STUDENT MANAGEMENT SYSTEM

Rakib Hasan
dept.of IRE
Bangabandhu Sheikh Mujibur Rahman Digital University,
Bangladesh.
1801047@iot.bdu.ac.bd

Fardin Ahmed Ashan
dept.of IRE
Bangabandhu Sheikh Mujibur Rahman Digital University,
Bangladesh.
2001033@iot.bdu.ac.bd

***Abstract*—Unlocking the Future of Education: Our 'Student Management System' is a Java-based technological system designed to empower students and educators alike. With two distinct user roles, students and administrators, our system offers seamless registration, personalized profiles, and administrative control. Students can effortlessly sign up, manage their information, and choose any four subjects. Administrators gain the ability to create ,update ,delete and see student profiles. This paper showcases the development process, system features, and the transformational potential of our application in enhancing technology-driven education accessibility.**

***Index Terms*—Student management system, Object, Encapsulation, Data abstraction, Exception handling, file handling.**

I. Introduction

In an era characterized by the rapid integration of technology into every sector of our lives, the realm of education is no exception. Recognizing the transformative potential of technology-enhanced learning, we have developed a 'Student Management System' designed to revolutionize the educational landscape. This Java-based system caters to the dual needs of students and administrators, with a primary goal of empowering learning experiences. The application empowers students to seamlessly register, manage their profiles, and make informed choices about their education, while administrators gain a robust set of tools to oversee and facilitate this process.

This project is to demonstrate the application of object oriented programming (OOP) principles and techniques in developing a student management system. OOP is a programming paradigm that organizes data and code into reusable and modular units called objects. OOP provides several benefits, such as code reusability, data hiding, abstraction, and polymorphism. In this project, we used the following OOP concepts:

A. Inheritance

This is the capability of a class to derive properties and characteristics from another class. In our system, there isn't any explicit use of inheritance (e.g., extending classes), but we have a hierarchy of classes with 'User', 'Student', and 'UserManager'. 'UserManager' is responsible for managing user and student data.

B. Encapsulation

This is the wrapping up of data and code under a single unit. We used encapsulation to hide the internal details and implementation of the classes and objects from the outside world. We declared the data members and member functions of the classes as private or protected, and provided public methods to access and manipulate them. For example, The 'User' class and 'Student' class both use encapsulation by defining private fields (username, password, batch, and subjects) and providing public getter and setter methods to access and modify these fields. This encapsulation ensures that the internal state of these classes is controlled and accessed through well-defined methods.

C. Data abstraction

This is the providing of only essential information about the data to the outside world, hiding the background details or implementation. We used data abstraction to simplify the complexity and design of the system. We defined abstract classes and interfaces that represented the general concepts and behaviors of the system, and then we implemented them in concrete classes. For example, The 'UserManager' class abstracts the details of working with 'user' and 'student' data by providing methods like 'registerStudent', 'login', 'showStudents', etc.

II. Purpose

The purpose is to design a college or school website which contains up-to-date information about the college or school. That should improve the efficiency of college or school record management.

A. Objectives

- Increasing the efficiency of college or school record management.
- Decrease time required to access and deliver student records.
- To make the system more secure.

B. Organization of paper

The paper is organized as follows: Section (III) explains system design. Section (IV) provides the language used. Section (V) covers, methodology and Section (VI) the conclusion and section (vII) covers the References

III. System Design

This deals with data flow diagrams, detailed flow graphs, requirement analysis, and the design language of the student management system.

A. Data flow diagram

A Data Flow Diagram (DFD) is a graphical representation of the "flow" of a Student Management System. A data flow diagram can also be used for the visualization of Data Processing.DFD shows the interaction between the system and outside entities. This context-level DFD is then "exploded" to show more detail of the system being modeled. A DFD represents the flow of data through a system. Data flow diagrams are commonly used during problem analysis. It views a system as a function that transforms the given input into the required output. The movement of data through the different transformations or processes in the system is shown in the following Data Flow Diagrams.

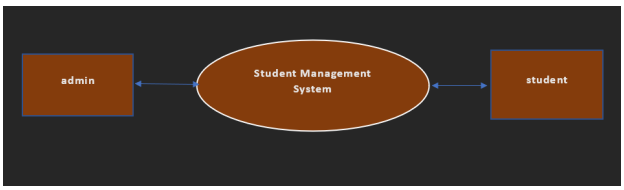

Fig. 1. Users of the student management system.

B. Admin's interface diagram

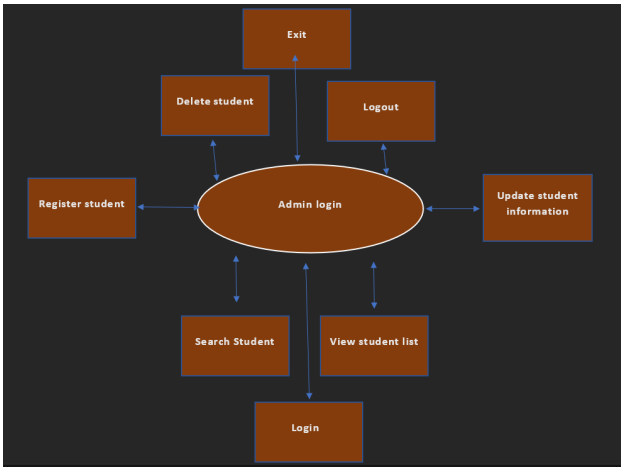

Fig. 2. Admin panel of the student management system.

The admin panel serves as a centralized command hub for administrators within the educational institution. This panel offers a wide range of tools and functionalities for efficient data management and administrative oversight. Administrators can access and manage the student list, creating and curating student profiles as needed. They can also update student status, ensuring that data remains accurate and up-to-date. The admin panel simplifies administrative tasks, promoting transparent communication and facilitating the effective management of educational processes within the institution.

C. Students' interface diagram

The student panel is tailored to provide a user-friendly and personalized experience for students. It allows students to complete the registration process seamlessly, providing their personal information, session details, and subject choices. After registration, students can log in to their profiles to view their sessions and the subjects they have chosen. This panel empowers students to shape their educational journey, fostering personalized learning experiences and putting control in their hands.

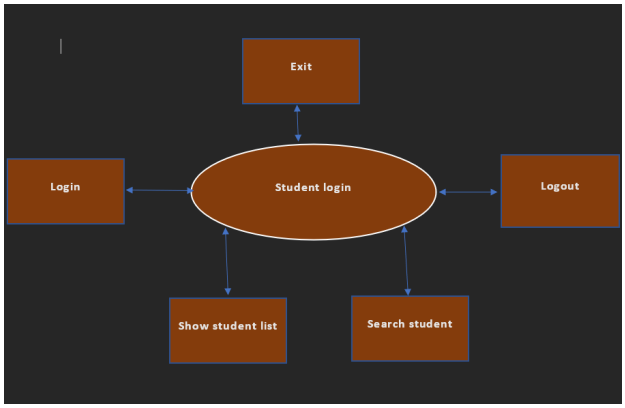

Fig. 3. Student panel of the student management system.

D. Details data flow diagram.

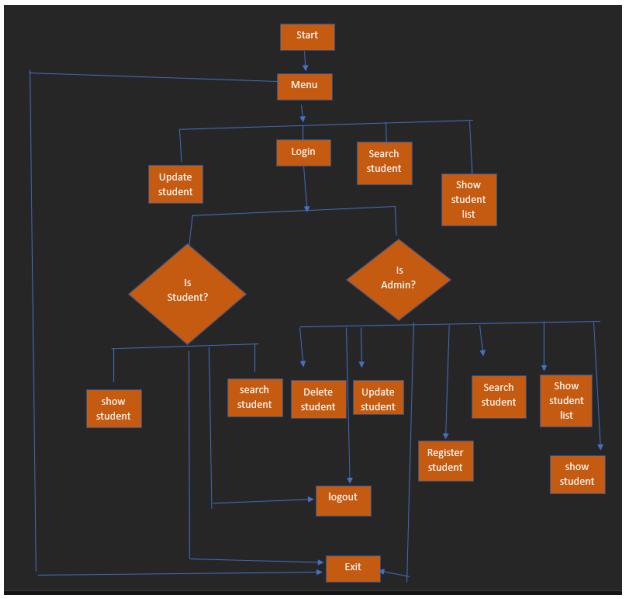

Fig. 4. Flow chart of the student management system.

The detailed flow graph is shown in Fig. 4 .The design of the student management system includes the design of the menu page which provides a way for all the students and admin to access the Student Management System. Every user of the SMS has a unique username and password provided by the admin of the college or school. The menu page mainly contains a login form through which a new user can register,or an existing user can login to the system by entering the username and password provided by the web master.

This paper mainly focuses on the management of the information of the students which is maintained by the school or college administration through various levels of control. The function of the individual entities will be explained in detail in the flow graph.

IV. Language

We built our project in java because Java is an object-oriented language, and its core features include encapsulation, inheritance, and polymorphism, enabling developers to create well-structured and reusable code. The language offers a rich set of standard libraries, known as the Java Standard Library, covering data structures, networking, and more. Java is widely used in web development, thanks to frameworks like Java EE (Enterprise Edition) for building enterprise-level web applications. It's also the foundation of the Android operating system, powering the majority of smartphones and tablets. Java's strong emphasis on security includes features like automatic memory management, runtime access controls, and exception handling, making it a trusted choice for critical applications. Additionally, Java supports multi-threading, facilitating concurrent programming. The language has a large and active developer community, offering an extensive array of tools, third-party libraries, and open-source projects. Java's enduring popularity, along with its robustness and portability, makes it a cornerstone of the software development world.

V. Methodology

The methodology for our 'Student Management System' project centers on the development and functionality of the menu-driven user interface, which serves as the primary interaction point for both administrators and students. The project initiates by presenting users with a menu bar that offers four main options: registration, login, displaying the user list, and logging out. Two distinct user roles are defined - admin and student. Admins are granted the privilege to create student accounts, using usernames and passwords, and they can subsequently log in using their credentials. Students, on the other hand, register and then log in with their own usernames and passwords. Once logged in, students can access a limited set of functions, primarily focused on viewing user lists, their current session, and the subjects they've selected. Meanwhile, administrators wield the power to create, update, delete, and display user information, with these actions resulting in the creation of relevant user data files on the local disk D. The methodology for this project combines a user-friendly menu interface, user role differentiation, data management, and file handling to create an effective and accessible system for educational institutions. It emphasizes ease of use, security, and data integrity, ultimately providing a practical and efficient tool for both students and administrators.

VI. Conclusion

In conclusion, the 'Student Management System' project stands as a testament to the power of technology in transforming the educational landscape. By offering students and administrators a user-centric, menu-driven interface, it simplifies and streamlines various aspects of educational management. The project's successful implementation ensures that students have easy registration, personalization, and access to their academic information, while administrators benefit from robust tools for data management and oversight. The application also emphasizes data integrity and file handling, which enhances security and accessibility. The 'Student Management System' serves as an effective bridge to an educational environment characterized by accessibility, personalization, and efficiency, opening new doors to the future of learning.

VII. References

1st ref= [1] 2nd ref= [2]

References

[1] SR Bharamagoudar, RB Geeta, and SG Totad. Web based student information management system. *International Journal of Advanced Research in Computer and Communication Engineering*, 2(6):2342–2348, 2013.

[2] AA Eludire. The design and implementation of student academic record management system. *Research Journal of Applied Sciences, Engineering and Technology*, 3(8):707–712, 2011.