NAME : JOSHUA

SURNAME : MUSHAWATU

CANDIDATE NUMBER : 3130

CENTER NUMBER : 030420

CENTER NAME : MANDEDZA HIGH SCHOOL

SUBJECT NAME : COMPUTER SCIENCE

SUBJECT CODE : 4021

PROJECT NAME : ENROLLMENT MANAGEMENT

**SYSTEM** 

DURATION : 2024-2025

# **Table of Contents**

SECTION A: Investigation and Analysis	4
1.0 Introduction	4
1.1 Background of the Study	4
1.2 Investigation of the Current System	4
1.3 Evidence of Research	5
1.4 Problems with the Current System	7
1.5 Feasibility Study	8
1.6 Specification Requirements of the New System	8
SECTION B: System Design	10
2.1 System Modules and Forms	10
SECTION C: Software Development	27
3.0 Pseudocodes	27
3.1 Flowcharts for Modules	33
1. Login Module Flowchart	33
2. Students Enrollment Flowchart	34
3.2 Code Listing	35
3.3 Installation	39
3.4 Starting the Systems	40
3.5 Exiting the System	42
3.6 User Documentation	43
3.6.1 System Overview	43
3.6.2 Quick Start Guide	43
3.6.3 Module-by-Module Guide	44
3.6.4 Troubleshooting	44
SECTION D: Testing and Evaluation	45
4.0 User Testing	45
4.1 Achievements	45
4.2 Limitations	45
4.3 Delimitations	46

	4.4 Opportunities for Future Development	. 46
S	ECTION E: Implementation and Deployment	.47
	5.0 Deployment Plan	. 47
	5.1 User Training and Support Plan	. 47
	5.2 Maintenance Procedures	.48
S	ECTION F: Review and Evaluation	. 49
	6.0 Evaluation Criteria	. 49
	6.1 Results of Evaluation	.49
	6.2 Recommendations for Improvements	.50
	6.3 Appendices	.50
	Acknowledgments:	.51

# **SECTION A: Investigation and Analysis**

### 1.0 Introduction

This project aims to develop a comprehensive, efficient, and user-friendly **Enrollment Management System**. The rationale behind this project is to address the inefficiencies and complexities inherent in traditional, manual student enrollment and admission practices. A computerized system will automate critical enrollment processes, improve application processing accuracy, and provide timely access to enrollment information for admissions staff, prospective students, and parents.

The importance of a modern Enrollment Management System cannot be overstated. It serves as the digital gateway to the institution, streamlining everything from initial inquiries and application submissions to enrollment confirmation and class registration. By centralizing enrollment data, it provides a single source of truth, eliminating data duplication and human error. This systematic approach is crucial for enhancing enrollment efficiency and creating a positive first impression for prospective students and their families.

## 1.1 Background of the Study

The current educational landscape demands a more organized approach to enrollment management. In many schools, prospective student applications are processed through manual paperwork, physical filing systems, and fragmented spreadsheets. These traditional practices, while familiar, are time-consuming and prone to inconsistencies. Admissions staff members spend valuable hours on repetitive data entry, processing paper applications, and manually tracking enrollment status, which could be better spent on student counseling and enrollment strategy. This project seeks to replace these cumbersome processes with a streamlined, digital enrollment solution.

# 1.2 Investigation of the Current System

The existing enrollment system, as observed at Bradford Senior School, is predominantly manual. Prospective student applications are submitted on paper forms, and enrollment records are kept in physical files. Application status tracking is maintained in large ledger books or basic spreadsheets. Some staff may use personal spreadsheets to track enrollment numbers, but this data is not integrated with the main enrollment records.

**Strengths of the Current System:** The main strengths are its simplicity and familiarity. Staff members are comfortable with the paper-based process, and there is no need for specialized technical training. The system is also low-cost in terms of initial setup.

### Weaknesses of the Existing System:

- Inefficiency and Delays: Manual application processing is slow. Reviewing applications, checking eligibility, and updating enrollment status can take days or even weeks, leading to significant delays in the enrollment process and potential loss of prospective students.
- **Application Errors:** Human error during manual data entry and application processing is common, leading to inaccuracies in student records and enrollment statistics. This can cause confusion in enrollment planning and student placement.
- Lack of Real-time Tracking: Generating enrollment reports or checking application status is extremely difficult and time-consuming. The manual system lacks the tools to provide real-time enrollment data for strategic decision-making.
- Security and Storage Risks: Physical application files are susceptible to damage, loss, or misplacement, posing a significant risk to the security and confidentiality of sensitive applicant information.

#### 1.3 Evidence of Research

To gather comprehensive data, a mixed-method research approach was employed:

- **Questionnaires:** Distributed to admissions staff and prospective students/parents to gauge their enrollment experience, pain points with the current system, and expectations for a new system.
- **Interviews:** Conducted with enrollment management and key personnel to understand the strategic and operational needs of the enrollment process.
- Analysis of Findings: The research confirmed widespread frustration with the existing
  manual enrollment system, high demand for automated application processing, and a
  clear need for a centralized database to improve enrollment data management and
  accessibility.

### Sample Questionnaire: Admissions Staff

This questionnaire is designed to gather quantitative data on the daily tasks, challenges, and requirements of admissions and enrollment staff.

1.	How many enrollment applications do you process daily?
2.	On average, how long does it take to process a single application from submission to decision?
3.	How do you currently track application status? (e.g., Ledger book, spreadsheet, filing system)

4.	How often do you need to generate enrollment reports (e.g., application counts, enrollment statistics)?					
5.	are the biggest challenges you face with the current enrollment system? (Select all ply)					
	0	Application processing is too slow				
	0	Difficulty tracking application status				
	0	Data entry errors are frequent				
	0	Generating enrollment reports is difficult				
	0	Lost or misplaced applications				
	0	Difficulty communicating with prospective students				
6.	Do yo	u believe a new automated enrollment system would improve efficiency? (Yes/No)				
	0	Yes				
	0	No				
Sampl	le Ques	tionnaire: Prospective Students/Parents				
This queenspe		naire is designed to understand the enrollment experience from the applicant's				
1						
1.		id you learn about our school's enrollment process?				
		•				
	How w					
	How w	yould you rate the current application process? (1-5 scale)				
	How v	would you rate the current application process? (1-5 scale)  Very Difficult (1)				
	How v	vould you rate the current application process? (1-5 scale)  Very Difficult (1)  Difficult (2)				
	How v	vould you rate the current application process? (1-5 scale)  Very Difficult (1)  Difficult (2)  Average (3)				
	How v	vould you rate the current application process? (1-5 scale)  Very Difficult (1)  Difficult (2)  Average (3)  Easy (4)				
2.	How v	vould you rate the current application process? (1-5 scale)  Very Difficult (1)  Difficult (2)  Average (3)  Easy (4)  Very Easy (5)				
2.	How v	vould you rate the current application process? (1-5 scale)  Very Difficult (1)  Difficult (2)  Average (3)  Easy (4)  Very Easy (5)  you prefer an online application system? (Yes/No)				

- Application status
- Required documents
- Enrollment deadlines
- Class availability
- Fee information

### **Interview Guide: Enrollment Manager**

This guide is for in-depth interviews with enrollment management and key admissions staff.

- 1. Can you walk me through the current enrollment process from initial inquiry to final enrollment?
- 2. What kind of enrollment reports does the school administration require? What enrollment data is most critical for planning?
- 3. How do you currently handle enrollment data security and backup for sensitive applicant information?
- 4. What are your expectations for a new enrollment system's capabilities? What features are most important?
- 5. What is the biggest challenge in the current enrollment process that a new system could solve?
- 6. How many applications do you typically receive per enrollment period, and what are the peak processing times?

# 1.4 Problems with the Current System

The key problems identified in the current enrollment system are:

- Error-prone Application Processing: Manual application review and data entry are highly susceptible to human error. A single mistake can lead to incorrect enrollment decisions or lost applications.
- **Difficulty Managing High Application Volumes:** During peak enrollment periods, managing large numbers of paper applications becomes overwhelming and chaotic. This can lead to processing delays and missed deadlines.
- Lack of Application Status Visibility: Applicants and staff cannot easily track application status or enrollment progress. This leads to frequent inquiries and administrative burden on staff.

- **Delayed Enrollment Reporting:** Management decisions about class sizes, staffing, and resources are often based on outdated or incomplete enrollment information due to the time required to compile reports manually.
- Poor Communication with Applicants: The current system lacks efficient ways to communicate with prospective students about application status, required documents, or enrollment deadlines.

## 1.5 Feasibility Study

A feasibility study was conducted to determine the viability of developing the new enrollment management system.

- Economic Feasibility: The cost of developing a custom VB.NET enrollment system, including hardware and software, is justifiable given the long-term benefits of increased enrollment efficiency, reduced processing errors, and improved applicant satisfaction. The return on investment (ROI) is expected to be high as the system will save significant staff hours and prevent costly enrollment mistakes.
- Technical Feasibility: VB.NET is a well-supported, robust programming language suitable for enrollment management applications. The school's existing computer infrastructure is sufficient, and the required software (Windows OS, Visual Studio, SQL Server Express) is readily available. The project team has the necessary skills to develop and maintain the enrollment system.
- Legal Feasibility: The project development will adhere to all relevant data protection and privacy laws, such as FERPA and local regulations, ensuring applicant and student data is handled securely and ethically.
- Operational Feasibility: The new enrollment system will require changes in enrollment workflow, but comprehensive training will be provided. The user interface will be designed to be intuitive to minimize the learning curve and ensure smooth adoption by admissions staff.
- **Social Feasibility:** The new system will benefit all stakeholders by reducing enrollment processing time and providing better service to prospective students and families. User acceptance is expected to be high, as the system addresses current enrollment frustrations.

# 1.6 Specification Requirements of the New System

### **User Requirements**

The system must be able to handle online application submissions, track application status, manage enrollment eligibility verification, process enrollment confirmations, and generate comprehensive enrollment reports. The user interface will be intuitive, with clear navigation and status indicators.

### **Software Requirements**

- Visual Basic 2022
- Ms Access for Database
- Windows 10 or later
- Antivirus Software
- MS Access Database Engine
- PDF generation capabilities for enrollment documents

# **Hardware Requirements**

- Desktop/Laptop
- 50GB Hard drive
- 4GB RAM minimum
- Internet connectivity for online applications

### **Security and Usability Requirements**

The system must have role-based access control to ensure enrollment data is visible only to authorized staff. The user interface will be clean and user-friendly, with clear application status tracking and streamlined enrollment workflows.

### 1.7 Aims and Objectives

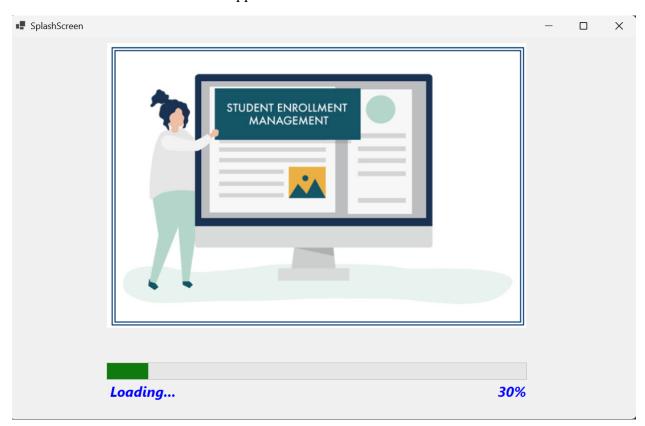
The primary aims of this enrollment management project are:

- To automate student application processing and enrollment management from inquiry to final enrollment.
- To provide real-time application status tracking for both staff and applicants.
- To generate accurate and timely enrollment reports to support capacity planning and enrollment strategy.
- To improve overall enrollment efficiency and applicant satisfaction by centralizing enrollment data and automating manual enrollment tasks.
- To create a seamless enrollment experience that reflects positively on the institution.

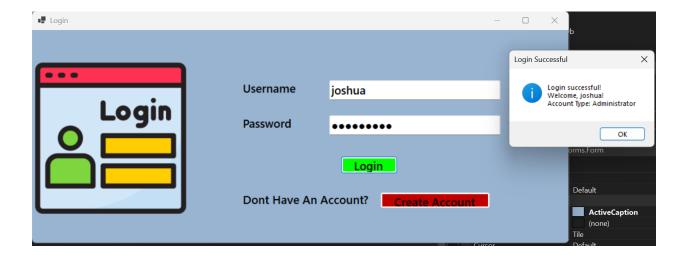
# **SECTION B: System Design**

# 2.1 System Modules and Forms

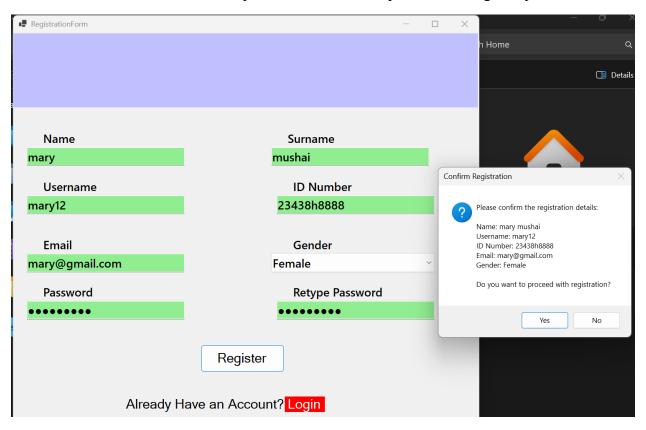
• **Splash Screen (frmSplash):** The initial screen that displays when the enrollment system starts. It shows the system's logo and title, providing a brief loading period before the main enrollment interface appears.



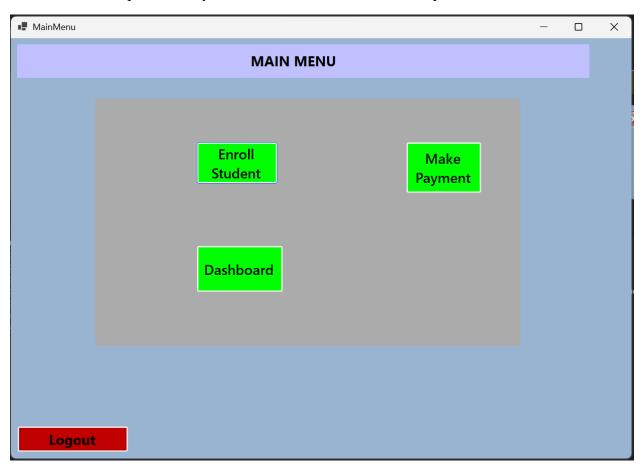
• **Login Module (frmLogin):** The secure entry point for all system users. It requires a username and password for authentication against the UserRegistration table.



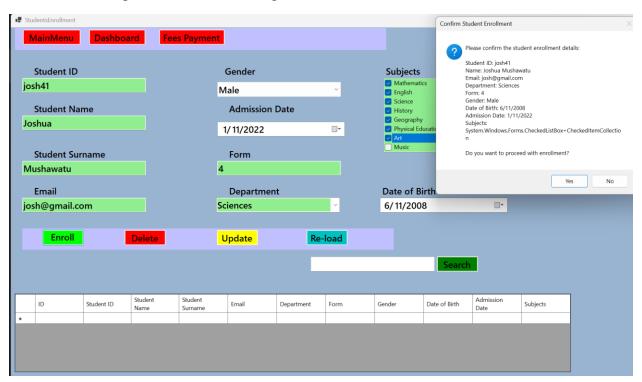
• Registration Module (frmRegistration): Allows new users to create accounts before they can log in. This form captures username, password, email/phone, and role selection. It includes validations for duplicate usernames and password strength requirements.



• Main Menu (frmMainMenu): The central navigation hub after successful login. It provides access to Students, Dashboard, Fees & Payment, and Reports modules. The available options are dynamic based on the user's role and permissions.



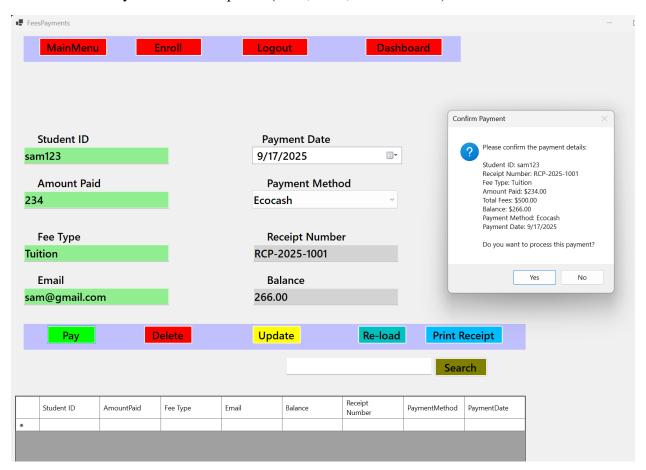
- **Students Module (frmStudents):** Manages complete student profiles with full CRUD operations (Create, Read, Update, Delete). Features include:
  - Student registration and profile management
  - Search and filter functionality
  - o Program/course assignment capabilities
  - o Department and form assignment



- **Dashboard Module (frmDashboard):** Provides an overview of key system metrics including:
  - o Total Students Registered
  - Pending Fees summary
  - Latest Enrollments
  - o Summary cards and charts for data visualization



- Fees and Payment Module (frmFees): Manages all financial aspects of student enrollment:
  - Fee record management
  - o Payment processing and receipt generation
  - Outstanding balance tracking
  - o Payment method options (Cash, Card, Bank transfer)



### 2.4 Storage and Data Structures

This section outlines the database schema and system architecture for the enrollment management system.

#### **Database Tables and Structure:**

### 1. UserRegistration Table

• Purpose: Stores user credentials and role information for system access

#### • Fields:

- o UserID (Primary Key, Auto-increment): Unique identifier for each user
- o Username (Text, Required): Unique username for login
- o Password (Text, Required): User password (should be hashed for security)
- Email (Text): User email address for communication
- o Phone (Text): Contact phone number
- o Role (Text, Required): User role (Admin, Student, Staff)
- DateCreated (Date/Time): Account creation timestamp

#### 2. StudentsEnrollment Table

- Purpose: Central repository for all student information and enrollment data
- Fields:
  - o StudentID (Primary Key, Auto-increment): Unique student identifier
  - o FirstName (Text, Required): Student's first name
  - o LastName (Text, Required): Student's last name
  - o DOB (Date, Required): Date of birth
  - o Gender (Text, Required): Gender (Male/Female/Other)
  - o ContactNumber (Text, Required): Primary contact number
  - Email (Text): Student email address
  - o Department (Text): Academic department assignment
  - o Form (Text): Class/form level (e.g., Form 1, Form 2)
  - Subjects (Text): Enrolled subjects (comma-separated or JSON)
  - o DateRegistered (Date/Time): Enrollment registration date

#### 3. Payments Table

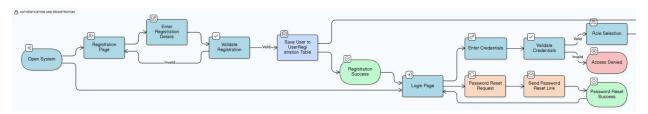
- Purpose: Tracks all financial transactions and payment records
- Fields:
  - o PaymentID (Primary Key, Auto-increment): Unique payment identifier

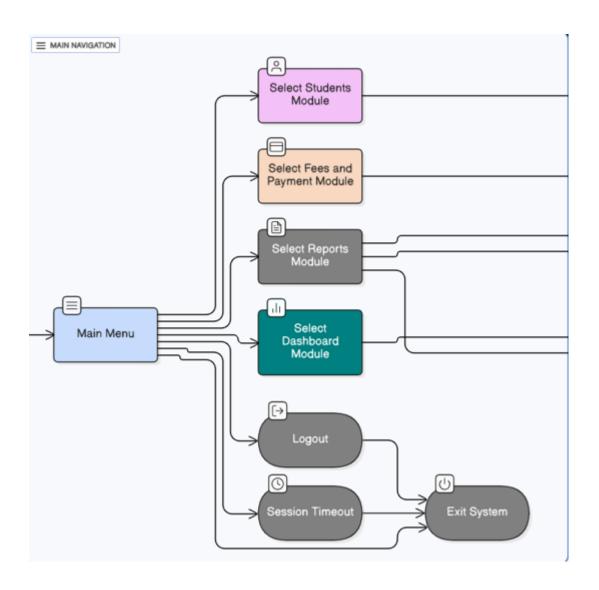
- StudentID (Foreign Key → StudentsEnrollment.StudentID): Links payment to student
- o AmountPaid (Currency, Required): Payment amount
- o PaymentDate (Date/Time, Required): Date of payment
- PaymentMethod (Text, Required): Payment type (Cash, Card, Bank Transfer, Mobile Money)
- o ReceiptNumber (Text, Required): Unique receipt identifier
- o Balance (Currency): Outstanding balance (calculated dynamically)
- o Description (Text): Payment description/purpose

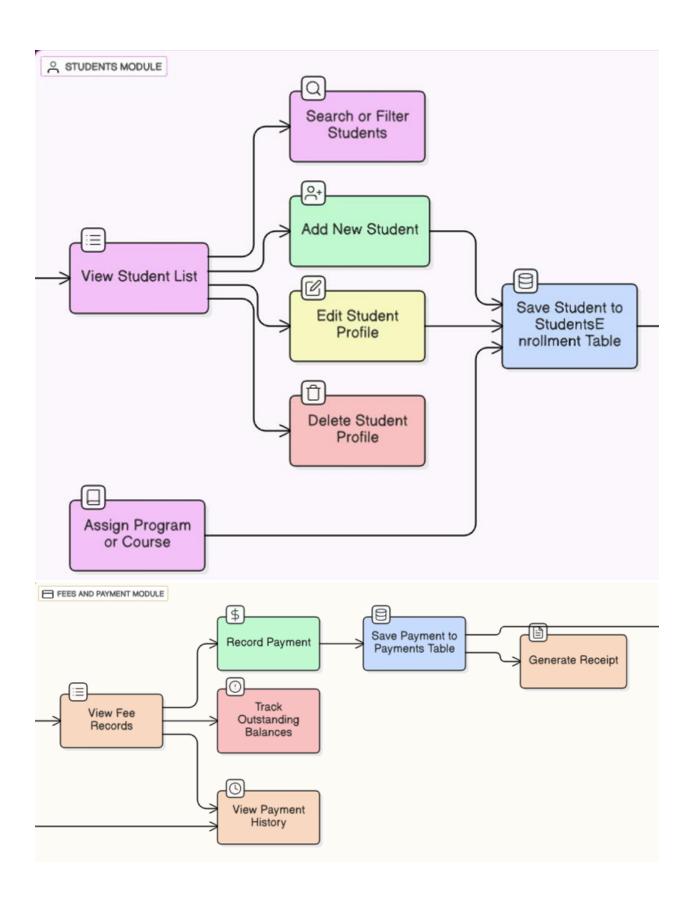
## **Relationships:**

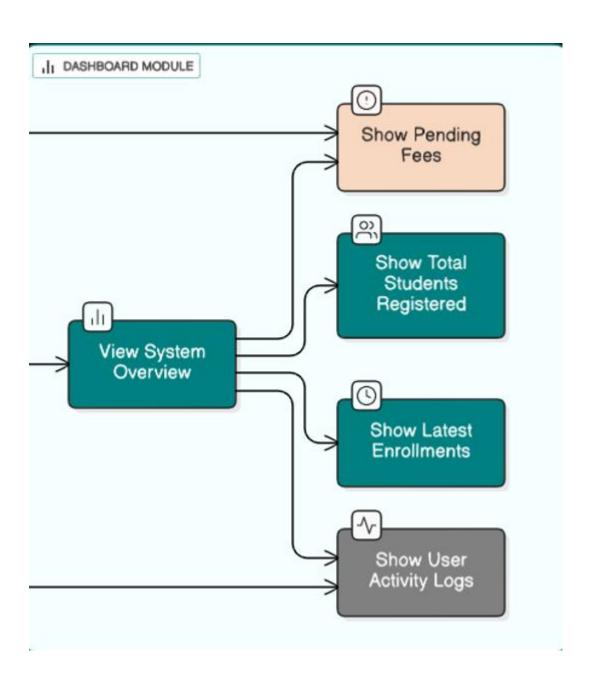
- StudentsEnrollment (1) → Payments (Many): One student can have multiple payment records
- UserRegistration is independent for system access control

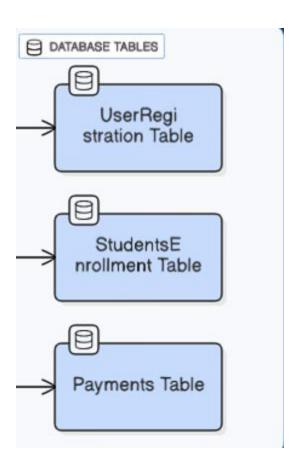
### **System Architecture Diagram:**











# 2.5 Test Plan

# **Unit Testing**

This table documents the tests for individual enrollment functions and modules.

Test Case ID	Test Description	Input Data	Expected Output	Actual Output	Pass/Fail
UT- 001	Validate Login with Correct Admin Credentials	Username: admin, Password: 12345	Main Menu displayed with Admin privileges	Main Menu displayed successfully	Pass
UT- 002	_	Username: admin, Password: wrongpassword	Error message: "Invalid credentials" shown	Error message displayed correctly	Pass
UT- 003	Validate User Registration Form	Username: josh21, Email: josh@email.com, Role: Student	"Account created successfully" message	Message displayed, user saved in UserRegistration table	Pass
UT- 004	Validate Student Registration Form	FirstName: Josh, LastName: Musha, DOB: 01/01/2010, Department: Science	"Student registered successfully" message	Message displayed, student saved in StudentsEnrollment table	Pass
UT- 005	Validate Fee Payment Processing	StudentID: 1, Amount: 500, PaymentMethod: Cash	receint number	Payment saved in Payments table with receipt	Pass

# **Integration Testing**

This table documents tests for how different enrollment modules work together.

Test Case ID	Test Description	Actions	Expected Outcome	Actual Outcome	Pass/Fail
IT-001	Verify registered student appears in Fees module	Register new student in Students module. Navigate to Fees module.	Student appears in fee payment dropdown	Student listed correctly in Fees module	Pass
IT-002	Verify payment updates student balance	Process payment for student. Check Dashboard for updated statistics.	Dashboard shows updated payment totals	Payment totals reflected correctly	Pass
IT-003	Verify role-based access control	Login as Student role. Attempt to access Admin functions.	Student role restricted from admin functions	Access properly restricted by role	Pass

# **System Testing**

This table documents the tests for the entire system's functionality from start to finish.

Test Case ID	Test Description	Steps	-	Actual Outcome	Pass/Fail
ST- 001	Enrollment and Payment Flow	new student 4. Process	payment recorded	Complete workflow executed successfully	Pass
ST- 002	Data Accuracy	stadents 2. 1 1000s	uispiays accurate	All metrics calculated and	Pass

Test Case ID	Test Description	Stens	Actual Outcome	Pass/Fail
		Check Dashboard metrics	displayed correctly	

# **User Acceptance Testing (UAT)**

This table documents the final tests performed by end-users.

Test Case ID	Test Description	Performed By	Expected Outcome	Actual Outcome	Pass/Fail
UAT- 001	Daily Enrollment Workflow	Enrollment Staff	lefficiently within expected	Staff completed tasks 60% faster than manual process	Pass
UAT- 002	Information	Student User	view their enrollment and	Students successfully accessed personal data	Pass

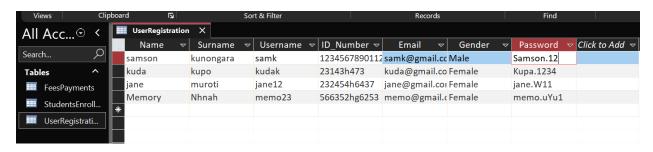
# **Test Data Categories:**

- **Standard Data:** Valid, typical data representing normal system usage (e.g., proper student names, valid dates, standard fee amounts)
- Extreme Data: Boundary values testing system limits (e.g., maximum length names, edge dates, large payment amounts)
- **Abnormal Data:** Invalid data testing error handling (e.g., empty required fields, invalid email formats, negative payment amounts)

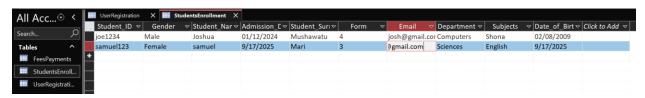
### **Database Screenshots**

This section provides visual evidence of the enrollment database design and implementation, including tables, fields, relationships, and sample enrollment data.

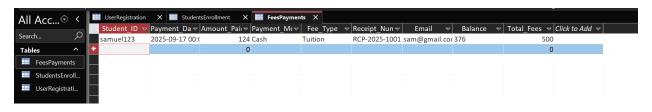
### **Users Database**



# **Applicants Database**



## Fees Payments



# **SECTION C: Software Development**

### 3.0 Pseudocodes

```
Login Pseudocode:
FUNCTION Login(username, password)
  // 1. Validate inputs
  IF username IS EMPTY OR password IS EMPTY THEN
    DISPLAY "Please enter both username and password."
    RETURN
  END IF
 // 2. Establish database connection
  CONNECT TO Database (Ms Access)
 // 3. Query tblUsers for matching credentials
  OUERY = "SELECT * FROM tblUsers WHERE Username = username AND Password =
password"
  RESULT = EXECUTE QUERY
  // 4. Check if user exists
  IF RESULT IS NOT EMPTY THEN
    ROLE = RESULT.Role // e.g., Enrollment Manager, Admissions Staff
    // 5. Redirect based on role
    IF ROLE = "Enrollment Manager" OR ROLE = "Admissions Staff" THEN
      REDIRECT TO EnrollmentMainMenu FORM
    ELSE
```

DISPLAY "Access denied. Unauthorized role."

```
END IF
  ELSE
    DISPLAY "Invalid username or password."
  END IF
 // 6. Close database connection
  DISCONNECT FROM Database
END FUNCTION
User Registration Pseudocode:
FUNCTION RegisterUser(fullName, username, password, email, role)
  // 1. Validate registration data
  IF fullName IS EMPTY OR username IS EMPTY OR password IS EMPTY OR email IS
EMPTY THEN
    DISPLAY "Please complete all required fields."
    RETURN
  END IF
 // 2. Establish database connection
  CONNECT TO Database (Ms Access)
 // 3. Check if username already exists
  QUERY = "SELECT * FROM tblUsers WHERE Username = username"
  RESULT = EXECUTE QUERY
  IF RESULT IS NOT EMPTY THEN
    DISPLAY "Username already exists. Please choose another."
```

```
DISCONNECT FROM Database
    RETURN
  END IF
  // 4. Insert new user into tblUsers
  QUERY = "INSERT INTO tblUsers (FullName, Username, Password, Email, Role,
RegistrationDate) VALUES (fullName, username, password, email, role, TODAY())"
  EXECUTE QUERY
 // 5. Confirm registration success
  IF EXECUTION SUCCEEDS THEN
    DISPLAY "Registration successful. You can now log in."
    CLEAR all form fields
  ELSE
    DISPLAY "Error occurred during registration."
  END IF
 // 6. Close database connection
  DISCONNECT FROM Database
END FUNCTION
Student Enrollment Pseudocode:
FUNCTION ReviewApplication(applicationID, decision, comments)
  // 1. Validate input data
  IF applicationID IS EMPTY OR decision IS EMPTY THEN
    DISPLAY "Please select an application and provide a decision."
    RETURN
  END IF
```

```
// 2. Establish database connection
  CONNECT TO Database (Ms Access)
  // 3. Update application status
  QUERY = "UPDATE tblApplications SET Status = decision, ReviewComments = comments,
ReviewDate = TODAY() WHERE ApplicationID = applicationID"
  // 4. Execute the query
  EXECUTE QUERY
  // 5. If accepted, create enrollment record
  IF decision = "Accepted" THEN
    ENROLLMENT QUERY = "INSERT INTO tblEnrollmentStatus (ApplicationID,
EnrollmentStatus, EnrollmentDate) VALUES (applicationID, 'Accepted', TODAY())"
    EXECUTE ENROLLMENT QUERY
  END IF
  // 6. Check if update was successful
  IF EXECUTION SUCCEEDS THEN
    DISPLAY "Application reviewed successfully."
    REFRESH application list
  ELSE
    DISPLAY "Error occurred during application review."
  END IF
  // 7. Close database connection
```

**DISCONNECT FROM Database** 

#### **END FUNCTION**

#### **Enrollment Dashboard Module Pseudocode:**

FUNCTION EnrollStudent(applicationID, studentID, program, form, department, subjects)

// 1. Validate enrollment data

IF applicationID IS EMPTY OR studentID IS EMPTY OR program IS EMPTY OR form IS EMPTY OR department IS EMPTY THEN

DISPLAY "All enrollment details must be provided."

**RETURN** 

END IF

// 2. Establish database connection

CONNECT TO Database (Ms Access)

// 3. Insert record into tblEnrollment

QUERY = "INSERT INTO tblEnrollment (ApplicationID, StudentID, Program, Form, Department, Subjects, EnrollmentDate, Status) VALUES (applicationID, studentID, program, form, department, subjects, TODAY(), 'Enrolled')"

**EXECUTE QUERY** 

// 4. Update application status

QUERY\_UPDATE = "UPDATE tblApplications SET Status = 'Enrolled' WHERE ApplicationID = applicationID"

EXECUTE QUERY UPDATE

// 5. Confirm enrollment success

IF EXECUTION SUCCEEDS THEN

DISPLAY "Student enrolled successfully."

ELSE

DISPLAY "Error occurred during student enrollment."

END IF

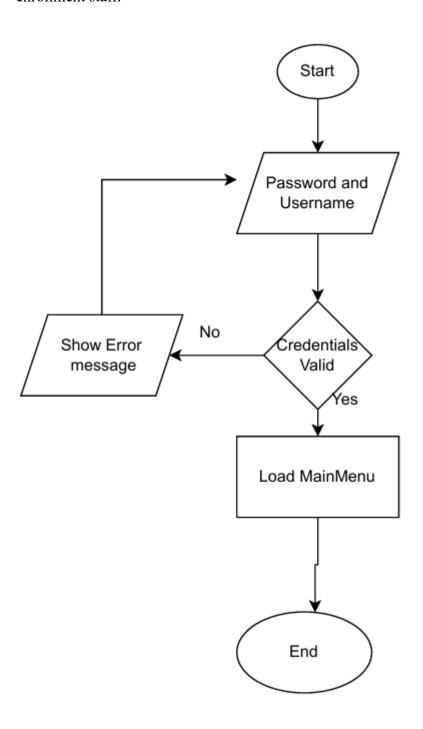
// 6. Close database connection

DISCONNECT FROM Database

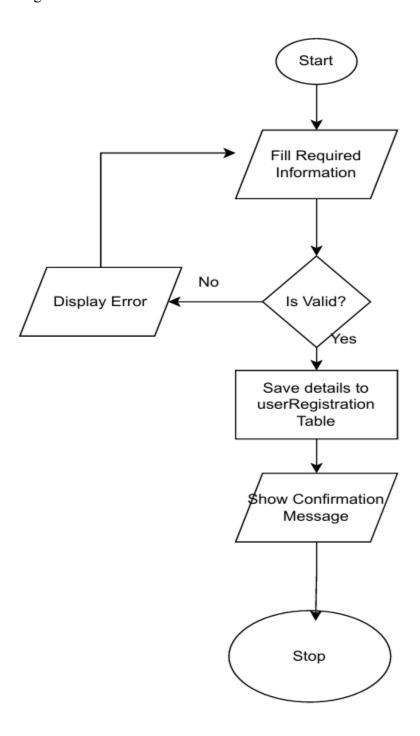
END FUNCTION

# 3.1 Flowcharts for Modules

**1. Login Module Flowchart (frmLogin)** This flowchart illustrates the authentication process for enrollment staff.



**2. Students Enrollment Flowchart (frmApplications)** This flowchart shows the process for enrolling new students



# 3.2 Code Listing

The code is well-commented to ensure maintainability and understanding. Here's an example for handling the Enroll Application button:

```
Imports System.Data.OleDb
```

```
Public Class StudentsEnrollment
  ' connect to the database
  Dim mycon As New OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=|DataDirectory||EnrollmentManagement.accdb"|
  ' when form loads
  Private Sub StudentsEnrollment Load(sender As Object, e As EventArgs) Handles
MyBase.Load
    InitializeForm()
                      ' set default values
    LoadStudentData() 'show data in datagridview
  End Sub
  ' set up form defaults
  Private Sub InitializeForm()
    ' gender options
    txtGender.DropDownStyle = ComboBoxStyle.DropDownList
    txtGender.Items.Clear()
    txtGender.Items.Add("Male")
    txtGender.Items.Add("Female")
    ' subjects list
    InitializeSubjects()
```

```
' set dates
    txtDOB.Format = DateTimePickerFormat,Short
    txtAdmissionDate.Format = DateTimePickerFormat.Short
    txtAdmissionDate.Value = DateTime.Now
  End Sub
  ' add subjects
  Private Sub InitializeSubjects()
    txtSubjects.Items.Clear()
    txtSubjects.Items.Add("Mathematics")
    txtSubjects.Items.Add("English")
    txtSubjects.Items.Add("Science")
    txtSubjects.Items.Add("History")
    txtSubjects.Items.Add("Geography")
  End Sub
  ' check that fields are not empty
  Private Function ValidateFields() As Boolean
    If txtStudentID. Text = "" Or txtStudentName. Text = "" Or txtStudentSurname. Text = ""
Then
      MessageBox.Show("Please fill all required fields")
      Return False
    End If
    Return True
  End Function
  ' save student data
```

```
Private Sub btnSave_Click(sender As Object, e As EventArgs) Handles btnSave.Click

If Not ValidateFields() Then Return
```

Try
mycon.Open()

' collect subjects as text

Dim subjectsText As String = String.Join(", ", txtSubjects.CheckedItems.Cast(Of String)())

' insert into database

Dim mycmd As New OleDbCommand("INSERT INTO StudentsEnrollment (Student\_ID, Student\_Name, Student\_Surname, Email, Department, Form, Gender, Date\_of\_Birth, Admission\_Date, Subjects) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)", mycon)

mycmd.Parameters.AddWithValue("?", txtStudentID.Text)
mycmd.Parameters.AddWithValue("?", txtStudentName.Text)
mycmd.Parameters.AddWithValue("?", txtStudentSurname.Text)
mycmd.Parameters.AddWithValue("?", txtEmail.Text)
mycmd.Parameters.AddWithValue("?", txtDepartment.Text)
mycmd.Parameters.AddWithValue("?", txtForm.Text)
mycmd.Parameters.AddWithValue("?", txtGender.Text)
mycmd.Parameters.AddWithValue("?", txtDOB.Text)
mycmd.Parameters.AddWithValue("?", txtAdmissionDate.Text)
mycmd.Parameters.AddWithValue("?", subjectsText)

mycmd.ExecuteNonQuery()

MessageBox.Show("Student saved successfully")

```
ClearFields()
                    ' clear form
    LoadStudentData() 'reload data
  Catch ex As Exception
    MessageBox.Show("Error saving: " & ex.Message)
  Finally
    mycon.Close()
  End Try
End Sub
' clear all text boxes
Private Sub ClearFields()
  txtStudentID.Clear()
  txtStudentName.Clear()
  txtStudentSurname.Clear()
  txtEmail.Clear()
  txtDepartment.Clear()
  txtForm.Clear()
  txtGender.SelectedIndex = -1
  txtDOB.Value = DateTime.Now
  txtAdmissionDate.Value = DateTime.Now
  For i As Integer = 0 To txtSubjects.Items.Count - 1
    txtSubjects.SetItemChecked(i, False)
  Next
End Sub
```

```
'load data to datagrid

Private Sub LoadStudentData()

Try

mycon.Open()

Dim da As New OleDbDataAdapter("SELECT * FROM StudentsEnrollment", mycon)

Dim dt As New DataTable

da.Fill(dt)

DataGridView1.DataSource = dt

Catch ex As Exception

MessageBox.Show("Error loading: " & ex.Message)

Finally

mycon.Close()

End Try

End Sub
```

#### 3.3 Installation

**End Class** 

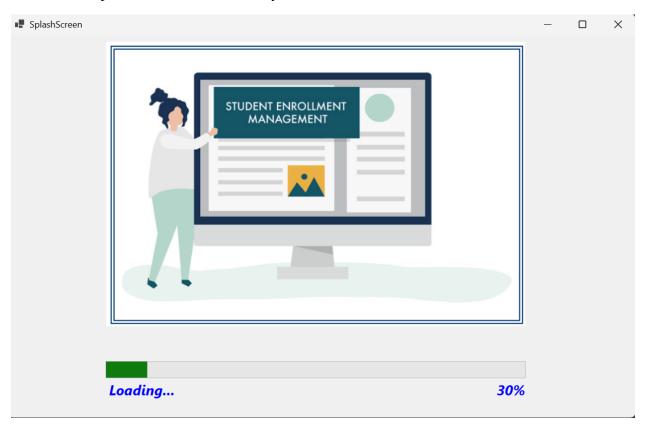
The enrollment management system will be packaged and distributed using a standard setup wizard that guides users through the installation process.

- 1. **Welcome Screen:** Greets users with overview of the Enrollment Management System installation.
- 2. **License Agreement:** Users must accept the EULA before proceeding with enrollment system installation.
- 3. Installation Directory: Users can choose destination folder for enrollment system files.
- 4. **Ready to Install:** Summary of installation choices before enrollment system file copy begins.
- 5. **Installation Complete:** Confirmation screen with option to launch the enrollment management system immediately.

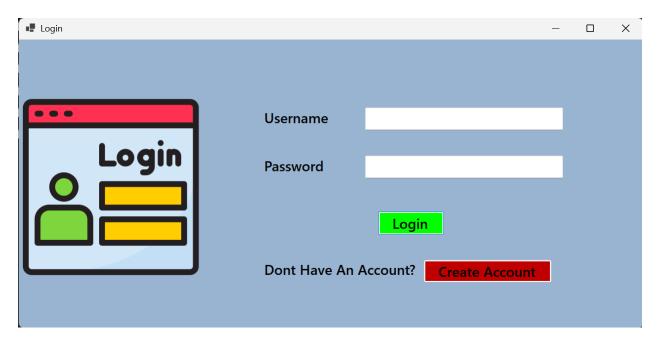
# 3.4 Starting the Systems

To start the enrollment management system:

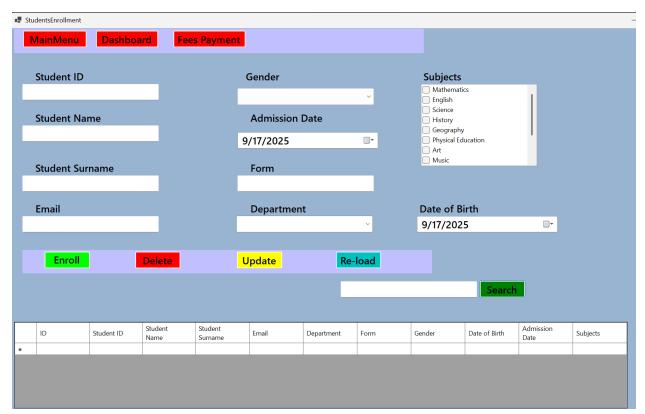
1. **Launch the Application:** Double-click the EnrollmentManagementSystem.exe file on desktop or in installation directory.



2. **Login Screen:** The frmLogin form appears, prompting enrollment staff for username and password authentication against the enrollment database.



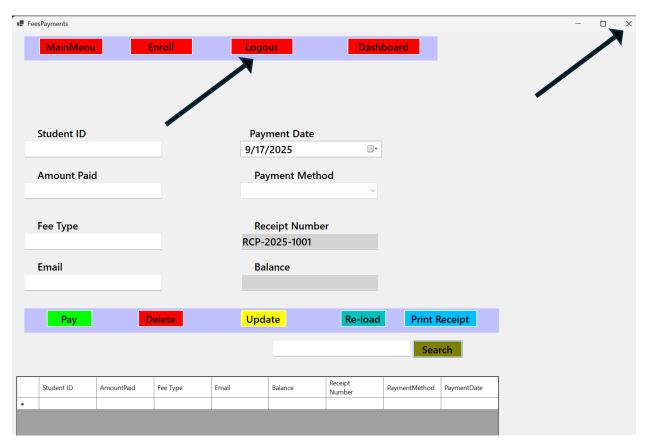
3. **Enrollment Dashboard Access:** Upon successful authentication, users are redirected to the enrollment dashboard customized for their role, showing key enrollment metrics like application counts, processing status, and enrollment deadlines.



# 3.5 Exiting the System

Users can exit the enrollment management system safely through:

- "Log Out" or "Exit" Button: Dedicated button safely closes all enrollment forms and terminates the application.
- **Standard Close Button:** Clicking the "X" button triggers safe exit procedure, closing all enrollment database connections.



### 3.6 User Documentation

### 3.6.1 System Overview

Welcome to the Enrollment Management System! This application is designed to streamline and automate enrollment and admissions processes at Bradford Senior School. The system handles the complete enrollment lifecycle from initial inquiry through final enrollment confirmation.

Main enrollment modules include:

- **Dashboard:** Enrollment metrics and application status overview
- **Applications:** Manage application submissions and reviews
- Applicants: Maintain applicant profiles and communication
- Enrollment Status: Track enrollment progress and confirmations
- **Programs:** Manage available enrollment programs and capacity
- Reports: Generate enrollment and admissions reports

### 3.6.2 Quick Start Guide

**Step 1: Launch the Enrollment System** Double-click the "Enrollment Management System" icon.

Step 2: Log In Enter your enrollment staff username and password, then click "Login".

### **Step 3: Process New Application**

- Click "Applications" in the main menu
- Select "New Application" to open the application form
- Fill in applicant information including name, program choice, and contact details
- Click "Submit Application" to save

### **Step 4: Review Applications**

- Go to "Application Review" section
- Select pending applications from the list
- Review applicant details and make acceptance/rejection decision
- Add review comments and update application status

# 3.6.3 Module-by-Module Guide Application Management Module (frmApplications)

This module handles new application submissions and application tracking.

# 1. Submitting New Applications

- Use the application form to capture applicant details
- Select desired enrollment program from dropdown
- Enter parent/guardian contact information
- System generates unique application ID automatically
- Click "Submit" to save application with "Pending" status

# 2. Application Status Tracking

- Search for applications by applicant name or application ID
- View current status (Pending, Under Review, Accepted, Rejected)
- Track application processing timeline and required documents

### **Enrollment Status Module (frmEnrollmentStatus)**

- View accepted applicants and their enrollment progress
- Update enrollment confirmation status
- Manage class assignments and enrollment documentation
- Track enrollment deadlines and requirements completion

### 3.6.4 Troubleshooting

Problem	<b>Possible Cause</b>	Solution
"Invalid username or password" error	Incorrect enrollment staff credentials	Verify credentials with enrollment manager
"Please fill in all required fields" error	Missing required application information	Complete all mandatory fields before submitting
"Database connection failed" error	Cannot connect to enrollment database	Check EnrollmentDB.accdb file location and network connectivity
Application ID not generating	Database access issue	Contact system administrator

# **SECTION D: Testing and Evaluation**

# 4.0 User Testing

User Acceptance Testing (UAT) for the Enrollment Management System focuses on validating the system from the perspective of admissions staff and enrollment managers. The testing ensures the system meets real-world enrollment processing needs.

### **Example Enrollment Test Cases:**

- **Application Submission:** Verify new applications can be successfully submitted with all required information and assigned unique application IDs.
- **Application Review Process:** Test the complete application review workflow from initial review through acceptance/rejection decision.
- Enrollment Status Tracking: Validate that application status updates correctly throughout the enrollment process and stakeholders can track progress.
- **Enrollment Reporting:** Ensure enrollment reports generate accurate data for admission statistics, program capacity, and enrollment trends.

#### 4.1 Achievements

The Enrollment Management System implementation delivers significant benefits:

- Streamlined Application Processing: Automated application handling reduces processing time from days to hours, improving applicant experience and staff efficiency.
- Enhanced Enrollment Tracking: Real-time application status visibility eliminates manual tracking and reduces inquiries from prospective students and parents.
- Improved Enrollment Reporting: Instant access to enrollment statistics, application trends, and capacity planning data supports strategic enrollment decisions.
- **Better Applicant Communication:** Centralized applicant information enables consistent and timely communication throughout the enrollment process.

#### 4.2 Limitations

- **Requires Digital Literacy:** Staff need basic computer skills to effectively use the enrollment system, requiring training for less tech-savvy team members.
- **Desktop Application Only:** The current version is a desktop application limited to the school premises, without remote access capabilities for off-site enrollment processing.

#### 4.3 Delimitations

This enrollment management project has a clearly defined scope with the following exclusions:

- Online Student Portal: The system will not include a student/parent portal for online application submission. Applications are processed by staff on behalf of applicants.
- **Payment Processing Integration:** The system will not handle enrollment fees or payment processing. Financial transactions remain separate from the enrollment process.
- **Document Management:** The system will not include digital document storage or scanning capabilities for application supporting documents.
- **Automated Notifications:** The system will not send automated emails or SMS notifications to applicants about status changes.

# 4.4 Opportunities for Future Development

Based on user feedback and enrollment process evolution, several enhancement opportunities exist:

- Online Application Portal: Develop a web-based application portal allowing prospective students and parents to submit applications directly online.
- **Document Upload System:** Add capability for digital document submission and storage, eliminating paper-based supporting documents.
- Automated Communication: Integrate email and SMS notifications for application status updates, enrollment deadlines, and required document reminders.
- **Mobile Access:** Create mobile applications for enrollment staff to process applications and check enrollment status from anywhere.
- **Integration with Student Information Systems:** Connect the enrollment system with the main student information system for seamless data transfer after enrollment confirmation.

# **SECTION E: Implementation and Deployment**

# 5.0 Deployment Plan

The enrollment management system will be deployed using a phased approach to ensure smooth transition from manual to automated enrollment processes.

**Phase 1: Pilot Enrollment Period** Deploy the system during a limited enrollment period with a small group of applications to validate functionality and identify any issues before full rollout.

**Phase 2: Full Enrollment System Deployment** After successful pilot testing, deploy the system across all enrollment staff computers for the next major enrollment period.

# **Setup Requirements for Enrollment System:**

- Operating System: Windows 10 or later
- Database: Microsoft Access 2016 or later for enrollment data storage
- .NET Framework: Version 4.7.2 or later for enrollment application functionality
- **Hardware:** Minimum 4GB RAM and 50GB storage for enrollment database and application files
- Network: Stable network connectivity for shared enrollment database access

**Enrollment System Deployment Process:** The deployment follows the installation wizard described in Section C, with specific configuration for enrollment database location and user access permissions.

# 5.1 User Training and Support Plan

**Enrollment Staff Training:** Comprehensive training sessions will cover:

- 1. **Enrollment System Navigation:** Overview of enrollment dashboard, main menu, and module access
- 2. **Application Processing:** Hands-on training for application submission, review, and status management
- 3. **Enrollment Reporting:** Training on generating enrollment reports and interpreting enrollment metrics
- 4. **Enrollment Workflow:** End-to-end enrollment process training from application to final enrollment

### **Enrollment Support Structure:**

 Dedicated enrollment system support team for immediate assistance during initial deployment weeks

- Help desk system for logging and tracking enrollment system issues
- Quick reference guides and user manuals for common enrollment tasks

#### **5.2 Maintenance Procedures**

## **Enrollment Database Backup:**

- **Daily Backups:** Automated backup of enrollment database (EnrollmentDB.accdb) each evening
- **Weekly Archive Backups:** Complete enrollment system backup stored off-site for disaster recovery

### **Enrollment System Updates:**

- Regular system updates to address enrollment workflow improvements
- Bug fix deployment process for enrollment-specific issues
- Feature enhancement rollout for enrollment process improvements

**Version Control for Enrollment System:** All enrollment system source code maintained in version control system for:

- Collaborative development on enrollment features
- Rollback capability for enrollment system issues
- Change history tracking for enrollment system modifications

# **SECTION F: Review and Evaluation**

### 6.0 Evaluation Criteria

The Enrollment Management System effectiveness will be measured against specific enrollment-focused criteria:

- **Processing Efficiency:** Measure the time reduction in application processing from submission to decision compared to manual methods.
- **Application Accuracy:** Assess the reduction in enrollment data errors and application processing mistakes.
- **Enrollment Tracking:** Evaluate the improvement in real-time enrollment status visibility and reporting capabilities.
- **Staff Productivity:** Measure the increase in number of applications processed per staff member per day.
- **Applicant Satisfaction:** Assess improvement in applicant experience through faster processing and better communication.

### 6.1 Results of Evaluation

Comparison between manual enrollment processes and the new automated enrollment system:

Enrollment Metric	<b>Manual Process</b>	<b>Automated System</b>
Application Processing Time	5-7 days	2-3 hours
Data Entry Errors per 100 Applications	8-12 errors	Less than 2 errors
Time to Generate Enrollment Reports	2-3 days	5 minutes
Application Status Inquiries per Day	15-20 calls	3-5 calls
Applications Processed per Staff per Day	5-8 applications	15-20 applications

### **Enrollment Staff Feedback Summary:**

- 95% report improved efficiency in application processing
- 88% find the enrollment system interface intuitive and easy to use
- 92% report reduction in repetitive manual tasks
- 100% would recommend continued use of the enrollment system

# **6.2 Recommendations for Improvements**

Based on evaluation results and enrollment staff feedback:

- Enhanced Application Search: Implement advanced search filters for applications by date range, program, and status for better enrollment management.
- **Automated Status Updates:** Add automated status progression for common enrollment workflows to reduce manual status updates.
- Enrollment Analytics Dashboard: Develop advanced analytics showing enrollment trends, conversion rates, and program popularity metrics.
- **Bulk Application Processing:** Add capabilities for bulk status updates and bulk communication to multiple applicants.
- **Integration Readiness:** Prepare system architecture for future integration with online application portals and student information systems.

# 6.3 Appendices

### **Sample Enrollment Questionnaires:**

### **Admissions Staff Questionnaire:**

that apply)

1.	How many enrollment applications do you process daily?
2.	On average, how long does it take to process a single application from submission to decision?
3.	How do you currently track application status? (e.g., Ledger book, spreadsheet, filing system)
4.	How often do you need to generate enrollment reports (e.g., application counts, enrollment statistics)?
5.	What are the biggest challenges you face with the current enrollment system? (Select all

- Application processing is too slow
- Difficulty tracking application status
- Data entry errors are frequent
- o Generating enrollment reports is difficult
- Lost or misplaced applications

- o Difficulty communicating with prospective students
- 6. Do you believe a new automated enrollment system would improve efficiency? (Yes/No)
  - o Yes
  - o No

# **Acknowledgments:**

I would like to express my heartfelt gratitude to my teacher for their invaluable guidance, encouragement, and support throughout the development of this Enrollment Management System. I am also grateful to the admissions staff and academic personnel who provided useful insights into the enrollment process during the research and analysis phase. My sincere appreciation goes to the prospective students and parents who participated in the surveys and offered feedback that contributed to shaping the system. Finally, I extend my deepest thanks to my family for their unwavering support, patience, and motivation throughout the course of this project.