CHAPTER 4 SYSTEM ANALYSIS AND DESIGN

4.1 Introduction

This chapter outlines the study and design process for the Adami Tuition Management System. The analysis step is critical in designing an application since it ensures that the application constructed satisfies all of the defined requirements.

The object-oriented technique is used in this project to provide the analyzed data. As a result, this chapter creates UML diagrams for the proposed system. Section 4.2 discussed the examination of system requirements analysis, including both functional and non-functional needs. Following that, Section 4.3 explores UML diagrams, including a use case diagram, a sequence diagram, and an activity diagram. Section 4.4 describes the proposed system's flowchart, database design, and interface design, the latter of which comprises the class diagram, database schema, and data dictionary. Finally, Section 4.5 summarized this chapter.

4.2 System Requirements Analysis

A software requirement is a functional or non-functional requirement for the system to implement. Functionality refers to the provision of a specific service to the user. Additionally, a software requirement can be non-functional; for example, it can be a performance requirement.

4.2.1 Functional Requirement

The primary goal of functional requirements in a requirement specification document is to define all of the system's activities or processes. These are derived from encounters with system users. The functional specifications. Table 4.1 below summarizes the proposed system.

No.	Modules	Function	User
1.0	Registration Module	• Enable user to create a new account by entering personal information • Display message if the user has any invalid input • Display message the user if registration is successful	• Student • Parent • Tutor
2.0	Login Module	• User can log in with username and password with associated database record • The system will send authentication code in email • After a successful login, the program should redirect users to the dashboard.	• Admin • Tutor • Student • Parent
3.0	Manage user data record module	• The system allow user to input, edit and update.	• Tutor • Student • Parent
4.0	Manage Subject	• User can choose the subject • User can add, update and delete for the chosen subject	• Student
5.0	Manage Class	• System allows to view list of classes	• Tutor
6.0	Manage Timetable	• The system shall enable the user to create the tuition center's schedule. • The system shall enable the user to delete the tuition center's schedule.	• Admin
7.0	Manage Attendance	• User generate QR Code • User allowed to scan attendance using QR Code • User allowed to take information from another user	• Tutor • Student
8.0	Record Payment	• System display student payment details • System display tutor wages and overtime payment • System display summary payment	• Admin
9.0	View Homework	• User generate homework assignment • User allowed to upload homework in the system	• Tutor • Student
10.0	Report	• System display report monthly	• Admin

Table 4.1: The proposed system's functional requirements

According to Table 4.1, the proposed system modules are as follows: registration module, login module, manage user data record module, manage subject, manage class, manage timetable, manage attendance, record payment, view homework and report. Additionally, Table 4.1 describes the functionality of each module and user that associated with each module.

4.2.2 Non-Functional Requirement Analysis

i) Security requirement

- The functional requirement specifies the security needs for software.
- The system enables authorized users to log in and perform system functions.
- The system verifies the user login credentials entered by users against any database record.

ii) Interface requirement

- Aesthetically pleasing interface that is presented for the user.

iii) Performance requirement

- For users, the website's load time should not exceed two second.
- The system must be available 24 hours a day.

iv) Reliability

- The system performs without failure for 1000 user below.

v) Data Integrity

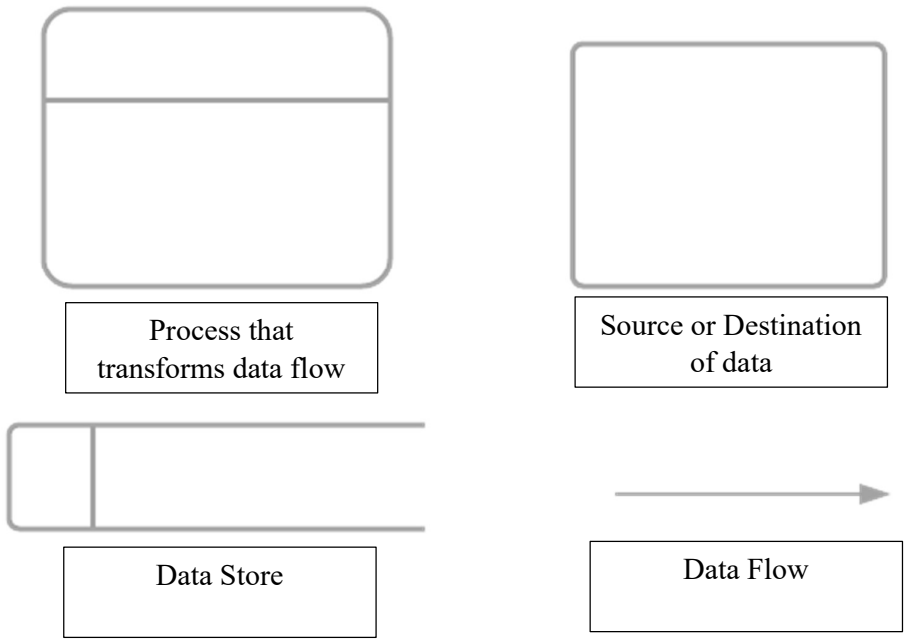
- The system makes sure data integrity by periodically backing up all database updates for each record transaction.

4.3 Data Flow Diagram

The primary function of design is to process and communicate information. Planning the information flow would result in a significantly more efficient design process. Mapping the information flows between designers would aid in comprehending the design process. DFD is an appropriate technique for modelling the design system because it is composed of processes connected by interfaces[1]. A DFD, alternatively referred to as a "bubble Chart," is used to define system requirements and indicate significant transformations that will become programmed during system design. Thus, it serves as the starting point for the design down to the smallest detail. A DFD is composed of a succession of bubbles connected by the system's data flows.

The construction of a DFD comprises of the following:

1. A square denotes the originator (source) or destination (destination) of system data.
2. An arrow indicates the direction of data flow. It is the conduit through which data passes.
3. A circle or a bubble illustrates the process by which entering data flows are transformed into exiting data flows.
4. An open rectangle is a data storage device, data in transit, or a temporary data repository.



4.3.1 Data Flows Diagrams

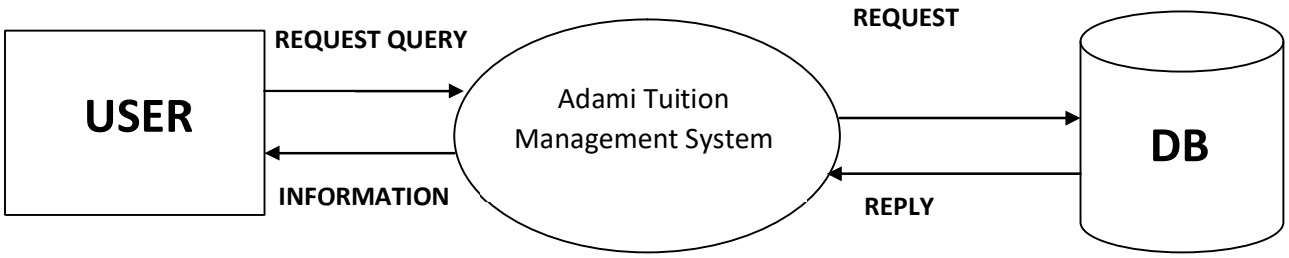


Figure 4.2: Data Flows Diagrams

4.3.2 DFD Level 1

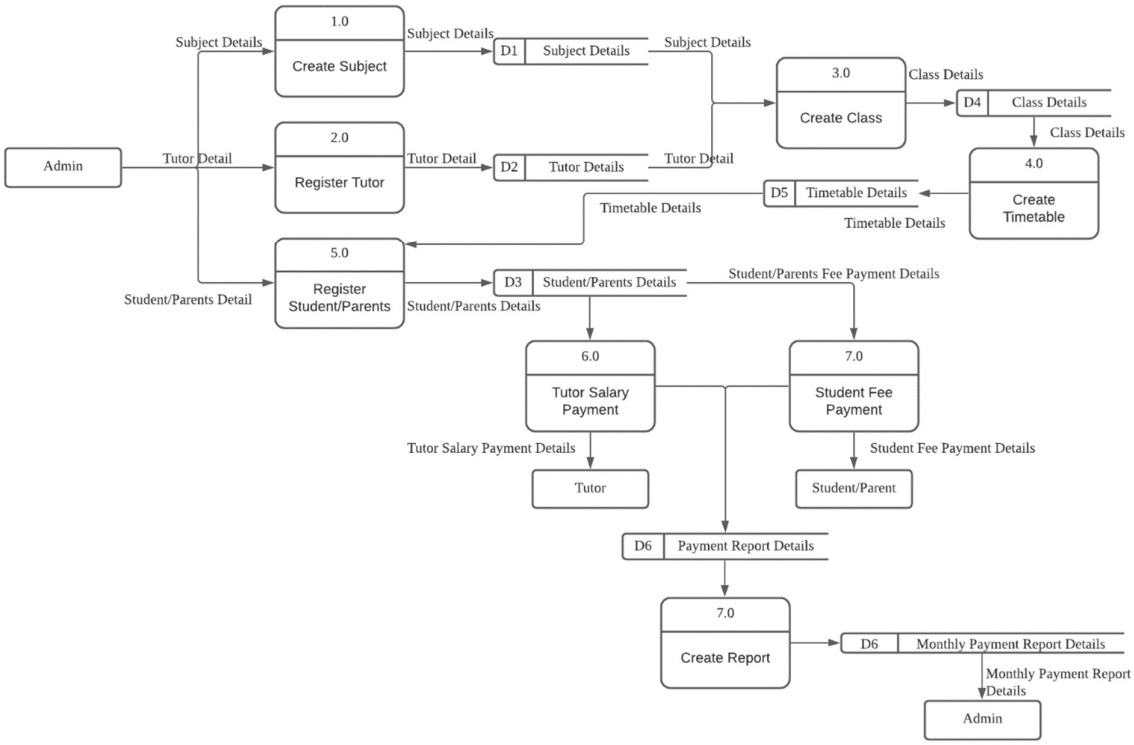


Figure 4.3: DFD Level 1

4.3.3 Activity Diagram

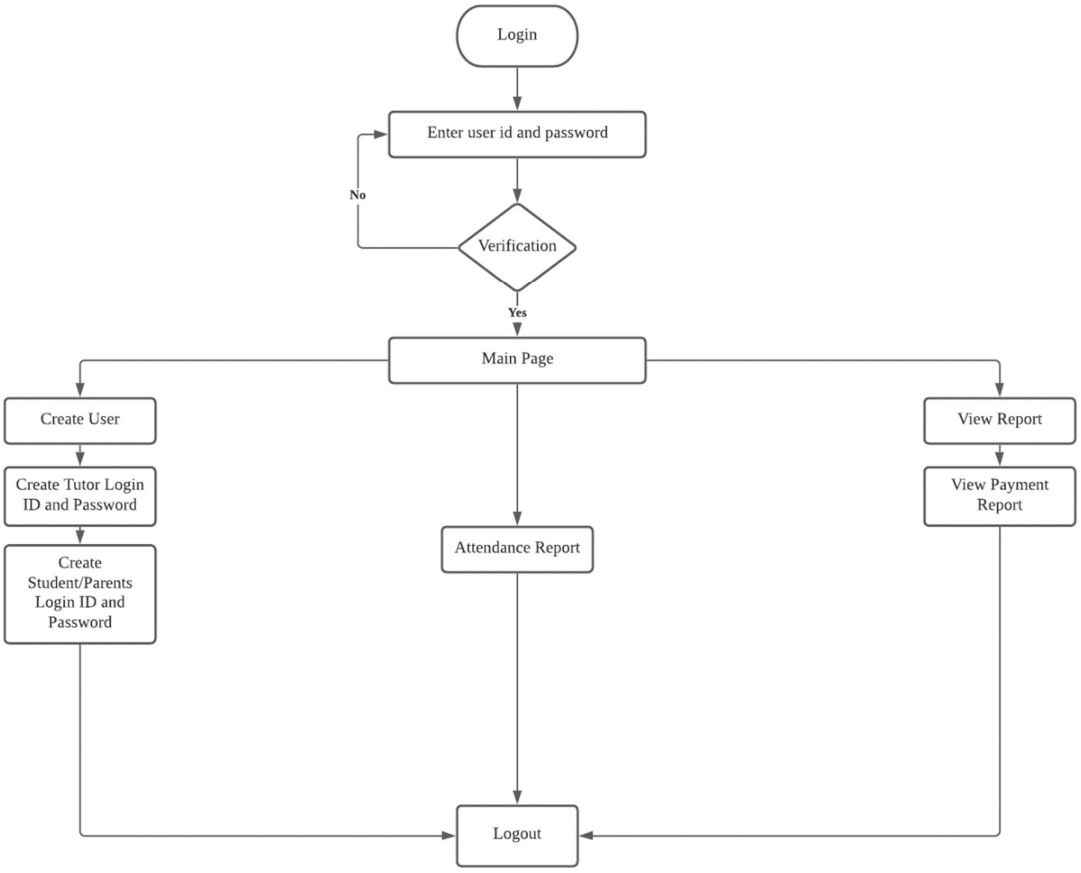


Figure 4.4: Activity Diagram for Admin

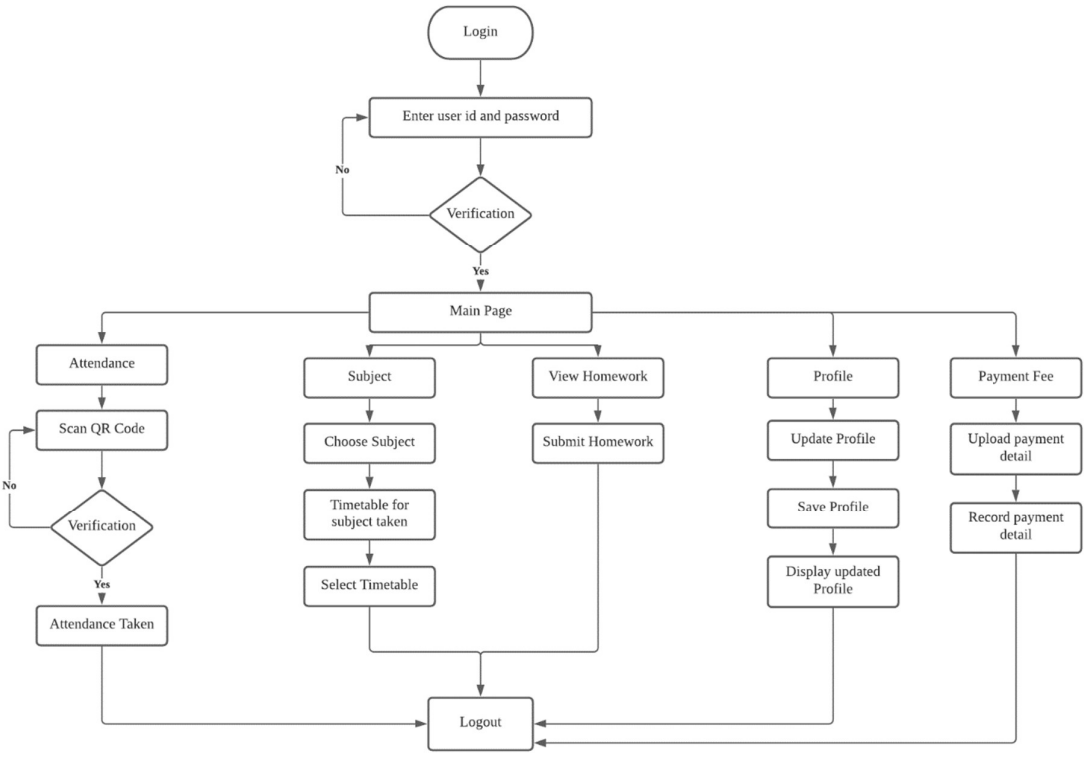


Figure 4.5: Activity Diagram for Student

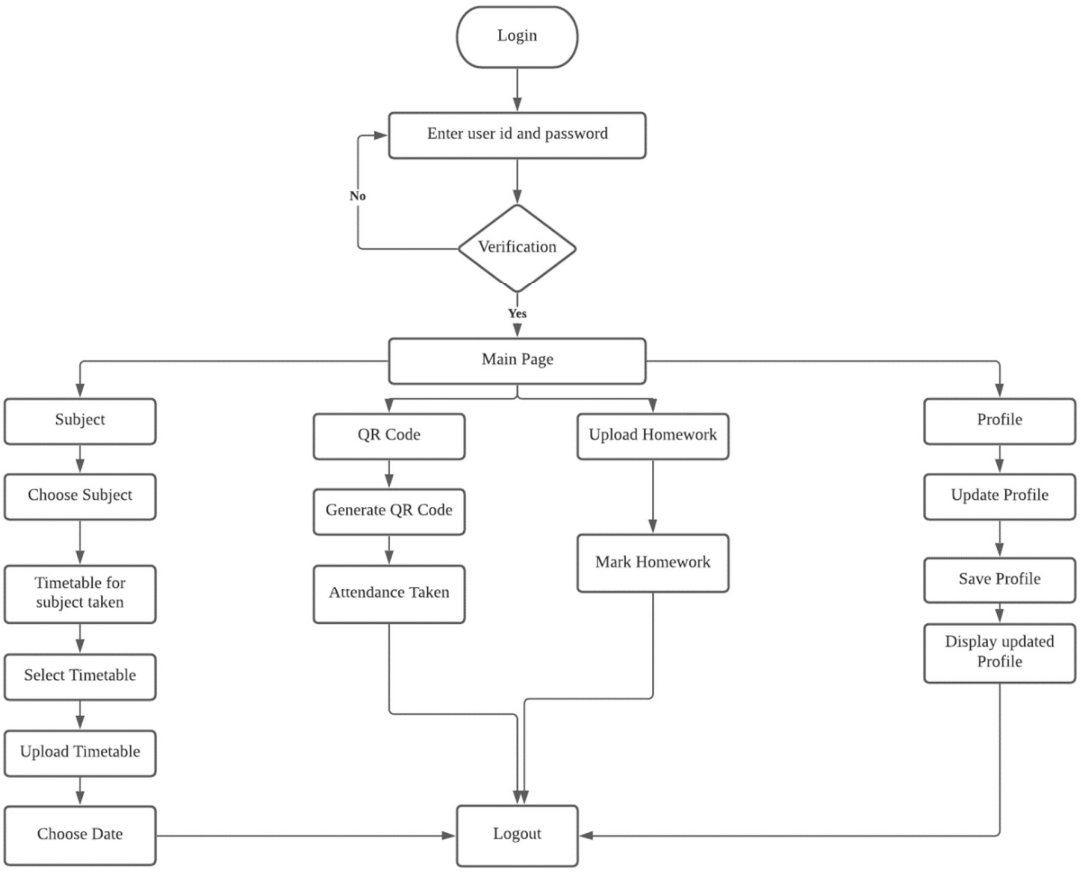


Figure 4.6: Activity Diagram for Tutor

4.4 Interface Design

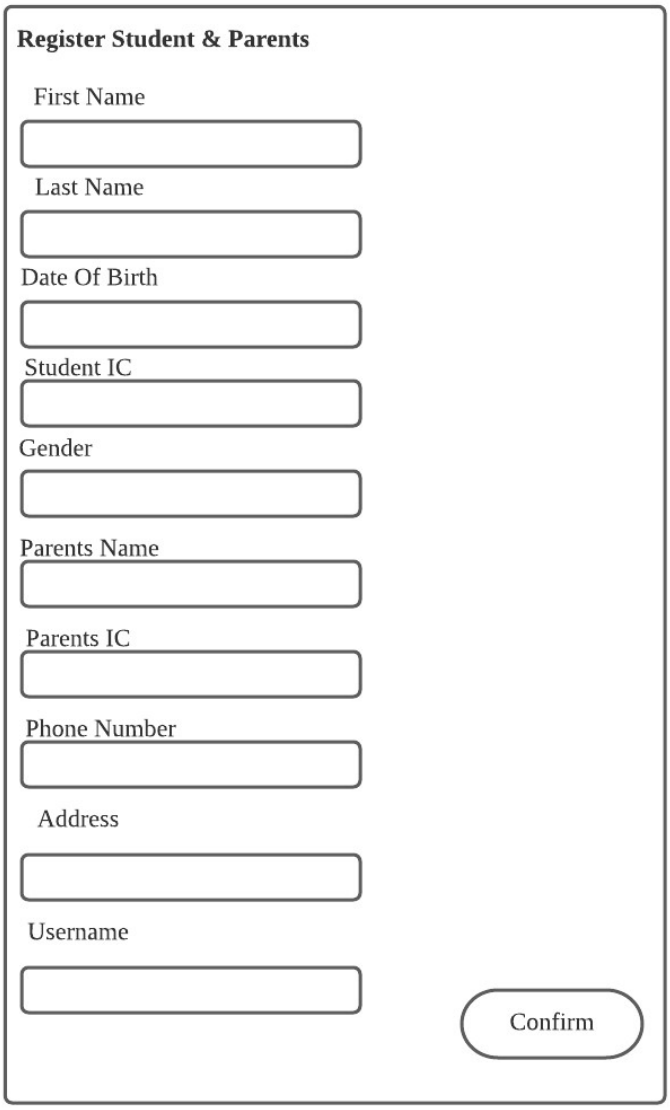


Figure 4.7: Register Page for Student/Parents

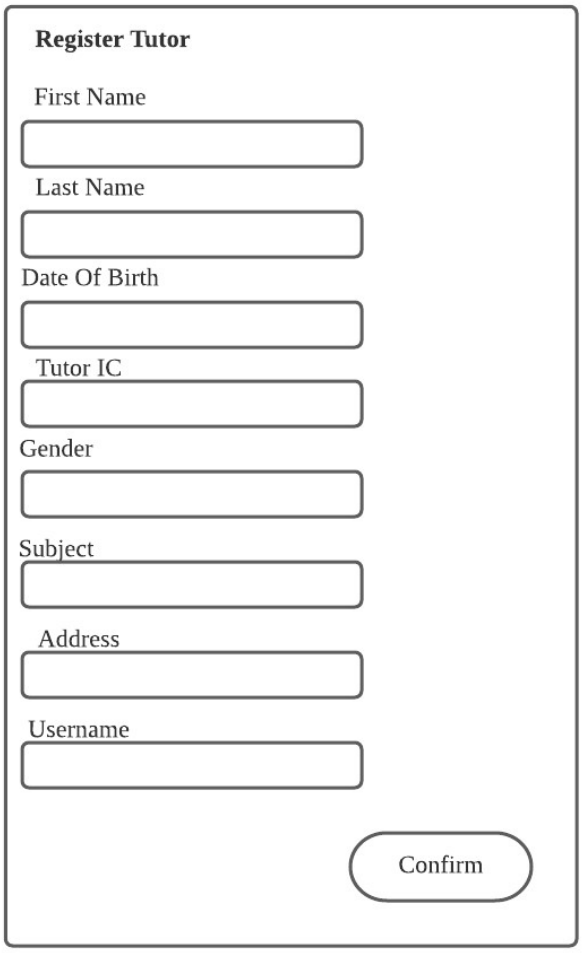


Figure 4.8: Register Page for Tutor

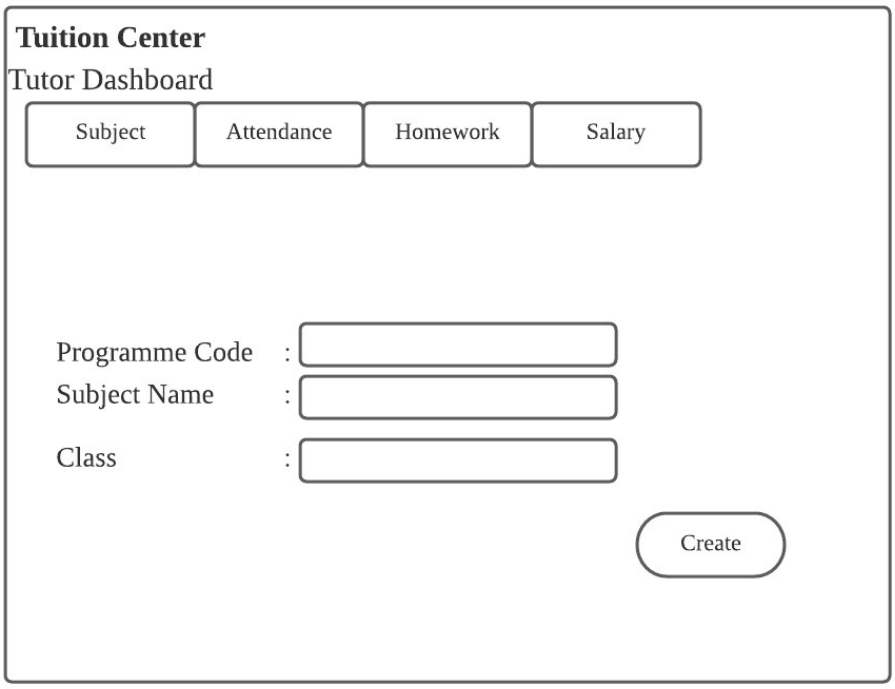


Figure 4.8: Register Page for Subject

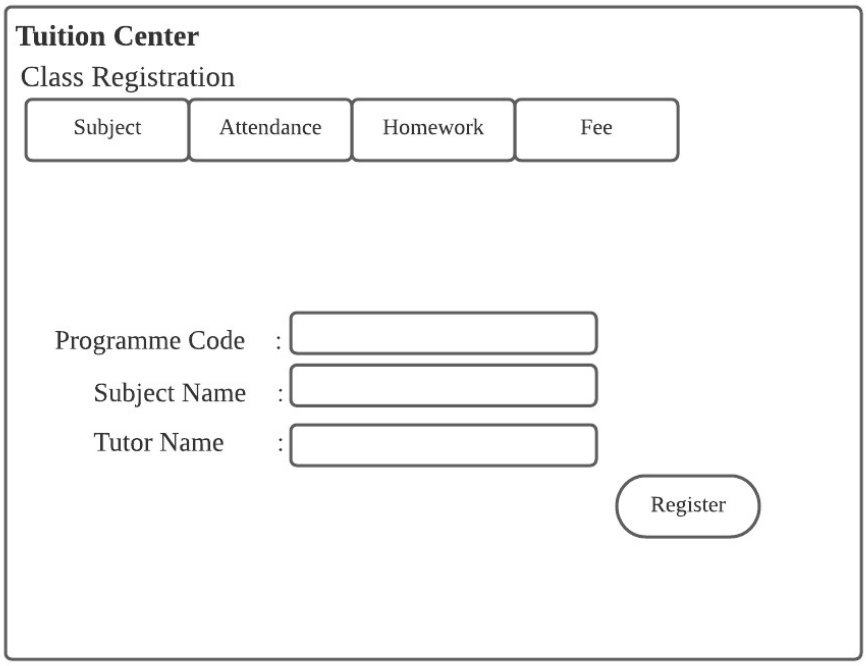


Figure 4.9: Register Subject for Student

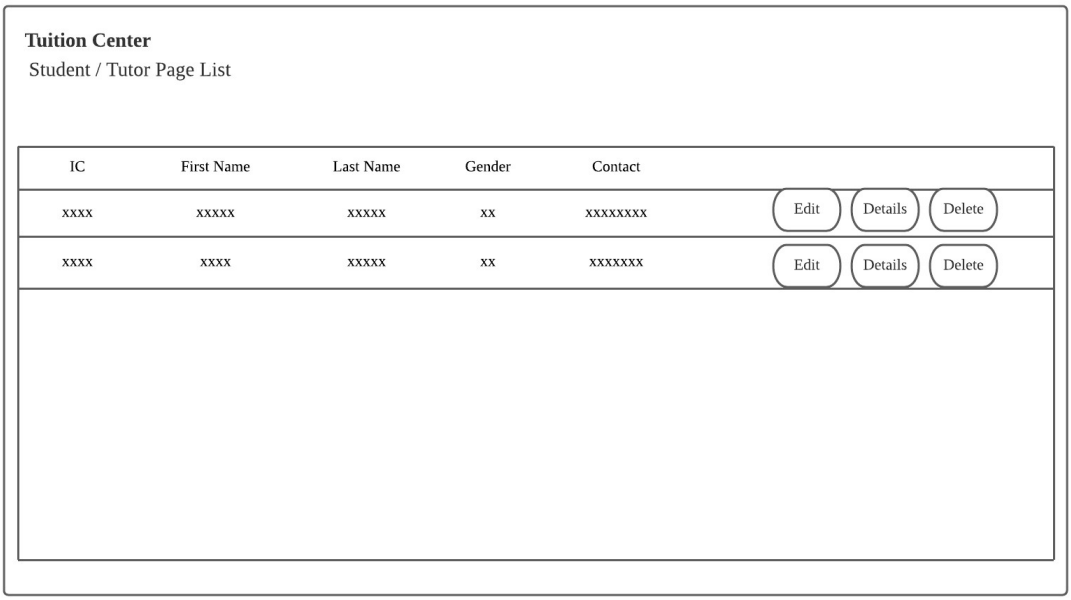


Figure 4.10: Student/Tutor Page List

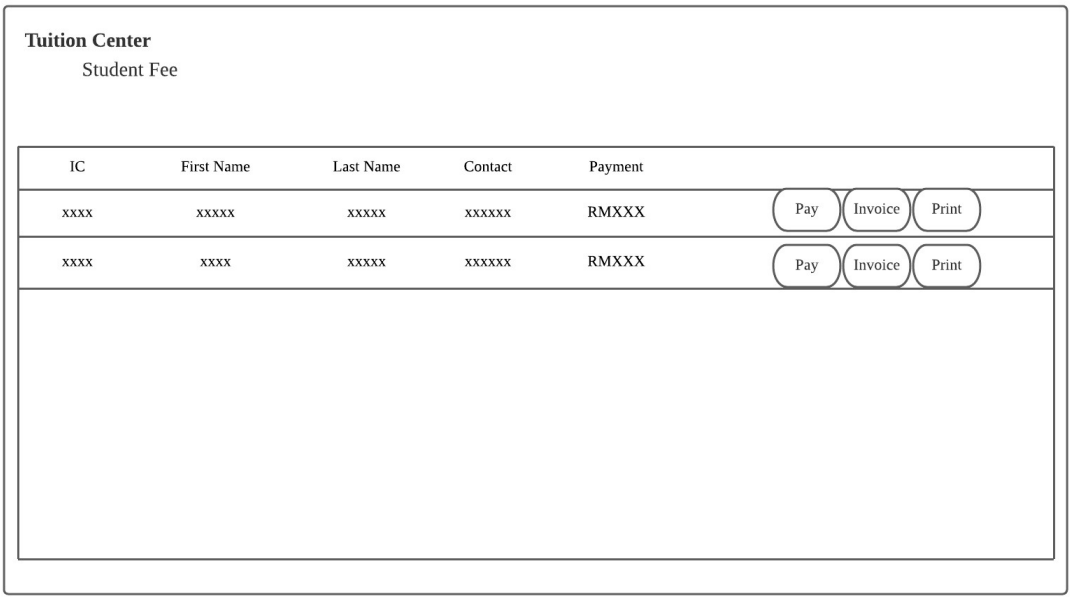


Figure 4.11: Payment Fee Page

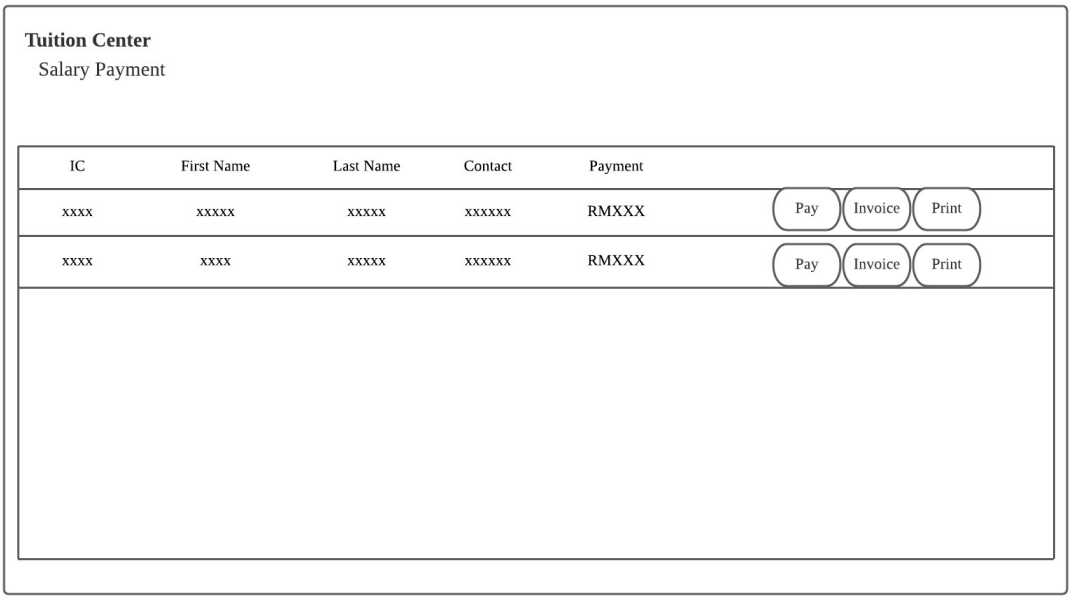


Figure 4.12: Salary Payment Page

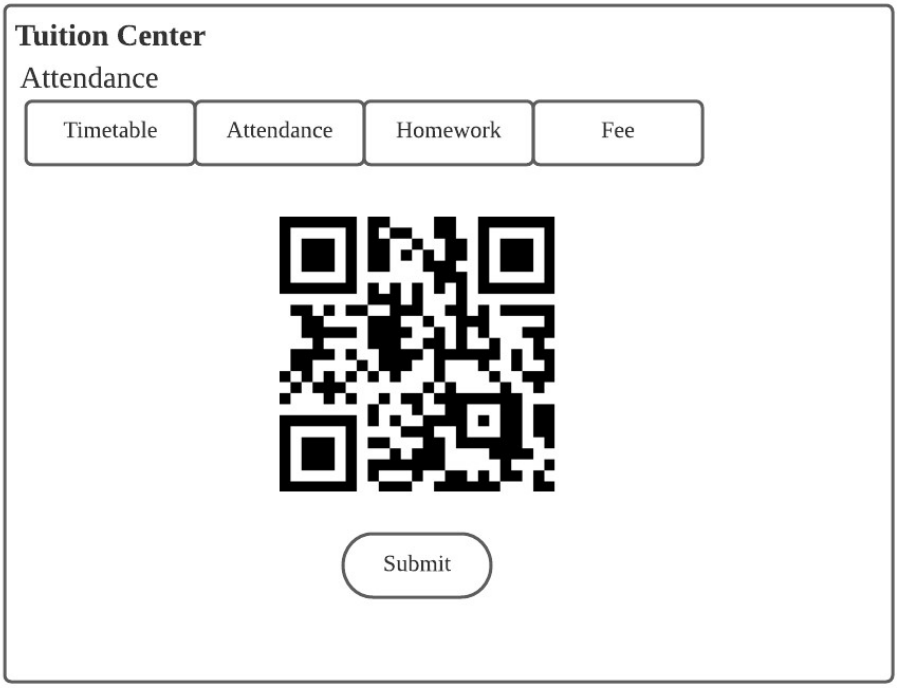


Figure 4.13: Generate Attendance QR Code

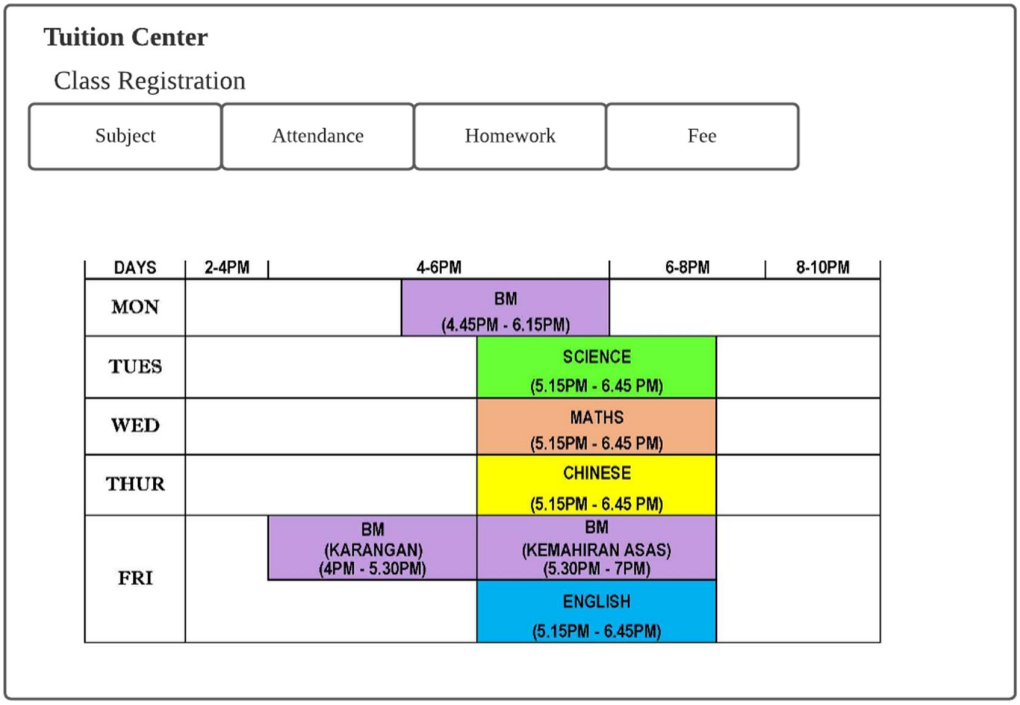
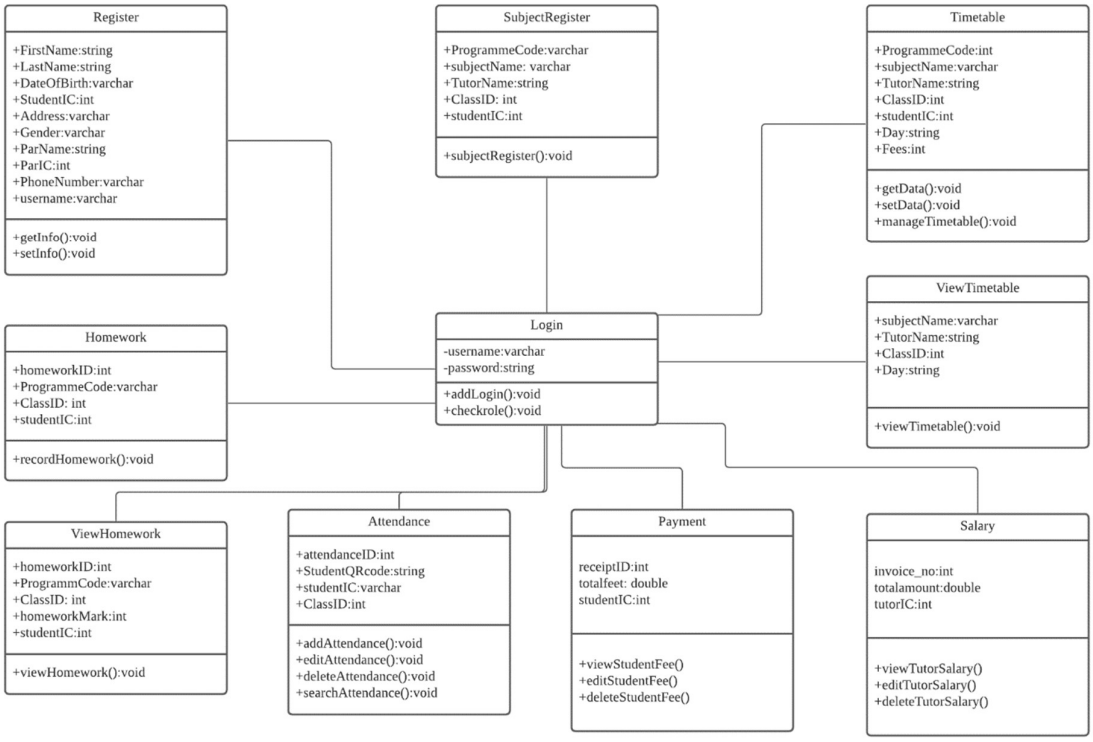


Figure 4.10: Timetable Page

4.5 Class Diagram



Adami Tuition Management Centre entities and attributes:

- Register: (FirstName, LastName, DateOfBirth, StudentIC, Address, Gender, ParName, ParIC, PhoneNumber, username)
- SubjectRegister: (ProgrammeCode, subjectName, TutorName, ClassID, StudentIC)
- Timetable: (ProgrammeCode, subjectName,TutorName,ClassID,Day,Fees)
- ViewTimetable: (ProgrammeCode, subjectName,TutorName,ClassID,Day)
- Salary: (invoice_no, totalamount,tutorIC)
- Payment: (receiptID, totalfee, studentIC)
- Attendance: (attendanceID, StudentQRCode, studentIC, ClassID)
- Homework: (homeworkID, ProgrammeCode, ClassID, studentIC)
- ViewHomework: (homeworkID, ProgrammeCode, ClassID, studentIC)
- Login: (username, password)

4.5.1 DATA DESIGN

Data design is the process through which the information domain model is transformed into data structures. The relationships between data objects are established in the ERD, which serves as the foundation for data design. I utilize a relational model to describe the entities and their characteristics in this system. To convert the ERD to a relational model, I assigned characteristics to each entity. Below are the table of attribute, data type, length, key and description.

Attribute	Data Type	Length	Key	Description
FirstName	String	30		User first name
LastName	String	30		User last name
DateOfBirth	Date			User date of birth
studentIC	int	12	PK	Student identification card
Address	Varchar	50		User address
ParName	String	30		User's parent name
ParIC	int	12		Parent identification card
PhoneNumber	Varchar	15		User's phone number
username	Varchar	15		Username for login to the system

Table 4.1: Student Table

Attribute	Data Type	Length	Key	Description
ProgrammeCode	Varchar	10	PK	Subject code
subjectName	Varchar	20		Subject name
TutorName	String	30		Tutor name for selected subject
ClassID	int	10	PK	Class ID for selected subject
studentIC	int	12	FK	Student identification card

Table 4.2: Subject Register Table

Attribute	Data Type	Length	Key	Description
ProgrammeCode	Varchar	10	FK	Subject code
subjectName	Varchar	20		Subject name
TutorName	String	30		Tutor name for selected subject
ClassID	int	10	FK	Class ID for selected subject
studentIC	int	12	FK	Student identification card
Day	String	10		Timetable day
Fees	Int	10		Student fees for selected subject

Table 4.3: Timetable

Attribute	Data Type	Length	Key	Description
homeworkID	Int	10	PK	Homework ID
subjectID	Int	10	FK	Subject ID
ClassID	Int	10	FK	Class ID for selected subject
StudentIC	Int	12	FK	Student identification card

Table 4.4: Homework Table

Attribute	Data Type	Length	Key	Description
attendanceID	Int	10	PK	Attendance ID
StudentQRcode	String	30		Quick Response code for attendance
studentIC	Int	12	FK	Student identification card
ClassID	Int	10	FK	Class ID for selected subject

Table 4.5: Attendance Table

Attribute	Data Type	Length	Key	Description
receiptID	Int	10	PK	Receipt ID
Totalfee	Double	20		Monthly fee for student
studentIC	Int	12	FK	Student identification card

Table 4.6: Payment Table

Attribute	Data Type	Length	Key	Description
Invoice_no	Int	10	PK	Invoice number code
Totalamountpayment	Double	20		Monthly salary for tutor
tutorIC	Int	12	FK	Tutor identification card

Table 4.7: Salary Table

Attribute	Data Type	Length	Key	Description
programmeCode	Varchar	10	FK	Subject code
subjectName	Varchar	10		Subject name
TutorName	String	30		Tutor name
ClassID	Int	10		Class ID for selected subject
Day	String	10		Timetable day

Table 4.8: View Timetable Table

Attribute	Data Type	Length	Key	Description
homeworkID	Int	10	PK	Homework ID
subjectID	Int	10	FK	Subject ID
ClassID	Int	10	FK	Class ID for selected subject
StudentIC	Int	12	FK	Student identification card
homeworkMark	Int	100		Mark for submitted homework

Table 4.9: View Homework

4.6 CONCLUSION

The initial analysis and design of the proposed system have been completed in this chapter to ensure the system can perform properly during the implementation phase. This chapter discuss the details of system requirement, which include both functional and non-functional needs. Additionally, this chapter analyses and visualizes the proposed system's architecture using an object-oriented methodology. The following chapter will discuss implementation and testing phase.

Reference

[1] L. J. M. C. G.-B. B. C. P. M. O. S. Díaz Vicente García, “14.2.1 What is UML?,” *Advances and Applications in Model-Driven Engineering*. IGI Global, 2014. [Online]. Available: https://app.knovel.com/hotlink/khtml/id:kt00U7R14A/advances-applications/what-is-uml