# DESIGN AND IMPLEMENTATION OF DAYCARE MANAGEMENT SYSTEM

**BY**

# ATIKU SUMAYYAH BU/8A/IT/3037

**DEPARTMENT OF COMPUTER SCIENCE BAZE UNIVERSITY**

# ABUJA

**SEPTEMBER, 2020**

# DESIGN AND IMPLEMENTATION OF DAYCARE MANAGEMENT

**SYSTEM**

# THESIS SUBMITTED IN PARTIAL FULLFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF

**B.Sc.**

# In COMPUTER SCIENCE

**BY**

# ATIKU SUMAYYAH BU/8A/IT/3037

**TO**

# THE DEPARTMENT OF COMPUTER SCIENCE BAZE UNIVERSITY, ABUJA

**SEPTEMBER, 2020**

# DECLARATION

This is to certify that this Report entitled [Design and implementation of daycare management system], which is submitted by [Sumayyah Atiku] in partial fulfillment of the requirement for the award of degree for B.Sc. Computer Science to the Department of Computer Science, Baze University Abuja, Nigeria, comprises of only my original work and due acknowledgement has been made in the text to all other materials used.

Date: September, 2020 Name of Student: Sumayyah Atiku

**APPROVED BY** ……………………

# Head Department of Computer Science

**CERTIFICATION**

This is to certify that this Report entitled [Design and implementation of daycare management system], which is submitted by [Sumayyah Atiku] in partial fulfillment of the requirement for the award of degree for B.Sc. in Computer Science to the Department of Computer Science, Baze University Abuja, Nigeria is a record of the candidate’s own work carried out by the candidate under my/our supervision. The matter embodied in this thesis is original and has not been submitted for the award of any other degree.

Date: Supervisor: Mr. Nasiru Aboki

Date: Supervisor: Dr Moses Ubaru

# APPROVAL

This is to certify that the research work, Daycare Management System and the subsequent preparation by [Sumayyah Atiku] with [BU/18A/IT/3037] has been approved by the Department of Computer Science, Faculty of Computing and Applied Science, Baze University, Abuja, Nigeria.

By

Dr. Moses Ubaru

1st Supervisor Date/Sign

Mr Nasiru Aboki

2nd Supervisor Date/Sign

Dr.C.V.Uppin

Head of Department Date/Sign

Prof. Mohammed Hammawa Baba

Dean, Faculty of Computing and Applied Science Date/Sign

Prof. Ahmed Baita Garko

External Examiner Date/Sign

# DEDICATION

I dedicate this project to Allah (SWA) my creator, my source of inspiration, wisdom, knowledge and understanding. He has been the source of my strength throughout this program and on His wings only have I soared. I also dedicate this work to my parents; Retired Air Commodore UJ Atiku and Mrs. Fatima Atiku who encouraged me all the way and made sure that I give it all it takes to finish that which I have started. To my dear husband in person of Dr. Aminu Ibrahim who has been affected in every way possible by this quest. My love for you all is immeasurable. Thank you and God bless.

# ABSTRACT

*child daycare system is the care of a child during the day by a person other than the child’s legal guardian, basically performed by someone outside the child’s immediate family. it is typically ongoing for a specific period, such as parents work time. the system will be used to register a child, monitor and verify who drops and picks the child, log of child’s activity, parents/guardian details, parents will be able to update a child’s details (e.g. record allergy, drug administration time if any etc.). it can also take a formal structure with education, child development and discipline.*

# TABLE OF CONTENTS

|  |  |
| --- | --- |
| Title Page | - - - - - - - - - - i |
| Cover Page | - - - - - - - - - - ii |
| Declaration | - - - - - - - - - - iii |
| Certification | - - - - - - - - - - iv |
| Approval | - - - - - - - - - - v |
| Dedication | - - - - - - - - - - vi |
| *ABSTRACT* | *- - - - - - - - - - VII* |

CHAPTER ONE: INTRODUCTION

[1.1 Overview - - - - - - - - - 1](#_TOC_250014)

* 1. Background and Motivation - - - - - - - 1
  2. [Statement of Problem - - - - - - - - 2](#_TOC_250013)
  3. [Aim and Objective - - - - - - - - 2](#_TOC_250012)
  4. [Significance of the Project - - - - - - - 2](#_TOC_250011)
  5. [Risk Assessment - - - - - - - - 3](#_TOC_250010)
  6. [Scope/Project Organization - - - - - - - 3](#_TOC_250009)

CHAPTER TWO: LITERATURE REVIEW

[2.0 Overview - - - - - - - - - 4](#_TOC_250008)

* 1. [Introduction - - - - - - - - - 4](#_TOC_250007)
  2. [Types of day care - - - - - - - - 5](#_TOC_250006)
  3. [Related Work - - - - - - - - - 5](#_TOC_250005)
     1. [EZcare childcare management software - - - - - 6](#_TOC_250004)
     2. [The Pre School Partner - - - - - - - 6](#_TOC_250003)
     3. [Power School Kindergarten System - - - - - - 6](#_TOC_250002)
     4. [Proposed solution - - - - - - - - 7](#_TOC_250001)

[2.4 Summary - - - - - - - - - 7](#_TOC_250000)

CHAPTER THREE: REQUIREMENTS ANALYSIS AND DESIGN

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 3.1 | Overview | - | - | - | - | - | - | - | - | - | 8 |
| 3.2 | Proposed Methodology | | | - - - - - - - | | | | | | | 8 |
| 3.3 | Approach to chosen Methodology - | | | | | - | - | - | - | - | 10 |
| 3.4 | Tool and Techniques - - - | | | | | - | - | - | - | - | 10 |
| 3.5 | Requirement Analysis - - | | | | | - | - | - | - | - | 10 |
| 3.6 | Requirement specifications - - | | | | | - | - | - | - | - | 11 |
| 3.6.1 | Functional Requirements Specification - | | | | | - | - | - | - | - | 11 |
| 3.6.2 | Non-functional Requirement specifications | | | | | - | - | - | - | - | 12 |
| 3.7. | Application Architecture | | | - | - | - | - | - | - | - | 12 |
| 3.7.1 | Use Case Diagram - | | | - | - | - | - | - | - | - | 14 |
| 3.7.2 | Class Diagram- - | | | - | - | - | - | - | - | - | 17 |
| 3.7.3 | Database Design - | | | - | - | - | - | - | - | - | 18 |
| 3.7.4 | Dataflow Diagram (DFD) | | | - | - | - | - | - | - | - | 19 |
| 3.7.5 | Activity Diagram - | | | - | - | - | - | - | - | - | 21 |
| 3.6 | Summary - - | | | - | - | - | - | - | - | - | 22 |

CHAPTER FOUR: IMPLEMENTATION AND TESTING

4.1 Overview - - - - - - - - - 23

* 1. Development tools and technologies - - - - - - 23
  2. Implementation Problems - - - - - - - 23
  3. Overcoming Implementation Problems - - - - - 24

4.5 Testing - - - - - - - - - 25

4.5.1 Test Plans - - - - - - - - - 25

4.5.2 Test Traceability Matrix - - - - - - - 26

4.6 Test Report Summary - - - - - - - - 28

4.7 User Guide - - - - - - - - - 28

4.8 Summary- - - - - - - - - - 28

CHAPTER FIVE: DISCUSSION, CONCLUSION, AND RECOMMENDATIONS

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 5.1 | Overview | - - - - - - - - - | | | | | | | | | 29 |
| 5.2 | Evaluation | - - - - - - - - - | | | | | | | | | 29 |
| 5.2.1 | Objective Assessment - | | | - | - | - | - | - | - | 29 |  |
| 5.3 | Limitations and challenges | | | - | - | - | - | - | - | - | 29 |
| 5.4 | Future Enhancements - | | | - | - | - | - | - | - | - | 30 |
| 5.5 | Recommendation - | | | - | - | - | - | - | - | - | 30 |
| 5.6 | Conclusion | - | - | - | - | - | - | - | - | - | 30 |
| REFERENCES | | - | - | - | - | - | - | - | - | - | 31 |
| APPENDICES | | - | - | - | - | - | - | - | - | - | 32 |

# CHAPTER ONE INTRODUCTION

# Overview

Early education and care system play a very significant role in the growth of children, preparing them for school and warranting parents the opportunity to engage in the workforce. It is believed that children need a warm, safe, colorful environment and diversified experience that focuses attention on ‘play’. By making these things available, a child will grow and develop at a pace that is just right for them.

The daycare strives to provide quality care for all the children through age appropriate activities that will help the children grow physically, emotionally, mentally and socially. High standards for all staffs is maintained and child ratio is kept small to ensure adequate care for each child. This system will make it easy to track and keep record of children and their parents.

# Background and Motivation Background

In Nigeria, organized education of the child below primary school age did not receive official recognition until very recently; it got the attention it deserves. The concept of infant schools was introduced in Nigeria by the missionaries in the early 20th century when such schools were set up in the Western and Eastern regions of Nigeria. Nowadays, early childhood educational institutions are located in various places and buildings campus of universities and Colleges, premises of some industries and business organizations, church premises, residential buildings with unprecedented expansion owing to the high demand for early childhood care and education by parents (Ejieh, 2006).

It is acceptable to say the application of computer technology in any venture would help in making that venture easier and the benefits of using a computer to manage children data in daycare clearly outweighs that of manual method which involves written documents.

# Motivation

The motivation behind this project is the creche located on the campus of baze university abuja. Considering the fact that there no such thing as children’s records or any form of children activity log that can be presented to the parents or guardians of the children. I got the inspiration to develop software that can track, register and provide information about the children.

# Statement of Problem

The issues that lead to this project are experienced by the management, staff and parents of children, they include.

* + - Lack of efficient database storage system.
    - Duplicate entry or registration.
    - Poor graphical user interface.
    - Time delay.
    - No provision for allergy indication.

# Aim and Objective

* + - To Design a system that will contain reliable database of all children admitted into the daycare.
    - To have a system that will be used to establish an efficient means of tracking children in daycare.
    - Verification of check in and out.
    - Identification of children with allergy.
    - Log of child’s activity while at the daycare.
    - To allow room for parents to update a child’s record.

# Significance of the Project

The implementation project will make it easy for daycare owners, staffs and parent to track children, provide adequate information about children, and register children both new and returning. It will allow parents to log into a digital system when to see the record of utilization pattern of the school, i.e., the child’s activity for that entire day.

It will limit the report of file loss thereby improving the security of the records the research project would serve as reference to others interested I further research of the topic.

# Risk Assessment

Risk assessment assures that a daycare environment is safe and ideal for children, staff and parents.

Power outage: loss of power is one the major issues that could affect this system, to prevent this problem, laptops and desktops should always be fully charged when is power supply. UPS, inverters or generators should be provided if possible.

Loss of data: to prevent any loss of information, a piece of data should be backed up as it is registered.

# Scope/Project Organization

This project is restricted to the creation of children daycare management system (A case study of AIM daycare center Kaduna) data used for the course of this project was gathered from this source.

# CHAPTER TWO:

**LITERATURE REVIEW**

# Overview

This chapter is concerned with the literature review. In this chapter, there are three subsections. The first subsection talks about the types of daycare that exist. The second subsection talks about other works that are related to this project, their advantages and lacks. The last subsection contains the summary an outline of the entire chapter.

# Introduction

According to (Thomas, 2011) an information system (IS) can be defined as a set of interrelated components that retrieve, process, store and distribute information to aid decision making and control within an organization. In addition, information system involves a variety of information technologies such as computers, software, database, internet, mobile devices and much more, to perform specific tasks, interact with and inform various actors.

According to (Laudon, 2005) Management information system (MIS) is a computerized database of financial information organized and programmed in such a way that it produces regular reports on every operation for every management level in an organization or company. Usually, it is also possible to acquire special reports from the system with ease. The major purpose of MIS is to provide managers with feedback on their own performance, the management can monitor the organization as a whole. The information presented by the MIS typically displays “actual” data over “planned” results and results from the previous year. Therefore, it measures progress against goals.

In this day and age, there is increasing need for day care centers due to working parents and the need for adequate care for children. There has to be an efficient management to keep up with the demands. For the purpose of making work easy in day care centers, day care owners subscribe to or purchase childcare management system.

Day care management system is basically a management tool for day care a management unit that allows them to virtually control every aspect of the day care management with a user friendly multi user interface. For parents, it serves as a tool to monitor their children’s activity during the day, for the management, it helps to determine the aim of the day care, making long term plans, determine performance of staffs and success of the day care.

# Types of day care

The day care industry is like an extension of parental care to a larger, controlled institutions. A large majority of childcare is still performed by parents, in-house nannies or relatives. Mentioned below are of the various kind of day care we have

Non-profit day care: This kind day care renders services for free, it is more likely to produce a good quality environment in which children can thrive. Not for profit day care has some structural advantage over for profit day care. They may receive better treatments especially if it is attached to a religious organization like church or mosque.

The location of the day care within a school may provide the advantage of coordinated school program and the advantage of same location for parents who have older school kids, work in the school or parents who are students themselves.

Parents are monotonously the legal owners of the non-profit day care and will regularly provide service in areas they specialize in, for instance, accounting and finance, legal advice etc. for free.

**For-profit day care:** for profit day care on the other hand rely on parents’ payment to run. While this in no way determines the care provided by the care providers. The time spent on wondering how to stay ‘in the black’ would be better invested in a program or receive accreditation. The organization purchases better equipment, keep facilities upgraded. It benefits the little people for whom it is intended, the children.

# Related Work

A software tracking system has overtime proven to be very important component of any organization or institution. Several tracking systems have been proposed for various daycare. Below are some of them.

# EZcare childcare management software

EZcare childcare management software manages childcare, preschool, and aftercare needs with an easy-to-use tool that helps organize family and child data. it is a web-based solution for child care organizations to manage their marketing. it can manage parent inquiries. Once a child is enrolled, EZCare’s integration allows the ability to transfer parent and child information into EZCare, saving the time of having to re-enter this information and reducing errors.

Advantage:

Provide efficient database for parents and children. Disadvantage:

Compulsory need for internet connection in order to access the product, no option for offline. Records vaccine but not allergy.

# The Pre School Partner

This is a software-based daycare management system. The system is available for a single PC, it contains a license key for 20 PC’s . It facilitates daily management of the daycare by tracking and providing all child’s, staff and parents or guardian information including drop off and pickup contacts, child schedules and attendance.

Advantage:

It is user friendly and work can be done offline Disadvantage:

The software is restricted to one PC. No record of child’s allergies if any

# Power School Kindergarten System

Power School is a cloud-based nursery and kindergarten management system. It is an excellent tool for school-parent communication. Parents conveniently and instantly get information from kindergarten on their children’s activities.

Advantage:

It is easy to use and efficient.

Disadvantage:

Needs internet at all time to operate.

# Proposed solution

Due to some of the disadvantages of the above related works, this system will solve the problem of registering children with allergies or other related issues, this will make it easy for the staff to know what to feed which child and how much to feed the child in order to avoid risking the child’s health in anyway. This will also enable parents to update their child’s record in case of any new development.

The system will also make storing of records easy and safe for long period of time. It will significantly improve the quality of work in the daycare.

# Summary

This chapter provides a review of current literature on recommended systems. the first section of this chapter talks about information system and management information system, their benefits and use in an organization. Next, the chapter talks about the types of daycare that exist which are the non-profit and the for-profit daycare. The third section of this chapter talks about other related works that have been done in relation to this topic.

# Overview

**CHAPTER THREE REQUIREMENTS ANALYSIS AND DESIGN**

This chapter gives account of analysis and design of daycare management system, the methodology used for the system development, the requirements of the daycare management system (functional and non-functional requirements), the tools used for the development of the system (hardware and software).

Furthermore, this chapter describes the model of the system in form of simplified, complete, and consistent abstraction of the system created for better understanding of the system using DFDs and UML diagrams. The system design; system architecture, user interfaces and the database design which are physical view of the system are also displayed in this chapter.

# Proposed Methodology

General System Development Life Cycle (SDLC)

System Development Life Cycle coheres to relevant phases that are important for developers, for instance planning, analysis, design, implementation and maintenance which are explained in the section below. There are various system development life cycle model available, the oldest one being the waterfall model originally referred to as the “System development life cycle” is a sequence of phases in which the output of each phase determines the input of the next stage. These stages follow the same basic steps. Other SDLC include: agile methodology, rapid application development methodology, prototyping methodology, spiral methodology. The choice of using any of these methodologies depends on the nature of the project.

However, the methodology chosen for this project is the waterfall model because it is a software development process that emphasizes logical progression of steps to be taken throughout the software development life cycle and also because it allows for early design changes.

Below are the stages involved in waterfall methodology:

Planning: In this first stage, there is a meeting with the customer to understand the requirements. This stage is the most fundamental phase as any miscommunication or misinterpretation (at this stage) may give rise to the software that is being developed. The key activity of project planning is the process of clear, discrete activities and the work needed to complete each activity within a single project.

Analyzing: As per the requirements, the software and hardware needed for the proper completion of the project is analyzed in this phase. Such features are decided at this stage; from deciding which computer language should be used for designing the software, to the database system that can be used for the smooth functioning of the software.

Design: the requirements that are gathered in the previous phase are broken down into logical units, so that the software process becomes easy for implementation. This is the stage, when the software requirements along with the hardware requirements for every unit are identified. Then the designs are made accordingly. The interrelation between the various units of the software are identified and the connections are made, using algorithms and diagrams. To sum it up, this is the phase, where the fundamental work for actual programming and implementation is done.

Implementation: In this phase the actual development of the software takes place. This phase is also known as coding and verification phase. Based on the algorithms written in the previous phase, software program is written. For every module, software code is written and tested, to check if the correct output is received.

Integration: And testing with the coding of the application complete, the testing of the written code now comes into scene. Testing checks if there are any flaws in the designed software and if the software has been designed as per the listed specifications. A proper execution of this stage ensures that the customer interested in the created software, will be satisfied with the finished product. If there are any flaws, the software development process must step back to the design phase.

Maintenance: this makes for the final phase of the waterfall model, where the software is organized at the client’s side, after it has undergone thorough testing. After the deployment of the software, routine maintenance work is carried out. Once the software has been deployed, in case the customer asks for any changes or enhancements, then the entire process is restarted.

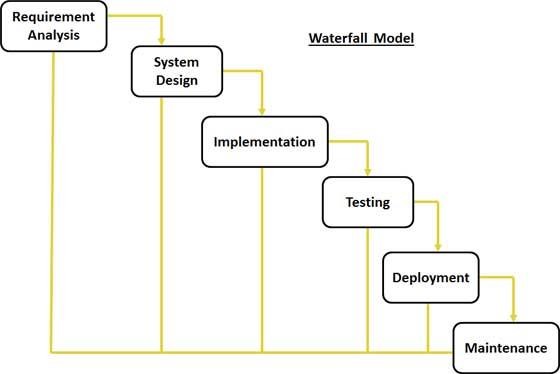


Figure 1. 1

# Approach to chosen Methodology

* 1. **Tool and Techniques**

The following tools are used for the purpose of this project

* + - A computer with 1.9Ghz of processor, 16GB of RAM, and 1TB of Hard drive
    - Erwin Data Modeler
    - UML Diagrams for the analysis and modeling of the system
    - Microsoft office programs for necessary documentations and visualizations
    - Microsoft Visio
    - Microsoft SQL server
    - Visual Studio

# Requirement Analysis

A software requirement specification is a complete description of the behavior of a system to be developed and may include a set of use cases that described the user will have with the software. It also contains non-functional requirements. non-functional requirements impose

constraint on the. Some of the commonly used requirement of analysis and specification methods are system design structured analysis design (SSAD) and object-oriented analysis design (OOAD).

# Requirement specifications

The tables below describe functional and non-functional requirements

# Functional Requirements Specification

**Table1 1 Functional Requirements**

|  |  |  |
| --- | --- | --- |
| Requirement No. | Description | Type |
| R-1 | A user should be able to register to the daycare |  |
| R-2 | A user should be able to view details of a staff/child |  |
| R-3 | Drop off and pickups should be verified |  |
| R-4 | Staff/child record should be able to be updated |  |
| R-5 | Staff should be able to view details of registered pupils |  |
| R-6 | System should allow creation of new account By admin |  |
| R-7 | System should show successful registrations |  |
| R-7 | Users should be able to log in using assigned password and username |  |
| R-8 | System should allow users to sign-out |  |
| R-9 | System should be able to save any change made |  |
| R- 10 | System should allow user to delete an account |  |

# Non-functional Requirement specifications Table1 2 No n-Functional requirements

|  |  |  |
| --- | --- | --- |
| Requirements NO. | Description | Type |
| R-1 | The system should keep running when started unless shutdown intentionally |  |
| R-2 | The system should be user friendly |  |
| R-3 | System should be reliable |  |
| R-4 | The system should run on any hardware. It should not conflict with other processes within  these environments |  |

* 1. **Application Architecture**

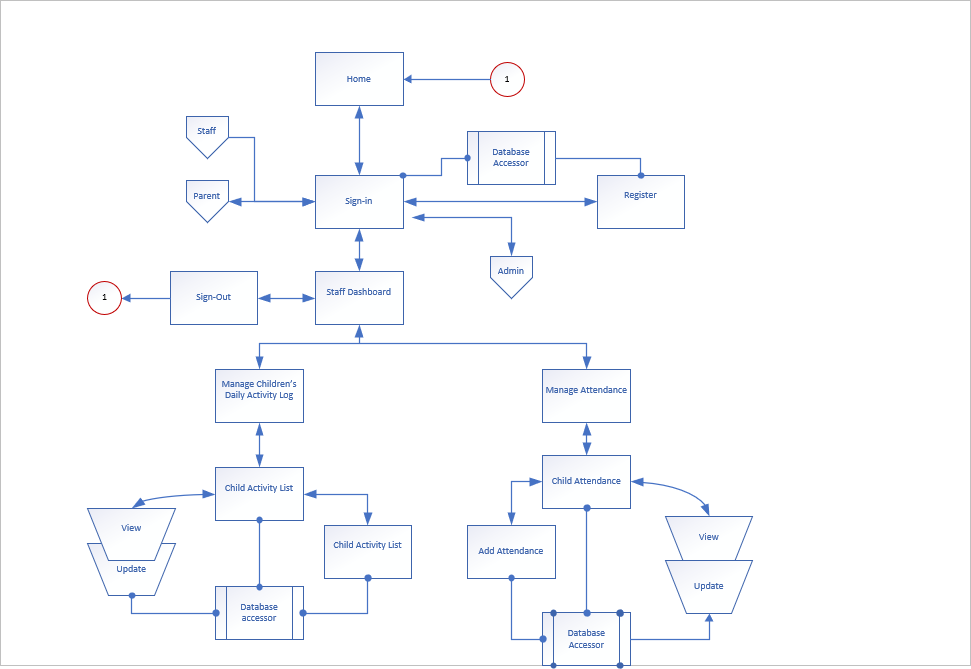


Figure 1. 2 Application Architecture

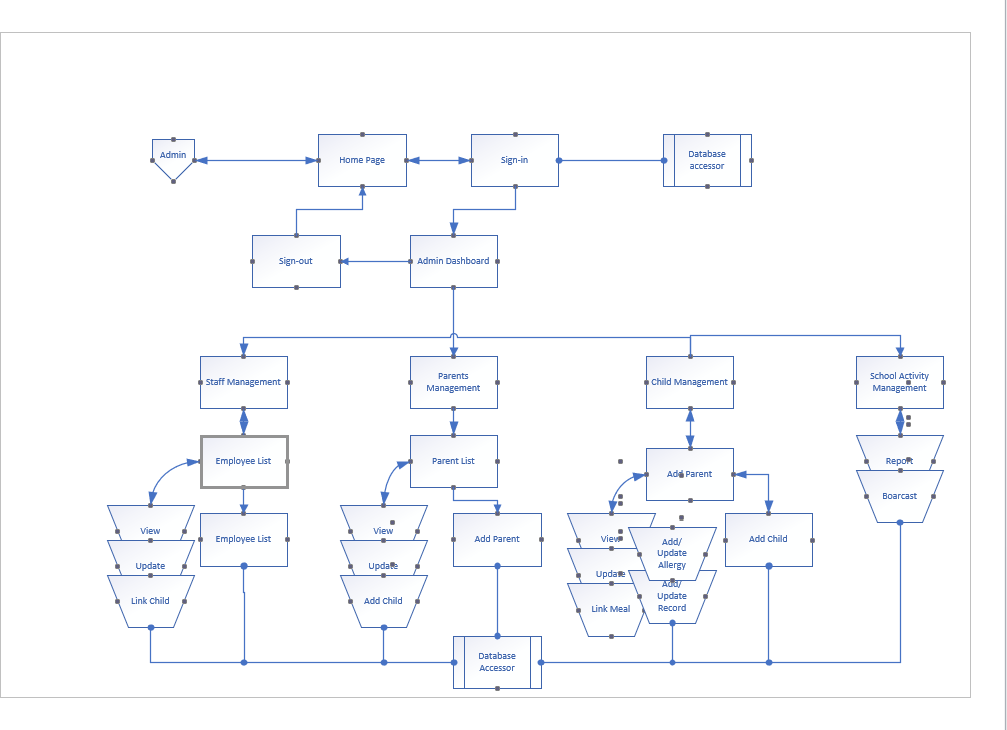


Figure 1. 3: Admin Application Architecture

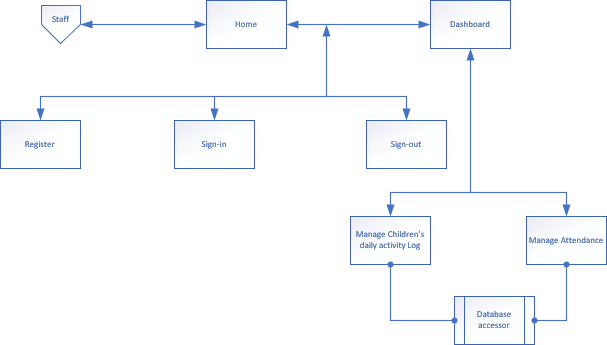


Figure 1. 4 Staff Application Architecture

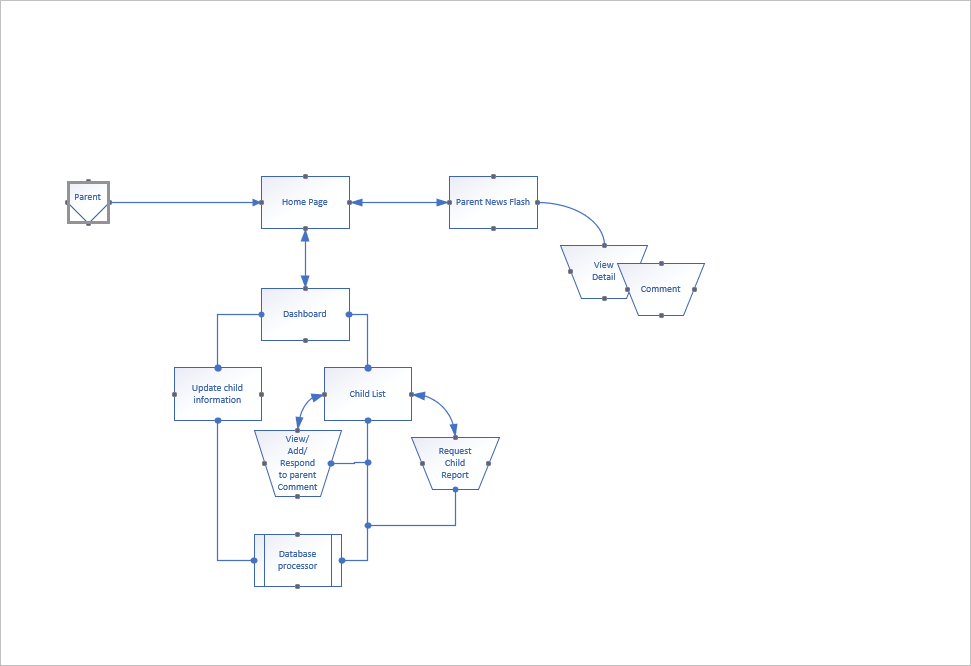


Figure 1.5 Parent Application Architecture

The application architectures above give a high-level representation of the system as well as represent the sub-processes, component and navigation of the system.

# Use Case Diagram

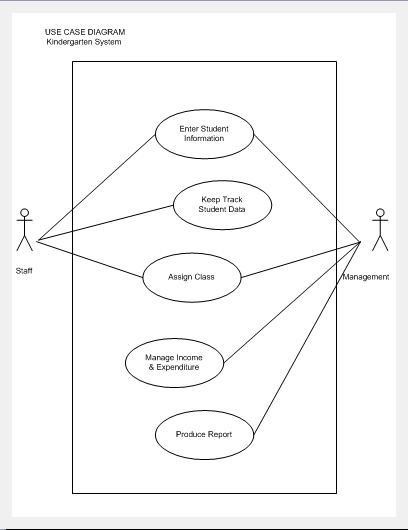


Figure 1.6

Table1.3 use case description

|  |  |
| --- | --- |
| Use Case Description | |
| Use Case Name: | Enter information |
| Scenario: | Enter pupil information |
| Triggering Event: | Staff and management will enter the pupil data |
| Brief Description: | New pupil will register for the new intake registration. The staff or |
| management will enter the pupil data according to the services  offer. For existing student, the system will update the data. They will assign and keep track of the pupil information. |

|  |  |  |
| --- | --- | --- |
| Actors: | Staff and management | |
| Stakeholder: | Parents: to give information about the pupil including medical information | |
| Preconditions: | pupil must exist |  |
| Enter information for new pupil | |
| Retrieve information for existing  pupil | |
| Postconditions: | Update information for existing pupil | |
| Flow of Activities: | Actor | System |
| 1. Admin will enter pupil | 1.1 Retrieve data for existing pupil |
| information | 1.2 Create new entry for new pupil |
| 2.Admin/ Staff will  keep track on | 2.1 Update latest  information |
| pupil information | 3.1 Display the information |
|  |  |

# Class Diagram

Figure 1. 7: class diagram

The figure above describes the class diagram of the system showing the classes, their attributes, operation and the relationship between them.

# Database Design

Figure 1. 8: Entity Relationship Diagram

The ERD above represents the entities, their attributes and the relationship between them. Eachentity has a unique primary keyWhich is the attribute that uniquely identifies the entity.

# Dataflow Diagram (DFD)

Identify

Children

Identification

and Tracking

Report

Parents

Children

Demand

Staff

Figure 1.9 context Level DFD

The above context diagram shows the overall view of the system, the flow of data between entities.

1.Identity

Processing

2. Activities

monitoring and Analyzing

Server Database

Report

3.User Interface

Demand

Demand

Report

Parent

Staff

Figure 2.1Level 0 DFD

# Activity Diagram



prompt for user ID/

Password

<3

Enter User ID

<3

Verify Password

incorrect

Check number of

wrong enteries



Lock User ID out of system

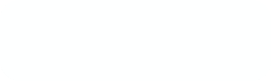
correct



Access granted

Figure 2.2 Login Activity Diagram

Request to enroll

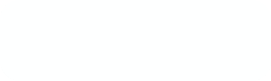


child

Check for vacancy

[>50]

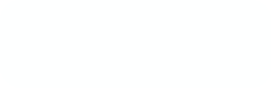
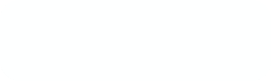
No vacancy



count pupils

vacancy

[< 50]



Add child

Record personal

information

Send confirmation to

parent



Dont enroll

Figure 2.3: Enrollment Activity Diagram

The above activity diagram shows the workflow within the system using, activities, actions, decisions…

# 3.6 Summary

This chapter describes the analysis and design of the system requirements and the design of the system that were developed using different UML diagrams and Dataflow diagram.

# Overview

**CHAPTER FOUR IMPLEMENTATION AND TESTING**

This chapter shows the implementation and testing of the Day care information management system, the technologies and tools used during the implementation of the system, the problems and errors encountered during the implementation and the solutions to the problems. This chapter also shows the implementation and functionalities of the system.

# Development tools and technologies

The following are the tools and technologies used for the development of the system.

* + - A laptop with 1.9GHz clock speed and 16GB RAM
    - Visual studio 2019 as IDE
    - Microsoft SQL Server Management Studio 18

# Implementation Problems

During the implementation of this desktop app, the following problems were encountered, the problems were eventually resolve. The issues has to do with both front-end and bank-end implementation.



Figure 2.4

This issue was encountered during the creation of a connection between the database and the application.

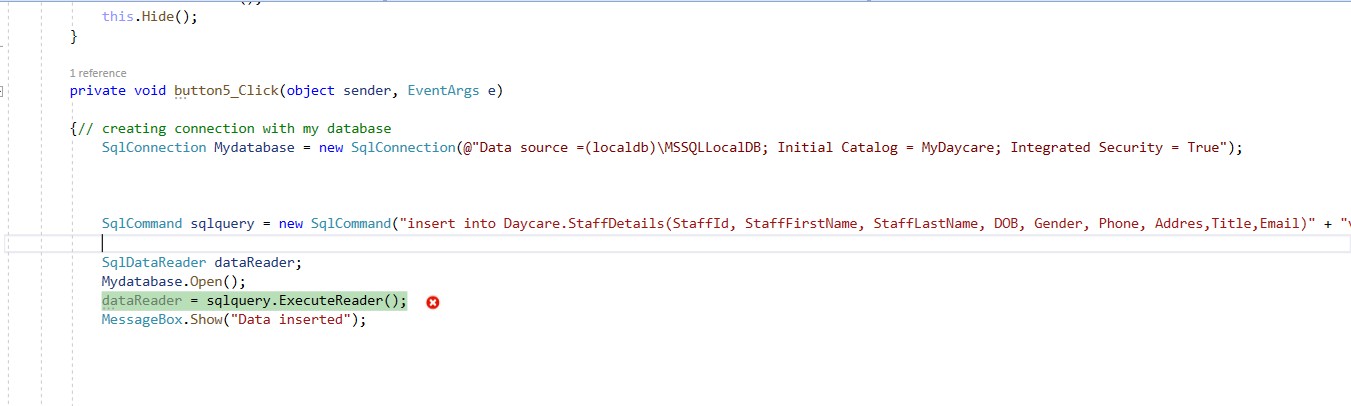


Figure 2. 5

this problem occurred due to conversion of data type. The conversion of a varchar data type to a date time data type resulted in an out-of-range value.

# Overcoming Implementation Problems`

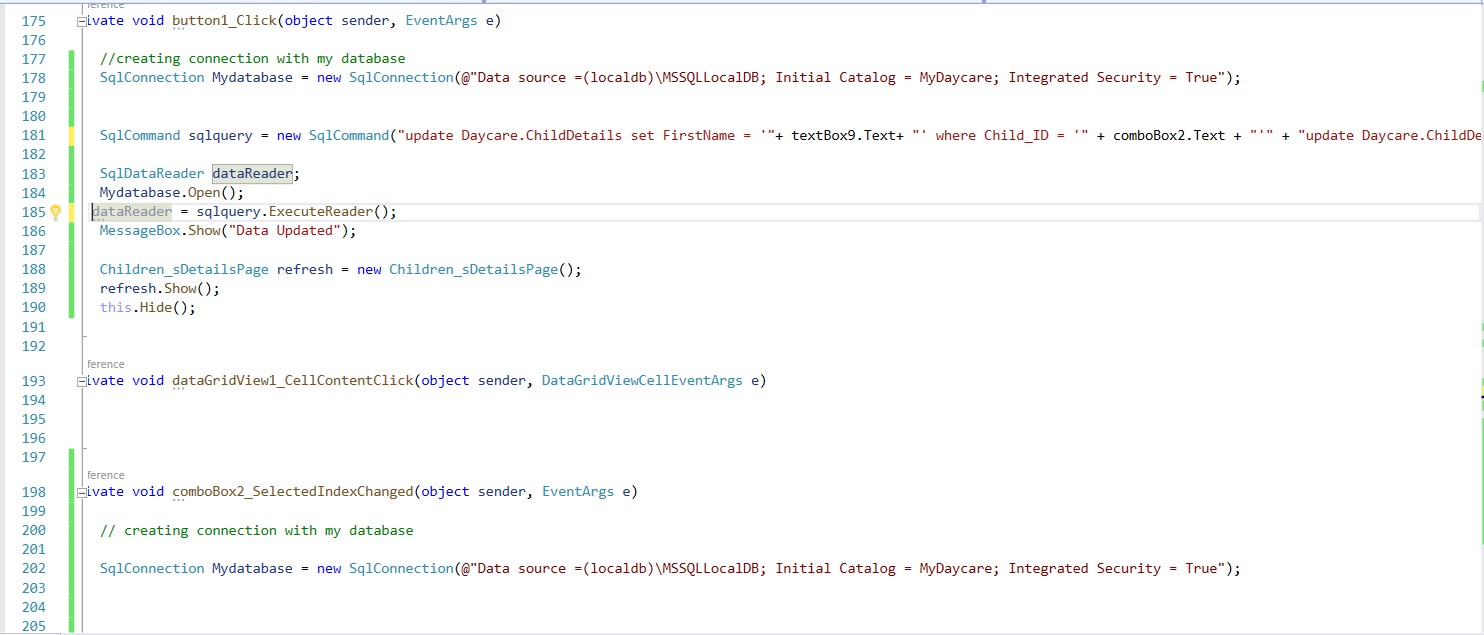


Figure 2. 6

Solution database connection problem



Figure 2.7Solution to data conversion problem

# Testing

This section shows the testing process carried out to check the application for errors. All functionalities were contained in the test plan to test for checking errors. System testing also checks to see if the system satisfies its requirements or not.

# Test Plans

Table1 4

|  |  |
| --- | --- |
| S/N | Administrator |
| 1 | Admin should be able to sign to the system |
| 2 | Admin should be able to sign out of the system |
| 3 | Admin should be able to edit his/her account |
| 4 | Admin should be able to manage users accounts |
| 5 | Admin should be able to manage user roles |
| 6 | Admin should be able to create a staff, Child or Parent Account |
| 7 | Admin should be able to manage Child, staff or parent details |
|  |  |
|  | Staff |

|  |  |
| --- | --- |
| 8 | Staff should be able to sign into the system |
| 9 | Staff should be able to add new child user |
| 10 | Staff should be able to manage attendance |
| 11 | Staff should be able to manage daily activity log |
| 12 | Staff should be able to add new parent |
|  |  |
|  | System |
| 13 | System should authenticate user login credentials |
| 14 | System should be able to hash user’s password |
| 15 | System should be able to check the role of a user to let him/her use some  functionalities |

# Test Traceability Matrix

Table1 5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Req.  No. | Description | Priority | Test Date | Test  Result |
| R-1 | Admin should be able to sign to the system | High | 7th September,  2020 | Pass |
| R-2 | Admin should be able to sign out of the system | High | 7th  September, 2020 | Pass |
| R-3 | Admin should be able to edit his/her account | High | 7th  September, 2020 | Pass |
| R-4 | Amin should be able to manage users accounts | High | 7th  September, 2020 | Pass |
| R-5 | Admin should be able to manage user roles | High | 7th  September, 2020 | Pass |
| R-6 | Admin should be able to create a staff, child or | High | 7th | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | parent account |  | September,  2020 |  |
| R-7 | Admin should be able to manage details staff, child or parent account | High | 7th  September, 2020 | Pass |
| R-8 | User should be able to login after account creation | High | 7th  September, 2020 | Pass |
| R-9 | Staff should be able to add a Child to the system | High | 7th  September, 2020 | Pass |
| R-10 | Staff should be able to manage child details | High | 7th  September, 2020 | Pass |
| R-11 | Staff should be able to manage child attendance | High | 7th  September, 2020 | pass |
| R-12 | Staff should be able to manage child daily activity log | High | 7th  September, 2020 | pass |
| R-13 | System should authenticate user login credentials | High | 7th  September, 2020 | Pass |
| R-14 | System should be able to hash user’s password | High | 7th  September, 2020 | Failed |
| R-15 | System should be able to check the role of a user to let him/her use some functionalities | High | 7th  September, 2020 | Pass |

# Test Report Summary

Table1 6

|  |  |
| --- | --- |
| Summary of Test Carried | Results |
| Tests carried out | 14 |
| Tests not carried out | 0 |
| Tests passed | 13 |
| Test failed | 1 |
| Percentage of tests carried out | 90 |
| Percentage of tests not carried out | 10 |
| Percentage of tests passed | 90 |
| Percentage of tests failed | 10 |

# User Guide

The user guide provides guidance on how to use the system to the user. It includes all methods for operating the system. It also provides a reference to difference aspects of the system. The user guide can be found in appendix E.

# Summary

This chapter shows the system implementation, the tools and technologies used during the implementation. This chapter also covers the problems that were encountered during the implementation and how the problems were solved, test plans, test traceability matrix and test report summary as well.

# CHAPTER FIVE

**DISCUSSION, CONCLUSION, AND RECOMMENDATIONS**

# Overview

This chapter covers the evaluation of the project, recommendations and conclusion. It presents the objective assessment which is related with the aim of the project and also the context of the project that gives the general conclusion of the project. The chapter further discusses the limitations of the project, the challenge encountered, future enhancement and recommendations.

# Evaluation

* + 1. **Objective Assessment**

The Daycare Information Management system has accomplished some its aims, at this point, the application can:

Allow Admin, Staff and parents to login and out of the system. Allow Admin to manage other users.

Allow the administrator to add and manage staffs, children, and parents accounts.

Enable the staff to manage children details including records of allergies and daily activities. Enable Admin and staff to manage attendance.

Enable admin to add new user and also assign username and password.

# Limitations and challenges

This system is developed for a Nigerian based day care, the system is therefore limited within Nigerian as it is developed to follow the Nigerian daycare processes. For a user to able to the system, an account must be created and a username and password must be assigned to that user.

During the analysis and implementation of this system, a couple of challenges were faced. When analyzing the system requirement, it was difficult to understand what the stakeholder wanted because they probably didn’t know what they as the outcome of the system.

This challenged was solved by applying skills like; problem solving skill, good communication skill as well as critical thinking.

# Future Enhancements

The system can be further enhanced by making it both a web and mobile application. This will make it easy and more efficient as it can be used by multiple devices at the same time.

# Recommendation

To make the system more efficient, below are some recommendations and future works.

As education is central to development, there should be availability of facility that will stakeholders participate in school improvement programs and decision making. To facilitate easy information access to such bodies, the desktop application could be further enhanced by integrating additional reports required by daycares or preschools.

# Conclusion

In here, I would like to conclude my work. The project Daycare Management Information System is a computerized Package designed to fulfil all the required task and activities in the Daycare. It is a great improvement over the manual system.

# REFERENCES

Woods, K. (2019, June 12). Company: PowerSchool. Retrieved June m13, 2019, from PowerSchool Website:

<https://www.powerschool.com/company/about-us/>

Ezcare care childcare software (2019, June 13). About Us retrieved from Ezcare website: <https://www.ezcaresoftware.com/client-services/>

The preschool partner (2019 June, 12). About us retrieved from preschool partner Website:<https://www.preschool-partners.org/about>

Ali, A. (2019, june 12). Kindergarten System. (A. Mohamed, Interviewer) Amber, l. (2009). The Systems. Sydney: The Globe and Mail.

Henderson, T. (2019, June 13). Home: Elvis. Retrieved June 13, 2018, from Elvis Website: <http://www.elviserp.com/Home>

Thomas, H. (2011). Information Systems. London: Olympia Publishers. Woods, K. (2017, June 12). Company:

Woods, K. (2019, June 12). Company: PowerSchool. Retrieved June 13, 2019, from PowerSchool Website:

<https://www.powerschool.com/company/about-us/>

Ezcare care childcare software (2019, June 13). About Us retrieved from Ezcare website: <https://www.ezcaresoftware.com/client-services/>

Ladjamudin, A. (2005). Analysis and Design of Information Systems. Yogyakarta: Science House.

The preschool partner (2019 June, 12). About us retrieved from preschool partner Website:<https://www.preschool-partners.org/about>

*Connolly, T. & Begg, C. (2002) Database System: A practical Approach to Design, Implementation, and Management, (3rd Edition). England: Addison Wesley.*

# APPENDICES

Appendix A – Project Management Document

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Task Name | Duration | Start | Finish | Deliverable |
| 1 | Project Plan | 7 days | 3rd March, 2020 | 10th March, 2020 | Work breakdown structure and Gantt Chart |
|  |  |  |  |  |  |
| 2 | Introduction | 14 days | 10th March, 2020 | 24th March,2020 | Introductory Chapter |
|  |  |  |  |  |  |
| 3 | Literature Review | 30 days | 24th March, 2020 | 24th April, 2020 | Reviewed Literature |
|  |  |  |  |  |  |
| 4 | System Analysis and Design | 30 days | 24th April, 2020 | 24th May, 2020 | System requirement specification which includes functional requirements, non-functional requirements and their models (Activity Diagram, Class Diagram, Use Case, Data flow diagram, Entity Relationship Diagram, etc.) |
|  |  |  |  |  |  |
| 5 | Implementation | 134 days | 26th May, 2020 | 31st August, 2020 | Coding |
|  |  |  |  |  |  |
| 6 | Testing | 8 days | 1st  September, 2020 | 8th September, 2020 | System Testing, Unit Testing, Integration Testing |
|  |  |  |  |  |  |

Appendix B: Interview questions

Q1. What are the objectives of your organization? Q2. What is the mission of your organization?

Q3. How many branches does your organization have? Q4. How many staffs do you have?

Q5. How does your current system work? Q6. What are the duties of a staff?

Q7. Where do you keep Staff and children information?

Q8. How do you keep track children attendance and daily activities? Q9. How do you generate Staff and children account?

Appendix C: Source codes Admin main page

privatevoid button2\_Click\_1(object sender, EventArgs e)

{

// creating new instance to load Staff page StaffDetails Staff = newStaffDetails(); Staff.Show();

this.Hide();

}

privatevoid button3\_Click(object sender, EventArgs e)

{

// creating new instance to load SignIn page Sign\_InPage SignIn = new Sign\_InPage(); SignIn.Show();

this.Hide();

}

privatevoid button4\_Click(object sender, EventArgs e)

{

// creating new instance to load admin attendance

StaffChildAttendanceMngt AdminAttendance = newStaffChildAttendanceMngt(); AdminAttendance.Show();

this.Hide();

}

privatevoid button5\_Click\_1(object sender, EventArgs e)

{

// creating new instance to load admin child details

Children\_sDetailsPage ChildDetails = new Children\_sDetailsPage(); ChildDetails.Show();

this.Hide();

}

privatevoid button1\_Click(object sender, EventArgs e)

{

AdminRegister register = newAdminRegister(); register.Show();

this.Hide();

}

privatevoid button6\_Click(object sender, EventArgs e)

{

// creating new instance to load page

Children\_sActivityLog ActivityLog = new Children\_sActivityLog(); ActivityLog.Show();

this.Hide();

}

privatevoid button7\_Click(object sender, EventArgs e)

{

// creating new instance to load Home Sign\_InPage sign\_In = new Sign\_InPage(); sign\_In.Show();

this.Hide();

}

Admin profile

namespace WindowsFormsApp1

{

publicpartialclassAdminProfile : Form

{

publicAdminProfile()

{

InitializeComponent();

}

privatevoid button2\_Click(object sender, EventArgs e)

{

Parent parentPage = newParent(); parentPage.Show();

this.Hide();

}

privatevoid textBox7\_TextChanged(object sender, EventArgs e)

{

}

privatevoid textBox3\_TextChanged(object sender, EventArgs e)

{

}

privatevoid AdminProfile\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'mydaycareDataSet28.StaffDetails' table. You can move, or remove it, as needed. this.staffDetailsTableAdapter.Fill(this.mydaycareDataSet28.StaffDetails);

comboBox1.Text = Sign\_InPage.mystring;

Admin Registration page namespace WindowsFormsApp1

{

publicpartialclassAdminRegister : Form

{

publicAdminRegister()

{

InitializeComponent();

}

privatevoid label5\_Click(object sender, EventArgs e)

{

}

privatevoid label8\_Click(object sender, EventArgs e)

{

}

privatevoid button5\_Click(object sender, EventArgs e)

{

// creating connection with my database

SqlConnection Mydatabase = newSqlConnection(@"Data source

=(localdb)\MSSQLLocalDB; Initial Catalog = MyDaycare; Integrated Security = True");

SqlCommand sqlquery = newSqlCommand("insert into Daycare.Login(username, password, Role)" + "values('" + textBox5.Text + "','" + textBox2.Text + "', '" + comboBox1.Text + "')", Mydatabase);

SqlDataReader dataReader; Mydatabase.Open();

dataReader = sqlquery.ExecuteReader(); MessageBox.Show("User Created"); AdminRegister refresh = newAdminRegister();

refresh.Show(); this.Hide();

}

Child Activity Log

namespace WindowsFormsApp1

{

publicpartialclassChildren\_sActivityLog : Form

{

publicChildren\_sActivityLog()

{

InitializeComponent();

textBox5.Text = DateTime.Now.ToLongTimeString(); textBox2.Text = DateTime.Now.ToLongTimeString();

textBox3.Text = DateTime.Now.ToLongTimeString(); textBox4.Text = DateTime.Now.ToLongTimeString();

}

privatevoid panel1\_Paint(object sender, PaintEventArgs e)

{

}

privatevoid label8\_Click(object sender, EventArgs e)

{

}

privatevoid label9\_Click(object sender, EventArgs e)

{

}

privatevoid Children\_sActivityLog\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'mydaycareDataSet10.ChildActivityLog' table. You can move, or remove it, as needed. this.childActivityLogTableAdapter1.Fill(this.mydaycareDataSet10.ChildActivityLog);

// TODO: This line of code loads data into the 'mydaycareDataSet9.ChildDetails' table. You can move, or remove it, as needed. this.childDetailsTableAdapter.Fill(this.mydaycareDataSet9.ChildDetails);

}

privatevoid label2\_Click(object sender, EventArgs e)

{

}

privatevoid label1\_Click(object sender, EventArgs e)

{

}

privatevoid pictureBox5\_Click(object sender, EventArgs e)

{

}

privatevoid pictureBox4\_Click(object sender, EventArgs e)

{

}

privatevoid pictureBox3\_Click(object sender, EventArgs e)

{

}

privatevoid button3\_Click(object sender, EventArgs e)

{

// creating new instance to load staff dashboard StaffDashboard StaffDash = newStaffDashboard(); StaffDash.Show();

this.Hide();

}

privatevoid button5\_Click(object sender, EventArgs e)

{

// creating connection with my database

SqlConnection Mydatabase = newSqlConnection(@"Data source

=(localdb)\MSSQLLocalDB; Initial Catalog = MyDaycare; Integrated Security = True");

SqlCommand sqlquery = new SqlCommand("insert into Daycare.ChildActivityLog( ChildName, Date, MealTime, NapTime, learningTime, DrugAdministrationTime)" + "values('" + comboBox2.Text + "', '" + dateTimePicker5.Text + "', '" + textBox5.Text + "','" + textBox2.Text + "', '" + textBox3.Text + "','" + textBox4.Text + "')", Mydatabase);

SqlDataReader dataReader; Mydatabase.Open();

dataReader = sqlquery.ExecuteReader(); MessageBox.Show("Data inserted");

Children\_sActivityLog refresh = new Children\_sActivityLog(); refresh.Show();

this.Hide();

}

privatevoid textBox5\_TextChanged(object sender, EventArgs e)

{

}

privatevoid button1\_Click(object sender, EventArgs e)

{

}

}

}

Children details page

namespace WindowsFormsApp1

{

publicpartialclassChildren\_sDetailsPage : Form

{

publicChildren\_sDetailsPage()

{

InitializeComponent();

}

privatevoid button3\_Click(object sender, EventArgs e)

{

// creating new instance to load admin main page AdminMainPage AdminHome = newAdminMainPage(); AdminHome.Show();

this.Hide();

}

privatevoid Children\_sDetailsPage\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'mydaycareDataSet27.ChildDetails' table. You can move, or remove it, as needed. this.childDetailsTableAdapter3.Fill(this.mydaycareDataSet27.ChildDetails);

// TODO: This line of code loads data into the 'mydaycareDataSet26.ChildDetails' table. You can move, or remove it, as needed. this.childDetailsTableAdapter2.Fill(this.mydaycareDataSet26.ChildDetails);

}

privatevoid panel1\_Paint(object sender, PaintEventArgs e)

{

}

privatevoid textBox6\_TextChanged(object sender, EventArgs e)

{

}

privatevoid textBox8\_TextChanged(object sender, EventArgs e)

{

}

privatevoid textBox5\_TextChanged(object sender, EventArgs e)

{

}

privatevoid textBox4\_TextChanged(object sender, EventArgs e)

{

}

privatevoid textBox3\_TextChanged(object sender, EventArgs e)

{

}

privatevoid label7\_Click(object sender, EventArgs e)

{

}

privatevoid label10\_Click(object sender, EventArgs e)

{

}

privatevoid label11\_Click(object sender, EventArgs e)

{

}

privatevoid label12\_Click(object sender, EventArgs e)

{

}

privatevoid label3\_Click(object sender, EventArgs e)

{

}

privatevoid label5\_Click(object sender, EventArgs e)

{

}

privatevoid label9\_Click(object sender, EventArgs e)

{

}

privatevoid label13\_Click(object sender, EventArgs e)

{

}

privatevoid dateTimePicker1\_ValueChanged(object sender, EventArgs e)

{

}

privatevoid button2\_Click(object sender, EventArgs e)

{

}

privatevoid textBox1\_TextChanged(object sender, EventArgs e)

{

}

privatevoid label15\_Click(object sender, EventArgs e)

{

}

privatevoid textBox14\_TextChanged(object sender, EventArgs e)

{

}

privatevoid pictureBox1\_Click(object sender, EventArgs e)

{

}

privatevoid pictureBox2\_Click(object sender, EventArgs e)

{

}

privatevoid button4\_Click(object sender, EventArgs e)

{

}

privatevoid button5\_Click(object sender, EventArgs e)

{

string Information = comboBox2.Text + " " + textBox9.Text + " " + textBox6.Text; textBox10.Text = Information;

// creating connection with my database

SqlConnection Mydatabase = newSqlConnection(@"Data source

=(localdb)\MSSQLLocalDB; Initial Catalog = MyDaycare; Integrated Security = True");

SqlCommand sqlquery = new SqlCommand("insert into Daycare.ChildDetails(Child\_ID, FirstName,LastName,ParentName,DOB,Age,Gender,PhoneNum,adress,Allergy,ChildInform ation)" + "values('" + comboBox2.Text + "','" + textBox9.Text + "', '" + textBox8.Text + "','"

+ textBox6.Text + "', '" + dateTimePicker1.Text + "', '" + textBox7.Text + "','" + textBox5.Text + "', '" + textBox3.Text + "' , '" + textBox2.Text + "','" + textBox4.Text + "' ,'"

+ textBox10.Text + "' )", Mydatabase);

SqlDataReader dataReader; Mydatabase.Open();

dataReader = sqlquery.ExecuteReader(); MessageBox.Show("Data Added");

Children\_sDetailsPage refresh = new Children\_sDetailsPage(); refresh.Show();

this.Hide();

}

privatevoid textBox7\_TextChanged(object sender, EventArgs e)

{

}

privatevoid button1\_Click(object sender, EventArgs e)

{

//creating connection with my database

SqlConnection Mydatabase = newSqlConnection(@"Data source

=(localdb)\MSSQLLocalDB; Initial Catalog = MyDaycare; Integrated Security = True");

SqlCommand sqlquery = new SqlCommand("update Daycare.ChildDetails set FirstName = '"+ textBox9.Text+ "' where Child\_ID = '" + comboBox2.Text + "'" + "update Daycare.ChildDetails set LastName = '" + textBox8.Text + "' where Child\_ID = '" + comboBox2.Text + "'" + "update Daycare.ChildDetails set ParentName = '" + textBox6.Text

+ "' where Child\_ID = '" + comboBox2.Text + "'" + "update Daycare.ChildDetails set DOB = '" + dateTimePicker1.Text + "' where Child\_ID = '" + comboBox2.Text + "'" + "update Daycare.ChildDetails set Age = '" + textBox7.Text + "' where Child\_ID = '" + comboBox2.Text + "'" + "update Daycare.ChildDetails set Gender = '" + textBox5.Text + "' where Child\_ID = '" + comboBox2.Text + "'" + "update Daycare.ChildDetails set PhoneNum

= '" + textBox3.Text + "' where Child\_ID = '" + comboBox2.Text + "'" + "update Daycare.ChildDetails set adress = '" + textBox2.Text + "' where Child\_ID = '" + comboBox2.Text + "'" + "update Daycare.ChildDetails set Allergy = '" + textBox4.Text + "' where Child\_ID = '" + comboBox2.Text + "'" + "update Daycare.ChildDetails set ChildInformation = '" + textBox10.Text + "' where Child\_ID = '" + comboBox2.Text + "'", Mydatabase);

SqlDataReader dataReader; Mydatabase.Open();

dataReader = sqlquery.ExecuteReader(); MessageBox.Show("Data Updated");

Children\_sDetailsPage refresh = new Children\_sDetailsPage(); refresh.Show();

this.Hide();

}

privatevoid dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

privatevoid comboBox2\_SelectedIndexChanged(object sender, EventArgs e)

{

// creating connection with my database

SqlConnection Mydatabase = newSqlConnection(@"Data source

=(localdb)\MSSQLLocalDB; Initial Catalog = MyDaycare; Integrated Security = True");

SqlCommand sqlquery = newSqlCommand("Select \* from Daycare.ChildDetails where Child\_ID = '"+ comboBox2.Text+"' ", Mydatabase);

SqlDataReader dataReader; Mydatabase.Open();

dataReader = sqlquery.ExecuteReader(); while (dataReader.Read())

{

string Child\_ID = dataReader.GetInt32(0).ToString(); string FirstName = dataReader.GetString(1);

string Lastname = dataReader.GetString(2); string Parentname = dataReader.GetString(3);

string DOB = dataReader.GetDateTime(4).ToShortDateString(); string Age = dataReader.GetInt32(5).ToString();

string Gender = dataReader.GetString(6);

string PhoneNum = dataReader.GetInt32(7).ToString(); string Adress = dataReader.GetString(8);

string Allergy = dataReader.GetString(9);

string ChildInformation = dataReader.GetString(10);

comboBox2.Text = Child\_ID; textBox9.Text = FirstName; textBox8.Text = Lastname; textBox6.Text = Parentname; dateTimePicker1.Text = DOB;

textBox7.Text = Age; textBox5.Text = Gender; textBox3.Text = PhoneNum; textBox2.Text = Adress; textBox4.Text = Allergy; textBox10.Text = ChildInformation;

Home Page

namespace WindowsFormsApp1

{

publicpartialclassHome : Form

{

publicHome()

{

InitializeComponent();

}

privatevoid button5\_Click(object sender, EventArgs e)

{

// creating new instance to load Signin page Sign\_InPage SignIn = new Sign\_InPage(); SignIn.Show();

this.Hide();

}

privatevoid button3\_Click(object sender, EventArgs e)

{

// creating new instance to load signin page Sign\_InPage SignIn = new Sign\_InPage(); SignIn.Show();

this.Hide();

}

privatevoid button6\_Click(object sender, EventArgs e)

{

// creating new instance to load sign in page Sign\_InPage SignIn = new Sign\_InPage(); SignIn.Show();

this.Hide();

}

privatevoid pictureBox5\_Click(object sender, EventArgs e)

{

}

privatevoid button7\_Click(object sender, EventArgs e)

{

// creating new instance to load Signin page Sign\_InPage SignIn = new Sign\_InPage(); SignIn.Show();

this.Hide();

}

privatevoid button4\_Click(object sender, EventArgs e)

{

// creating new instance to load Signin page Sign\_InPage SignIn = new Sign\_InPage(); SignIn.Show();

this.Hide();

}

privatevoid button2\_Click(object sender, EventArgs e)

{

// creating new instance to load Sign in page Sign\_InPage SignIn = new Sign\_InPage(); SignIn.Show();

this.Hide();

}

privatevoid panel1\_Paint(object sender, PaintEventArgs e)

{

}

privatevoid button1\_Click(object sender, EventArgs e)

{

//loads signin page

Sign\_InPage sign\_In = new Sign\_InPage(); sign\_In.Show();

this.Hide();

}

privatevoid Home\_Load(object sender, EventArgs e)

{

}

}

}

Sign-in page

namespace WindowsFormsApp1

{

publicpartialclassSign\_InPage : Form

{

publicSign\_InPage()

{

InitializeComponent();

}

publicstaticstring mystring = "";

privatevoid Sign\_InPage\_Load(object sender, EventArgs e)

{

}

privatevoid button1\_Click(object sender, EventArgs e)

{

SqlConnection Mydatabase = newSqlConnection(@"Data source

=(localdb)\MSSQLLocalDB; Initial Catalog = MyDaycare; Integrated Security = True"); Mydatabase.Open();

string query = ("select count (\*) from Daycare.Login where username = '" + Username.Text

+ "' and password = '" + Password.Text + "' and Role = '" + Role.Text + "'"); SqlDataAdapter sqlquery = newSqlDataAdapter(query, Mydatabase);

DataTable dataContainer = newDataTable(); sqlquery.Fill(dataContainer);

if (dataContainer.Rows[0][0].ToString() == "1")

{

if (Role.Text == "Admin”)

{

mystring = Username.Text;

AdminMainPage admin = newAdminMainPage(); admin.Show();

this.Hide();

}

elseif (Role.Text == "Staff")

{

mystring = Username.Text;

StaffDashboard staff = newStaffDashboard(); staff.Show();

this.Hide();

}

}

elseif (Role.Text == "parent")

{

Parent parentpage = newParent();

parentpage.Show(); this.Hide();

}

else

}

MessageBox.Show("Incorrect Username or Password");

privatevoid label2\_Click(object sender, EventArgs e)

{

}

privatevoid Username\_TextChanged(object sender, EventArgs e)

{

}

privatevoid Role\_SelectedIndexChanged(object sender, EventArgs e)

{

}

privatevoid pictureBox1\_Click(object sender, EventArgs e)

{

}

privatevoid label4\_Click(object sender, EventArgs e)

{

}

privatevoid button2\_Click(object sender, EventArgs e)

{

Home home = newHome(); home.Show();

this.Hide();

}

privatevoid textBox1\_TextChanged(object sender, EventArgs e)

{

}

privatevoid button3\_Click(object sender, EventArgs e)

{

}

}

}

Attendance management

namespace WindowsFormsApp1

{

publicpartialclassStaffChildAttendanceMngt : Form

{

publicStaffChildAttendanceMngt()

{

InitializeComponent();

textBox2.Text = DateTime.Now.ToLongTimeString(); textBox6.Text = DateTime.Now.ToLongTimeString(); textBox11.Text = DateTime.Now.ToLongTimeString(); textBox5.Text = DateTime.Now.ToLongTimeString();

}

privatevoid button3\_Click(object sender, EventArgs e)

{

// creating new instance to load admin main page AdminMainPage AdminHome = newAdminMainPage(); AdminHome.Show();

this.Hide();

}

privatevoid label2\_Click(object sender, EventArgs e)

{

}

privatevoid label3\_Click(object sender, EventArgs e)

{

}

privatevoid label10\_Click(object sender, EventArgs e)

{

}

privatevoid button1\_Click(object sender, EventArgs e)

{

}

privatevoid button2\_Click(object sender, EventArgs e)

{

}

privatevoid AdminAttendanceMngt\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'mydaycareDataSet19.ChildAttendance' table. You can move, or remove it, as needed. this.childAttendanceTableAdapter1.Fill(this.mydaycareDataSet19.ChildAttendance);

// TODO: This line of code loads data into the 'mydaycareDataSet18.StaffAttendance' table. You can move, or remove it, as needed. this.staffAttendanceTableAdapter2.Fill(this.mydaycareDataSet18.StaffAttendance);

// TODO: This line of code loads data into the 'mydaycareDataSet17.StaffAttendance' table. You can move, or remove it, as needed. this.staffAttendanceTableAdapter1.Fill(this.mydaycareDataSet17.StaffAttendance);

// TODO: This line of code loads data into the 'mydaycareDataSet16.StaffDetails' table. You can move, or remove it, as needed. this.staffDetailsTableAdapter.Fill(this.mydaycareDataSet16.StaffDetails);

// TODO: This line of code loads data into the 'mydaycareDataSet15.ChildDetails' table. You can move, or remove it, as needed. this.childDetailsTableAdapter.Fill(this.mydaycareDataSet15.ChildDetails);

// TODO: This line of code loads data into the 'mydaycareDataSet13.StaffAttendance' table. You can move, or remove it, as needed. this.staffAttendanceTableAdapter.Fill(this.mydaycareDataSet13.StaffAttendance);

// TODO: This line of code loads data into the 'mydaycareDataSet12.ChildAttendance' table. You can move, or remove it, as needed. this.childAttendanceTableAdapter.Fill(this.mydaycareDataSet12.ChildAttendance);

}

privatevoid label6\_Click(object sender, EventArgs e)

{

}

privatevoid button1\_Click\_1(object sender, EventArgs e)

{

//creating connection with my database

SqlConnection Mydatabase = newSqlConnection(@"Data source

=(localdb)\MSSQLLocalDB; Initial Catalog = MyDaycare; Integrated Security = True");

SqlCommand sqlquery = new SqlCommand("insert into Daycare.ChildAttendance(ChildId, ChildName, Dayoftheweek, dropoffPerson, DropoffTime, pickupTime, StaffName)" + "values('" + comboBox1.Text + "','" + textBox1.Text + "', '" + dateTimePicker2.Text + "', '" + textBox9.Text + "','" + textBox2.Text + "', '" + textBox6.Text

+ "', '" + textBox4.Text + "')", Mydatabase);

SqlDataReader dataReader; Mydatabase.Open();

dataReader = sqlquery.ExecuteReader();

MessageBox.Show("Checked In");

StaffChildAttendanceMngt staffChild = newStaffChildAttendanceMngt(); staffChild.Show();

this.Hide();

}

privatevoid label15\_Click(object sender, EventArgs e)

{

}

privatevoid label18\_Click(object sender, EventArgs e)

{

}

privatevoid button5\_Click(object sender, EventArgs e)

{

//creating connection with my database

SqlConnection Mydatabase = newSqlConnection(@"Data source

=(localdb)\MSSQLLocalDB; Initial Catalog = MyDaycare; Integrated Security = True");

SqlCommand sqlquery = newSqlCommand("insert into Daycare.StaffAttendance( StaffId, StaffName, SignInTime, SignOutTime, Dayoftheweek)" + "values('" + comboBox2.Text + "', '" + textBox12.Text + "', '" + textBox11.Text + "','" + textBox5.Text + "', '" + dateTimePicker1.Text + "')", Mydatabase);

SqlDataReader dataReader; Mydatabase.Open();

dataReader = sqlquery.ExecuteReader(); MessageBox.Show("Checked In");

StaffChildAttendanceMngt staffChild = newStaffChildAttendanceMngt(); staffChild.Show();

this.Hide();

}

privatevoid comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

//creating connection with my database

SqlConnection Mydatabase = newSqlConnection(@"Data source

=(localdb)\MSSQLLocalDB; Initial Catalog = MyDaycare; Integrated Security = True");

SqlCommand sqlquery = newSqlCommand("select \* from Daycare.ChildDetails where Child\_ID = '" + comboBox1.Text + "'", Mydatabase);

SqlDataReader dataReader; Mydatabase.Open();

dataReader = sqlquery.ExecuteReader();

while (dataReader.Read()) {

string ChildId = dataReader.GetInt32(0).ToString(); string ChildName = dataReader.GetString(1);

comboBox1.Text = ChildId; textBox1.Text = ChildName;

}

}

privatevoid comboBox2\_SelectedIndexChanged(object sender, EventArgs e)

{

//creating connection with my database

SqlConnection Mydatabase = newSqlConnection(@"Data source

=(localdb)\MSSQLLocalDB; Initial Catalog = MyDaycare; Integrated Security = True");

SqlCommand sqlquery = newSqlCommand("select \* from Daycare.StaffDetails where StaffId = '" + comboBox2.Text + "'", Mydatabase);

SqlDataReader dataReader; Mydatabase.Open();

dataReader = sqlquery.ExecuteReader();

while (dataReader.Read())

{

string StaffId = dataReader.GetInt32(0).ToString(); string StaffName = dataReader.GetString(1);

comboBox2.Text = StaffId; textBox12.Text = StaffName;

}

}

privatevoid button4\_Click(object sender, EventArgs e)

{

//creating connection with my database

SqlConnection Mydatabase = newSqlConnection(@"Data source

=(localdb)\MSSQLLocalDB; Initial Catalog = MyDaycare; Integrated Security = True");

SqlCommand sqlquery = newSqlCommand("update Daycare.ChildAttendance set pickupTime = '"+ textBox6.Text + "' where ChildId ='" + comboBox1.Text + "' ", Mydatabase);

SqlDataReader dataReader; Mydatabase.Open();

dataReader = sqlquery.ExecuteReader(); MessageBox.Show("Checked Out");

StaffChildAttendanceMngt staffChild = newStaffChildAttendanceMngt(); staffChild.Show();

this.Hide();

}

privatevoid button2\_Click\_1(object sender, EventArgs e)

{

//creating connection with my database

SqlConnection Mydatabase = newSqlConnection(@"Data source

=(localdb)\MSSQLLocalDB; Initial Catalog = MyDaycare; Integrated Security = True");

SqlCommand sqlquery = newSqlCommand("update Daycare.StaffAttendance set SignOutTime = '" + textBox5.Text + "' where StaffId ='" + comboBox2.Text + "' ", Mydatabase);

SqlDataReader dataReader; Mydatabase.Open();

dataReader = sqlquery.ExecuteReader(); MessageBox.Show(" Staff Checked Out");

StaffChildAttendanceMngt staffChild = newStaffChildAttendanceMngt(); staffChild.Show();

this.Hide();

}

} }

Staff Dashboard

namespace WindowsFormsApp1

{

publicpartialclassStaffDashboard : Form

{

publicStaffDashboard()

{

InitializeComponent();

}

privatevoid button3\_Click(object sender, EventArgs e)

{

// creating new instance to load staff dashboard StaffDetails StaffD = newStaffDetails(); StaffD.Show();

this.Hide();

}

privatevoid button3\_Click\_1(object sender, EventArgs e)

{

// creating new instance to load staff dashboard Sign\_InPage sign\_In = new Sign\_InPage(); sign\_In.Show();

this.Hide();

}

Staff details

{// creating connection with my database

SqlConnection Mydatabase = newSqlConnection(@"Data source

=(localdb)\MSSQLLocalDB; Initial Catalog = MyDaycare; Integrated Security = True");

SqlCommand sqlquery = new SqlCommand("insert into Daycare.StaffDetails(StaffId, StaffFirstName, StaffLastName, DOB, Gender, Phone, Addres,Title,Email)" + "values('" + comboBox1.Text + "', '" + Firstname.Text + "', '" + Lastname.Text + "', '" + dateTimePicker1.Text + "', '" + Gender.Text + "','" + phone.Text + "', '" + Address.Text + "', '" + title.Text + "' , '" + Email.Text + "')", Mydatabase);

SqlDataReader dataReader; Mydatabase.Open();

dataReader = sqlquery.ExecuteReader(); MessageBox.Show("Data Added");

StaffDetails refresh = newStaffDetails();

refresh.Show(); this.Hide();

}

privatevoid StaffDetails\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'mydaycareDataSet22.StaffDetails' table. You can move, or remove it, as needed. this.staffDetailsTableAdapter1.Fill(this.mydaycareDataSet22.StaffDetails);

// TODO: This line of code loads data into the 'mydaycareDataSet.StaffDetails' table. You can move, or remove it, as needed. this.staffDetailsTableAdapter.Fill(this.mydaycareDataSet.StaffDetails);

}

privatevoid phone\_TextChanged(object sender, EventArgs e)

{

}

privatevoid title\_TextChanged(object sender, EventArgs e)

{

}

privatevoid button1\_Click(object sender, EventArgs e)

{

//creating connection with my database

SqlConnection Mydatabase = newSqlConnection(@"Data source

=(localdb)\MSSQLLocalDB; Initial Catalog = MyDaycare; Integrated Security = True");

SqlCommand sqlquery = new SqlCommand("update Daycare.StaffDetails set StaffFirstName = '" + Firstname.Text + "' where StaffId = '" + comboBox1.Text + "'" + "update Daycare.StaffDetails set StaffLastName = '" + Lastname.Text + "' where StaffId = '"

+ comboBox1.Text + "'" + "update Daycare.StaffDetails set DOB = '" + dateTimePicker1.Text + "' where StaffId = '" + comboBox1.Text + "'" + "update Daycare.StaffDetails set Gender = '" + Gender.Text + "' where StaffId = '" + comboBox1.Text + "'" + "update Daycare.StaffDetails set Phone = '" + phone.Text + "' where StaffId = '" + comboBox1.Text + "'" + "update Daycare.StaffDetails set Addres = '" + Address.Text + "' where StaffId = '" + comboBox1.Text + "'" + "update Daycare.StaffDetails set Title = '" + title.Text + "' where StaffId = '" + comboBox1.Text + "'" + "update Daycare.StaffDetails set Email = '" + Email.Text + "' where StaffId = '" + comboBox1.Text + "'", Mydatabase);

SqlDataReader dataReader;

Mydatabase.Open();

dataReader = sqlquery.ExecuteReader(); MessageBox.Show("Data Updated");

StaffDetails refresh = newStaffDetails(); refresh.Show();

this.Hide();

}

privatevoid comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

// creating connection with my database

SqlConnection Mydatabase = newSqlConnection(@"Data source

=(localdb)\MSSQLLocalDB; Initial Catalog = MyDaycare; Integrated Security = True");

SqlCommand sqlquery = newSqlCommand("Select \* from Daycare.StaffDetails where StaffId = '" + comboBox1.Text + "' ", Mydatabase);

SqlDataReader dataReader; Mydatabase.Open();

dataReader = sqlquery.ExecuteReader(); while (dataReader.Read())

{

string StaffId = dataReader.GetInt32(0).ToString(); string FirstName = dataReader.GetString(1);

string LastName = dataReader.GetString(2);

string DOB = dataReader.GetDateTime(3).ToShortDateString(); string gender = dataReader.GetString(4);

string PhoneNum = dataReader.GetInt32(5).ToString(); string Addres = dataReader.GetString(6);

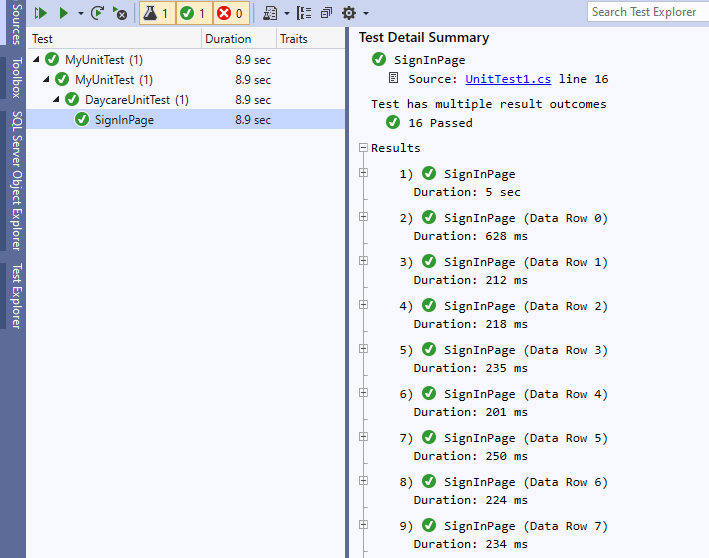
string Title = dataReader.GetString(7); string email = dataReader.GetString(8);

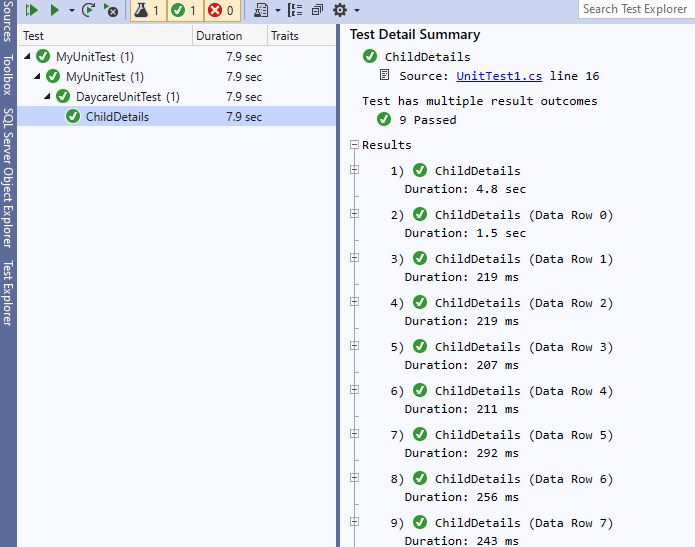
comboBox1.Text = StaffId; Firstname.Text = FirstName; Lastname.Text = LastName; dateTimePicker1.Text = DOB; Gender.Text = gender; Phone.Text = PhoneNum; Address.Text = Addres;

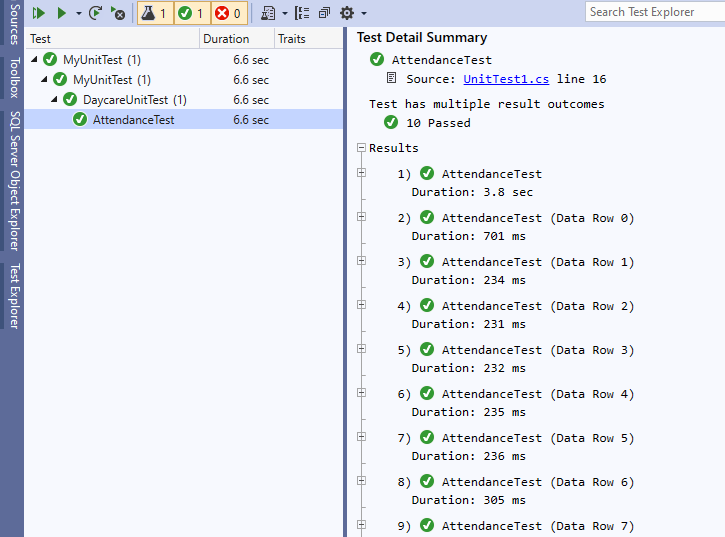
title.Text = Title;

Email.Text = email;

# Appendix D: Test cases







**Appendix E: User Guide**

# system overview

This application is a desktop application which means it can only run on PCs. The system functions in a way that a user will have to register with the system, registration can only be done by the admin, followed by sign-in. After signing into the system, the user will see home page where he/she can decide on which activity to perform.

* 1. organization of the guide

the user guide consists of the following sections: System summary, which shows the functions of the system’s hardware and software requirements, system configuration and system behavior.

# System summary

the summary shows the general overview of the system. It outlines the which shows the functions of the system’s hardware and software requirements, system configuration and system behavior.

# System configuration

The daycare system can operate on any desktop device. It does not require any internet service to save, add or update data contained in the system.

# User Access Levels

A user must be registered in the system to be able to use the system. For instance, a username and password be generated for the user.

# Getting started

This section describes how to access and use the system.

# Searching and Signing into the system

After launching the system, it can be searched on the PC by searching for it through the windows search, after which a user can sign into the system after attaining a username and password.

# Appendix F: User Interfaces

