open Door Now(Enter QR code while camera had broken) | Go to Open Door Now layout | Login successful | At the user or admin layout click on button “ Open Door Now” | Go to Open Door Now layout | NamNT | NamNT | Pass |
| 22 | Open Door without enter Key | At the Open Door Now layout | Click on button Open Door without enter key | Show dialog “Please input enough information!!!”. | NamNT | NamNT | Pass |
| 23 | Open Door with wrong key | At the Open Door Now layout | Enter wrong key and click on button Open Door | Show Toast “FAIL\_INVALID\_KEY” | NamNT | NamNT | Pass |
| 24 |  |  | Open Door with right key | At the Open Door Now layout | Enter right key and click on “Open Door” | Door has opened and come back user or admin layout | NamNT | NamNT | Pass |
| 25 | Logout | Logout | Login successful | User click on “Logout” | Come back Login layout | NamNT | NamNT | Pass |
| 26 | Reset Password( only for admin) | Go to “Reset Password User” layout | Login successful | At the admin layout click on button “Reset Password User” | Go to “Reset Password User” layout | NamNT | NamNT | Pass |
| 27 |  |  | Reset password without information of user | At the Reset Password User layout | Click on Reset button without information of user | Show Dialog “Please input enough information!!!”. | NamNT | NamNT | Pass |
| 28 | Reset password with wrong information of user | At the Reset Password User layout | Click on Reset button with wrong information of user | Show Dialog “FAIL\_USER\_NOT\_EXIST”. | NamNT | NamNT | Pass |
| 29 | Reset password with right information of user | At the Reset Password User layout | Click on Reset button with right information of user | Come back admin layout | NamNT | NamNT | Pass |
| 30 |  | Create User (only for admin) | Go to “Create User” layout | Login successful | At the admin layout click on button “Create User” | Go to “Create User” layout | NamNT | NamNT | Pass |
| 31 | Create user without information of user | At the Create User layout | Click on button Create New User without enter information of new user | Show Dialog “Please input enough information!!!”. | NamNT | NamNT | Pass |
| 32 | Create User with special character in FullName | At the Create User layout | Click on button Create New User with special character in FullName | Show Dialog “Invalid!!! Don’t input special character in name”. | NamNT | NamNT | Pass |
| 33 |  |  | Create User with confirm wrong password | At the Create User layout | Click on button Create New User with confirm wrong password | Show Dialog “Please input right confirm password!!!”. | NamNT | NamNT | Pass |
| 34 | Create User with enter username existed | At the Create User layout | Click on button Create New User with enter username existed | Show Toast “FAIL\_USER\_EXIST”. | NamNT | NamNT | Pass |
| 35 | Create User successful | At the Create User layout | Click on button Create New User with right information of new user | Come back admin layout | NamNT | NamNT | Pass |
| 36 |  | Lock User (only for admin) | Go to Lock User layout | Login successful | At the admin layout click on “Lock User” button | Go to Lock User layout | NamNT | NamNT | Pass |
| 37 | Lock User | At the Lock User layout | Admin choose an username and click on Lock | Come back admin layout, user looked will be delete on database. | NamNT | NamNT | Pass |
| 38 |  | Setting Lock (only for admin) | Go to Setting Lock Layout | Login successful | At the admin layout click on “Setting Lock” button | Go to Setting Lock layout | NamNT | NamNT | Pass |
| 39 |  |  | Setting Lock without information | At the Setting Lock layout | Click on “Setting” button without information | Show error dialog “Please input enough information!!!”. | NamNT | NamNT | Pass |
| 40 | Setting lock with wrong format information | At the Setting Lock layout | Click on “Setting” button with wrong format information | Show error dialog format. | NamNT | NamNT | Pass |
| 41 | Setting Lock successful | At the Setting Lock layout | Click on “Setting” button with right information | Come back admin layout and apply new setting | NamNT | NamNT | Pass |

### 2. Hardware test case

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Function | | | Precondition | Step of testing | Expected Result | Created by | Executed by | Result |
| Large Function | Medium Function | Small Function |
| 1 | Power Supplier | Power Supplier | Power from adapter 12VDC | Divice is ready | Plugin adapter on Power Supplier board | Led Power adapter on board is on. | Namnt | Namnt | Pass |
| 2 | Power from Pin 12VDC | Divice is ready | Plugin pin 12VDC on Power Supplier | Lock is unlock. | Namnt | Namnt | Pass |
| 3 | Output Power 5VDC for BeagleBone | Divice is ready  Power Supplier have power supplies | Plugin jack DC from Power Supplier board to BeagleBone | Led Power of BeagleBone is on. | Namnt | Namnt | Pass |
| 4 | Ouput Power 12VDC for Solenoid | Divice is ready  Power Supplier have power supplies | Use DC Voltmeter to measure volt at output DC 12V | DC Voldmeter show 12V | Namnt | Namnt | Pass |
| 5 | Controller Lock | Controller Lock | Divice is ready  Power Supplier have power supplies | Push a pulse to input controller lock | Lock is unlock. | Namnt | Namnt | Pass |
| 6 | Infrared Sensor | Infrared Sensor | Infrared Sensor | Supply power | Bring a book go near pair Diode | Light signal is on. | Namnt | Namnt | Pass |
| 7 | Camera | Camera | Camera | Supply power | Get data from camera | Get image from camera | Namnt | Namnt | Pass |

# SYSTEM USER’S MANUAL

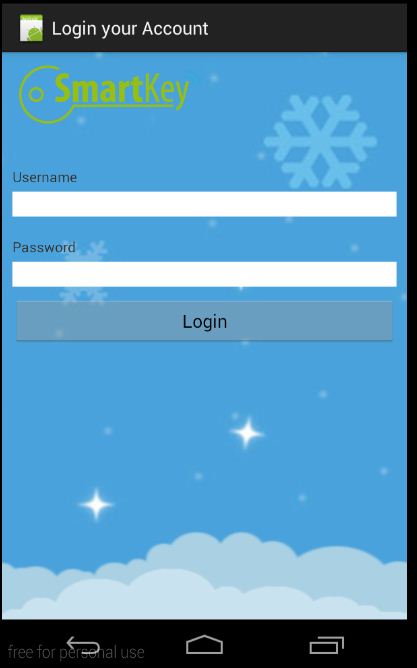
## I.Installation Guide

## II. User’s Guide

### 1. Mobile application

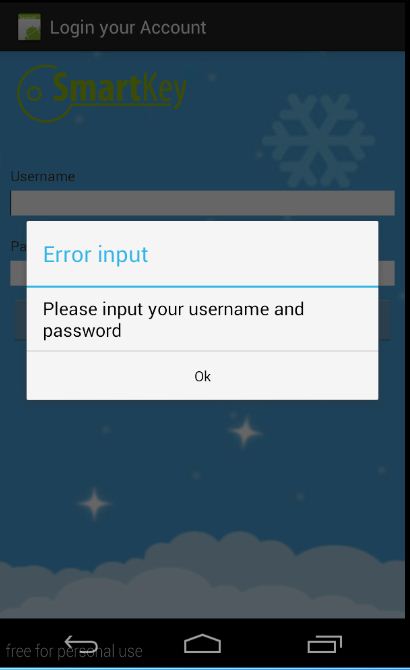
**Login application android:**

Enter Username and Password then click on button “Login” to login.



*Figure 76: Login*

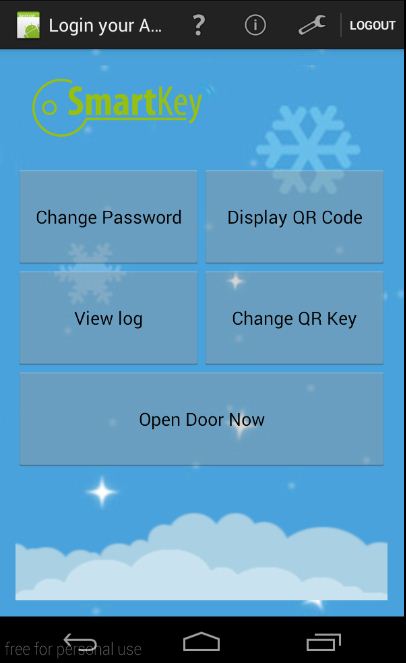
If Click “Login” button without Username and Password. App will show dialog “ Please input your username and password”.



*Figure 77: Re input information*

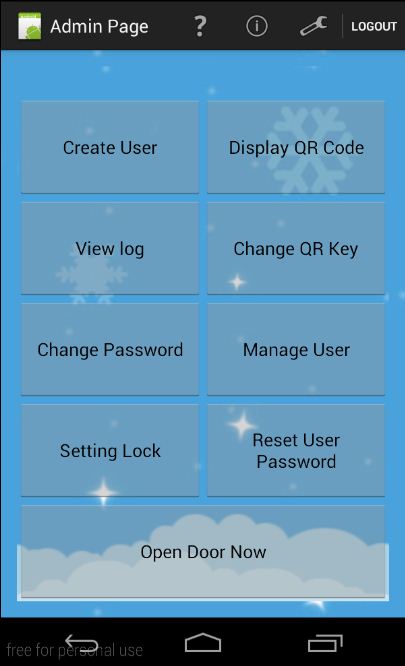
If Login successful app will go page for user or admin depend on kind of account.

User’s page: with some functions Change Password, Display QR Code, View Log, Change QR Key, Open Door Now and Logout.



*Figure 78: User page*

Admin page: with some function Change Password, Display QR Code, View Log, Change QR Key, Open Door Now, Create User, Setting Lock, Lock User, Reset User Password and Logout.



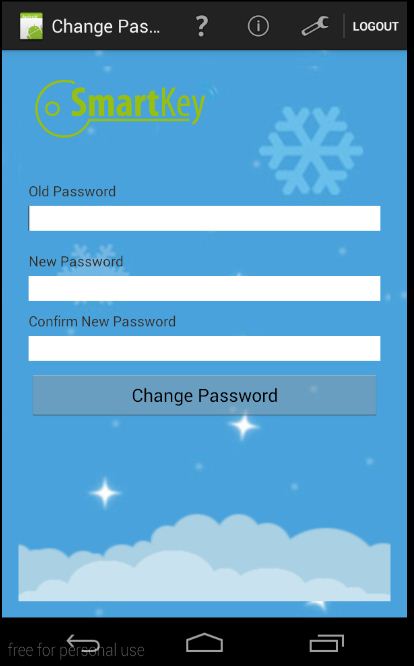
*Figure 79: Admin Page*

For the User, they have some functions listed :

#### Change Password:

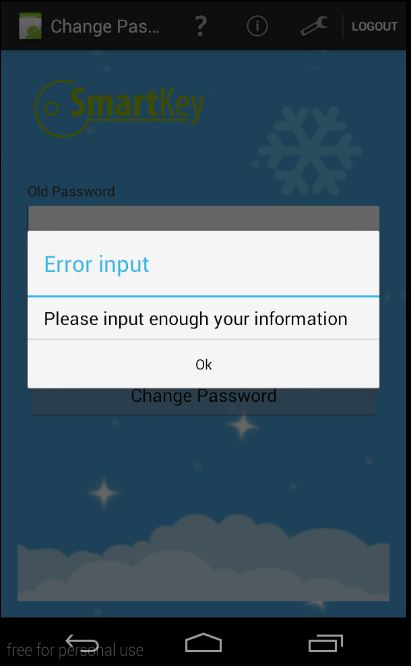
User click on “Change Password” button from User’s Page to go Change Password page.

User have to enter right Old Password, New Password, Confirm New Password then click on button “ Change Password”.



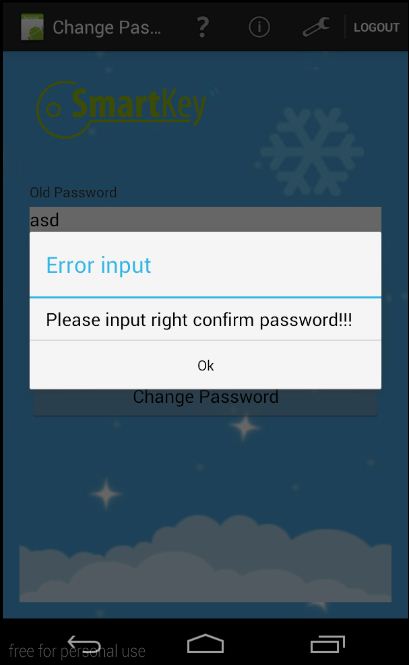
*Figure 80: Change Password*

If any field haven’t filled. App will show error notice : “Please input enough your information”



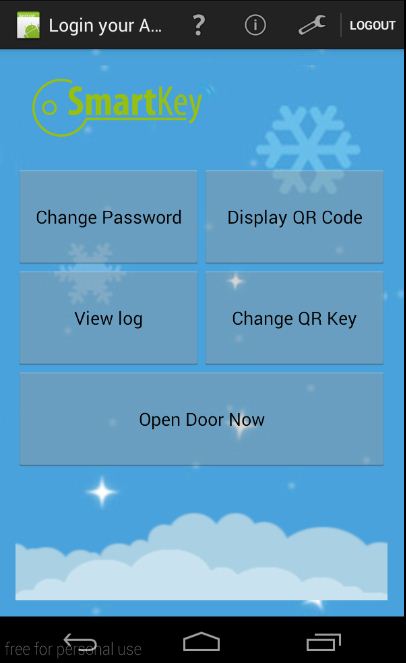
*Figure 81: Reinput information change password*

If User enter wrong Confirm New Password wrong. App will show error : “Please input right confirm password!!!”



*Figure 82: Wrong confirm new password*

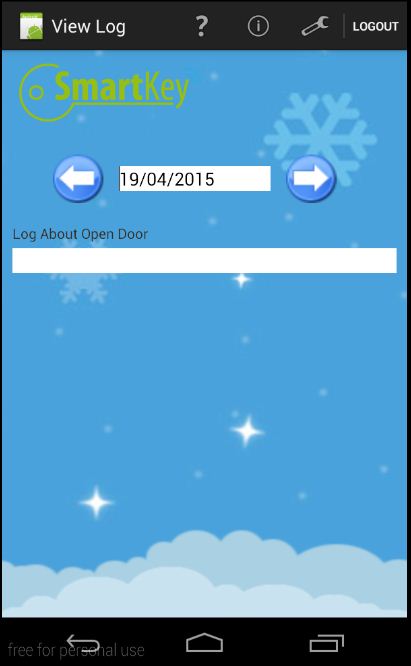
If user completed change password app will come back User’s Page.



*Figure 83: Come back User page after change password*

#### View Log:

User click on button “View Log” from User’s Page to go View Log page.

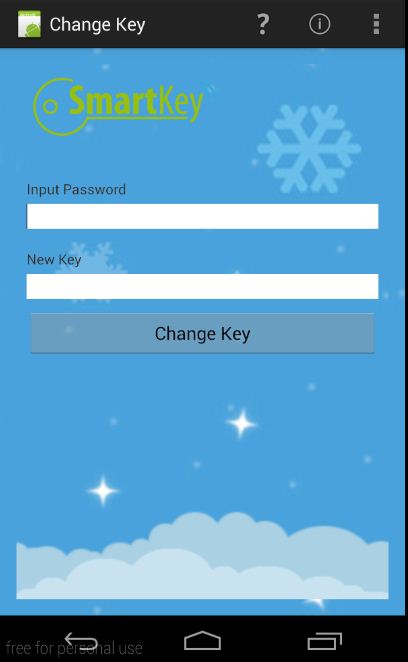


*Figure 84: View Log*

User use date time picker to view log arranged by day.

#### Change QR Code:

User click on button “Change QR Code” from User’s Page to go Change QR Code.

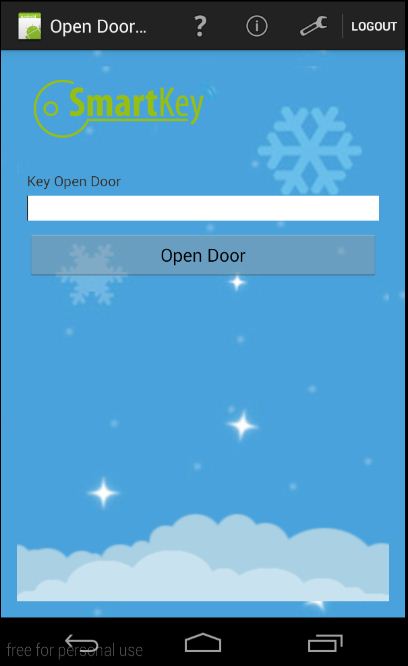


*Figure 85: Change QR code*

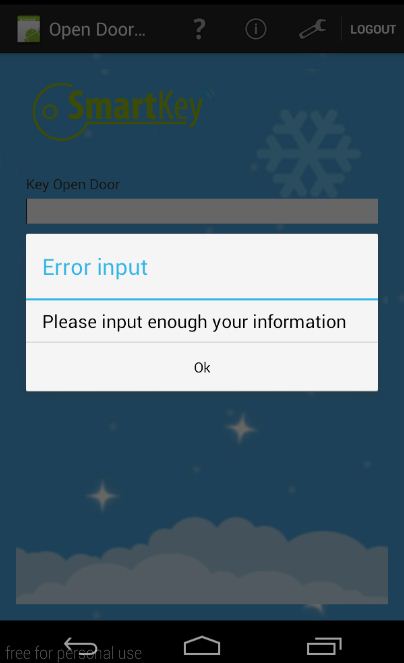
User enter old QR code (Input Password) and new QR code (New Key), then click on button “change Key” to finish. After that app will come back User’s Page.

#### Open Door :

User open door by enter QR by hand on the mobile. To use this function, user click on button “Open Door Now” from User’s Page.



*Figure 86: Open Door Now*

User Enter QR code by hand, then click “Open Door” button to open the door by mobile. If user don’t enter QR code , app will show notice :”Please input enough information”. 

*Figure 87: Reinput QR code to open door*

#### Display QR Code:

To use this function, user click on “Display QR Code” button from User’s Page. App will go “Display QR Code” and show QR code of user.

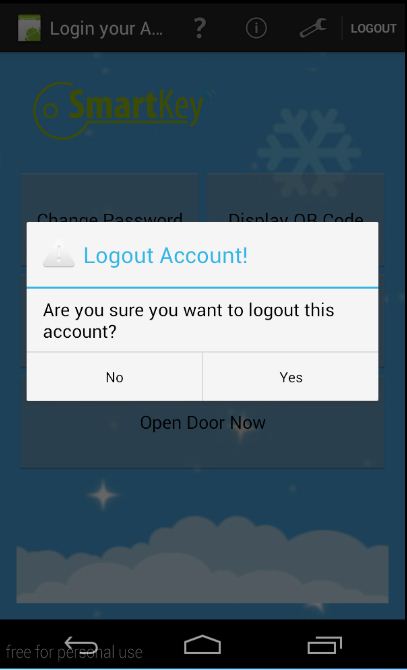
**Display QR Code:** To use this function, user click on “Display QR Code” from User’s Page. App will go forward Display QR Code page and show QR code. User can use it to open door outside.



*Figure 88: Display QR code*

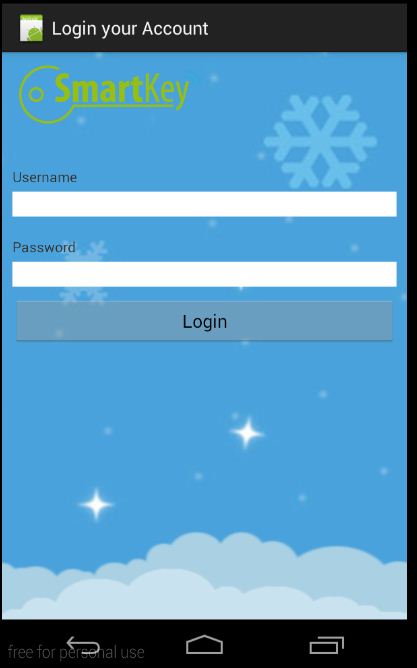
#### Logout:

To use this function, user click on “Logout” label on the top right of screen and confirm to logout and come back login page.



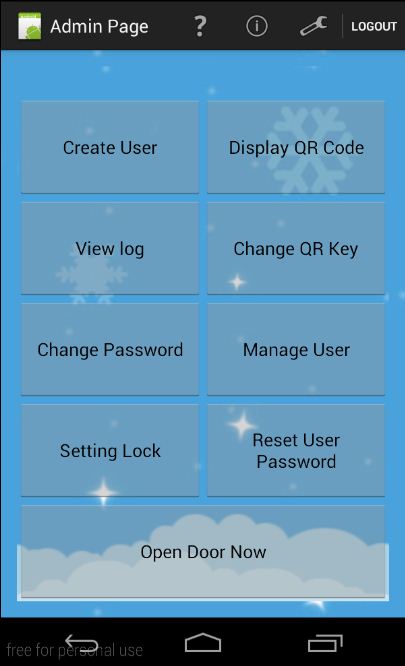
*Figure 89: Confirm Logout*

After logout come back Login page.



*Figure 90: After Logout will come back Login page*

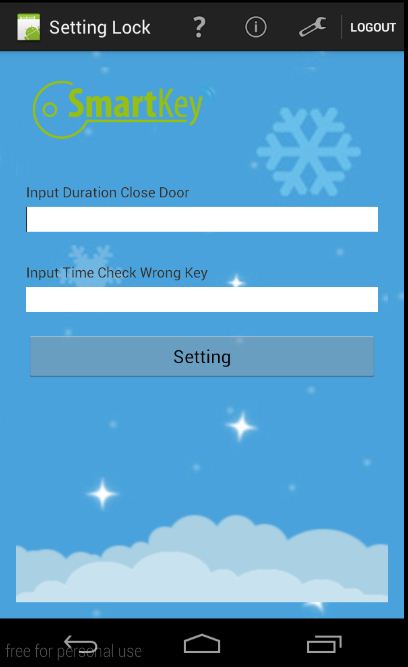
**For admin:** Admin have some function similar user such as : Change Password, Change QR Code, View Log, Display QR Code, Open Door Now, Logout. Beside admin have some function only for admin : Create User, Setting Lock, Reset User Password, Manager User**.**

****

*Figure 91: Admin page*

#### Setting Lock:

To use this function , admin have to click on Setting Lock from Admin Page. App will go Setting Lock page.

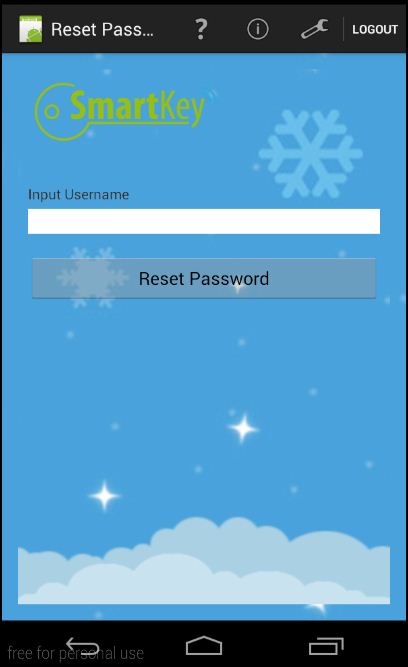


*Figure 92: Setting Log*

In this function admin can set T time open the door and times to scan wrong QR code. If user check wrong QR over times admin setting, camera will pause in 5 minutes. Each time wrong code , Device will play sound to warning user.

#### Reset User Password:

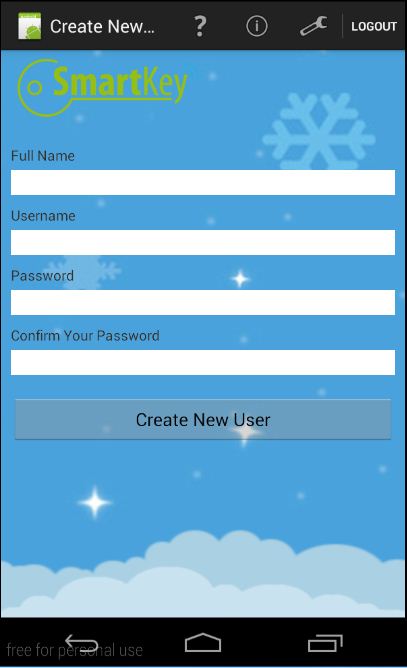
To use this function admin have to click on button “Reset User Password” from Admin Page. App will go Reset Password page. Admin will enter username of user want reset password, password change Default.



*Figure 93:Reset User Password*

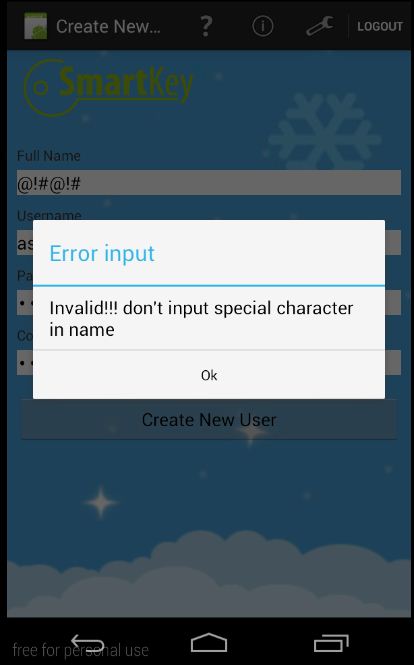
#### Create User

To use this function admin have to click on button “Create User” from Admin Page. App will go Create User page.



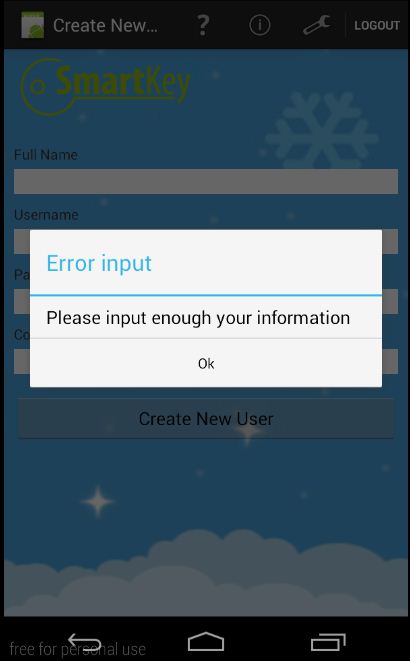
*Figure 94: Create User*

Admin enter Full Name of new user, Username, Password and Confirm Password. Special character not accepted for all field.



*Figure 95: Wrong format information*

An error will be showed if any field aren’t filled.



*Figure 96: Lack information*

#### Manager User

To use this function admin have to click on button “Manager User” from Admin Page. App will go Manager User page.



*Figure 97: Manager User*

In this function admin can Lock or Remove user display on screen.

### 2. Lock Control module

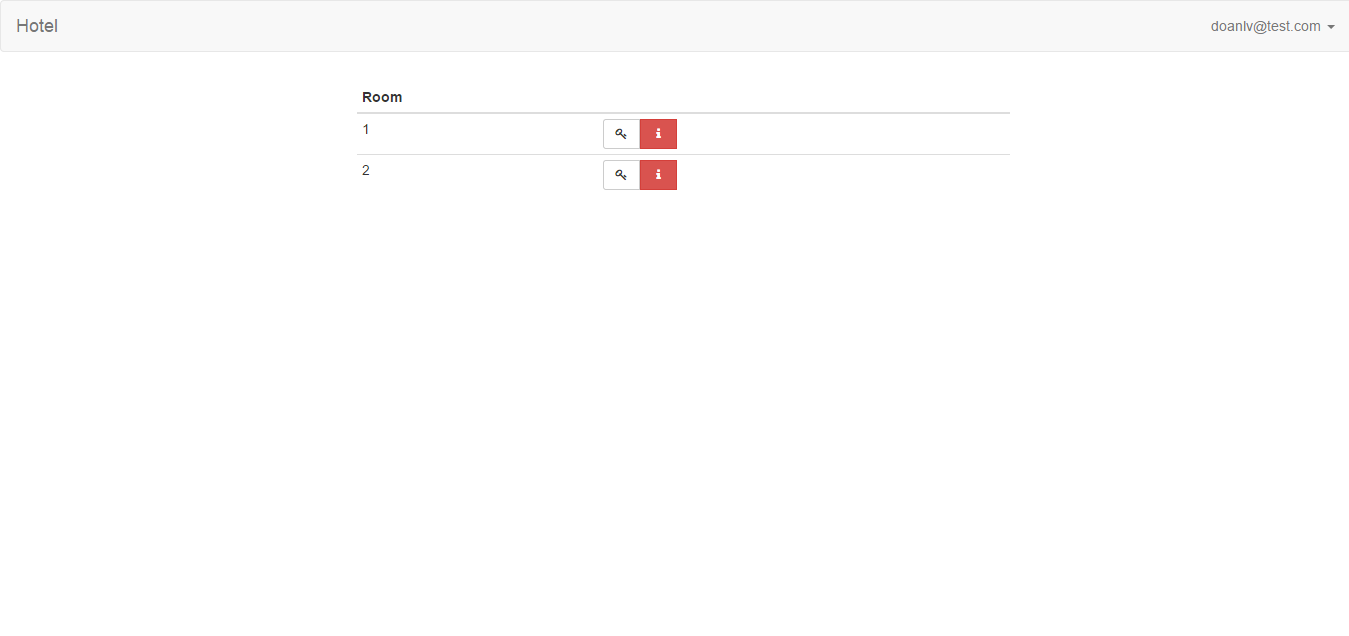
### 3. Web application (for use in hotel)

#### 3.1 Access

*Figure 98: Access*

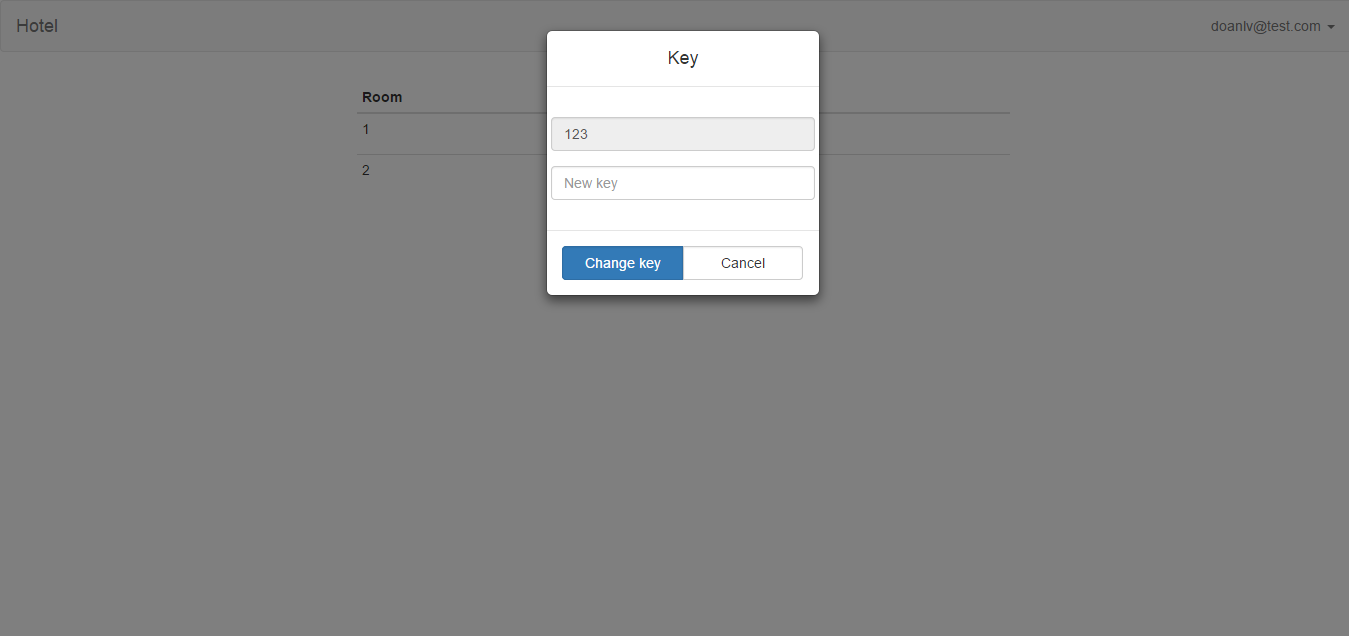
You have to login to the system to use functions. At the login screen, you ought to fill two fields E-mail and Password with your own information that registered or granted before.

#### 3.2 Manager



*Figure 99: Manage room*

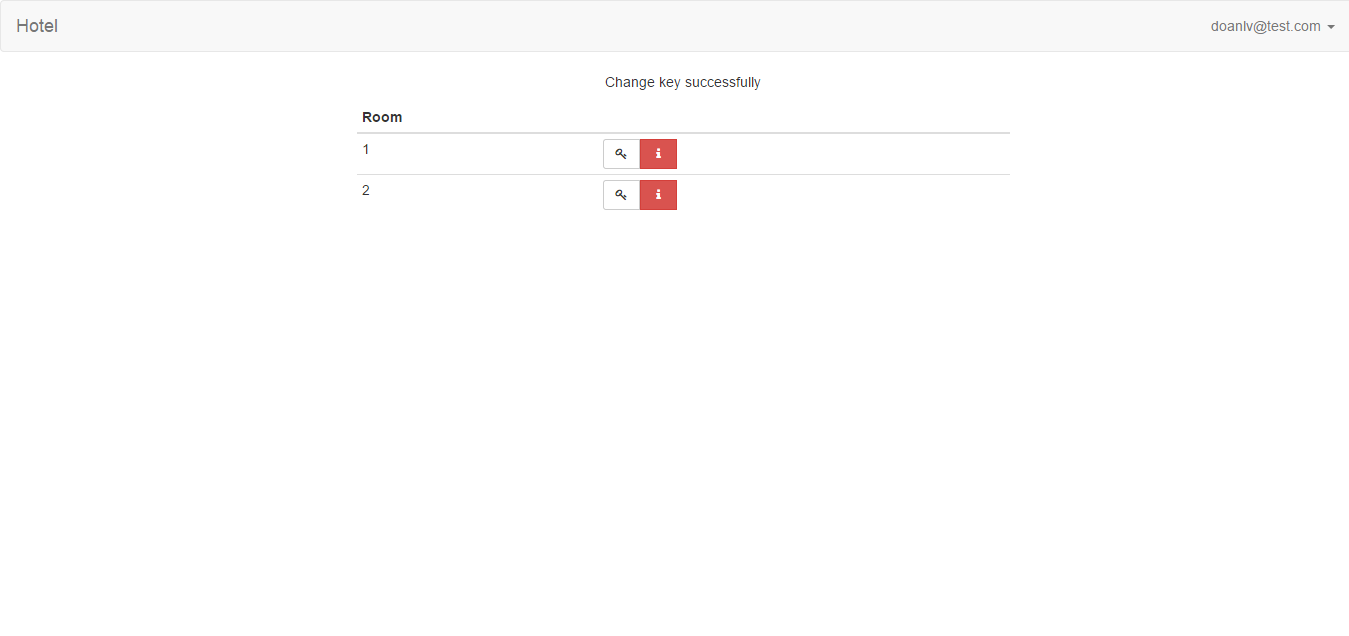
You will be redirect to Manager page from Login page if your role is manager. You can easily see that there is list of rooms. For each room, you have to option followed by two buttons, button with key icon or one with I (means information) icon. First button allows you view static key of room and change key, second one allows you to view list of reservations of room.



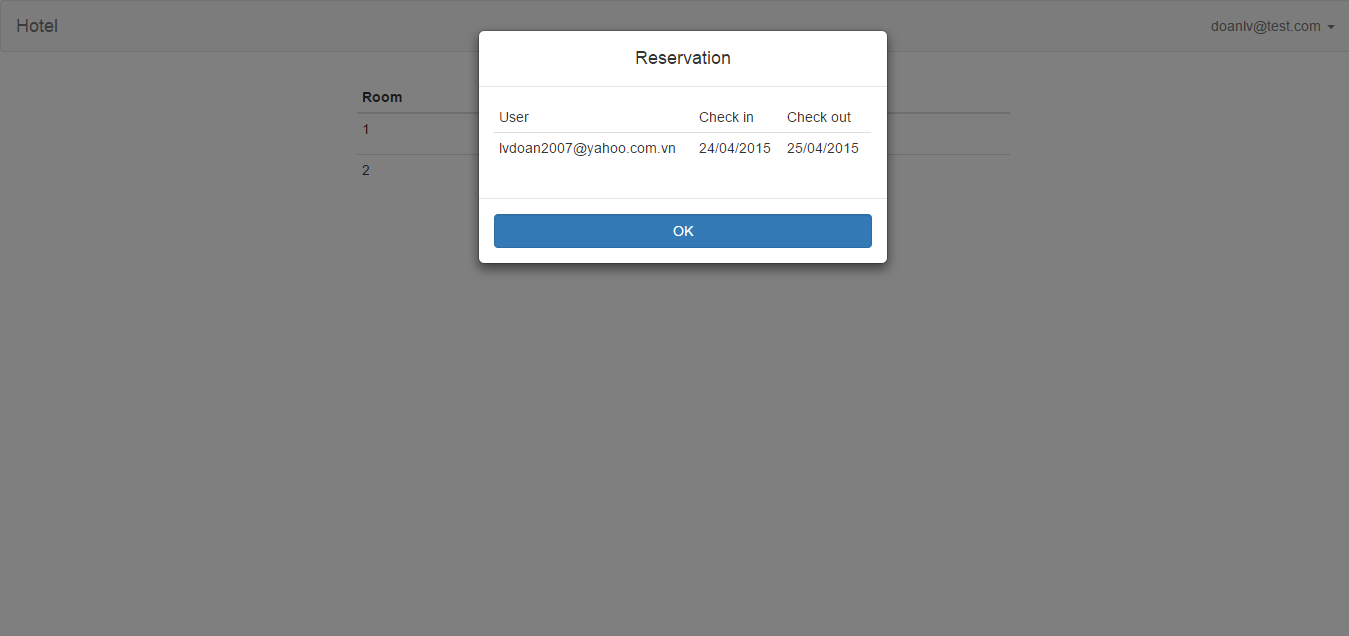
*Figure 100: Manage Key*

Click on button with key icon, small window will be shown with current static key of room.

You can change the static key of room by inputting new key to “New key” field and click Change key. The status message will be appeared on the main page.

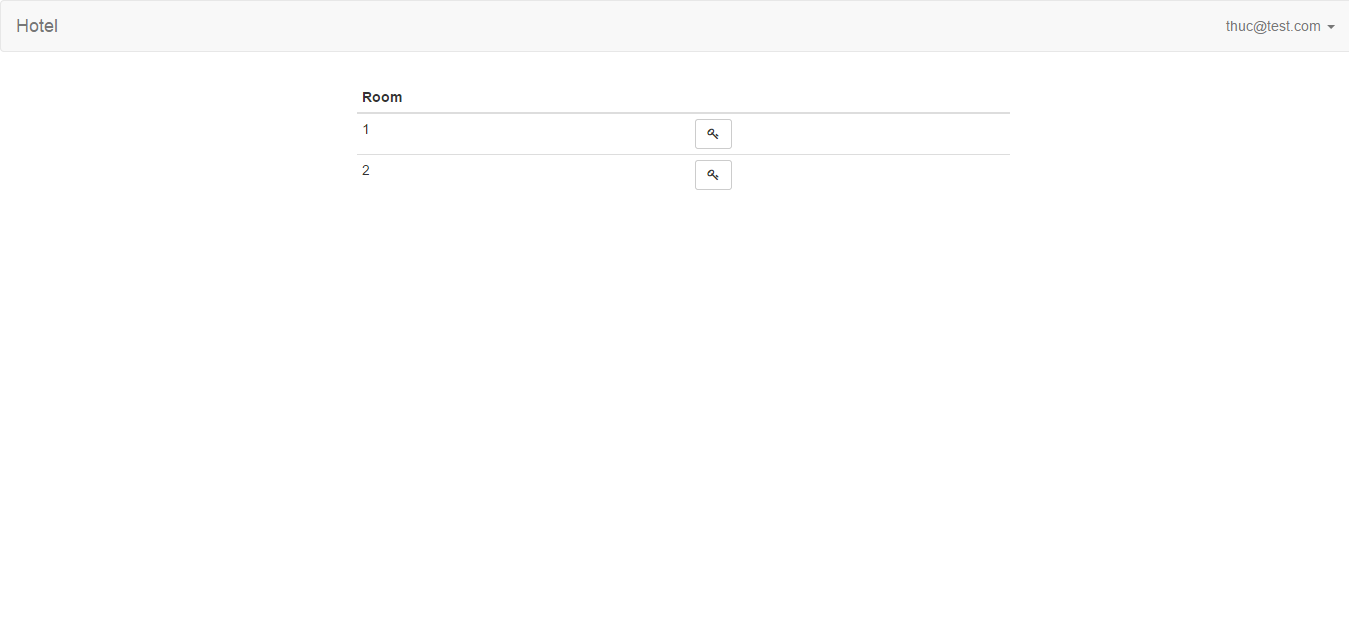
Click on button with I icon, another window will be appeared. This window have list of latest reservations of current room.

*Figure 101: Comeback Manage Page*



*Figure 102: Manage reservations*

#### 3.3 Staff

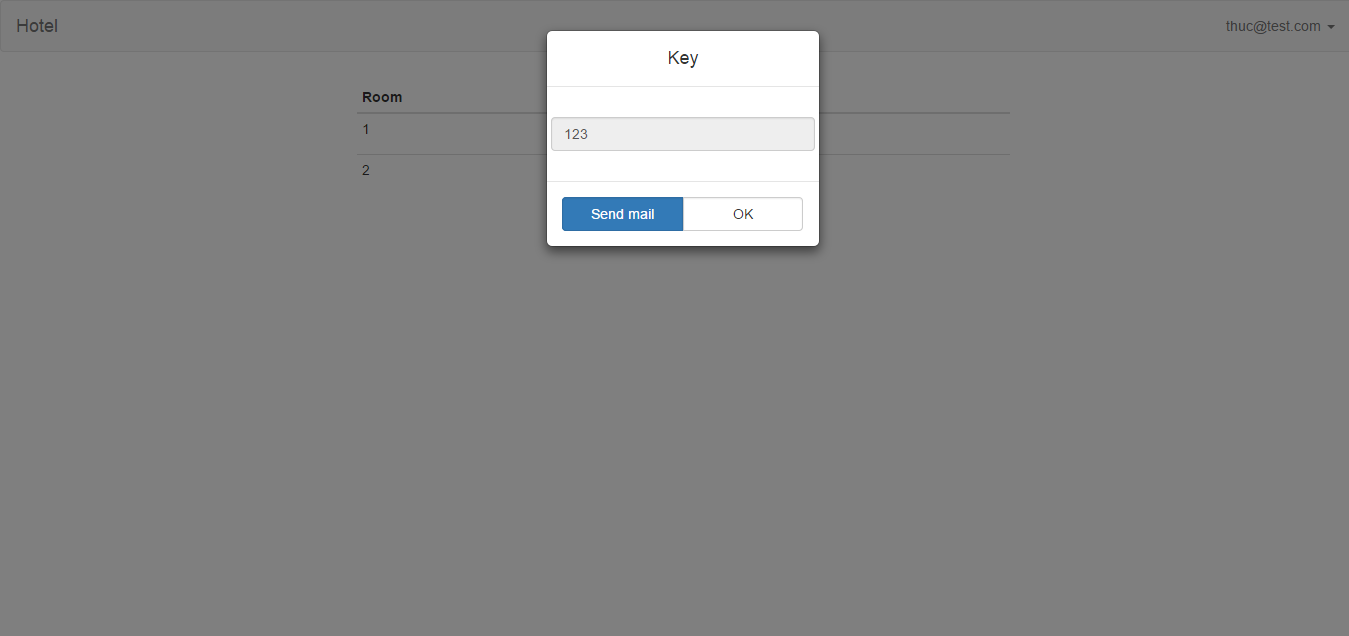


*Figure 103: Display Key*

If you login with role “staff”, this page will be shown.

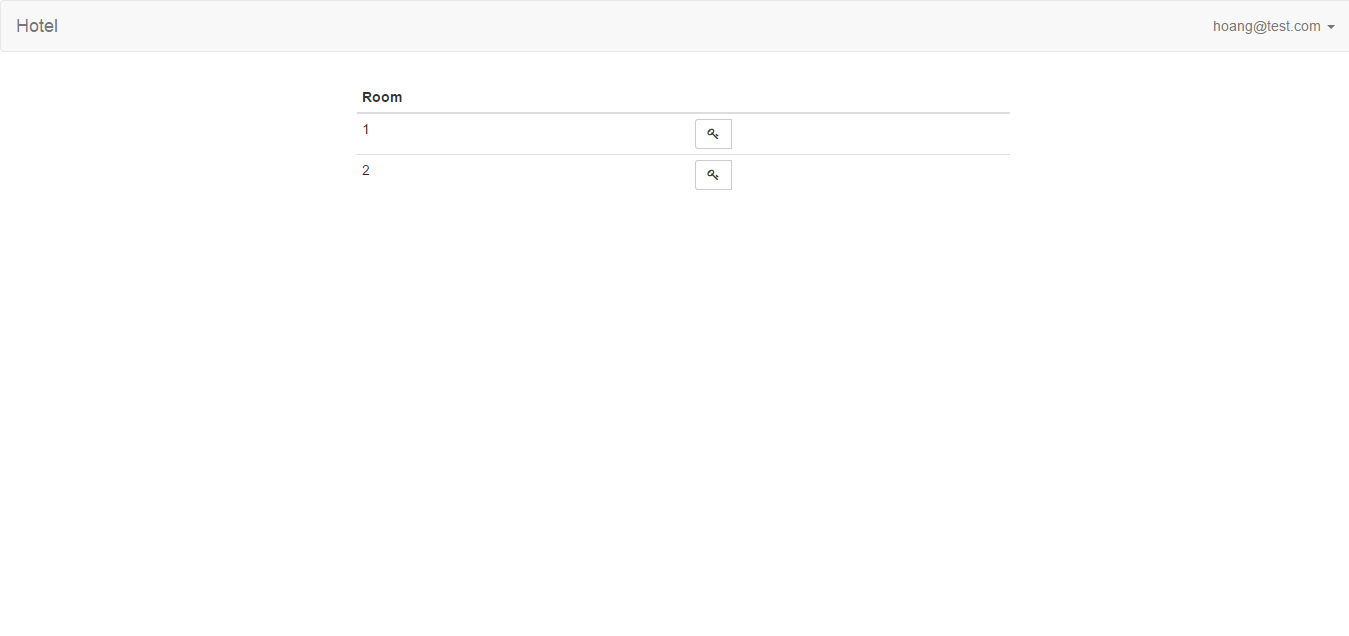
Click on button with key icon, static key is placed on field.

Send key button allows “staff” send to your own email a message containing QR code using to open the door.



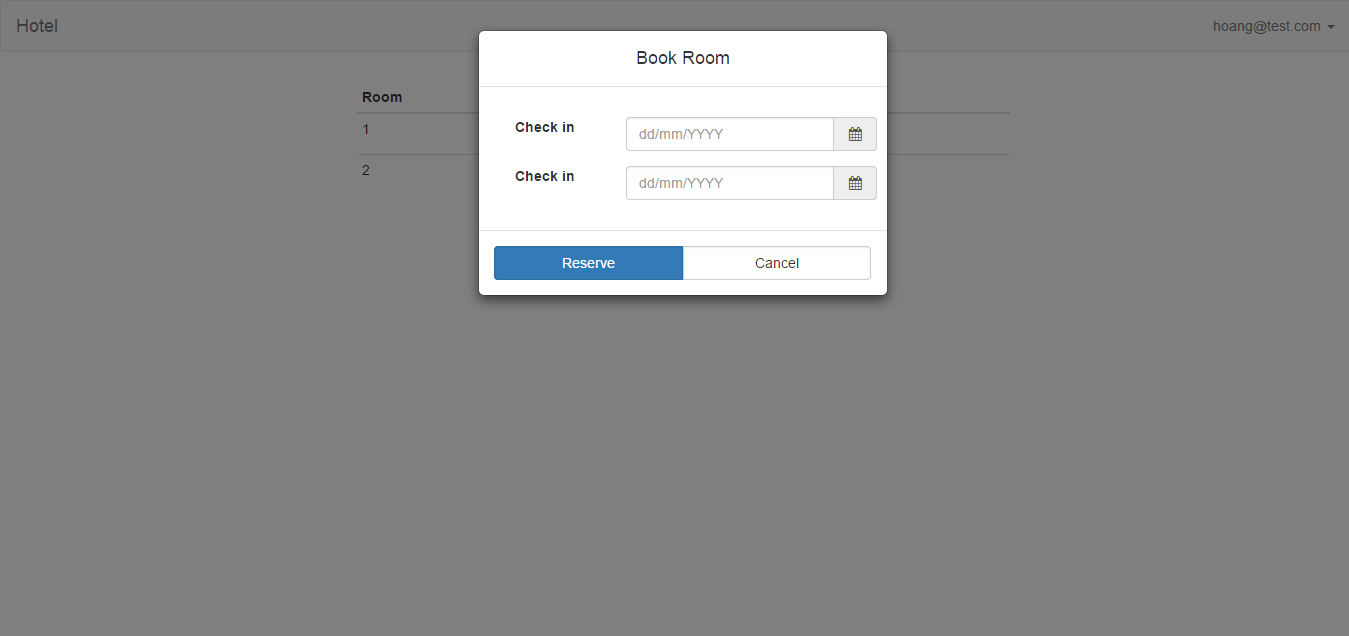
*Figure 104: Send key to email staff*

#### 3.4 Guest



*Figure 105: View room*

You are a guest. This page will help you book room. Click button on room’s row you need to reserve, a form will be shown and let you book room by fulfilling two fields “Check in” and

“Check out”, then click “Reserve” button.  


*Figure 106: Booking room*

# REFERENCES

<http://www.dientuchiase.com/2014/12/mach-chuyen-nguon-12v-khi-mat-ien.html>

<http://imgarcade.com/1/ir-receiver-circuit-using-photodiode/>

<http://elecrom.com/2008/02/19/how-to-make-simple-infrared-sensor-modules/>

<http://tme.vn/>

<https://www.wikipedia.org/>