

**Final Project Report**

for

**CELS Management System**

By

Peaky Blinders

WD 08

Client - Cambridge English Language School

Version 1.0

Prepared by

H G L S Nawarathna

Project Manager of Peaky Blinders

# Declaration

We hereby declare that the project work entitled “CELS Management System”, submitted to the SLIIT Academy (Pvt.) Ltd. a subsidiary of Sri Lanka Institute of Information Technology is a record of an original work done by us, under the guidance of our lecturer in charge Mr. D.M.R.P. Jayawardana and supervisor Ms. Lakna Gammedda. This project work is submitted in the partial fulfillment of the requirement for the award of the Diploma in Information Technology. The Results embodied in this report have not been submitted to any other University or Institution for the award of any degree or diploma. Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.

Date of Submission: 29th November 2020

Name of the supervisor: Ms. Lakna Gammedda

………………………………………………

Signature of the Supervisor.

# Acknowledgement

First and foremost, we would like to thank our lecturer in charge Mr. Roshan Jayawardana and supervisor Ms. Lakna Gammedda for your assistance and dedicated involvement in every step throughout the process. Without your active guidance, help and encouragement we would not have made the headway in the project.

We are extremely thankful for the Principle and the staff of Cambridge English Language School who provided us with the opportunity to develop this project with regard to their institution. They have been unwavering in their personal and professional support during requirement gathering sessions.

We would also like to show gratitude to all our lecturers /instructors at SLIIT Academy who have always supported us in our academics.

Most importantly, none of this could have happened without our families that has been kind and supportive to us over the project development period.

At last but not least gratitude goes to all our friends who directly and indirectly supported us in completing this project.

Thank you.

Table of Contents

[Declaration ii](#_Toc57570660)

[Acknowledgement iii](#_Toc57570661)

[List of Figures vi](#_Toc57570662)

[List of Tables vi](#_Toc57570663)

[Acronyms and Abbreviations ix](#_Toc57570664)

[Revision History x](#_Toc57570665)

[Group Members x](#_Toc57570666)

[1 Executive Summary 1](#_Toc57570667)

[2 Project Background 2](#_Toc57570668)

[2.1 Introduction 2](#_Toc57570669)

[2.2 Current System 2](#_Toc57570670)

[2.3 System to be Developed 3](#_Toc57570671)

[3 Problem Definition and Requirement Analysis 5](#_Toc57570672)

[3.1 Problem Definition 5](#_Toc57570673)

[3.1.1 People Affected 5](#_Toc57570674)

[3.1.2 Limitations of Current System 5](#_Toc57570675)

[3.2 Requirement Analysis 6](#_Toc57570676)

[3.2.1 Gathering Requirements 6](#_Toc57570677)

[3.2.2 General Objectives 6](#_Toc57570678)

[3.2.3 Specified Objectives 7](#_Toc57570679)

[4 Design and Implementation 8](#_Toc57570680)

[4.1 High Level Architecture Design 8](#_Toc57570681)

[4.2 Hardware, Software and Communication Interfaces 9](#_Toc57570682)

[4.2.1 Hardware Requirements 9](#_Toc57570683)

[4.2.2 Software Requirements 9](#_Toc57570684)

[4.2.3 Liveware Requirements 9](#_Toc57570685)

[4.3 Non-Functional Requirements 10](#_Toc57570686)

[4.3.1 Operational Requirements 10](#_Toc57570687)

[4.3.2 Performance Requirements 10](#_Toc57570688)

[4.3.3 Security Requirements 10](#_Toc57570689)

[4.4 Implementation 10](#_Toc57570690)

[4.4.1 Development Methodology 10](#_Toc57570691)

[4.4.2 Development Technologies 12](#_Toc57570692)

[4.4.3 Development Tools 13](#_Toc57570693)

[5 Testing and Deployment 14](#_Toc57570694)

[5.1 Login 14](#_Toc57570695)

[5.2 Create Profile 18](#_Toc57570696)

[5.3 Student Register 27](#_Toc57570697)

[5.4 Add Staff Member 42](#_Toc57570698)

[5.5 Add Class 49](#_Toc57570699)

[5.6 New Course 51](#_Toc57570700)

[5.7 Add an Exam 54](#_Toc57570701)

[5.8 Document 58](#_Toc57570702)

[5.9 Transaction 62](#_Toc57570703)

[5.10 Send Announcement 65](#_Toc57570704)

[6 Future Enhancement 68](#_Toc57570705)

[7 Summary and Discussion 69](#_Toc57570706)

[8 References 70](#_Toc57570707)

[Appendix A 71](#_Toc57570708)

[Appendix B 72](#_Toc57570709)

[Appendix C 73](#_Toc57570710)

# List of Figures

[figure 4.1. 1 High-level architecture Diagram 9](#_Toc57507801)

[figure 4.4.1. 1 Waterfall Methodology 13](#_Toc57507811)

[figure 4.4.2.2. 1 Database environment 14](#_Toc57508029)

# List of Tables

[table 5.1 1 User Name validation 15](#_Toc57508252)

[table 5.1 2 Password validation 16](#_Toc57508253)

[table 5.1 3 Test case for Login 18](#_Toc57508254)

[table 5.2. 1 Profile name validation 19](#_Toc57508131)

[table 5.2. 2 Profile user validation 20](#_Toc57508132)

[table 5.2. 3 Password & confirmation validation 21](#_Toc57508133)

[table 5.2. 4 Security question validation 22](#_Toc57508134)

[table 5.2. 5 Answer validation 23](#_Toc57508135)

[table 5.2. 6 User type validation 23](#_Toc57508136)

[table 5.2. 7 Test case for profile 28](#_Toc57508137)

[table 5.3. 1 Student name validation 29](#_Toc57508147)

[table 5.3. 2 Student address validation 29](#_Toc57508148)

[table 5.3. 3 Student contact number validation 30](#_Toc57508149)

[table 5.3. 4 Student email validation 31](#_Toc57508150)

[table 5.3. 5 Student birthday validation 31](#_Toc57508151)

[table 5.3. 6 Student guardian validation 32](#_Toc57508152)

[table 5.3. 7 Student occupation validation 33](#_Toc57508153)

[table 5.3. 8 Student gender validation 33](#_Toc57508154)

[table 5.3. 9 Student grade validation 34](#_Toc57508155)

[table 5.3. 10 Student class validation 35](#_Toc57508156)

[table 5.3. 11 Student courses validation 35](#_Toc57508157)

[table 5.3. 12 student Test Case 43](#_Toc57508158)

[table 5.4. 1 Staff name validation 44](#_Toc57508159)

[table 5.4. 2 Staff address validation 44](#_Toc57508160)

[table 5.4. 3 Staff email validation 45](#_Toc57508161)

[table 5.4. 4 Staff contact validation 46](#_Toc57508162)

[table 5.4. 5 Staff NIC validation 46](#_Toc57508163)

[table 5.4. 6 Staff test case 49](#_Toc57508164)

[table 5.5. 1 Class time validation 51](#_Toc57508167)

[table 5.5. 2 Class test case 52](#_Toc57508168)

[table 5.6. 1 New course time validation 53](#_Toc57508175)

[table 5.6. 2 Course test case 55](#_Toc57508176)

[table 5.7 1 Exam type validation 56](#_Toc57508182)

[table 5.7 2 Exam hall validation 57](#_Toc57508183)

[table 5.7 3 Exam time validation 57](#_Toc57508184)

[table 5.7 4 Exam test case 59](#_Toc57508185)

[table 5.8. 1 Document file location validation 59](#_Toc57508189)

[table 5.8. 2 Document test case 61](#_Toc57508190)

[table 5.8. 3 Document open validation 62](#_Toc57508191)

[table 5.8. 4 Document open test case 63](#_Toc57508192)

[table 5.9. 1 Transaction test case 64](#_Toc57508197)

[table 5.9. 2 Transaction print receipt test case 65](#_Toc57508198)

[table 5.10. 1 Announcement subject validation 66](#_Toc57508210)

[table 5.10. 2 Announcement message validation 67](#_Toc57508211)

[table 5.10. 3 Announcement test case 68](#_Toc57508212)

# Acronyms and Abbreviations

IDE : Integrated Development Environment 9, 13

MSSQL : Microsoft Structured Query Language 9

SDLC : Software Development Life Cycle 10

CELS : Cambridge English Language School i, ii, 1, 2, 3, 5, 6, 8, 16

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision made by** | **Date** | **Reason for Changes** | **Version** |
| H G L S Nawarathna | 29/11/2020 | Initial Draft | 1.0 |

# Group Members

|  |  |  |
| --- | --- | --- |
| **Name** | **Role** | **Responsibilities** |
| H G L S Nawarathna  (IT 19 1021 60) | Project Manager | Project management, documentations and leading the team |
| H A E I Jayawardhana  (IT 19 0306 54) | Programmer | Handling the coding side |
| N M V S Nawarathna  (IT 19 7770 78) | Database Manager | Database implementation and management |
| Y M L D K Yapa  (IT 19 0650 90) | UI Designer | Designing user interfaces |

# Executive Summary

Main goal of this system is backing the management of the Cambridge English Language School. Cambridge Language School is one of the institutes which provide valuable service to the both school children and employees in the Anuradhapura area. This institute was started by Mr. M.D. Sarathchandra Mapitiyage on 1997. On 2019, Mrs. S.C. Dilrukshi was appointed as the Principal of the institute. Currently this institute has a manual student management system, which can identify as a traditional paper-based system.

This system is known as CELS Management System. Principal of the CELS and the receptionist have the access to this system. This system can provide lots of functions to these users. By this, users can obtain benefits like time saving, space saving and money saving.

This system is a computer-based system. Management of students, staff, classes, exams, documents and transactions can be done through this system easily and quickly. As well as sending announcements and generating reports for different sections can be done with this CELS management system. To give a very user-friendly experience this system contains user understandable interfaces and contents.

# Project Background

## Introduction

Cambridge English Language School Management System is the future management system of the CELS institute. Currently this institute functions using manual system. They are using this manual system to manage and keep track of manage various records of students, staff, classes, exams, documents, announcements and transactions. Because of this manual system, principal of CELS and staff have faced various problems. There are so many drawbacks of this manual system.

CELS management system, being a computer-based system, directly connect with one database. This will help to manage so many things within the institute. Principal and receptionist will be able to work with this system. They have accounts which differ from each other. There are some limitations to the receptionist. But there is no ant limitation to the principal to access the system since he is playing the administrator role.

## Current System

Cambridge English Language School Management System is the proposed future management system for the CELS institute. Currently this institute use a file based manual system. They are using this manual system to manage and keep track of various records of students, staff, classes, exams, documents, announcements and transactions. The principal of CELS and staff have faced numerous problems because there are many drawbacks of this manual system like data redundancy, lack of security, requirement of lot of labor and data inconsistency.

CELS management system, being a computer-based system. Which directly connect with one database. This will help to store and retrieve database information in a fast and efficient manner. Principal and receptionist will be the persons who handle this system. They have two different roles which differ from each other. There are some limitations to the receptionist. But there is no ant limitation to the principal to access the system since he is the administrator.

## System to be Developed

After holding few meetings with Principal of CELS and Manager of the CELS we could identify their requirements. They needed system which can manage,

* + - * Students
      * Staff
      * Classes
      * Exams
      * Documents
      * Transactions
      * Announcements

With the help of student management function user can add students to the system, update student details, delete student from the system or view student details.

Users can use staff management function to add staff member to the system, update staff member details, delete staff member from the system and view staff member details.

Above two functions help to prevent wasting time, wasting money, unnecessary usage of resources and space management.

Classes management function helps to manage all the classes. This is like timetable. By this user can add new classes, view class details, edit class details and delete class from the system.

This institute offers exams for several course such as IELTS, FCE, KET, PET and Spoken English. User can add, update, delete and view exams with the help of this function. As well as user can register students for the exams.

This institute uses cupboards to keep copies of documents such as exam papers, certificates, tutorials. With this function they can keep copy of those documents in the system.

Only principal has access to the transaction management part. This function helps to keep transaction details securely and updated up to date.

With the help of announcement management system, user can send any announcement to selected students of staff members.

# Problem Definition and Requirement Analysis

## **Problem Definition**

Cambridge English Language School currently uses a fully manual paper-based system. Major drawbacks of this system are time wasting, losing of documents, resources wasting and unnecessary manpower usage.

### People Affected

* Principal of the CELS
* Receptionist
* Students
* Staff members

### Limitations of Current System

* Low security and privacy to the data
* Time wasting
* Cannot view details when needed
* Data redundancy
* Loss of documents
* Damaging documents
* Cannot update details easily
* Unnecessary usage of resources and space

## Requirement Analysis

### Gathering Requirements

Due to pandemic situation of the country, we had to face some difficulties when gathering requirements. As solution for that we held zoom meetings with Principal of CELS and Manager of CELS.

#### **Interviews**

We held many zoom meetings with our client and we could gather so many requirements from this (*appendix A 1*).

By this we could get basic ideas about client’s requirements. We identified what are the main functions they are willing through this system.

#### **Existing Documents**

To further usage, principal of CELS send us a letter (*appendix B 1*) and student registering applications (*appendix C 1*) to get better idea about their requirements

### General Objectives

* Increasing the security of storing data
* Reduce the manpower
* Reduce the time wasting
* Manage unnecessary usage of space
* Time saving
* Provide efficiency at storing data

### Specified Objectives

There are two main users of this system. They are Principal and receptionist of the institute. This is stand-alone system and Principal and receptionist have different access levels to the system. Admin of this system is principal and receptionist act as a user. And this system will be run on two separate computers. One for admin and one for user.

Both admin and user can:

* Add, update, delete and view student details.
* Mark the student attendance.
* Add, update, delete and view staff member details.
* Add, update, delete and view class details.
* Add, update, delete and view exam details.
* Add documents to the system or delete documents from the system.
* Send announcements to the students and staff members.

Only admin can:

* Add transaction details, update transaction details, delete transaction or view transaction history.
* Create user profiles.

# Design and Implementation

## **High Level Architecture Design**

CELS Management system is managed by two users; the Admin and the Receptionist. The data will be stored in the database and both can access to the database. The following high-level architecture diagram which includes hardware and software, shows the components of the system and their relationship.

Diagram

Description automatically generated

figure 4.1. High-level architecture Diagram

## Hardware, Software and Communication Interfaces

### Hardware Requirements

• 4GB RAM x 2

• 5GB space on hard drive x 2

• Keyboard x 2

• Mouse x 2

• Monitor x 2

• Intel dual core processor x 2

### Software Requirements

* Windows 7 or above operating system.
* Programming Language. Oracle Java SE development kit 8.
* NetBeans IDE 8.
* MSSQL server 2019.

### Liveware Requirements

• Need users with good knowledge about handling the system.

## Non-Functional Requirements

### Operational Requirements

* System should compatible with Windows operating system.

### Performance Requirements

* System should response to the user within few seconds.
* System should have the ability to generate the report to the user immediately.

### Security Requirements

* Admin should be able to access to every part in the system and should be able to add, update and deletion.
* Receptionist only have access to Student, attendance marking, transaction and to register a student for an exam. Other functions should not be available for the receptionist.

## Implementation

### Development Methodology

In the SDLC there are different methodologies that people follow when working on a project. In each methodology has different phases to follow. Each phase has different steps, deliverables and different techniques.

We have selected Waterfall methodology as our development methodology for our project. In Waterfall methodology, the development steps are moving from phase to phase. So each phase is begins or originates only after the previous phase has finished. With this development methodology, our team had to identify the system requirements before the development begin. So we could develop an efficient system according to our client’s requirements.

In waterfall methodology there are 4 different phases to follow. Planning, analysis, designing and implementation are the sequence of phases to follow in waterfall methodology.

During the Planning phase, we identify the opportunities and project manager create a work plan to control and direct the project along with the SDLC.

During the Analysis phase, we identify the users of the system, what they are expecting from the system and when they are going to use the system. This phase is conducted through interviews with the client and some questionnaires.

The next phase is Design. During design phase, we focus on our design strategy. Designing interfaces of the system and creating the database for the system will be done during this phase.

Next, we have to move to Implementation phase. In here we construct and install the system. During the construction period, programming and testing should be done. Developing the system and testing the system is done during construction period. Finally we can install the system and establish a support plan for the system. [1]

Diagram

Description automatically generatedfigure 4.4.1. Waterfall Methodology

### Development Technologies

#### **Coding Environment**

We select our coding environment as NetBeans using Java language. Java is Object oriented which means, everything in the language behaves like an object. So it is easy to develop the system according to our client’s requirements.

#### **Database Environment**

A database is a collection of components that comprise and regulates the group of data, management and use of data which consist of software, hardware, people and techniques of handling database. [2]

figure 4.4.2.2. Database environmentDiagram

Description automatically generated

### Development Tools

#### **Coding IDE**

* NetBeans IDE8.2

#### **Database Management System**

* Microsoft SQL Server Management studio 2018

#### **Reporting Tools**

* Generating Student details report
* Generating Staff member detail report
* Generating a receipt for payments
* Generating an admission for exam registration

# Testing and Deployment

## Login

* Test ID : 01
* Tester Name : H.G.L.S. Nawarathna
* Test Type :
  + - Character validations
    - Range Validations
    - Null Validations

User Name:

Invalid

Invalid

NULL

NULL

Valid

A Z

a z

0 9

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condition** | **Valid**  **Partition** | **Invalid**  **Partition** | **Valid**  **Boundaries** | **Invalid**  **Boundaries** |
| If (username! = null) | A-Z, a-z, 0-9 | Symbols and  Special  characters | A-Z, a-z, 0-9 | Symbols and Special  characters |

table 5.1 User Name validation

Password:

Invalid

Invalid

NULL

NULL

Valid

A Z

a z

0 9

Symbols

Special characters

Symbols

Special characters

table 5.1 Password validation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condition** | **Valid**  **Partition** | **Invalid**  **Partition** | **Valid**  **Boundaries** | **Invalid**  **Boundaries** |
| If (password! = null) | A-Z, a-z, 0-9, symbols and special characters |  | A-Z, a-z, 0-9 | A-Z, a-z, 0-9, symbols and special characters |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test**  **Scenario** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| 01 | Check Login with valid username and password | 1. 1. Go to Login 2. 2. Enter username 3. 3. Enter password   4. Click Login | Username:  User404  Password:  User@404 | User or Admin must direct to the Home of CELS management system. | As expected, | Pass |
| 02 | Check Login with invalid username | 1. 1. Go to Login 2. 2. Enter username 3. 3. Enter password   4. Click Login | Username:  User405  Password:  User@404 | Prompt error message “Invalid username or password” | As expected, | Pass |
| 03 | Check Login with invalid password | 1. 1. Go to Login 2. 2. Enter username 3. 3. Enter password   4. Click Login | Username:  User404  Password:  User403 | Prompt error message “Invalid username or password” | As expected, | Pass |
| 04 | Check Login with null username field | 1. 1. Go to Login 2. 2. Enter password   3. Click Login | Username:  null  Password:  User404 | Prompt error message “Invalid username or password” | As expected, | Pass |
| 05 | Check Login with null password field | 1. 1. Go to Login 2. 2. Enter password   3. Click Login | Username:  User404  Password:  null | Prompt error message “Invalid username or password” | As expected, | Pass |

table 5.1 Test case for Login

## Create Profile

* Test ID : 02
* Tester Name : H.G.L.S. Nawarathna
* Test Type :
  + - Character validations
    - Range Validations
    - Null Validations

Name:

Invalid

Invalid

NULL

NULL

Valid

A Z

a z

space space

table 5.2. Profile name validation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condition** | **Valid**  **Partition** | **Invalid**  **Partition** | **Valid**  **Boundaries** | **Invalid**  **Boundaries** |
| If (name! = null) | A-Z, a-z, Space | Symbols and  Special  Characters except space | A-Z, a-z | Symbols and  special  characters except  space |

User:

Invalid

Invalid

NULL

NULL

Valid

A Z

a z

0 9

table 5.2. Profile user validation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condition** | **Valid**  **Partition** | **Invalid**  **Partition** | **Valid**  **Boundaries** | **Invalid**  **Boundaries** |
| If (username! = null) | A-Z, a-z, 0-9 | Symbols and  Special  characters | A-Z, a-z, 0-9 | Symbols and Special  characters |

Password and Confirm password:

Invalid

Invalid

NULL

NULL

Valid

A Z

a z

0 9

Symbols Symbols

Special characters Special characters

table 5.2. Password & confirmation validation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condition** | **Valid**  **Partition** | **Invalid**  **Partition** | **Valid**  **Boundaries** | **Invalid**  **Boundaries** |
| If (password! = null) | A-Z, a-z, 0-9, symbols and special characters |  | A-Z, a-z, 0-9 | A-Z, a-z, 0-9, symbols and special characters |

Security Question:

Invalid

Invalid

NULL

NULL

Valid

What’s your mother’s name?

Last 4digits of your mobile no?

What’s your first school?

What’s your grade one teacher name?

What’s your mother’s name?

Last 4digits of your mobile no?

What’s your first school?

What’s your grade one teacher name?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condition** | **Valid**  **Partition** | **Invalid**  **Partition** | **Valid**  **Boundaries** | **Invalid**  **Boundaries** |
| If (securityQuestion! = null) | What is your mother's name?  What are the last four numbers of your mobile no.?  What is your first school?  what is your grade one teacher's name? |  | What is your mother's name?  What are the last four numbers of your mobile no.?  What is your first school?  what is your grade one teacher's name? |  |

table 5.2. Security question validation

Answer:

Invalid

Invalid

NULL

NULL

Valid

A Z

a z

0 9

table 5.2. Answer validation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condition** | **Valid**  **Partition** | **Invalid**  **Partition** | **Valid**  **Boundaries** | **Invalid**  **Boundaries** |
| If (answer! = null) | A-Z, a-z, 0-9, space | Symbols and  Special  Characters except space | A-Z, a-z, 0-9, space | Symbols and Special  Characters, except space |

Type:

Invalid

Invalid

NULL

NULL

Valid

Admin Admin

User User

table 5.2. User type validation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condition** | **Valid**  **Partition** | **Invalid**  **Partition** | **Valid**  **Boundaries** | **Invalid**  **Boundaries** |
| If (type ! = null) | Admin  User |  | Admin  User |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test**  **Scenario** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Pass/**  **Fail** |
| 01 | Check create profile with valid inputs | 1. 1. Go to create profile 2. 2. Enter name 3. 3. Enter user name 4. 4. Enter password   5. Enter confirm password  6. Select security question  7. Enter answer  8. Select type. | Name:  Anonymous  User name:  Admin001  Password:  Admin@001  Confirm Password:  Admin@001  Select security question: What are the last four numbers of your mobile no.?  Answer: 0000  Type: User | Prompt message “New User Added” | As expected, | Pass |
| 02 | Check create profile with invalid name | 1. 1. Go to create profile 2. 2. Enter name 3. 3. Enter user name 4. 4. Enter password   5. Enter confirm password  6. Select security question  7. Enter answer  8. Select type. | Name:  Anonymous 01  User name:  Admin001  Password:  Admin@001  Confirm Password:  Admin@001  Select security question: What are the last four numbers of your mobile no.?  Answer: 0000  Type: User | Prompt message “Invalid Name” | As expected, | Pass |
| 03 | Check create profile with invalid username | 1. 1. Go to create profile 2. 2. Enter name 3. 3. Enter user name 4. 4. Enter password   5. Enter confirm password  6. Select security question  7. Enter answer  8. Select type. | Name:  Anonymous  User name:  Admin@001  Password:  Admin@001  Confirm Password:  Admin@001  Select security question: What are the last four numbers of your mobile no.?  Answer: 0000  Type: User | Prompt error message “User name not valid” | As expected, | Pass |
| 04 | Check create profile with unmatching password and confirm password | 1. 1. Go to create profile 2. 2. Enter name 3. 3. Enter user name 4. 4. Enter password   5. Enter confirm password  6. Select security question  7. Enter answer  8. Select type. | Name:  Anonymous  User name:  Admin@001  Password:  Admin@001  Confirm Password:  Admin001  Select security question: What are the last four numbers of your mobile no.?  Answer: 0000  Type: User | Prompt error message “Check password again” | As expected, | Pass |
| 05 | Check create profile with invalid answer | 1. 1. Go to create profile 2. 2. Enter name 3. 3. Enter user name 4. 4. Enter password   5. Enter confirm password  6. Select security question  7. Enter answer  8. Select type. | Name:  Anonymous  User name:  Admin@001  Password:  Admin@001  Confirm Password:  Admin001  Select security question: What are the last four numbers of your mobile no.?  Answer: 0000@#  Type: User | Prompt error message “Invalid answer” | As expected, | Pass |

table 5.2. Test case for profile

## Student Register

* Test ID : 03
* Tester Name : H.G.L.S. Nawarathna
* Test Type :
  + - Character validations
    - Range Validations
    - Null Validation

Name:

Invalid

Invalid

NULL

NULL

Valid

A Z

a z

Space Space

table 5.3. Student name validation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condition** | **Valid**  **Partition** | **Invalid**  **Partition** | **Valid**  **Boundaries** | **Invalid**  **Boundaries** |
| If (name! = null) | A-Z, a-z, Space | Symbols and  Special  Characters except space | A-Z, a-z | Symbols and  special  characters except  space |

Address:

Invalid

Invalid

NULL

NULL

Valid

A Z

a z

Space Space

0 9

/ , / ,

table 5.3. Student address validation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condition** | **Valid**  **Partition** | **Invalid**  **Partition** | **Valid**  **Boundaries** | **Invalid**  **Boundaries** |
| If (address! = null) | A-Z, a-z,0-9, **/** , **,** , Space | Symbols and  Special  Characters except space,/ and , | A-Z, a-z,0-9, **/,** **,** , Space | Symbols and  Special  Characters except space,/ and , |

Contact Number:

Invalid

Invalid

NULL

NULL

Valid

0 9

table 5.3. Student contact number validation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condition** | **Valid**  **Partition** | **Invalid**  **Partition** | **Valid**  **Boundaries** | **Invalid**  **Boundaries** |
| If (contactNumber! = null) | 0-9 | Letters, Symbols and  Special  Characters | 0-9 | Letters, Symbols and  Special  Characters |

Email:

Invalid

Invalid

NULL

NULL

Valid

A Z

a z

@ and .

table 5.3. Student email validation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condition** | **Valid**  **Partition** | **Invalid**  **Partition** | **Valid**  **Boundaries** | **Invalid**  **Boundaries** |
| If (email! = null) | 0-9,A-Z,a-z, @ , . | Other symbols | 0-9,A-Z,a-z, @ , . | Other symbols |

Birthday:

Invalid

Invalid

NULL

NULL

Valid

0 9

.

table 5.3. Student birthday validation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condition** | **Valid**  **Partition** | **Invalid**  **Partition** | **Valid**  **Boundaries** | **Invalid**  **Boundaries** |
| If (birthday! = null) | 0-9, . | Other symbols, Letters | 0-9, . | Other symbols, Letters |

Guardian:

A Z

a z

space space

Invalid

Invalid

NULL

NULL

Valid

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condition** | **Valid**  **Partition** | **Invalid**  **Partition** | **Valid**  **Boundaries** | **Invalid**  **Boundaries** |
| If (guardian! = null) | A-Z, a-z, Space | Symbols and  Special  Characters except space | A-Z, a-z | Symbols and  special  characters except  space |

table 5.3. Student guardian validation

Occupation:

Invalid

Invalid

NULL

NULL

Valid

Student Student

Employee Employee

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condition** | **Valid**  **Partition** | **Invalid**  **Partition** | **Valid**  **Boundaries** | **Invalid**  **Boundaries** |
| If (occupation! = null) | Student  Employee | - | Student  Employee | - |

table 5.3. Student occupation validation

Gender:

Invalid

Invalid

NULL

NULL

Valid

Male Male

Female Female

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condition** | **Valid**  **Partition** | **Invalid**  **Partition** | **Valid**  **Boundaries** | **Invalid**  **Boundaries** |
| If (gender! = null) | Male  Female | - | Male  Female | - |

table 5.3. Student gender validation

Grade:

Invalid

Invalid

NULL

NULL

Valid

1 11

A/L A/L

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condition** | **Valid**  **Partition** | **Invalid**  **Partition** | **Valid**  **Boundaries** | **Invalid**  **Boundaries** |
| If (grade! = null) | 1-9, A/L | - | 1-9, A-L | - |

table 5.3. Student grade validation

Classes:

English

Science

Math

Geography

Combined math

Biology

Chemistry

Invalid

Invalid

NULL

NULL

Valid

English

Science

Math

Geography

Combined math

Biology

Chemistry

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condition** | **Valid**  **Partition** | **Invalid**  **Partition** | **Valid**  **Boundaries** | **Invalid**  **Boundaries** |
| If (class! = null) | English  Science  Math  Geography  Combined math  Biology  Chemistry | - | English  Science  Math  Geography  Combined math  Biology  Chemistry | - |

table 5.3. Student class validation

Courses:

Invalid

Invalid

NULL

NULL

Valid

IELTS

FCE

KET

PET

SPOKEN

IELTS

FCE

KET

PET

SPOKEN

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condition** | **Valid**  **Partition** | **Invalid**  **Partition** | **Valid**  **Boundaries** | **Invalid**  **Boundaries** |
| If (course! = null) | IELTS FCE  KET  PET  Spoken | - | IELTS FCE  KET  PET  Spoken | - |

table 5.3. Student courses validation

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test**  **Scenario** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Pass/**  **Fail** |
| 01 | Student registration with valid inputs | 1. 1. Student registration 2. 2. Enter name 3. 3. Enter address 4. 4. Enter contact number   5. Enter email  6. Enter birthday  7. Enter guardian  8. Select occupation  9. Select Gender  10. Select grade  11. Select classes (one or more)  12. Select courses (one or more) | Name:  Sahan Anjana  Address:  No 20/1, 2nd lane, Kandy  Contact number:  0771234567  Email:  studnet@gmail.com  Birthday:  2001.01.01  Guardian:  Mr. Perera  Occupation:  Student  Gender:  Male  Grade:  11  Classes:  Math, English  Courses:  FCE | Prompt message “Student added” | As expected, | Pass |
| 02 | Student registration with invalid name | 1. 1. Student registration 2. 2. Enter name 3. 3. Enter address 4. 4. Enter contact number   5. Enter email  6. Enter birthday  7. Enter guardian  8. Select occupation  9. Select Gender  10. Select grade  11. Select classes (one or more)  12. Select courses (one or more) | Name:  Sahan1  Address:  No 20/1, 2nd lane, Kandy  Contact number:  0771234567  Email:  studnet@gmail.com  Birthday:  2001.01.01  Guardian:  Mr. Perera  Occupation:  Student  Gender:  Male  Grade:  11  Classes:  Math, English  Courses:  FCE | Alert “Naming is incorrect” | As expected, | Pass |
| 03 | Student registration with invalid address | 1. 1. Student registration 2. 2. Enter name 3. 3. Enter address 4. 4. Enter contact number   5. Enter email  6. Enter birthday  7. Enter guardian  8. Select occupation  9. Select Gender  10. Select grade  11. Select classes (one or more)  12. Select courses (one or more) | Name:  Sahan Anjana  Address:  No 20/1#,  Contact number:  0771234567  Email:  studnet@gmail.com  Birthday:  2001.01.01  Guardian:  Mr. Perera  Occupation:  Student  Gender:  Male  Grade:  11  Classes:  Math, English  Courses:  FCE | Alert “Invalid address” | As expected, | Pass |
| 04 | Student registration with invalid  contact number | 1. 1. Student registration 2. 2. Enter name 3. 3. Enter address 4. 4. Enter contact number   5. Enter email  6. Enter birthday  7. Enter guardian  8. Select occupation  9. Select Gender  10. Select grade  11. Select classes (one or more)  12. Select courses (one or more)  5. Enter confirm password  6. Select security question  7. Enter answer  8. Select type. | Name:  Sahan Anjana  Address:  No 20/1, 2nd lane, Kandy  Contact number:  077a  Email:  studnet@gmail.com  Birthday:  2001.01.01  Guardian:  M. Perera  Occupation:  Student  Gender:  Male  Grade:  11  Classes:  Math, English  Courses:  FCE | Alert  “Incorrect contact number” | As expected, | Pass |
| 05 | Student registration with invalid email | 1. 1. Student registration   2. Enter name  3. Enter address  4. Enter contact number  5. Enter email  6. Enter birthday  7. Enter guardian  8. Select occupation  9. Select Gender  10. Select grade  11. Select classes (one or more)  12. Select courses (one or more)  5. Enter confirm password  6. Select security question  7. Enter answer  8. Select type. | Name:  Sahan Anjana  Address:  No 20/1, 2nd lane, Kandy  Contact number:  0771234567  Email:  studnet@#m  Birthday:  2001.01.01  Guardian:  M. Perera  Occupation:  Student  Gender:  Male  Grade:  11  Classes:  Math, English  Courses:  FCE | Alert “Naming is incorrect” | As expected, | Pass |
| 06 | Student registration with invalid birthday | 1. Student registration  2. Enter name  3. Enter address  4. Enter contact number  5. Enter email  6. Enter birthday  7. Enter guardian  8. Select occupation  9. Select Gender  10. Select grade  11. Select classes (one or more)  12. Select courses (one or more)  5. Enter confirm password  6. Select security question  7. Enter answer  8. Select type. | Name:  Sahan1  Address:  No 20/1#,  Contact number:  077123456  Email:  studnet@gmail.com  Birthday:  20013.01.01  Guardian:  M. Perera  Occupation:  Student  Gender:  Male  Grade:  11  Classes:  Math, English  Courses:  FCE | Alert “Incorrect birthday format” | As expected, | Pass |

table 5.3. student Test Case

## Add Staff Member

* Test ID: 04
* Tester Name: H A E I Jayawardhana
* Test type:
  + NULL validations
  + Character validations
  + Range validations

StaffID:

* Staff member ID cannot insert manually. It is generated by the Database automatically with auto increment function.

Name:

Invalid

Invalid

NULL

NULL

Valid

A Z

a z

Space Space

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condition** | **Valid partition** | **Invalid partition** | **Valid boundaries** | **Invalid boundaries** |
| If(name != null) | A-Z, a-z | Null, numbers, symbols | A-Z, a-z | Null, numbers, symbols |

table 5.4. Staff name validation

Address:

Invalid

Invalid

NULL

NULL

Valid

A Z

a z

Space Space

0 9

/ , / ,

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condition** | **Valid partition** | **Invalid partition** | **Valid boundaries** | **Invalid boundaries** |
| If(address!= null) | A-Z, a-z, 0-9, /, space, , | Symbols and  Special  Characters except space,/ and , | A-Z, a-z, 0-9, / , space, , | Symbols and  Special  Characters except space,/ and , |

table 5.4. Staff address validation

Email:

Invalid

Invalid

NULL

NULL

Valid

A Z

a z

@ and .

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condition** | **Valid**  **Partition** | **Invalid**  **Partition** | **Valid**  **Boundaries** | **Invalid**  **Boundaries** |
| If (email! = null) | 0-9,A-Z,a-z, @ , . | Other symbols | 0-9,A-Z,a-z, @ , . | Other symbols |

table 5.4. Staff email validation

Contact:

Invalid

Invalid

NULL

NULL

Valid

0 9

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condition** | **Valid**  **Partition** | **Invalid**  **Partition** | **Valid**  **Boundaries** | **Invalid**  **Boundaries** |
| If (contactNumber! = null) | 0-9 | Letters, Symbols and  Special  Characters | 0-9 | Letters, Symbols and  Special  Characters |

table 5.4. Staff contact validation

NIC:

Invalid

Invalid

NULL

NULL

Valid

0 9

V v

X x

10

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condition** | **Valid partition** | **Invalid partition** | **Valid boundaries** | **Invalid boundaries** |
| If(nic != null) | 0-9, V,v, X,x | Null, letters except V & x, symbols | 0-9, V,v, X,x | Null, letters except V & x, symbols |

table 5.4. Staff NIC validation

Type:

* Need to select the type of the staff member.
* If type is Academic, need to select Grade and Subject.

Grade:

* If the member type is Academic, need to select the grade of the staff member teaches.

Subject:

* If the member type is Academic, need to select the Subject of the staff member.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test case ID** | **Test case scenario** | **Test steps** | **Test data** | **Expected results** | **Actual result** | **Pass / Fail** |
| 01 | Add new staff member with Null data fields | 1. select staff menu  2.fill some fields and keep some  fields empty  3. select add button | Name not filled  Address not filled  Email not filled  Contact number not filled  NIC not filled | Display Error message  “Please Fill the remaining details” | As Expected | Pass |
| 02 | Add new Staff member with invalid data | 1. select staff menu  2. fill fields with some invalid data  3. select save button | Name not filled  Address not filled  Email:  sample#.com  contact:  0711234567q  NIC:  A14658647V | Display Error message  “Please Fill the remaining details”  “Please Fill the remaining details”  "Invalid email Address!! Please enter a valid email address."  "Invalid Phone number!! Please enter a valid Phone Number."  "Invalid NIC number!! Please enter a valid NIC." | As Expected | Pass |
| 03 | Add new Staff member with valid data | 1. select staff menu  2. fill fields with valid data  3.select save button | Name:  Mr.Kapila Ransinghe  Address:  F60,Government flats, Colombo 03  Email:  kapila89@gmail.com  Contact:  0711234567  NIC:  891458864V  Type:  Academic  Grade:  A/L  Subject:  Math | Add user to the System  "Staff member added!" | As Expected | Pass |

table 5.4. Staff test case

## Add Class

* Test ID: 05
* Tester Name: Y M L D K Yapa.
* Test Type:
* Null Validation.

Class No & Class Id:

* These fields are auto generated by the database

Subject name:

* Subjects should select by the admin or the receptionist when adding new Class to the system.

Grade:

* Grade also should be selected when adding a new Class.

Hall:

* Class hall should be selected when adding new class

Time:

Invalid

Invalid

NULL

NULL

Valid

08.00 17.00

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condition** | **Valid partition** | **Invalid partition** | **Valid boundaries** | **Invalid boundaries** |
| If(time != null) | 08.00-17.00 | Null, letters, symbols | 08.00-17.00 | Null, letters, symbols |

table 5.5. Class time validation

Day:

* Day should be selected when adding the class

Teacher:

* Teacher should be selected when adding the class

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test**  **Scenario** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| 01 | Add Class with correct data | 1. Go to Class Tab.  2. Now enter the correct Data in the related swing controls.  3. After entering data click the add button. | Subject= Math  Grade = 11  Hall = Hall 03  Time = 13.00-16.00  Day = Sunday  Teacher= Miss. Nimali | Class details  added successfully | As expected, | Pass |
| 02 | Add the Class with null fields | 1. Go to Class Tab.  2.Now enter the correct Data in the related swing controls.  3.After entering data click the add button. | ClassId(cannot type, its auto generated)  Subject = null  Grade = 11  Hall = Hall 03  Time = null  Day = Sunday  Teacher = Mr. Saman Kumara | Display error message | As expected | Pass |

table 5.5. Class test case

## New Course

* Test ID: 06
* Tester Name: Y M L D K Yapa.
* Test Type:
* Null Validation.
* Check is selected

Course No & Course Id:

* These values are generated automatically. Cannot add manually.

Hall:

* Class hall should be selected when adding new class

Time:

Invalid

Invalid

NULL

NULL

Valid

08.00 17.00

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condition** | **Valid partition** | **Invalid partition** | **Valid boundaries** | **Invalid boundaries** |
| If(time != null) | 08.00-17.00 | Null, letters, symbols | 08.00-17.00 | Null, letters, symbols |

table 5.6. New course time validation

Day:

* Day should be selected when adding the class

Teacher:

* Teacher should be selected when adding the class

Courses:

* Courses should select when adding the course.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test**  **Scenario** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| 01 | Add Course with correct data | 1. 1. Go to Class Tab. 2. 2. Now enter the correct Data in the related swing controls.   3.After entering data click the add button. | ClassId(cannot type, its auto generated)  Hall = Hall 02  Time = 13.00-15.00  Day = Wednesday  Teacher = Mr. Saman Bandara  Course = IELTS | Course added successfully | As expected | Pass |
| 02 | Add the Course with null fields | 1. Go to Class Tab.  2. Now enter the correct Data in the related swing controls.  3.After entering data click the add button | ClassId(cannot type, its auto generated)  Hall = Hall 02  Time = 13.00 – 15.00  Day = Wednesday  Teacher = Mr. Kumara  Course = IELTS | Display error message | As expected, | Pass |

table 5.6. Course test case

## Add an Exam

* + - * + Test ID: 07
        + Tester Name: N.M.V.S.Nawarathana
        + Test Types:
* Null Validation

Exam ID:

* Exam ID is Auto incrementable.

Exam type:

IELTS

FCE

PET

KET

Invalid

Invalid

NULL

NULL

Valid

IELTS

FCE

PET

KET

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condition** | **Valid**  **Partition** | **Invalid**  **Partition** | **Valid**  **Boundaries** | **Invalid**  **Boundaries** |
| If(Type!=null) | IELTS  FCE  PET  KET | Null | IELTS  FCE  PET  KET | Null |

table 5.7 Exam type validation

Exam Hall:

Hall 01

Hall 02

Hall 03

Invalid

Invalid

NULL

NULL

Valid

Hall 01

Hall 02

Hall 03

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condition** | **Valid**  **Partition** | **Invalid**  **Partition** | **Valid**  **Boundaries** | **Invalid**  **Boundaries** |
| If(Hall!=null) | Hall 01  Hall 02  Hall 03 | Null | Hall 01  Hall 02  Hall 03 | Null |

table 5.7 Exam hall validation

Time:

Invalid

Invalid

NULL

NULL

Valid

08.00 17.00

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condition** | **Valid partition** | **Invalid partition** | **Valid boundaries** | **Invalid boundaries** |
| If(time != null) | 08.00-17.00 | Null, letters, symbols | 08.00-17.00 | Null, letters, symbols |

table 5.7 Exam time validation

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test**  **Scenario** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Pass/**  **Fail** |
| 01 | Add Correct Exam Data | 1. 1. Go to Exam Tab. 2. 2. Select the Add radio button 3. 3. Now enter the correct Data.   4. After entering data click the add button. | ExamId(cannot type, its auto generated)  Type = IELTS  Hall = Hall 01  Time = 10.30 A.M  Date = 2020-11-21 | Exam Added Successfully | As expected, | Pass |
| 02 | Add the Exam with null fields | 1. 1. Go to Exam Tab. 2. 2. Select the Add radio button 3. 3. Now enter the correct Data.   4. After entering data click the add button. | ExamId(cannot type, its auto generated)  Type = null  Hall = null  Time = 10.30 A.M  Date = 2020-11-21 | Please Select Correct Value | As expected, | Pass |

table 5.7 Exam test case

## Document

* + - * Test ID: 08
      * Tester Name: N.M.V.S.Nawarathana
      * Test Types:
* Null Validation

File Location:

Invalid

Invalid

NULL

NULL

Valid

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condition** | **Valid**  **Partition** | **Invalid**  **Partition** | **Valid**  **Boundaries** | **Invalid**  **Boundaries** |
| If(path.isEmpty()) | Valid Path | Null | Valid Path | Null |

table 5.8. Document file location validation

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test**  **Scenario** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Pass/**  **Fail** |
| 01 | Add Correct Document. | 1. 1. Go to Document Tab. 2. 2. Select the Select radio button 3. 3. Now select correct document.   4. After Selecting correct document click the Save button. | Exam.pdf | Path Added | As expected, | Pass |
| 02 | Add the Document with null fields. | 1. 1. Go to Document Tab. 2. 2. Select the Select radio button 3. 3. Now select correct document.   4. After Selecting correct document click the Save button. | null | Please Select Correct path. | As expected, | Pass |

table 5.8. Document test case

Open Document

* + - * Test ID: 09
      * Tester Name: N.M.V.S.Nawarathana
      * Test Types:
* Null Validation

Invalid

Invalid

NULL

NULL

Valid

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condition** | **Valid**  **Partition** | **Invalid**  **Partition** | **Valid**  **Boundaries** | **Invalid**  **Boundaries** |
| If(path.isEmpty()) | Valid Path | Null | Valid Path | Null |

table 5.8. Document open validation

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test**  **Scenario** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Pass/**  **Fail** |
| 01 | Open Added Document. | 1. 1. Go to Document Tab. 2. 2. Go to Document table and select the related row. 3. 2. Select the Open radio button | Exam.pdf | Exam.pdf opened. | As expected, | Pass |
| 02 | Open null fields. | 1. 1. Go to Document Tab. 2. 2. Go to Document table and select the related row.   2. Select the Open radio button | null | File is not Exit. | As expected , | Pass |

table 5.8. Document open test case

## Transaction

* Test ID: 10
* Tester Name: Y M L D K Yapa.
* Test Type:
* Validate the Insert Data
* Null Validation

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case** | **Test Scenario** | **Test Step** | **Test Data** | **Expected Result** | **Actual Result** | **Pas/**  **Fail** |
| 01 | Check Student Profile and generate the receipt with Valid Data. | 1.Goto Transaction Tab.  2.Select the Student from Student table  3.Fil Other text field for the generate bill (Receipt Id is auto generated  4. After Click Generate receipt button | Transaction ID = auto  Student ID = when the click student table  Receipt Id = auto  Student Name = when the click student table  Guardian Name = when the click student table  Fee Amount = 6000/=  Month = December  Description = Class Fee  Student E-mail = StudnetName12@gmail.com  Student Grade = 8  Student Subject = Maths | After the click Generate Receipt generate Receipt id and Receipt in txt area. | generated | Pass |

table 5.9. Transaction test case

Print receipt:

* Test ID: 11
* Tester Name: Y M L D K Yapa

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case** | **Test Scenario** | **Test Step** | **Test Data** | **Expected Result** | **Actual Result** | **Pas/**  **Fail** |
| 01 | Print Receipt. | Go to Transaction Tab.  Select the Student from Student table  Fil Other text field for the generate bill (Receipt Id is auto generated  After Click Generate receipt button  Click Add Button  8click print Button | Transaction ID = auto  Student ID = when the click student table  Receipt Id = auto  Student Name = when the click student table  Guardian Name = when the click student table  Fee Amount = 5000/=  Month = null  Description = null  Student E-mail = StudnetName12@gmail.com  Student Grade = A/L  Student Subject = Maths | Show about printing preview Massage | Show massage | Pass |

table 5.9. Transaction print receipt test case

## Send Announcement

* Test ID: 01
* Tester Name: H A E I Jayawardhana
* Test type:
* Null validations

Receivers:

* Receivers of the mail can be selected by class wise or to all.
* After selecting checkboxes, by clicking ‘Select receivers’ button system will fill the receiver’s area.

Subject:

Invalid

Invalid

NULL

NULL

Valid

A Z

a z

0 9

Symbols Symbols

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condition** | **Valid partition** | **Invalid partition** | **Valid boundaries** | **Invalid boundaries** |
| If(subject!= null) | A-Z,a-z,0-9, symbols | Null | A-Z, a-z,  0-9 ,symbols | Null |

table 5.10. Announcement subject validation

Message:

Invalid

Invalid

NULL

NULL

Valid

A Z

a z

0 9

Symbols Symbols

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condition** | **Valid partition** | **Invalid partition** | **Valid boundaries** | **Invalid boundaries** |
| If(subject!= null) | A-Z,a-z,0-9, symbols | Null | A-Z, a-z,  0-9 ,symbols | Null |

table 5.10. Announcement message validation

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test case ID** | **Test case scenario** | **Test steps** | **Test data** | **Expected results** | **Actual result** | **Pass / Fail** |
| 01 | Send Email with null values | 1.go to announcement menu  2.click ‘send’  button | Receivers not added  Subject not filled  Message is empty | Display an error message  “Please select receivers”  “Please enter the Subject”  "Mail body is Empty!" | As expect | Pass |
| 02 | Send email with  Valid values | 1.go to announcement menu  2.select receivers  3.enter a subject  4.enter message body  5.click ‘send’  button | Receivers:  test@test.com  subject:  test  message:  test mail | Email sends with Gmail  “Email sent successfully” | As expect | Pass |

table 5.10. Announcement test case

# Future Enhancement

Marking attendance manually is a time-consuming process using the provided interface in the system. As a solution for that, we are planning to add function like taking attendance with fingerprints or barcode reader. There is no access for student to this system. In future we have planned to allow the students to access the documents section and it should be implemented as a web-based platform for convenient use. Also, in future versions, more secure structure for the system will be provided because this contains personal and financial data.

# Summary and Discussion

Our main goal of this project was to build a computer-based management system which requested from our client Cambridge English Language School. This is a solution for problems in their current manual paper-based system.

We submitted our proposal on 27th July 2020. After got approve for the project we started implementing of the system. We held several meetings with Ms. Lakna Gammedda who was appointed by our lecturer in charge Mr. Roshan Jayawardana to get better idea about this project. After holding several meetings with our client, we could get better idea about their requirements. First of all, we tried to build completed prototype and it also succeeded.

After 12 weeks of working, we could finish our system as we proposed. This will be the fully completed solution for the problem of the client and this system consists all the requirements of the client.

# References

|  |  |
| --- | --- |
| [1] | B. H. W. R. M. R. Alan Dennis, System Analysis and Design, Wiley, 2014. |
| [2] | "w3schools," [Online]. Available: https://www.w3schools.in/dbms/database-environment/#:~:text=A%20database%20environment%20is%20a,database%2C%20and%20the%20data%20also.. |

# A screen shot of a person Description automatically generatedAppendix A

appendix A

# Text, letter Description automatically generatedAppendix B

appendix B

# Appendix C

Text

Description automatically generated

appendix C

Text

Description automatically generated