

# **DIGITAL DOOR SECURITY AND CONTROL SYSTEM**

**Name: Muhammad Sohaib Furqan**

**Session: F – 12 - 2012-2016**

**Roll No:111126-BSCS-12**



**COMPUTER SCIENCE DEPARTMENT**

## **DIGITAL DOOR SECURITY AND CONTROL SYSTEM**

| <b>Registration No:</b> | <b>Name:</b>              | <b>Degree</b> |
|-------------------------|---------------------------|---------------|
| 111126                  | Muhammad Sohaib<br>Furqan | BSCS          |

### **SUPERVISED BY:**

Dr. Muhammad Sanaullah

(This thesis submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science AIR UNIVERSITY, MULTAN)

© Copyright Air University Multan Campus

All Rights Reserved.

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

In the Name of Allāh, the Most Gracious, the Most Merciful

|   |
|---|
| <b>AIR UNIVERSITY MULTAN CAMPUS</b>                 |
| <b>PROJECT SUPERVISOR &amp; TOPIC APPROVAL FORM</b> |

|   |                                |                         |
|---|--------------------------------|-------------------------|
| <b><u>Student Name:</u></b><br>Muhammad Sohaib Furqan   | <b>Reg. No. 111126</b>         |                         |
| <b>Degree</b>   | <b>Contact No. 03028639288</b> |                         |
| <b>Batch of Joining</b>   | BSCS                           |                         |
| <b>Batch of Joining</b>   |                                |                         |
| <b>Supervisor Name:</b> Dr. Muhammad Sanaullah<br><b><u>Thesis/Project Title:</u></b> Digital Door Security and Control System<br><b><u>Comments:</u></b> |                                |                         |
|   |                                |                         |
|   |                                |                         |
| <b><u>Approval by FYP Committee:</u></b>  |                                |                         |
| _____<br>Dr. Muhammad Sanaullah   | _____<br>FYP/PMO Coordinator   | _____<br>Faculty Member |
| _____<br><b>Head of Department</b>  | _____<br><b>Director</b>       |                         |

**AIR UNIVERSITY MULTAN CAMPUS**  
**Department of Computer Sciences and Engineering**  
**PROJECT PROPOSAL SUBMISSION FORM**

|                             |                        |                 |        |             |      |
|-----------------------------|------------------------|-----------------|--------|-------------|------|
| <b>Personal Information</b> |                        |                 |        |             |      |
| <b>Name</b>                 | Muhammad Sohaib Furqan | <b>Reg. No.</b> | 111126 | <b>CGPA</b> | 2.76 |
| <b>Degree</b>               | BSCS                   |                 |        |             |      |
| <b>Batch of Joining</b>     | BSCS-Fall-2012         |                 |        |             |      |

|   |                        |
|---|------------------------|
| <b>Project Complete</b>   |                        |
| <b>On Deadline:</b>   |                        |
| <b><u>Research Project Information:</u></b>   |                        |
| <b><u>Project Title:</u></b> Digital Door Security and Control System   |                        |
| <b><u>Problem Statement:</u></b> To provide an affordable security solution to households   |                        |
| <b><u>Project Development:</u></b> Air University Multan Campus   |                        |
| <b><u>Supervisor Name:</u></b>  | Dr. Muhammad Sanaullah |
| <b><u>Supervisor Remarks:</u></b>   | _____                  |
|   | _____                  |
|   | _____                  |
| <b><u>Head of Department's Approval</u></b><br><div style="display: flex; justify-content: center; align-items: center; gap: 20px;"> <div style="border: 1px solid black; padding: 5px 10px;">Approved</div> <div style="font-size: 2em;">/</div> <div style="border: 1px solid black; padding: 5px 10px;">Not Approved</div> </div><br><b><u>Signature:</u></b> _____<br><br><b><u>Date:</u></b> _____ |                        |



**Department of Computer Science and Engineering**

**BSCS PROJECT AND THESIS DEFENSE APPROVAL SHEET**

The undersigned certify that they have read the following thesis, examined the defense, and are satisfied with the overall exam performance, and recommend thesis and project to Department of Computer Sciences and Engineering.

**Project Title:**

Digital Door Security and Control System

**Submitted by:** Muhammad Sohaib Furqan

**Registration No.:** 111126

**BS Computer Science**

Degree in Full (BS Computer Science)

**Computer Science**

Name of Discipline

**Dr. Muhammad Sanaullah**

Name of Supervisor

\_\_\_\_\_  
Signature of Supervisor

\_\_\_\_\_  
Name of External Examiner

\_\_\_\_\_  
Signature of External Examiner

**Dr. Amanullah Khan**

Name of Head of Department

\_\_\_\_\_  
Signature of Head of Department

Date: \_\_\_\_\_

# Attestation of Authorship

I, Muhammad Sohaib Furqan, Roll Number 111126, Student of Bachelor of Science in the subject of Computer Science Session Fall-2012, hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person except that which appears in the citations and acknowledgements. Nor does it contain material, which to a substantial extent I have submitted for fulfilling requirements of any other degree in any other university or other institute of higher studies.

Dated \_\_\_\_\_

Signature of the student \_\_\_\_\_

**Name:** Muhammad Sohaib Furqan

**Program:** BS Computer Science

## **RESEARCH COMPLETION CERTIFICATE**

It is certified that the thesis entitled: “Digital Door Security and Control System”

Registration No. 111126 under my supervision. It is also certified that the thesis is based on original Project/Research work and meet all criteria and standards laid down for BS Computer Science Degree.

The following areas have been critically monitored:-

- Conformance of IEEE format.
- Precision and correctness of the language.
- Literature Review is relevant and comprehensive.
- Relevance of reference with the text.
- Methodology is appropriate.

---

Dated

---

Signature

---

Supervisor Name and Designation

---

PMO, Department of Computer  
Science, Air University Multan  
Campus



## Dedications

*I dedicate this work to my parents, wife, sisters, teachers and friends and to everyone who contributed in making me better and stronger and helped me with the different storms of life.*

## **ACKNOWLEDGEMENTS**

First and foremost, I would like to express heartfelt gratitude to my supervisor, Dr. Muhammad Sanaullah, without whose continuous support, efforts and encouragement, this work would not have been possible. His untiring guidance and attempts to improve my work has definitely contributed a lot in my development as an individual. Thank you sir for all the times you have tolerated even my most annoying questions and being humble and kind enough to take me step by step through the steep learning curve. Your contributions will never be forgotten.

Apart from my supervisor, I would thank the very honorable Head of Department Dr. Aman Ullah Khan and all other teachers who have worked on me through the four years of my stay at this prestigious institute. Ms. Ayesha Afzal, Dr. Abdul Mueed, Dr. Mehdi Hassan, Mr. Ahmad Mohsin, Mr Falak Nawaz and Mr. Muzaffar Hameed are all worthy of a mention. Thank you for guiding me along every step and enabling me to transform into a better professional and individual.

I would also like to acknowledge with utmost sincerity and thankfulness the never-ending belief of my father, mother and the entire family. Thank you dad for being a role-model and an inspiration and for bearing with me in the toughest of times. I would not be writing this dissertation had you not believed in me. Thank you mama for the countless times you tolerated my short-temper and childishness, for helping me out when I felt like giving up everything and for all the restless days and sleepless nights you had to endure for my betterment. Both of you have made me believe that physical disability doesn't count and can be overcome with mental toughness and self-belief. I hope I can make you proud one day, God be willing. A big thank you to my wife, who encouraged me with this work whenever I was stuck and pushed me on. Also, to all three of my sisters whose jokes and quarrels provided me with an opportunity to get my mind off the busy schedules and tight deadlines.

Last but definitely not the least, I would like to thank all my classmates for all the cooperation. All the fun we had in the last four years will be remembered. Thank you for all the memories. I hope we stay connected always. I would specifically like to thank Sarib Niazi and Naeem Khawaja for their continuous support and the sincerest friendship there can be. I would not have completed my degree with ease had you two not supported me and stuck with me in times of need. You equally share my success and I pray for you to progress by leaps and bounds. Ameen!

# Table of Contents

|   |              |
|---|--------------|
| <b>RESEARCH COMPLETION CERTIFICATE .....</b>  | <b>viii</b>  |
| <b>Dedications.....</b>                       | <b>ix</b>    |
| <b>ACKNOWLEDGEMENTS.....</b>                  | <b>x</b>     |
| <b>List of Tables .....</b>                   | <b>xvi</b>   |
| <b>List of Figures.....</b>                   | <b>xvi</b>   |
| <b>Abstract.....</b>                          | <b>xviii</b> |
| <b>Chapter 1 .....</b>                        | <b>1</b>     |
| <b>1.0 Introduction:.....</b>                 | <b>2</b>     |
| <b>1.1 Objectives: .....</b>                  | <b>3</b>     |
| <b>1.2 Scope: .....</b>                       | <b>3</b>     |
| <b>Check In Module .....</b>                  | <b>4</b>     |
| <b>Check Out Module .....</b>                 | <b>4</b>     |
| <b>Administrator Module.....</b>              | <b>4</b>     |
| <b>1.3 Literature Review: .....</b>           | <b>4</b>     |
| <b>1.4 Problem Statement: .....</b>           | <b>6</b>     |
| <b>1.5 Summary: .....</b>                     | <b>6</b>     |
| <b>Chapter 2 .....</b>                        | <b>7</b>     |
| <b>2.0 Introduction:.....</b>                 | <b>8</b>     |
| <b>Project Deliverables:.....</b>             | <b>8</b>     |
| <b>Schedule and Budget Summary:.....</b>      | <b>8</b>     |
| <b>Budget Summary: .....</b>                  | <b>8</b>     |
| <b>Tentative Schedule:.....</b>               | <b>9</b>     |
| <b>2.1 Evolution of the plan:.....</b>        | <b>9</b>     |
| <b>2.2.1 Project Organization.....</b>        | <b>10</b>    |
| <b>External interfaces .....</b>              | <b>10</b>    |
| <b>Internal interfaces .....</b>              | <b>10</b>    |
| <b>Roles and Responsibilities: .....</b>      | <b>10</b>    |
| <b>2.2.2 Managerial Process Plans:.....</b>   | <b>10</b>    |
| <b>2.2.2.1 Work Plan:.....</b>                | <b>10</b>    |
| <b>2.2.2.2 Resource Allocation: .....</b>     | <b>11</b>    |
| <b>2.2.2.3 Budget Allocation:.....</b>        | <b>11</b>    |
| <b>2.3 Control Plan.....</b>                  | <b>11</b>    |
| <b>2.3.1 Requirements Control Plan: .....</b> | <b>11</b>    |

|   |   |    |
|---|---|----|
| 2.3.2   | Schedule Control Plan: .....            | 11 |
| 2.3.3   | Budget Control Plan: .....              | 11 |
| 2.3.4   | Quality Control Plan:.....              | 12 |
| 2.3.5   | Reporting Plan: .....                   | 12 |
| 2.3.6   | Metrics Collection Plan: .....          | 12 |
| 2.4   | Risk Management Plan: .....             | 12 |
| 2.5   | Closeout Plan:.....                     | 12 |
| 2.6   | Technical Process:.....                 | 13 |
| 2.6.1   | Process Model:.....                     | 13 |
| 2.6.2   | Methods, tools and techniques: .....    | 14 |
| 2.6.3   | Infrastructure Plan: .....              | 14 |
| 2.6.4   | Product Acceptance Plan: .....          | 15 |
| 2.7   | Support Process Plans: .....            | 15 |
| 2.7.1   | Configuration Management Plan: .....    | 15 |
| 2.7.2   | Validation and Verification Plan: ..... | 15 |
| 2.7.3   | Documentation Plan: .....               | 15 |
| 2.7.4   | Quality Assurance Plan: .....           | 15 |
| 2.7.5   | Review and Audits: .....                | 15 |
| 2.7.6   | Problem Resolution Plan:.....           | 16 |
| 2.7.7   | Subcontractor Management Plan:.....     | 16 |
| 2.7.8   | Process Improvement Plan: .....         | 16 |
| Summary:  | .....                                   | 16 |
| Chapter 3                                       | .....                                   | 17 |
| 3.0   | Introduction:.....                      | 18 |
| 3.1   | Business Requirements:.....             | 18 |
| 3.2   | User Requirements: .....                | 19 |
| Administrator Module:                           | .....                                   | 19 |
| Check-In Module:                                | .....                                   | 19 |
| Check-Out Module:                               | .....                                   | 19 |
| 3.3   | Functional Requirements:.....           | 19 |
| FR01: Administrator Login.....                  | 19                                      |    |
| FR02: Add or Edit users.....                    | 20                                      |    |
| FR03: View Logs and Generate Reports.....       | 20                                      |    |
| FR04: Edit User Privileges .....                | 20                                      |    |
| FR05: Set users as ‘active’ or ‘inactive’ ..... | 20                                      |    |
| FR06: Reactivate the System.....                | 21                                      |    |

|  |           |
|--|-----------|
| FR07: Allow Registered Users to Check In to the System.....  | 21        |
| FR08: Allow Registered Users to Check Out of the System..... | 21        |
| <b>3.4 Non-Functional Requirements:.....</b>                 | <b>21</b> |
| NFR01: Performance.....                                      | 21        |
| NFR02: Security .....  | 21        |
| NFR03: Availability .....                                    | 22        |
| NFR04: Documentation .....                                   | 22        |
| NFR05: Disaster Management.....                              | 22        |
| NFR06: Usability .....                                       | 22        |
| NFR07: Defects Maintenance.....                              | 22        |
| <b>3.5 Assumptions and constraints: .....</b>                | <b>22</b> |
| Development Language and tools:.....                         | 22        |
| Operating System:.....                                       | 23        |
| Browser Support: .....                                       | 23        |
| <b>3.6 Actors: .....</b>                                     | <b>23</b> |
| Administrator:.....  | 23        |
| Regular Users .....  | 23        |
| <b>3.7 Use-Cases: .....</b>                                  | <b>23</b> |
| Use-Case diagram: .....                                      | 25        |
| Descriptive Use-Cases:.....                                  | 26        |
| UC01-Check-In/Check-Out.....                                 | 26        |
| UC02-Unlock door: .....                                      | 26        |
| UC03-Lock door:.....   | 27        |
| UC04- Log in to the system's admin panel .....               | 28        |
| UC05- Edit User Details.....                                 | 28        |
| UC06- Add users: .....                                       | 29        |
| UC07- Edit users: .....                                      | 30        |
| UC08- Edit administrative rights for user: .....             | 30        |
| UC09- View Reports: .....                                    | 31        |
| UC10- Reactivate the system.....                             | 32        |
| <b>Chapter 4 .....</b>                                       | <b>33</b> |
| <b>4.0 Introduction .....</b>                                | <b>34</b> |
| Test Case 01 .....   | 34        |
| Test Case 02 .....   | 35        |
| Test Case 03 .....   | 36        |
| Test Case 04 .....   | 37        |

|  |    |
|--|----|
| Test Case 05 .....   | 39 |
| Test Case 06 .....   | 40 |
| Test Case execution .....                                      | 41 |
| Post-Conditions: .....   | 42 |
| Test Case 07 .....   | 42 |
| Post-Conditions: .....   | 43 |
| Chapter 5 .....  | 44 |
| 5.0 Introduction:.....   | 45 |
| 5.1 Administrator Login:.....                                  | 45 |
| 5.2 Traceability Matrix:.....                                  | 46 |
| FR01: Administrator Login .....                                | 46 |
| FR02: Add or Edit Users .....                                  | 46 |
| FR03: View Logs and Generate Reports.....                      | 47 |
| FR04: Edit User Privileges .....                               | 48 |
| FR05: Set Users as ‘Active’ or ‘Inactive’ .....                | 49 |
| FR06: Reactivate the System.....                               | 50 |
| FR07: Allow Registered Users to Check-In To the System .....   | 51 |
| FR08: Allow Registered Users to Check Out From the System..... | 51 |
| 5.3 Entity Relationship Diagram: .....                         | 52 |
| 5.4 Graphical User Interface:.....                             | 53 |
| 5.5 Design Goals: .....  | 56 |
| ➤ Performance: .....   | 56 |
| ➤ Security: .....  | 56 |
| ➤ Usability: .....   | 57 |
| ➤ Maintainability:.....  | 57 |
| ➤ Economical: .....  | 57 |
| Summary: .....   | 57 |
| Chapter 6 .....  | 58 |
| 6.0 Introduction:.....   | 59 |
| 6.1 App.config.....  | 59 |
| 6.2 Program.cs .....   | 59 |
| Glossary: .....  | 62 |
| References:.....   | 63 |

## **List of Tables**

|  |          |
|--|----------|
| <b>TABLE 1: BUDGET SUMMARY .....</b>       | <b>8</b> |
| <b>TABLE 2: DELIVERABLE SCHEDULE .....</b> | <b>9</b> |

## **List of Figures**

|  |           |
|--|-----------|
| <b>FIGURE 1 CHAPTER 2 WORK PLAN .....</b>                  | <b>10</b> |
| <b>FIGURE 2 CHAPTER 2 INCREMENTAL MODEL.....</b>           | <b>13</b> |
| <b>FIGURE 3 CHAPTER 5 LOGIN USE CASE .....</b>             | <b>45</b> |
| <b>FIGURE 4CHAPTER 5: ENTITY RELATIONSHIP DIAGRAM.....</b> | <b>52</b> |
| <b>FIGURE 6 GUI 1 CHECK-IN.....</b>                        | <b>53</b> |
| <b>FIGURE 7 GUI 2 CHECK-OUT .....</b>                      | <b>53</b> |
| <b>FIGURE 8 GUI 3 ADMIN PANEL LOGIN FORM.....</b>          | <b>54</b> |
| <b>FIGURE 9 GUI 4 SELECT OPERATION .....</b>               | <b>54</b> |
| <b>FIGURE 10 GUI 5 USERS MANAGEMENT .....</b>              | <b>55</b> |
| <b>FIGURE 11 GUI 6 SELECT REPORT .....</b>                 | <b>56</b> |





## **Abstract**

Digital Door Security and Control System is developed for use as a domestic security system, primarily for main entrances to homes. It consists of two views, the user view and the admin view. The user view is split into two sub-modules, Check In and Check Out. The motivation behind this project is to provide an inexpensive security solution to people so that they can have control and surveillance over who enters or exits their house, and to provide complete record of all the instances of entry and exit. Administrators can add other administrators, or other categories of users such as family members or workers and can view logs and generate reports. Registered users can check in within the time they have been allocated. The system captures a hidden image of the user at the time of check in to enhance security. Moreover, administrators also have the option to make a user inactive so that the system doesn't recognize them, thus making sure unauthorized users can't enter the household.

# **Chapter 1**

## **Introduction**

## **1.0 Introduction:**

In this era of Information Technology, security has become an entity of central focus. With the increasing violence and crime rate, it is essential that even residences have adequate security measures in place. It is impossible in today's day and age of schedules and deadlines to constantly sit in one place, and people frequently leave their houses unattended. In under-developed countries such as Pakistan, practices such as these can lead to immediate chaos and confusion and increased threat. Moreover, even within a home, only selected people must have the authority to access the details of entrance and exit to and from the house. Any given home has a different set of users must be classified accordingly. Digital Door Security and Control System provides a software-hardware solution by using Arduino micro-controller to control entry and exit to the home supported with an admin Panel which enables authorized users (administrators) to register other users, change details of already registered users, or make a user inactive.

Currently, three types of users can be registered with the system: administrators, family members and workers. Each registered user is assigned allowed timings and can check in to the system only within the allowed timeframe. If a user tries to log in at a time other than the allowed time interval, entry is denied by the system for that user.

To enhance security, the system captures a hidden image of the user which is in front of the camera at the time of check-in. This captured image is currently visible neither to the user who is checking in nor to the administrators, but will be used when face recognition and matching features are incorporated in the product features in the future. The system also records the number of unsuccessful attempts each time a check-in operation is performed and halts itself after three invalid attempts so that any suspicious activity can be avoided. An administrator will then have to access the system's admin panel and re-enable the system before it can be used regularly again.

## **1.1 Objectives:**

The motivation behind the development of this project is to implement a cost-cutting, efficient and reliable solution for common household use so that users can stay in sync with the details of who is accessing their residence.

The main objective of Digital Door Security and Control System is to ensure time based access to the residence and provide complete logging and reporting functionality so that all details can be viewed by the administrators efficiently and conveniently. Administrators can also change existing user details or make a user inactive so that the system does not recognize the inactive user.

### **Key benefits of the system:**

- Affordable and reliable
- Time based control
- Reporting facilities to admin
- Audio based welcome message
- System automatically disabled after three invalid attempts
- Hidden image capture
- Easy operation and simple interface

## **1.2 Scope:**

The scope of this project is to allow household users to secure access to their homes through use of micro-controller technology.

The stakeholders for this project are administrators, family members and workers. The project has three views.

## **Check In Module**

- Users registered with the system can use their PIN to check in to the household
- Users need to enter PIN in the specified format or else input will not show on screen

## **Check Out Module**

- Users registered with the system can use their PIN to check out of the household
- Users need to enter PIN in the specified format or else input will not show on screen

## **Administrator Module**

- Users registered with the system can use their username and PIN to log in to the admin panel
- Users need to enter username and PIN in the specified format or else won't be logged in
- Administrators can add new users, assign timings and update existing user details
- Administrators can also mark users active or inactive
- If the system is halted, administrators can also use their privileges to re-enable the system

## **1.3 Literature Review:**

In order to gather ideas and get a notion of how products similar to the one discussed in this SRS have been designed in the past, I have reviewed many research papers, all of which have some common features discussed below:

***Security and usability Improvement on a digital door lock system based on Internet of Things, Ilkyu Ha, Kyungil University, Gyeongsan.***

***[http://www.sersc.org/journals/IJSIA/vol9\\_no8\\_2015/5.pdf](http://www.sersc.org/journals/IJSIA/vol9_no8_2015/5.pdf)***

This system can send recorded images to the user's mobile device when user attempts an illegal operation. It can also send alarm notification when the lock is physically damaged. The main features in this system include connection to mobile devices, key sharing Access Notification, image transfer, door opening and closing by speech recognition etc.

***Web-based online embedded door access control and home security system based on face-recognition, Mrutyunjaya Sahani, Chiranjiv Nanda, Abhijeet Kumar Sahu and Biswajeet Pattnaik, Department of Electrical and Electronics engineering, Sikha "O" Anusandhan University, Bhubaneswar, Odisha, India***

***[http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=7159473](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7159473)***

This system is a wireless access control system controlled by Raspberry Pi designed for a smart home environment. The system transfers the visitor's image through email or an alert SMS via GSM network.

This system consists of two components, Wireless Control Units (WCU) and a Wireless Information Unit (WIU) linked by a radio transceiver.

**Design and Implementation of a Door Lock Control Based on a Near Field Communication of a Smartphone, Chi-Huang Hung \*, Ying-Wen Bai, Je-Hong Ren  
Department of Electrical Engineering, Fu Jen Catholic University, New Taipei City, Taiwan Graduate Institute of Applied Science and Engineering \*, Fu Jen Catholic University, Taiwan Department of Information Technology \*, Lee-Ming Institute of Technology, New Taipei City, Taiwan**

***[http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=7216992](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7216992)***

This system uses the integration of a Near Field Communication (NFC) and a smartphone to achieve a door lock control system by means of a Logical Link Control Protocol (LLCP). The NFC uses Radio Frequency Identification Device technology to perform non-contact standard data exchange between two NFC devices.

## **1.4 Problem Statement:**

Although there are many security solutions available in the market in the modern era, most of them are being put to commercial use. There is a need of an affordable, easy to use, efficient and reliable security solution which can be utilized as a cost effective security solution for common, residential use. The system must be such that it takes into perspective the different types of home users, and prevents unauthorized access. The system must also allow editing of user details. There should also be a feature allowing administrators to render a user inactive if they do not want an existing user to be authenticated by the system any longer. The system should present administrators with well-defined reporting capabilities to help them view the entry/exit activities that take place within the household.

## **1.5 Summary:**

This section covers the introduction of the project “Digital door security and control system” which has been developed with special consideration to related work done in the past and the security needs of home users. Care has been taken to ensure that the project fulfills all the functional and non-functional requirements identified.



# **Chapter 2**

## **Planning and Methodology**

## 2.0 Introduction:

This chapter is related to project scheduling, describing the deliverables, their submission to the supervisor and the different methodologies adopted to achieve the tasks.

### Project Deliverables:

The list of project deliverables is:

- Project Management Plan
- Software Requirements Specification
- Software Design Description
- Software Quality Assurance Plan (including verification and validation plan and test design document)
- Working system with relational database design
- Final thesis document

### Schedule and Budget Summary:

#### Budget Summary:

| Item              | Price              |
|-------------------|--------------------|
| Arduino Mega 2560 | PKR 4500.00        |
| 4-port relay      | PKR 600.00         |
| 5V Motor          | PKR 50.00          |
| Solenoid Lock     | PKR 2200.00        |
| 12V adapter       | PKR 250.00         |
| Jumper wires      | PKR 50.00          |
| Breadboard        | PKR 150.00         |
| <b>Total</b>      | <b>PKR 7800.00</b> |

Table 1: Budget Summary

The above table lists the budget spent in gathering necessary equipment for development of the project. Approximate price of each item has been listed.

### **Tentative Schedule:**

| <b>Item</b>                                    | <b>Due Date</b>                 |
|--|---------------------------------|
| <b>Software Project Management Plan</b>        | 23 <sup>rd</sup> November, 2015 |
| <b>Software Requirements Specification 1.0</b> | 08 <sup>th</sup> December, 2015 |
| <b>Software Requirements Specification 2.0</b> | 15 <sup>th</sup> December, 2015 |
| <b>Software Requirements Specification 3.0</b> | 1 <sup>st</sup> January, 2016   |
| <b>Complete SRS</b>                            | 14 <sup>th</sup> January, 2016  |
| <b>Complete Database Design Document</b>       | 16 <sup>th</sup> March, 2016    |
| <b>Complete Test Case Document</b>             | 09 <sup>th</sup> May, 2016      |
| <b>Project Thesis Document</b>                 | 1 <sup>st</sup> June, 2016      |
| <b>Demo and Presentation</b>                   | 7 <sup>th</sup> June, 2016      |

**Table 2: Deliverable Schedule**

The above table lists the names and submission dates of the different deliverables constituting the project. Each of these deliverables were submitted to the supervisor on or before the due date and feedback was taken on the next meeting date.

## **2.1 Evolution of the plan:**

The preliminary draft of the thesis submitted to supervisor Dr. Muhammad Sanaullah and after approval, copies of the same distributed to members of the committee on the date as referred to in table 2.1

## 2.2 Project Plan:

### 2.2.1 Project Organization

The project organization defines the human infrastructure of the project. The task is designed to define the project organization chart, the roles, and the relationship of the project team.

### External interfaces

The external interfaces for the project would be members of the project committee.

### Internal interfaces

None

### Roles and Responsibilities:

The developer is responsible for all documentation developed and for all work done.

### 2.2.2 Managerial Process Plans:

#### 2.2.2.1 Work Plan:

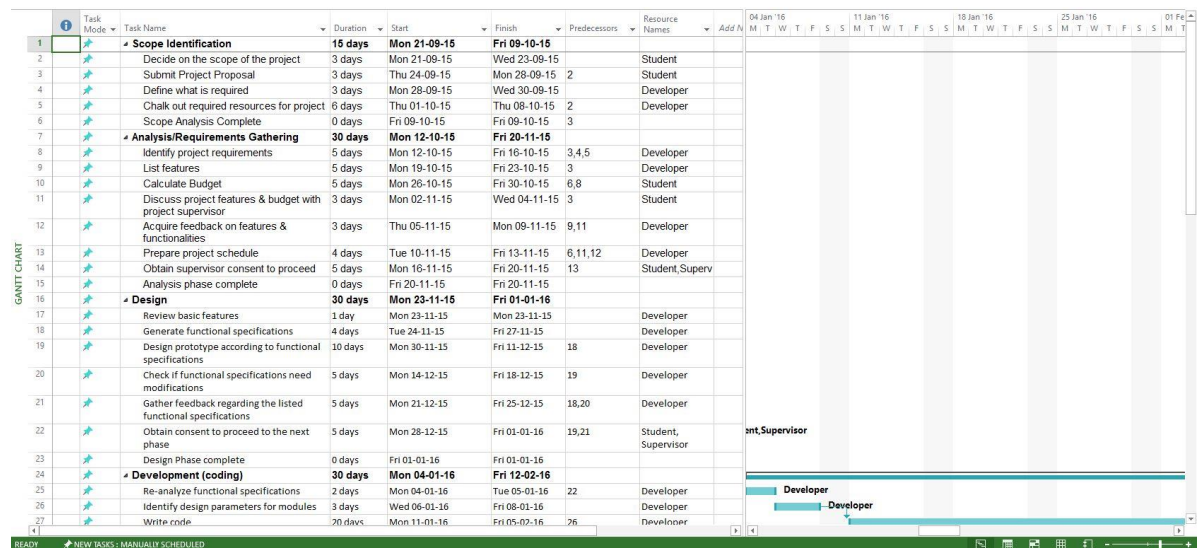


Figure 1 Chapter 2 Work Plan

The above image is a snapshot from the work plan that was developed at the beginning of the project. It is intended to provide the reader of the dissertation with a basic idea of the flow of the development process and the various activities involved and their duration.

#### **2.2.2.2 Resource Allocation:**

This project used resources in the form of time, effort and budget I have spent in preparing the project deliverables.

#### **2.2.2.3 Budget Allocation:**

Refer to table 1 above.

### **2.3 Control Plan**

#### **2.3.1 Requirements Control Plan:**

When changes are to be made in the requirements after the SRS has been released, the changes are brought into the attention of the committee and discussed. Any changes that are to be made, with the prior approval of the committee and only if feasible and permissible within the project constraints, and resources in terms of knowledge and skill of the developer required. Once the changes made to the SRS document, an updated version of the SRS released and circulated to the committee.

#### **2.3.2 Schedule Control Plan:**

To meet the deadlines, extra time was spent on the project. Most of the documentation phase was completed in phase 1 of the Final Year Project.

#### **2.3.3 Budget Control Plan:**

PKR 10,000 were allocated for the project at the start of the first phase of the Final Year Project. It was made sure that the project was completed within the budget allocated and the resources defined.

### **2.3.4 Quality Control Plan:**

To ensure quality of the product black box testing performed on final product.

### **2.3.5 Reporting Plan:**

Weekly meetings were held with the supervisor in which the deliverables assigned were reviewed and corrected and new deliverables were assigned.

### **2.3.6 Metrics Collection Plan:**

None.

## **2.4 Risk Management Plan:**

**Risk # 1** “Deficiency in the knowledge and understanding of the problem and its solution” indicates that the developer does not have proper understanding of the problem, due to which the quality of the project will be affected in terms of the requirements of the project and their fulfilment, which is not desirable. To overcome this, regular meetings with the supervisor will be conducted to help the developer in delivering an efficient and quality product.

**Risk # 2** “Lack of skills and knowledge of tools and devices needed for project development” which means that the developer does not have knowledge about the tools that will be used in the development process. In this case, the developer is expected to update his/her knowledge about the tools available and decide which one will be used for the project. The developer may consult the supervisor for this purpose.

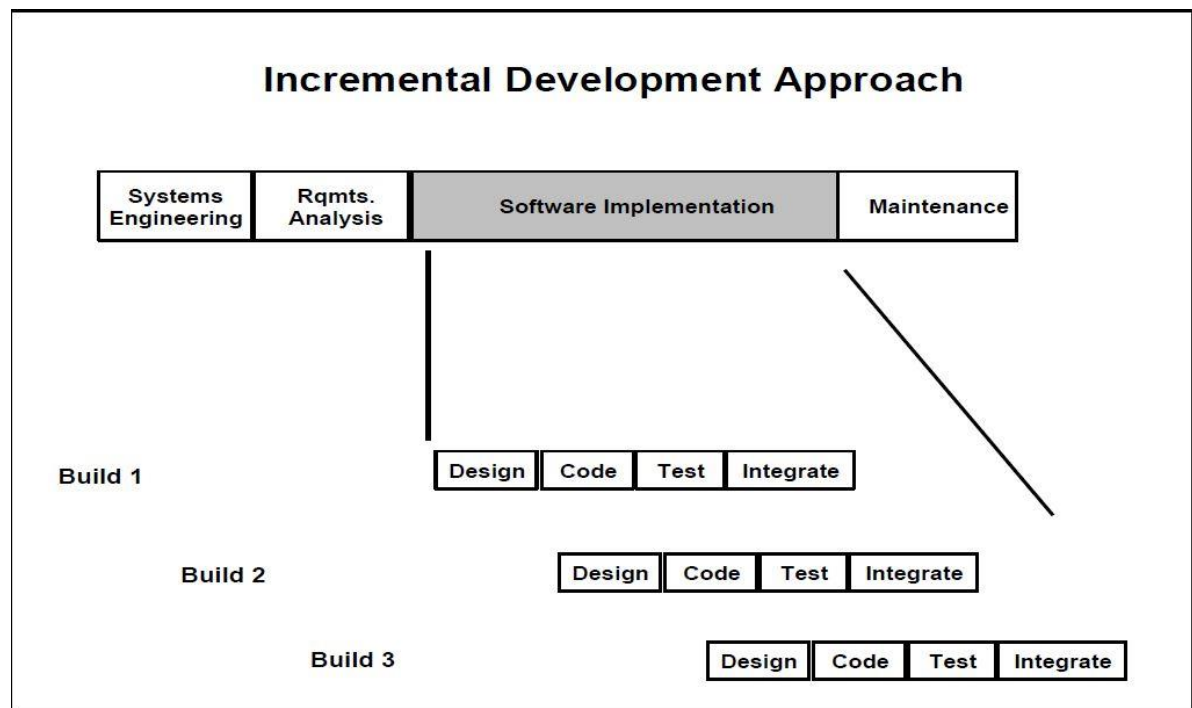
## **2.5 Closeout Plan:**

All the details about the briefings, report on the lessons learnt, project objectives and milestones achieved mentioned as part of the Final Year Project thesis.

## 2.6 Technical Process:

### 2.6.1 Process Model:

The system is built using the incremental model which is a variety of the waterfall process model. It is useful for projects such as this in which requirements change from time to time. It is shown in the figure below. Under this approach, the project is broken down into smaller deliverables called builds. The different software phases are repeated for each build. It also provides the opportunity to deliver limited functionality early in the project. Moreover, if the projects runs overtime and must be terminated before all the requirements implemented, only the functionality with low priority is sacrificed. For this reason, earlier builds are used to develop the more stable requirements and the later builds can be used for building the smaller requirements.



*Figure 2 Chapter 2 Incremental Model*

The figure shows the structure of the incremental process model of software engineering.

This product is split up into a number of deliverables, each of which is developed separately. This model helps ease the process of development through step by step approach rather than building a new product all at once.

### **Advantages:**

It is generally easier to test and debug than other methods of software development because relatively smaller changes are made during each iteration. This allows for more targeted and rigorous testing of each element within the overall product.

### **2.6.2 Methods, tools and techniques:**

Development Languages and Tools

#### **Languages:**

- C# .NET
- Microsoft T-SQL
- Arduino C

#### **Tools:**

- Microsoft Visual Studio C# 2015 Community
- Microsoft SQL server 2014
- Arduino compiler

### **2.6.3 Infrastructure Plan:**

The hardware resources for this system are personal computers with 1GB RAM, 250 GB Hard Drive and Dual Core or higher processor. System is developed on a PC with 4GB RAM, 500GB Hard Drive and i5 processor.



#### **2.6.4 Product Acceptance Plan:**

The product will be accepted after testing of the product/application by the approval committee.

### **2.7 Support Process Plans:**

#### **2.7.1 Configuration Management Plan:**

All project deliverables are to be considered as configuration items. The configuration item as well as its file named after the document followed by the version number. Updates to the same item will be numbered 1.1, 1.2 and the next full update will be numbered as 2.0.

#### **2.7.2 Validation and Verification Plan:**

A validation and verification plan as a part of Software Quality Assurance and Verification and Validation plan will be developed using recommended departmental standards.

It will be ensured that all standards are met in the best way possible.

#### **2.7.3 Documentation Plan:**

It were ensured that the entire documentation conforms to the IEEE recommended standards. All the documents discussed and reviewed by the advisor before issuance and distribution to the members of the committee on due dates.

#### **2.7.4 Quality Assurance Plan:**

For quality assurance of product, black box testing techniques adopted.

#### **2.7.5 Review and Audits:**

Review and Audits addressed as a part of the Software Quality Assurance and Verification & Validation Plan that developed by following recommended departmental standards.

### **2.7.6 Problem Resolution Plan:**

The problems resolved informally between the developer, the advisor and the project committee.

### **2.7.7 Subcontractor Management Plan:**

None

### **2.7.8 Process Improvement Plan:**

None.

## **Summary:**

A project plan lays down detailed information about the start and end of the different deliverables. This acts as a timetable, helping the developer to achieve goals set in an estimated time. Methodology is an important factor to describe the way in which goals will be achieved. Incremental process model is used due to the changing requirements and division into deliverables.

# **Chapter 3**

## **System Specification**

### **3.0 Introduction:**

For many, requirements engineering remains the core of the Software Development Life Cycle. Development any system implies that the requirements of the system be identified first, including system functionality and user expectations. Better understanding of the requirements will lead to better product quality and in turn better user satisfaction. This section lists in detail the requirements specified in the SRS document, with necessary updates according to the latest modifications in the project.

Properly designed SRS helps developers to achieve goals without hindrance. This section lists all the required information for development of the system. To make things simple, clear and concise, the process of SRS document development was divided into three deliverables.

The scope of the SRS is to specify requirements for the product being developed. This document will be confined to the identification of higher level business, user, functional and non-functional requirements, their descriptions, assumptions, and constraints applicable to the system. Use Cases, use case diagram, and their description will also be covered.

This document covers:

1. Business Requirements
2. User Requirements
3. Functional Requirements
4. Non-Functional Requirements
5. Use Cases, Diagram and Descriptions
6. Constraints and assumptions

### **3.1 Business Requirements:**

There is a need for security and authentication to be increased in homes, offices and other buildings. A security system is needed for this purpose which increases ease of use by

allowing the user to control access to residences. The system should be able to allow or deny entry to users based on correct or incorrect authentication respectively. System should allow user(s) to set or reset authentication credentials. The system should automatically take preventive measures in case a suspicious activity is anticipated.

## **3.2 User Requirements:**

### **Administrator Module:**

- Administrator login
- Add or edit users
- View logs and generate Reports
- Edit privileges for users
- Set users as 'active' or 'inactive'
- Reactivate the system after it is halted
- Allow registered users to check in to the system
- Allow registered users to check out from the system

### **Check-In Module:**

- Check in to the household

### **Check-Out Module:**

- Check Out of the household

## **3.3 Functional Requirements:**

### **FR01: Administrator Login**

|          |   |
|----------|---|
| FR01- 01 | System shall allow administrator to log in                                      |
| FR01- 02 | System shall get username from administrator                                    |
| FR01- 03 | System shall get password from administrator                                    |
| FR01- 04 | System shall authenticate administrator when he/she clicks on the submit button |

## **FR02: Add or Edit users**

|          |  |
|----------|--|
| FR02- 01 | System shall allow for adding new users or removing existing ones                                  |
| FR02- 02 | System shall require the Administrator login before removing existing users                        |
| FR02- 03 | System shall allow administrator to add or a new record or select an existing record from the grid |
| FR02- 04 | System shall allow user to save details for new user or update details for existing users          |
| FR02- 05 | System shall perform the action as indicated by the user   |

## **FR03: View Logs and Generate Reports**

|          |   |
|----------|---|
| FR03- 01 | System shall allow administrator to select the desired report to view                           |
| FR03- 02 | System shall generate the report selected by administrator                                      |
| FR03- 03 | System shall allow administrator to print the report, refresh the report or view another report |
| FR03- 04 | System shall display reports in a proper, uncluttered format                                    |

## **FR04: Edit User Privileges**

|          |   |
|----------|---|
| FR04- 01 | System shall allow the administrators to add or remove administrative rights on other users |
| FR04- 02 | System shall require the administrator to select the desired user                           |
| FR04- 03 | System shall allow the administrator to change category of the desired user                 |
| FR04- 04 | System shall allow administrator to click the update button                                 |
| FR04- 05 | System shall perform the desired action and enforce the specified changes                   |

## **FR05: Set users as ‘active’ or ‘inactive’**

|          |  |
|----------|--|
| FR05- 01 | System shall allow for marking existing users as active or inactive                                |
| FR05- 02 | System shall require the Administrator login before allowing to change status                      |
| FR05- 03 | System shall allow administrator to add or a new record or select an existing record from the grid |
| FR05- 04 | System shall display details of the selected user in the appropriate textboxes                     |
| FR05- 05 | System shall wait for user action  |
| FR05- 06 | System shall change PIN of user to 0000 (make inactive) if the delete button is pressed            |

### **FR06: Reactivate the System**

|          |  |
|----------|--|
| FR06- 01 | System shall disable keypad after three unsuccessful check-in/check-out attempts |
| FR06- 02 | System shall require administrator login once keypad is disabled                 |
| FR06- 03 | System shall give administrator the choice to reactivate system upon login       |
| FR06- 04 | System shall perform the desired action and enable the keypad                    |

### **FR07: Allow Registered Users to Check In to the System**

|          |   |
|----------|---|
| FR06- 01 | System shall require user to enter PIN                                      |
| FR06- 02 | System shall authenticate user once PIN is entered                          |
| FR06- 03 | System shall verify if user is checking in within the allotted timeframe    |
| FR06- 04 | System shall open lock and record time in case of successful authentication |

### **FR08: Allow Registered Users to Check Out of the System**

|          |   |
|----------|---|
| FR07- 01 | System shall require user to enter PIN                                      |
| FR07- 02 | System shall authenticate user once PIN is entered                          |
| FR07- 03 | System shall open lock and record time in case of successful authentication |

## **3.4 Non-Functional Requirements:**

### **NFR01: Performance**

|           |  |
|-----------|--|
| NFR01- 01 | The average start time of the entire system should be less than 60 seconds                       |
| NFR01- 02 | The mean time to failure for the system should not be more than 5 seconds within 24 hours of use |
| NFR01- 03 | The average system response time should be less than 5 seconds                                   |
| NFR01- 04 | Only one user should be able to access admin panel at a given time                               |

### **NFR02: Security**

|           |  |
|-----------|--|
| NFR02- 01 | System must provide access to only authenticated users who access the system admin panel                           |
| NFR02- 02 | System must not allow ANY user EXCEPT the designated user to add new users or remove existing ones                 |
| NFR02- 03 | System must not allow ANY user EXCEPT the designated user to add or remove administrator privileges on other users |

|           |  |
|-----------|--|
| NFR02- 04 | The system shall ensure that no data is saved to the computer at the end of the user session |
| NFR02- 05 | All access to the system shall be logged and a record shall be maintained                    |
| NFR02- 06 | All possible measures shall be taken to prevent hack attacks on the system in any form       |

### **NFR03: Availability**

|           |                                     |
|-----------|-------------------------------------|
| NFR03- 01 | The system should be available 24/7 |
|-----------|-------------------------------------|

### **NFR04: Documentation**

|           |   |
|-----------|---|
| NFR04- 01 | All functionalities of the system must be available to the user in the form of user documentation |
| NFR04- 02 | Online help should be available In case of any problem  |

### **NFR05: Disaster Management**

|           |  |
|-----------|--|
| NFR05- 01 | In case of crash the system should try to recover the maximum amount of data |
|-----------|--|

### **NFR06: Usability**

|           |   |
|-----------|---|
| NFR06- 01 | The system should be easy to use for a novice                           |
| NFR06-02  | The time required to get a hang of the overall system should be minimal |

### **NFR07: Defects Maintenance**

|           |   |
|-----------|---|
| NFR07- 01 | Post-release defects for the system must not exceed 3 critical bugs per month |
| NFR07- 02 | Fixing of post-release bugs should not take more than 24 hours                |

## **3.5 Assumptions and constraints:**

### **Development Language and tools:**

- Microcontroller: Arduino Mega (C)



- Mobile Platform: None
- Front End: Microsoft Visual Studio C#.NET
- Back End: Microsoft SQL Server 2014

### **Operating System:**

- Any modern windows OS will be supported

### **Browser Support:**

- None

## **3.6 Actors:**

The actors which are likely to interact with the system are listed below:

### **Administrator:**

The administrator will have full access to all the features and functionalities of the admin panel. The administrator will be one of the end users of the system.

A list of all the features and functionalities available to the administrator are listed in the user requirements section.

### **Regular Users**

Regular users of the system not having access to the admin panel. A list of all the features and functionalities available to the regular users are listed in the user requirements section.

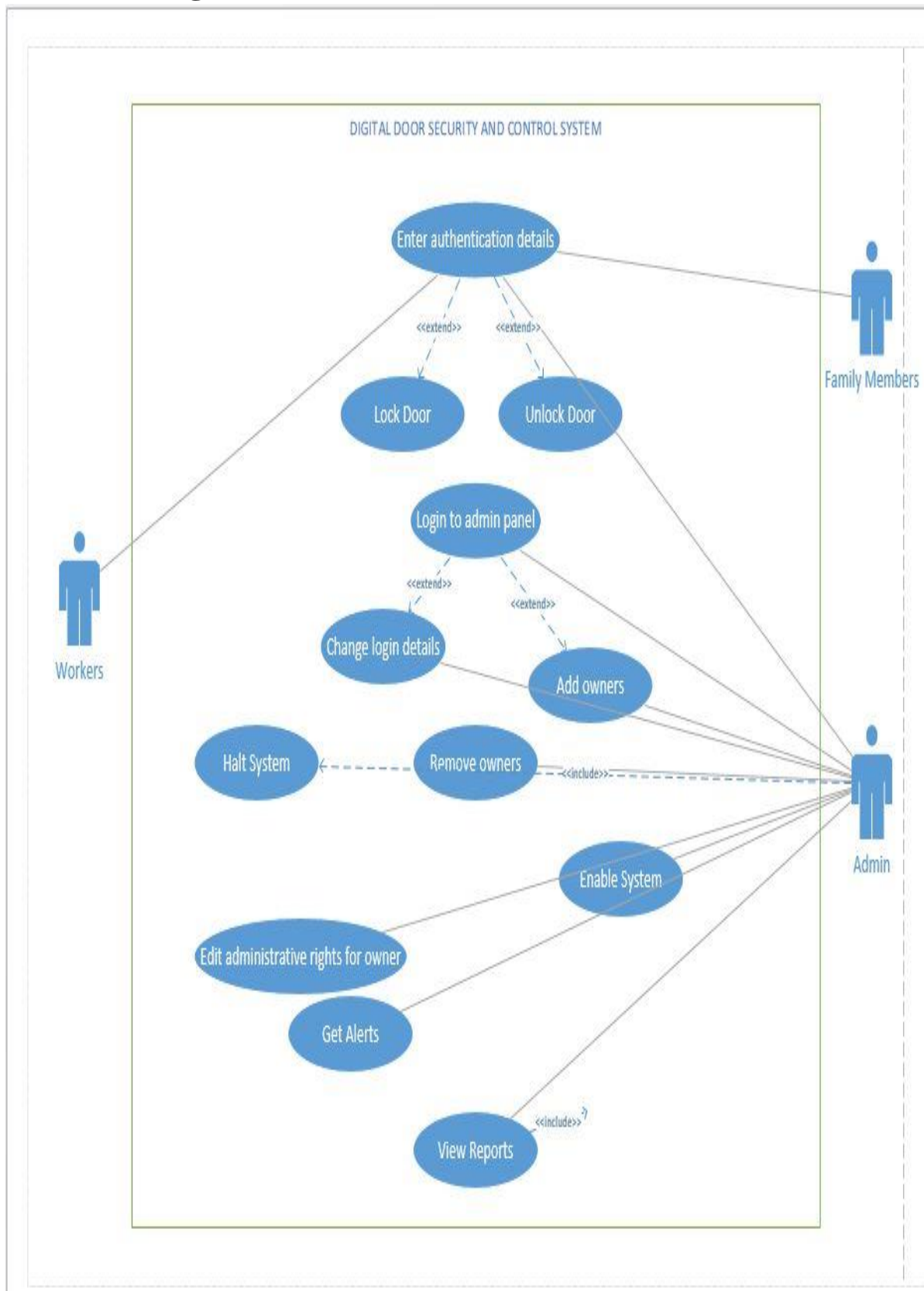
## **3.7 Use-Cases:**

The system will have the following use-cases:

- Check-In/ Check-Out
- Unlock Door
- Lock Door
- Log In To The System's Admin Panel

- Change User Details
- Add Users
- Edit Users
- Edit Administrative Rights For users
- View Reports
- Activate halted system

## Use-Case diagram:



## **Descriptive Use-Cases:**

### **UC01-Check-In/Check-Out**

#### **Brief Description:**

This use-case defines the process by which users/visitors will authenticate themselves to unlock the door on the entrance.

**Actors:** Visitor, Administrators

#### **Pre-Condition(s):**

1. User(s) must know the correct authentication details

#### **Main flow:**

1. Use-case starts when a user arrives at the entrance and wants to unlock the door
2. System shall provide an interface for the user to enter the PIN
3. User shall enter the PIN through keypad
4. System shall authenticate the entered PIN and allow entry to the user, thus unlocking the entrance
5. Use-case terminates with the user being able to successfully unlock the door and enter the premises

#### **Alternative flows:**

1. User enters incorrect PIN several times which results in the system being halted

#### **Exceptions:**

1. The system is down due to maintenance work and/or faulty behavior

#### **Post-Condition(s):**

1. The user successfully authenticates as valid

### **UC02-Unlock door:**

#### **Brief Description:**

This use-case defines the process by which the door is unlocked after the user enters authentication details

**Actors:** Visitor, user(s)

#### **Pre-Condition(s):**

1. The user must have entered the correct PIN into the authentication panel at the entrance

#### **Main flow:**

1. The use case begins when a user has been authenticated and identified as valid
2. System shall authenticate the PIN entered by the user at the authentication panel
3. System shall unlock the door and allow entry to the user
4. The use case terminates when the user is allowed to enter

**Alternative flows:**

1. The door unlocks but the user does not enter and goes back
2. The user enters the wrong authentication PIN and is asked to retry
3. The user enters the wrong PIN several times and the system halts

**Exceptions:**

1. The system is faulty or under maintenance

**Post-Condition(s):**

1. The door unlocks and the user successfully enters

**UC03-Lock door:**

**Brief Description:**

This use-case defines the process and the situations under which the door is unlocked after the user enters

**Actors:** Visitor, user(s)

**Pre-Condition(s):**

1. The user must have checked-in previously

**Main flow:**

1. The use-case begins when the user has checked-in or a certain time has elapsed
2. System shall wait for a certain time to elapse
3. If the user has entered, system must update the database to reflect the latest changes and lock the door
4. If the user has not entered, system must lock the door again after keeping it unlocked unless opened again
5. The use-case terminates when the door is locked again

**Alternative flows:**

1. The time is just about to elapse when system goes down

**Exceptions:**

1. System is down or under-maintenance

**Post-Condition(s):**

1. The door is locked again and the security reinforced

## **UC04- Log in to the system's admin panel**

### **Brief Description:**

This use-case defines the process by which an administrator logs in to the admin panel of the system

**Actors:** User(s)

### **Pre-Condition(s):**

1. The user must know the log in details for the admin panel

### **Main flow:**

1. The use- case begins when the user wants to access the admin panel of the system
2. User shall access the admin panel of the system through the provided button
3. The system shall wait for the user to enter login details for admin panel
4. System shall authenticate the login details entered by the user
5. System shall take user to the next form
6. The use-case terminates when the user is successfully logged in to the admin panel of the system

### **Alternative flows:**

1. User forgets admin panel login details

### **Exceptions:**

1. The entire system is down or under maintenance

### **Post-Condition(s):**

1. User successfully logs in to the admin panel of the system

## **UC05- Edit User Details**

### **Brief Description:**

This use case defines the process by which an administrator can edit details of existing users

**Actors:** administrator

### **Pre-Condition(s):**

Administrator must be logged into the admin panel

### **Main flow:**

1. The use-case begins with the administrator attempting to change log in credentials for admin panel
2. The system shall display users management form
3. The system shall allow administrator to select a user from the grid.
4. The system shall show the details for the selected user in the corresponding text boxes
5. The system shall wait for the administrator to update the details

6. The system shall put the newly changed user details into play
7. The use-case terminates when the user details have been successfully changed

**Alternative flows:**

1. The administrator logs into admin panel but does not do anything
2. The administrator selects a user from the grid but does not update anything

**Exceptions:**

1. The admin panel or the entire system is down

**Post-Condition(s):**

1. The log in details for the admin panel are changed successfully

**UC06- Add users:**

**Brief Description:**

This use case describes the process by which an administrator adds new users for the system

**Actors:** Administrator

**Pre-Condition(s):**

1. The administrator must be logged in to the system's admin panel

**Main flow:**

1. The use-case begins when the administrator attempts to add other users for the authentication system
2. The system shall open the user management form
3. The administrator shall wait for the administrator to fill in details for the new user
4. The administrator shall add new users when save button is clicked
5. System shall add the new entries to the list of existing users and update the grid to reflect the changes made
6. The use-case terminates when new users have been added

**Alternative flows:**

1. The administrator selects the option to add new users for admin panel and fills up the details but does not click the save button

**Exceptions:**

1. The admin panel or the entire system is down
2. The maximum number of users have already been added

**Post-Condition(s):**

1. The new users have successfully been added and can now access the admin panel

## **UC07- Edit users:**

### **Brief Description:**

This use case describes the process by which an administrator adds new users for the system

**Actors:** administrator

### **Pre-Condition(s):**

1. The administrator must be logged in to the system's admin panel

### **Main flow:**

1. The use-case begins when the administrator attempts to edit user details for the system
2. The system shall open the user management page
3. The administrator shall select the desired user from grid
4. The system shall populate the text boxes with the details for the selected user
5. The system shall wait for the admin to update the details
6. The system shall make the changes when the update button is clicked
7. The use-case terminates when the details of the selected users have been modified and updated

### **Alternative flows:**

1. The administrator selects the option to remove existing users for admin panel but does not do anything
2. The administrator selects the option to remove existing users for admin panel and selects them but does not click the next button

### **Exceptions:**

1. The admin panel or the entire system is down

### **Post-Condition(s):**

1. The selected users have successfully been removed from the system and cannot access the admin panel

## **UC08- Edit administrative rights for user:**

### **Brief Description:**

This use case describes the process by which an administrator edits administrative rights for other users for the system

**Actors:** administrator

### **Pre-Condition(s):**

1. The administrator must be logged in to the system's admin panel

### **Main flow:**



1. The use-case begins when the administrator attempts to change administrative rights for users of the system
2. The system shall open the user management form
3. The administrator shall select the desired user from the grid
4. The system shall populate text boxes with the details of the selected user
5. The system shall allow administrator to change administrative privileges for existing users
6. The administrator shall update category for selected users and click update
7. System shall update administrator privileges for the selected users
8. The use-case terminates when the administrator privileges have been updated for selected users

**Alternative flows:**

1. The administrator selects the user but does make any changes
2. The administrator makes the necessary changes but does not click the update button

**Exceptions:**

1. The admin panel or the entire system is down

**Post-Condition(s):**

1. The selected users have successfully been restricted as specified by the updated privileges

## **UC09- View Reports:**

**Brief Description:**

This use case describes the process by which an administrator views the various reports which the system can generate

**Actors:** administrator

**Pre-Condition(s):**

1. The administrator must be logged in to the system's admin panel
2. The database must reflect the latest changes made to the system

**Main flow:**

1. The use-case begins when the user attempts to view reports
2. The administrator shall select the option to view reports
3. The system shall display view reports page
4. The administrator shall select the desired report by clicking on it
5. The system shall generate desired report and show it to the user
6. The use-case terminates when the report has been displayed to the user

**Alternative flows:**

1. The administrator selects the report to view but no entries exist in the database

**Exceptions:**

1. The admin panel or the entire system is down

**Post-Condition(s):**

1. The selected report is displayed

**UC10- Reactivate the system****Brief Description:**

This use case describes the process by which an administrator reactivates a halted system

**Actors:** administrator

**Pre-Condition(s):**

1. The administrator must be logged in to the system's admin panel

**Main flow:**

1. The use-case begins when the administrator attempts to reactivate the system in halted state
2. The system shall disable the entire keypad except the admin panel button
3. The administrator shall click on the admin panel button
4. The system shall require administrator to log in
5. The administrator shall enter details and log in to the admin panel
6. System shall display reactivate system button to the administrator
7. The administrator shall click the button to reactivate the system
8. The use-case terminates when the system is reactivated and the keypad is enabled

**Alternative flows:**

1. The user logs in but doesn't reactivate the system

**Exceptions:**

1. The admin panel or the entire system is down
2. The system is up and working but the admin panel cannot be accessed due to connectivity issues

**Post-Condition(s):**

1. The system has been reactivated.

# **Chapter 4**

## **Test Cases**

## 4.0 Introduction

This section contains test cases generated against the major functionalities of the system. The test cases will be as follows:

### Test Case 01

|  |
|--|
| <b>Test Case ID:</b> tc_01   |
| <b>Test Priority (Low/Med/High):</b> High                          |
| <b>Module Name:</b> Authentication numeric keypad                  |
| <b>Test Title:</b> Verify successful authentication with valid PIN |
| <b>Description:</b> Test user authentication                       |
| <b>Use Case ID:</b> UC01   |
| <b>Functional Requirements Involved:</b> FR04, FR06                |

### Test Case execution:

| Step | Test Steps                                  | Test Data/<br>Inputs | Expected<br>Result                       | Actual<br>Result                         | Status<br>(Pass/Fail) | Notes |
|------|---|----------------------|--|--|-----------------------|-------|
| 1    | Start the application                       |                      |  |  |                       |       |
| 2    | Enter PIN<br>(Invalid)                      | *9999#               | Match Not<br>Found                       | Match Not<br>Found                       | Pass                  |       |
| 3    | Enter PIN<br>(Valid)                        | *5055#               | Door<br>Open                             | Door<br>Open                             | Pass                  |       |
| 4    | Enter PIN<br>in the<br>incorrect<br>Pattern | Xyz123               | Nothing<br>showing<br>on input<br>screen | Nothing<br>showing<br>on input<br>screen | Pass                  |       |

**Post-conditions:**

User is successfully authenticated and door is opened

**Test Case 02**

|   |
|---|
| <b>Test Case ID:</b> tc_02                                |
| <b>Test Priority (Low/Med/High):</b> High                 |
| <b>Module Name:</b> Admin Panel Login Screen              |
| <b>Test Title:</b> Verify successful Login to admin panel |
| <b>Description:</b> Test Admin Login                      |
| <b>Use Case ID:</b> UC01, UC04                            |
| <b>Functional Requirements Involved:</b> FR01             |

**Test Case execution:**

| Step | Test Steps  | Test Data/<br>Inputs | Expected<br>Result | Actual<br>Result | Status<br>(Pass/Fail) | Notes |
|------|---|----------------------|--------------------|------------------|-----------------------|-------|
| 1    | Click<br>Admin<br>Panel<br>button on<br>numeric<br>keypad |                      |                    |                  | Pass                  |       |
| 2    | Enter valid<br>username                                   | Xyz                  |                    |                  | Pass                  |       |
| 3    | Enter<br>Valid<br>Password                                | *5055#               |                    |                  | Pass                  |       |

|   |                          |        |                               |   |      |  |
|---|--------------------------|--------|-------------------------------|---|------|--|
| 4 | Click on login button    |        | Admin should be able to login | User is directed to admin panel operations screen | Pass |  |
| 5 | Provide Invalid Username | 123xyz | Error message                 | Error message shown to user                       | Pass |  |
| 6 | Provide Invalid PIN      | WWxx   | Error Message                 | Error message shown to user                       | Pass |  |

### Post-Conditions:

User is validated and successfully logged into admin panel

### Test Case 03

|   |
|---|
| <b>Test Case ID:</b> tc_03  |
| <b>Test Priority (Low/Med/High):</b> Low                                  |
| <b>Module Name:</b> Admin Panel Operations screen                         |
| <b>Test Title:</b> Verify operation is chosen successfully                |
| <b>Description:</b> Test operations screen for admin panel                |
| <b>Use Case ID:</b> UC07,UC08   |
| <b>Functional Requirements Involved:</b> FR02, FR03, FR04, FR07,FR08,FR09 |

**Test Case execution:**

| Step | Test Steps                           | Test Data/<br>Inputs | Expected<br>Result                                     | Actual<br>Result                                | Status<br>(Pass/Fail) | Notes |
|------|--------------------------------------|----------------------|--|---|-----------------------|-------|
| 1    | Login to<br>admin<br>panel           |                      |  |   | Pass                  |       |
| 2    | Choose an<br>operation<br>to perform |                      | User<br>should be<br>navigated<br>to desired<br>screen | The<br>desired<br>screen is<br>shown to<br>user | Pass                  |       |

**Post-Conditions:**

User is re-directed to desired screen

**Test Case 04**

|  |
|--|
| <b>Test Case ID:</b> tc_04                               |
| <b>Test Priority (Low/Med/High):</b> High                |
| <b>Module Name:</b> Add new user                         |
| <b>Test Title:</b> Verify new user is added successfully |
| <b>Description:</b> Successfully add new user            |
| <b>Use Case ID:</b> UC06                                 |
| <b>Functional Requirements Involved:</b> FR07            |

**Test Case execution:**

| Step | Test Steps                                   | Test Data/<br>Inputs | Expected<br>Result                               | Actual<br>Result                                    | Status<br>(Pass/Fail) | Notes |
|------|--|----------------------|--|---|-----------------------|-------|
| 1    | Login to<br>admin panel                      |                      |  |   | Pass                  |       |
| 2    | Navigate to<br>users<br>management<br>screen |                      |  |   | Pass                  |       |
| 3    | Enter name<br>of user                        | Jack                 |  |   | Pass                  |       |
| 4    | Enter<br>unique<br>username                  | Jackjack             |  |   | Pass                  |       |
| 5    | Enter valid<br>PIN                           | *2222#               |  |   | Pass                  |       |
| 6    | Choose user<br>category                      | Administrator        |  |   | Pass                  |       |
| 7    | Upload user<br>photo                         | Any image            | Image<br>should be<br>shown in<br>picture<br>box | Added<br>image is<br>displayed<br>in picture<br>box | Pass                  |       |
| 8    | Set Begin<br>time for user                   | 12:40:39             |  |   | Pass                  |       |
| 9    | Set end time<br>for user                     | 23:30:00             |  |   | Pass                  |       |



|    |                      |  |  |  |      |  |
|----|----------------------|--|--|--|------|--|
| 10 | Click the add button |  | New user should be added in the database | New user is added and user is informed | Pass |  |
|----|----------------------|--|--|--|------|--|

### Post-Conditions:

New user is registered with the system

### Test Case 05

|   |
|---|
| <b>Test Case ID:</b> tc_05                              |
| <b>Test Priority (Low/Med/High):</b> High               |
| <b>Module Name:</b> Update user record                  |
| <b>Test Title:</b> Verify user is updated successfully  |
| <b>Description:</b> Successfully update user record     |
| <b>Use Case ID:</b> UC05,UC08                           |
| <b>Functional Requirements Involved:</b> FR03,FR07,FR08 |

### Test Case execution:

| Step | Test Steps           | Test Data/<br>Inputs | Expected Result | Actual Result | Status (Pass/Fail) | Notes |
|------|----------------------|----------------------|-----------------|---------------|--------------------|-------|
| 1    | Login to admin panel |                      |                 |               | Pass               |       |
| 2    | Navigate to users    |                      |                 |               | Pass               |       |

|   |                           |  |  |  |      |                  |
|---|---------------------------|--|--|--|------|------------------|
|   | management screen         |  |  |  |      |                  |
| 3 | Select row from grid      |  | Fields will be populated with data from selected row | Data from selected row appears in the fields | Pass |                  |
| 4 | Update any values desired |  |  |  | Pass |                  |
| 5 | Click the update button   |  | Record should be updated in the database             | Program requires image path again            | Fail | Check code logic |

### Post-Conditions:

Existing record of the user is updated to reflect new values.

### Test Case 06

|   |
|---|
| <b>Test Case ID:</b> tc_06  |
| <b>Test Priority (Low/Med/High):</b> High   |
| <b>Module Name:</b> Delete user (Make inactive)                                     |
| <b>Test Title:</b> Verify user is deleted (status changed to inactive) successfully |
| <b>Description:</b> Successfully make selected user inactive                        |
| <b>Use Case ID:</b> UC10  |

|   |
|---|
| <b>Functional Requirements Involved: FR06</b> |
|---|

**Test Case execution:**

| Step | Test Steps                          | Test Data/<br>Inputs | Expected Result   | Actual Result   | Status (Pass/Fail) | Notes |
|------|-------------------------------------|----------------------|---|---|--------------------|-------|
| 1    | Login to admin panel                |                      |   |   | Pass               |       |
| 2    | Navigate to users management screen |                      |   |   | Pass               |       |
| 3    | Select row from grid                |                      | Fields will be populated with data from selected row    | Data from selected row appears in the fields                      | Pass               |       |
| 4    | Click the delete button             |                      | PIN should be changed to 0000 and user status should be | User is made inactive and cannot be authenticated with the system | Pass               |       |

|  |  |  |                        |  |  |  |
|--|--|--|------------------------|--|--|--|
|  |  |  | changed to<br>inactive |  |  |  |
|--|--|--|------------------------|--|--|--|

### Post-Conditions:

User is made inactive and cannot be authenticated with the system

### Test Case 07

|   |
|---|
| <b>Test Case ID:</b> tc_07  |
| <b>Test Priority (Low/Med/High):</b> High                             |
| <b>Module Name:</b> Generate reports                                  |
| <b>Test Title:</b> Generate reports                                   |
| <b>Description:</b> : Generate reports to show current database state |
| <b>Use Case ID:</b> UC14  |
| <b>Functional Requirements Involved:</b> FR10                         |

### Test Case execution:

| Step | Test Steps                                    | Test<br>Data/<br>Inputs | Expected<br>Result | Actual<br>Result | Status<br>(Pass/Fail) | Notes |
|------|---|-------------------------|--------------------|------------------|-----------------------|-------|
| 1    | Login to<br>admin panel                       |                         |                    |                  | Pass                  |       |
| 2    | Navigate to<br>report<br>generation<br>screen |                         |                    |                  | Pass                  |       |

|   |   |  |  |                                     |      |  |
|---|---|--|--|-------------------------------------|------|--|
| 3 | Select report<br>type from<br>menustrip on<br>top |  | Selected<br>report<br>should be<br>generated | A report is<br>shown on the<br>form | Pass |  |
|---|---|--|--|-------------------------------------|------|--|

**Post-Conditions:**

A crystal report reflecting the latest state of the database is generated and shown on the form

# **Chapter 5**

## **System Design**

## 5.0 Introduction:

This chapter will cover the design phase in order to understand the flow of development process before coding.

## 5.1 Administrator Login:

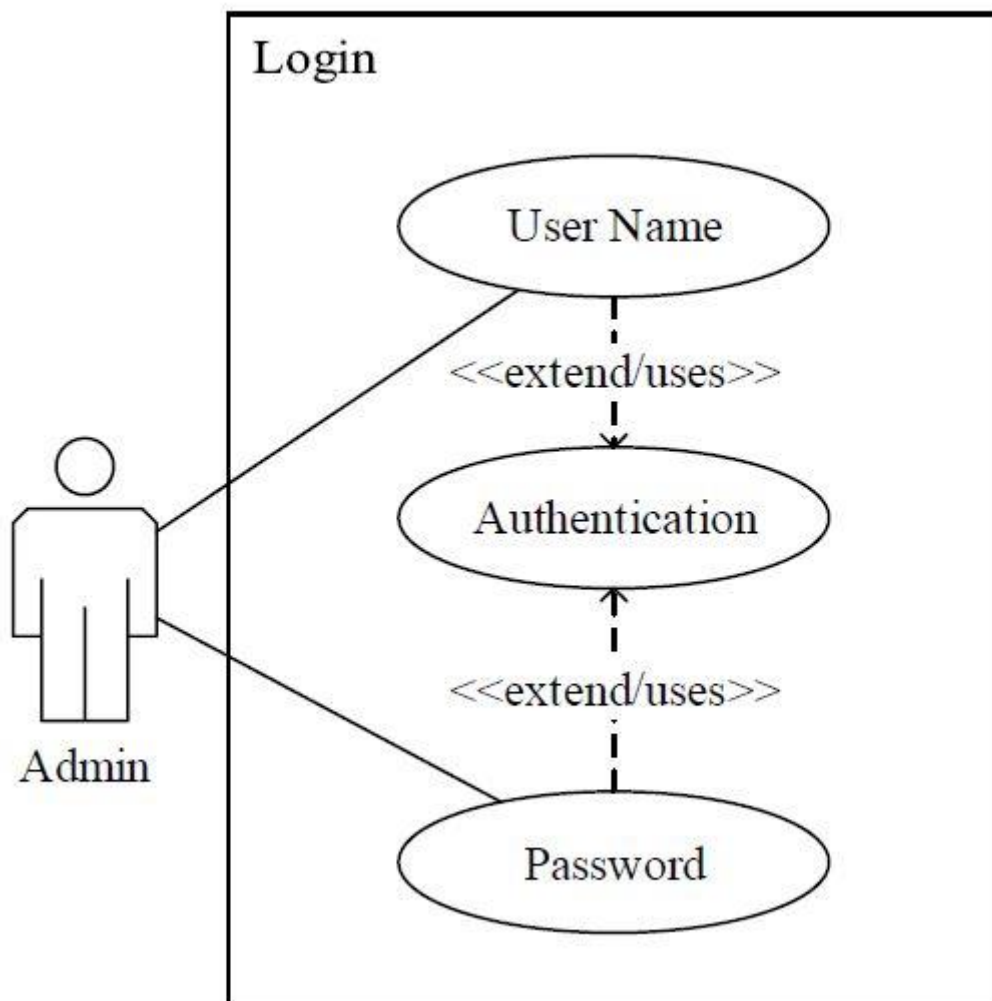


Figure 3 Chapter 5 Login Use Case

An extract of a use case diagram showing the actor “Administrator” and its various roles along with their relationships.

## 5.2 Traceability Matrix:

### FR01: Administrator Login

| Req. No  | Actor         | Functional Requirements   | UC#  | Non-Functional Requirements  | GUI#  |
|----------|---------------|---|------|--|-------|
| FR01- 01 | Administrator | System shall allow administrator to log in                                      | UC04 | NFR01-01<br>NFR01-02<br>NFR01-03<br>NFR01-04<br>NFR02-02<br>NFR02-03<br>NFR02-04 | GUI 2 |
| FR01- 02 | Administrator | System shall get username from administrator                                    | UC04 | NFR01-01<br>NFR01-02<br>NFR01-03<br>NFR01-04<br>NFR02-02<br>NFR02-03<br>NFR02-04 | GUI 2 |
| FR01- 03 | Administrator | System shall get username from administrator                                    | UC04 | NFR01-01<br>NFR01-02<br>NFR01-03<br>NFR01-04<br>NFR02-02<br>NFR02-03<br>NFR02-04 | GUI 2 |
| FR01- 04 | Administrator | System shall authenticate administrator when he/she clicks on the submit button | UC04 | NFR01-01<br>NFR01-02<br>NFR01-03<br>NFR01-04<br>NFR02-02<br>NFR02-03<br>NFR02-04 | GUI 2 |

### FR02: Add or Edit Users

| Req. No  | Actor         | Functional Requirements   | UC#           | Non-Functional Requirements  | GUI#  |
|----------|---------------|---|---------------|--|-------|
| FR02- 01 | Administrator | System shall allow for adding new users or removing existing ones | UC06,<br>UC07 | NFR01-01<br>NFR01-02<br>NFR01-03<br>NFR01-04<br>NFR02-02<br>NFR02-03 | GUI 4 |



|          |               |  |               |  |       |
|----------|---------------|--|---------------|--|-------|
|          |               |  |               | NFR02-04   |       |
| FR02- 02 | Administrator | System shall require the Administrator login before removing existing users                        | UC06,<br>UC07 | NFR01-01<br>NFR01-02<br>NFR01-03<br>NFR01-04<br>NFR02-02<br>NFR02-03<br>NFR02-04 | GUI 4 |
| FR02- 03 | Administrator | System shall allow administrator to add or a new record or select an existing record from the grid | UC06,<br>UC07 | NFR01-01<br>NFR01-02<br>NFR01-03<br>NFR01-04<br>NFR02-02<br>NFR02-03<br>NFR02-04 | GUI 4 |
| FR02- 04 | Administrator | System shall allow user to save details for new user or update details for existing users          | UC06,<br>UC07 | NFR01-01<br>NFR01-02<br>NFR01-03<br>NFR01-04<br>NFR02-02<br>NFR02-03<br>NFR02-04 | GUI 4 |
| FR02- 05 | Administrator | System shall perform the action as indicated by the user   | UC06,<br>UC07 | NFR01-01<br>NFR01-02<br>NFR01-03<br>NFR01-04<br>NFR02-02<br>NFR02-03<br>NFR02-04 | GUI 4 |

### FR03: View Logs and Generate Reports

| Req. No  | Actor         | Functional Requirements   | UC#  | Non-Functional Requirements  | GUI#  |
|----------|---------------|---|------|--|-------|
| FR03 01  | Administrator | System shall allow administrator to select the desired report to view | UC09 | NFR01-01<br>NFR01-02<br>NFR01-03<br>NFR01-04<br>NFR02-02<br>NFR02-03<br>NFR02-04 | GUI 5 |
| FR03- 02 | Administrator | System shall generate the report selected by administrator            | UC09 | NFR01-01<br>NFR01-02<br>NFR01-03<br>NFR01-04<br>NFR02-02<br>NFR02-03<br>NFR02-04 | GUI 5 |

|          |               |   |      |  |       |
|----------|---------------|---|------|--|-------|
| FR03- 03 | Administrator | System shall allow administrator to print the report, refresh the report or view another report | UC09 | NFR01-01<br>NFR01-02<br>NFR01-03<br>NFR01-04<br>NFR02-02<br>NFR02-03<br>NFR02-04 | GUI 5 |
| FR03- 04 | Administrator | System shall display reports in a proper, uncluttered format                                    | UC09 | NFR01-01<br>NFR01-02<br>NFR01-03<br>NFR01-04<br>NFR02-02<br>NFR02-03<br>NFR02-04 | GUI 5 |

#### **FR04: Edit User Privileges**

| <b>Req. No</b> | <b>Actor</b>  | <b>Functional Requirements</b>  | <b>UC#</b> | <b>Non-Functional Requirements</b>   | <b>GUI#</b> |
|----------------|---------------|---|------------|--|-------------|
| FR04- 01       | Administrator | System shall allow the administrators to add or remove administrative rights on other users | UC08       | NFR01-01<br>NFR01-02<br>NFR01-03<br>NFR01-04<br>NFR02-02<br>NFR02-03<br>NFR02-04 | GUI 4       |
| FR04- 02       | Administrator | System shall require the administrator to select the desired user                           | UC08       | NFR01-01<br>NFR01-02<br>NFR01-03<br>NFR01-04<br>NFR02-02<br>NFR02-03<br>NFR02-04 | GUI 4       |
| FR04- 03       | Administrator | System shall allow the administrator to change category of the desired user                 | UC08       | NFR01-01<br>NFR01-02<br>NFR01-03<br>NFR01-04<br>NFR02-02<br>NFR02-03<br>NFR02-04 | GUI 4       |
| FR04- 04       | Administrator | System shall allow administrator to click the update button                                 | UC08       | NFR01-01<br>NFR01-02<br>NFR01-03<br>NFR01-04<br>NFR02-02<br>NFR02-03<br>NFR02-04 | GUI 4       |

|          |               |   |      |  |       |
|----------|---------------|---|------|--|-------|
| FR04- 05 | Administrator | System shall perform the desired action and enforce the specified changes | UC08 | NFR01-01<br>NFR01-02<br>NFR01-03<br>NFR01-04<br>NFR02-02<br>NFR02-03<br>NFR02-04 | GUI 4 |
|----------|---------------|---|------|--|-------|

### **FR05: Set Users as ‘Active’ or ‘Inactive’**

| <b>Req. No</b> | <b>Actor</b>  | <b>Functional Requirements</b>   | <b>UC#</b> | <b>Non-Functional Requirements</b>   | <b>GUI#</b> |
|----------------|---------------|--|------------|--|-------------|
| FR05- 01       | Administrator | System shall allow for marking existing users as active or inactive                                | UC13       | NFR01-01<br>NFR01-02<br>NFR01-03<br>NFR01-04<br>NFR02-02<br>NFR02-03<br>NFR02-04 | GUI 4       |
| FR05- 02       | Administrator | System shall require the Administrator login before allowing to change status                      | UC13       | NFR01-01<br>NFR01-02<br>NFR01-03<br>NFR01-04<br>NFR02-02<br>NFR02-03<br>NFR02-04 | GUI 4       |
| FR05- 03       | Administrator | System shall allow administrator to add or a new record or select an existing record from the grid | UC13       | NFR01-01<br>NFR01-02<br>NFR01-03<br>NFR01-04<br>NFR02-02<br>NFR02-03<br>NFR02-04 | GUI 4       |
| FR05- 04       | Administrator | System shall display details of the selected user in the appropriate textboxes                     | UC13       | NFR01-01<br>NFR01-02<br>NFR01-03<br>NFR01-04<br>NFR02-02<br>NFR02-03<br>NFR02-04 | GUI 4       |
| FR05- 05       | Administrator | System shall wait for user action  | UC13       | NFR01-01<br>NFR01-02<br>NFR01-03<br>NFR01-04<br>NFR02-02<br>NFR02-03             | GUI 4       |

|          |               |   |      |  |       |
|----------|---------------|---|------|--|-------|
|          |               |   |      | NFR02-04   |       |
| FR05- 06 | Administrator | System shall change PIN of user to 0000 (make inactive) if the delete button is pressed | UC13 | NFR01-01<br>NFR01-02<br>NFR01-03<br>NFR01-04<br>NFR02-02<br>NFR02-03<br>NFR02-04 | GUI 4 |

### **FR06: Reactivate the System**

| <b>Req. No</b> | <b>Actor</b>  | <b>Functional Requirements</b>   | <b>UC#</b> | <b>Non-Functional Requirements</b>   | <b>GUI#</b> |
|----------------|---------------|--|------------|--|-------------|
| FR06- 01       | Administrator | System shall disable keypad after three unsuccessful check-in/check-out attempts | UC10       | NFR01-01<br>NFR01-02<br>NFR01-03<br>NFR01-04<br>NFR02-02<br>NFR02-03<br>NFR02-04 | GUI3        |
| FR06- 02       | Administrator | System shall require administrator login once keypad is disabled                 | UC10       | NFR01-01<br>NFR01-02<br>NFR01-03<br>NFR01-04<br>NFR02-02<br>NFR02-03<br>NFR02-04 | GUI 3       |
| FR06- 03       | Administrator | System shall give administrator the choice to reactivate system upon login       | UC10       | NFR01-01<br>NFR01-02<br>NFR01-03<br>NFR01-04<br>NFR02-02<br>NFR02-03<br>NFR02-04 | GUI 3       |
| FR06- 04       | Administrator | System shall perform the desired action and enable the keypad                    | UC10       | NFR01-01<br>NFR01-02<br>NFR01-03<br>NFR01-04<br>NFR02-02<br>NFR02-03<br>NFR02-04 | GUI 3       |

**FR07: Allow Registered Users to Check-In To the System**

| <b>Req. No</b> | <b>Actor</b>           | <b>Functional Requirements</b>  | <b>UC#</b> | <b>Non-Functional Requirements</b>   | <b>GUI#</b> |
|----------------|------------------------|---|------------|--|-------------|
| FR07- 01       | User,<br>Administrator | System shall require user to enter PIN                                      | UC01       | NFR01-01<br>NFR01-02<br>NFR01-03<br>NFR01-04<br>NFR02-02<br>NFR02-03<br>NFR02-04 | GUI 1       |
| FR07- 02       | User,<br>Administrator | System shall authenticate user once PIN is entered                          | UC01       | NFR01-01<br>NFR01-02<br>NFR01-03<br>NFR01-04<br>NFR02-02<br>NFR02-03<br>NFR02-04 | GUI 1       |
| FR07- 03       | User,<br>Administrator | System shall verify if user is checking in within the allotted timeframe    | UC01       | NFR01-01<br>NFR01-02<br>NFR01-03<br>NFR01-04<br>NFR02-02<br>NFR02-03<br>NFR02-04 | GUI 1       |
| FR07- 04       | User,<br>Administrator | System shall open lock and record time in case of successful authentication | UC01       | NFR01-01<br>NFR01-02<br>NFR01-03<br>NFR01-04<br>NFR02-02<br>NFR02-03<br>NFR02-04 | GUI 1       |

**FR08: Allow Registered Users to Check Out From the System**

| <b>Req. No</b> | <b>Actor</b>           | <b>Functional Requirements</b>         | <b>UC#</b> | <b>Non-Functional Requirements</b>   | <b>GUI#</b> |
|----------------|------------------------|--|------------|--|-------------|
| FR07- 01       | User,<br>Administrator | System shall require user to enter PIN | UC01       | NFR01-01<br>NFR01-02<br>NFR01-03<br>NFR01-04<br>NFR02-02<br>NFR02-03<br>NFR02-04 | GUI 1       |

|          |                        |   |      |  |       |
|----------|------------------------|---|------|--|-------|
| FR07- 02 | User,<br>Administrator | System shall<br>authenticate user<br>once PIN is entered                                | UC01 | NFR01-01<br>NFR01-02<br>NFR01-03<br>NFR01-04<br>NFR02-02<br>NFR02-03<br>NFR02-04 | GUI 1 |
| FR07- 03 | User,<br>Administrator | System shall open<br>lock and record<br>time in case of<br>successful<br>authentication | UC01 | NFR01-01<br>NFR01-02<br>NFR01-03<br>NFR01-04<br>NFR02-02<br>NFR02-03<br>NFR02-04 | GUI 1 |

### 5.3 Entity Relationship Diagram:

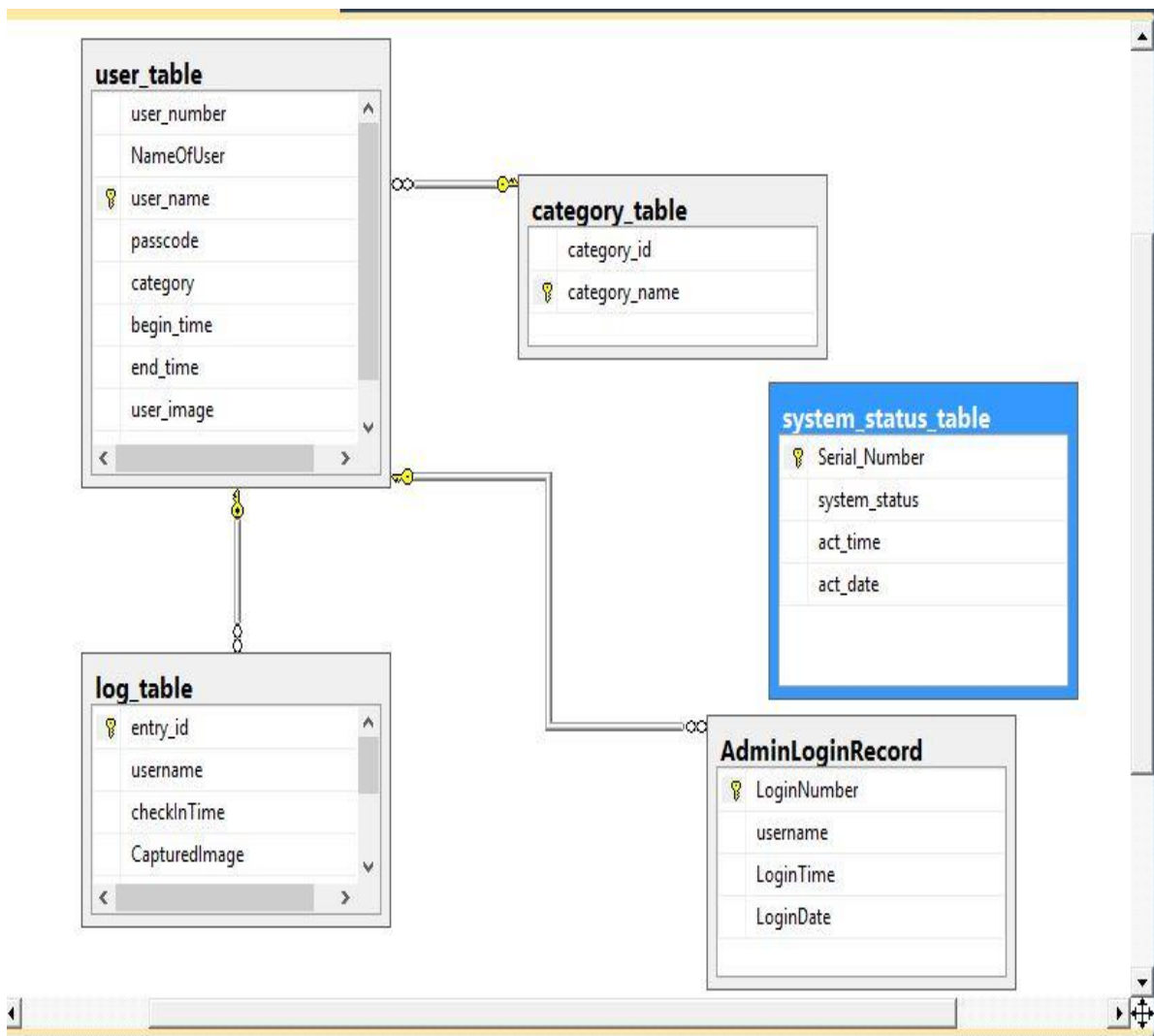


Figure 4Chapter 5: Entity Relationship Diagram

The above diagram shows the database diagram for this project. It is helpful in providing an overview of the design of various tables and their relationships with one another. It also provides knowledge of the data types of different attributes of each table.

## 5.4 Graphical User Interface:

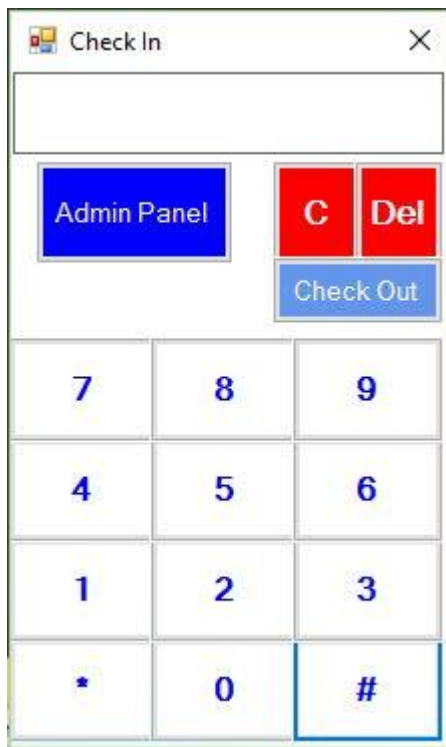


Figure 6 GUI 1 Check-In

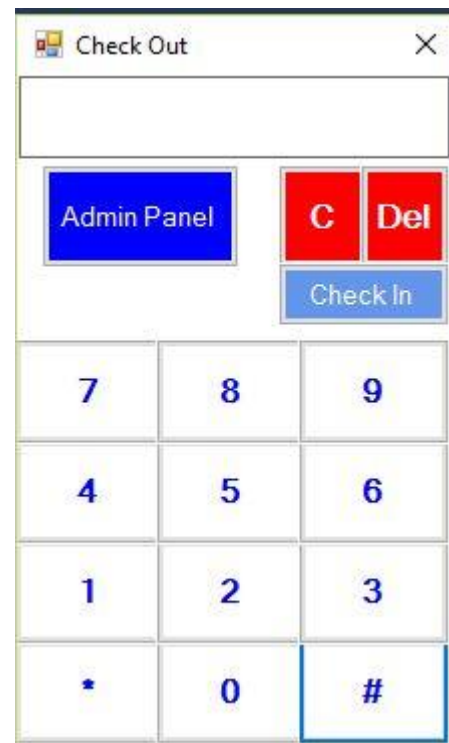


Figure 7 GUI 2 Check-Out

The figure on the left is the Check In interface. The system uses it to allow a user to check in to the household. Likewise, the figure on the left is the Check In interface. The system uses it to allow a user to check in to the household.



**Figure 8 GUI 3 Admin Panel Login Form**

The above snapshot shows the login screen for the admin panel of the system. Admin enters the credentials and is authenticated upon clicking the “login” button.



**Figure 9 GUI 4 Select Operation**



Shown above is the “select operation” screen of the admin panel. This is displayed right after successful login and allows for selecting one of the activities to perform. The system proceeds depending on the selection.

The screenshot shows a web application window titled "Admin Operations". It features a form on the left for entering user details and a table on the right displaying a list of users.

**Form Fields:**

- Enter Name:
- Enter Username:
- Enter PIN:
- Select Category:
- Begin Time:
- End Time:
- Browse Image:
- Save:
- Clear:

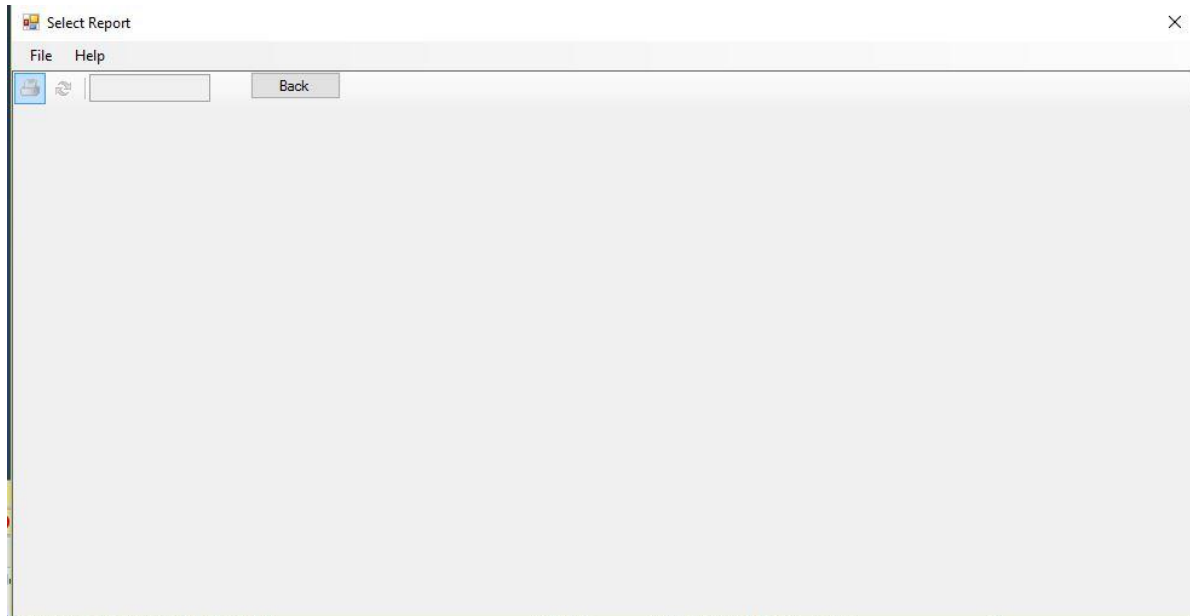
**User Management Table:**

| Name          | Username     | Category | Begin Time | End Time |
|---------------|--------------|----------|------------|----------|
| abc           | abc          | Admin    | 10:11:37   | 10:11:37 |
| asdf          | asdf         | Admin    | 18:54:41   | 18:54:41 |
| Jack          | jack         | Workers  | 11:17:31   | 11:17:31 |
| Naeem Khawaja | nkhawaja     | Admin    | 23:30:50   | 23:30:50 |
| Sarib Niazi   | saribniazi   | Admin    | 00:00:00   | 23:59:00 |
| Sohaib Furqan | sohaibfurqan | Admin    | 00:00:00   | 23:59:59 |
| sss           | sss          | Admin    | 18:04:12   | 19:04:12 |
| wxy           | wxy          | Workers  | 11:04:50   | 11:04:50 |
| xyz           | xyz          | Admin    | 23:41:37   | 23:41:37 |
| *             |              |          |            |          |

Navigation buttons: Back, Manage Log, Log Out

**Figure 10 GUI 5 Users Management**

The “Users Management” screen of the admin panel is shown above. It allows the user to add new records or edit existing records in one place. Users can select existing records from the grid on the right, causing the textboxes to be populated with the fields of the selected record. The textboxes can then be edited as required. The updated record will be stored to the database when the user clicks the update button. If the user selects a record and hits the delete button, the selected user will be changed to an inactive state, thus preventing them from using the system’s authentication panels.



**Figure 11 GUI 6 Select Report**

Above is the “select report” screen of the system. The user selects the report to be generated from the file menu on top and the corresponding crystal report is generated. The about menu lists the credits for the project.

## **5.5 Design Goals:**

### **➤ Performance:**

Performance is one of the significant aspects of this system, as it plays a very important role in the quality of any product. The system must perform efficiently and reliably in all scenarios, so as to enhance security to the fullest. Better performance here implies that the system will perform with the same reliability and efficiency every time.

### **➤ Security:**

Security is a major concern as the system will be implemented in the security domain. Users of the system have to be provided with the best preventive measures available

which will only be possible when security is given proper focus. Security considerations are a feature of development from the start.

➤ **Usability:**

The system must be user-friendly so that it can be easily operated by users of all kinds irrespective of the educational background. It should provide clear and concise instructions and follow a simplistic pattern. The GUI design should be such that no issues are faced while operating the system.

➤ **Maintainability:**

After the software deployment, if client want to add some more functionality to the software system, the software must be flexible enough to accomplish any change.

➤ **Economical:**

The system is designed to be economical and easily affordable so that it can be put to effective home use.

## **Summary:**

This chapter lays down in detail the traceability matrix, GUI, Design goals and all other aspects of the design of the system.

# **Chapter 6**

## **Coding & Development**

## 6.0 Introduction:

This chapter includes the code produced during the development process.

### 6.1 App.config

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
<configSections>
  </configSections>

  <connectionStrings>
    <add name="DigiDoorConnString" connectionString="Data Source= DESKTOP-
N9PERM3; Initial Catalog=DigiDoor; Integrated Security=true"
      providerName="System.Data.SqlClient" />
  </connectionStrings>
  <startup useLegacyV2RuntimeActivationPolicy ="true">
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.0"/>

  </startup>
</configuration>
```

### 6.2 Program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace DigiDoor_UPDATED
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new CheckInFom());
        }
    }
}
```

### 6.3 SelectAdminOperation.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
```

```

using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace DigiDoor_UPDATED
{
    public partial class SelectAdminOperation : Form
    {
        public SelectAdminOperation()
        {
            InitializeComponent();
        }

        public string setActiveStatus { get; set; }
        public string setSystemStatus { get; set; }

        private void ManageUsersBtn_Click(object sender, EventArgs e)
        {
            AdminOperations op = new AdminOperations();
            op.Show();
            this.Hide();
        }

        private void ViewReportsBtn_Click(object sender, EventArgs e)
        {
            // RPTViewerForm form = new RPTViewerForm();
            // form.Show();
            // this.Hide();

            SelectReportForm form = new SelectReportForm();
            form.Show();
            this.Hide();
        }

        private void LogOutBtn_Click(object sender, EventArgs e)
        {
            DialogResult dialog = MessageBox.Show("Do you really want to log out?", "Log Out", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
            if (dialog == DialogResult.Yes)
            {
                try
                {
                    loginForm form = new loginForm();
                    form.Show();
                    this.Hide();
                }
                catch (Exception ex)
                {
                    string lineNumber =
                    ex.StackTrace.Substring(ex.StackTrace.Length - 7, 7);
                    MessageBox.Show(ex.Message + "\n" + lineNumber, "Exception",
                    MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
                }
            }
        }
    }
}

```

```

        private void SelectAdminOperation_FormClosing(object sender,
FormClosingEventArgs e)
        {
            try
            {
                DialogResult dialog = MessageBox.Show("Do you really want to
exit?", "Exit", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
                if (dialog == DialogResult.Yes)
                {

                    Application.ExitThread();
                }
                else
                {
                    e.Cancel = true;
                }
            }

            catch (Exception ex)
            {
                string lineNumber = ex.StackTrace.Substring(ex.StackTrace.Length
- 7, 7);
                MessageBox.Show(ex.Message + "\n" + lineNumber, "Exception",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
            }
        }

        private void ActivateSystemButton_Click(object sender, EventArgs e)
        {
            setActiveStatus = "true";
            setSystemStatus = "active";
            MessageBox.Show("System
activated!", "Success", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
        private void SelectAdminOperation_Load(object sender, EventArgs e)
        {
            //CheckInFom adminop = new CheckInFom();
            //if (adminop.setActiveStatus.ToString() == "false" &&
adminop.setSystemStatus.ToString() == "Inactive")
            //{
                ActivateSystemButton.Visible = true;
            //}
        }
    }
}

```

## **Glossary:**



## References:

- visualcsharptutorials.com, 'Windows Forms' [2016]. Available: <http://www.visualcsharptutorials.com/windows-forms>. [Accessed: 03 April, 2016].
- Stack overflow, 'C# winforms' [2014]. Available: <http://www.stackoverflow.com/questions/.../c-sharp-get-line-number-which-threw-exception>. [Accessed: 12 April, 2016]
- W3schools, 'SQL server tutorials' [2016]. Available: <http://www.W3schools.com/sql>. [Accessed: 14 April, 2016]
- Google Scholar, 'Related Work' [2012-2016]. Available: [https://scholar.google.com.pk/scholar?hl=en&q=automatic+door+lock+&btnG=&as\\_sdt=1%2C5&as\\_sdt=](https://scholar.google.com.pk/scholar?hl=en&q=automatic+door+lock+&btnG=&as_sdt=1%2C5&as_sdt=) [Accessed: 26 March, 2016]
- Microsoft Developer Network, 'Developer Resources' [2012]. Available: <https://msdn.microsoft.com/en-us/library/67ef8sbd.aspx>. [Accessed: 11 May, 2016]
- Microsoft Developer Network, 'Developer Resources' [2012]. Available: [https://msdn.microsoft.com/en-us/library/618ayhy6\(v=vs.71\).aspx](https://msdn.microsoft.com/en-us/library/618ayhy6(v=vs.71).aspx). [Accessed: 11 May, 2016]