Student Portal Information Using Power app

Niagara College Toronto

Rapid Application Development (RAD) Client-Server Project (PROG1210-G1)

24 September 2024

**Group Member**

* Dinal Fagami
* Fellipe Macedo
* Jennifer Egbele
* Misan Regmi
* Siddhartha Gurung

**Executive Summary**

This proposal outlines the development of a **Student Information Portal** using **Microsoft Power Apps**. The development shall be done in a way that the portal manages all the data of students effectively within educational institutions through an effective, easy-to-use interface for students, faculty members, and administration staff. In today's modern world, most educational systems face major problems related to atomized systems, which are old and pose problems in accessing and managing student information with efficiency. The focus of the project is to ensure that all these challenges are addressed through the provision of a one-stop platform that avails, in real time, access to critical data regarding students.

The following might help develop operational efficiency, reduce the administrative burden, and further enhance the student experience. The Student Information Portal will utilize Power Apps-a low-code development platform-which will be highly integrated with other Microsoft products, such as SharePoint and Excel, for a rapid deployment, scalability, and at lower costs. The necessary functionalities provided in the portal will include course registrations, tracking grades, and attendance with proper profile management. It is strictly done with data security that has access-based control dependent upon roles. The **main goals** of the project are as follows:

1. **Centralization of Student Data**: Develop a unified platform that offers ease of access and management of student data to the student, teacher, and administrator. The Following are some of the features speculated for the platform:
2. **Improved Student Experience:** A personalized dashboard displaying grades, attendance, upcoming deadlines, and course registration management for the student.
3. **Administrative Efficiency:** Automate administrative processes of enrollment and course management to free up educators' and staff's time for other more pressing tasks.
4. **Assurance of data security:** Use role-based access control and data encryption to protect sensitive student information from being stored or accessed by unauthorized individuals.
5. **Scalability:** The portal should be easily updated and expanded; it needs the ability to support future enhancements with evolving institutional requirements.

By adopting the Student Information Portal through Power Apps, educational institutions will make a remarkable development in the process of managing the data stored within the portal by rendering it a more connected, user-friendly, and secure solution for all stakeholders.

**Project Background & Related Work**

SISs development has been a major focus of educational institutions worldwide to come up with better records management, administrative streamlining, and improvements to the students' experiences. Conventionally, every institution has used either a legacy system or a custom-developed system, most of which are characterized by high development and maintenance costs. During the recent era, low-code systems like Microsoft Power Apps have been amongst the most effectual means of building dynamic and scalable systems without taking an extensive basis in coding. This review, therefore, summarizes some of the key related works within the field of student information systems-major approaches, technologies, and trends-and how this project intends to innovate by leveraging Power Apps.

**1. Traditional SIS Platforms and Legacy Systems**

Early systems of student information were in-house and tailored, many times requiring high development investments and heavy IT support. While such systems work, maintenance problems, scalability issues, and integration with new cloud technologies have been some of the challenges. Various literature reviews, like the one by Zornada and Velkavrh (2005), describe those different kinds of legacy systems. These authors show that most of those systems are rigid in some aspects and hence not flexible enough to respond to the challenges of modern educational institutions. Although they might manage simple tasks of course registration and grade tracking, they are usually siloed and it may be difficult for other departments to share data. This project will look to exceed these limitations through exploitation of cloud integration and the flexibility of Power Apps, such that the data is stored centrally and retrieved from any department with much ease.

**2. Web-Based or Cloud-Hosted Student Information Systems**

Many institutions today have moved to modern cloud technologies by implementing web-based or cloud-hosted SIS solutions. A review by Bhatti et al. (2018) identified the benefits of cloud-based student management systems, especially in terms of cost efficiencies and scalability. They mentioned that the cloud-hosted system had facilitated the update process and allowed access from a distance, besides providing collaboration amongst the concerned stakeholders like students, faculty, and administration. Most of these applications require extensive development cost during the initial phase and also need professional IT support for customization. On the other hand, Power Apps is a low-code solution that allows institutions to develop and deploy custom applications much more quickly with lower upfront investment and even lower maintenance burden, thus reducing the overall cost of ownership.

3**. Low-Code Platforms for Education**

During the last few years, a number of low-code and no-code platforms have emerged as popular options because their mainstay is democratizing application development. In a research work carried out by Ebert and Duarte (2018), low-code platforms like Power Apps were surveyed to establish their potential in developing an application that suites specific needs in the shortest time without deep coding knowledge. The authors have indicated how educational institutions can utilize these platforms to effectively make the administrators develop and manage specific applications, tailor-made according to their requirements without actual complex IT infrastructures. In view of low code development as an increasingly active area, this project with Power Apps shall be presenting a customized Student Information Portal which works well with the prevailing Microsoft applications such as SharePoint and Excel.

**4. Student Information Systems using Microsoft Technologies**

From integration of the Microsoft technologies into educational platforms, the interest has been growing over time. Khatri and Soni, 2020, reviewed some Microsoft tools like Power BI, Power Automate, and SharePoint that are used in the management of educational data. These tools offer robust solutions for data visualization, workflow automation, and document management. Power Apps are at an advantage as, within the ecosystem, it is integrated with these tools to provide a holistic education platform. Herein, the project proposes that Power Apps leverage its integration with Power Automate for automating workflows and Power BI in real-time analytics and reporting the performance of the students for a more integrated approach with greater effectiveness as compared to older approaches.

**5. Mobile and User-Centric SIS Platforms**

Recently, the tendency is toward mobile and user-centered student information systems. Alharbi et al. (2019) researched the effectiveness of the design of the Student Portal using the mobile-first methodology since students have increased dependence on mobile phones to access services, but most existing systems cannot be easily optimized for mobile use due to the complexity of the legacy infrastructure. Power Apps is responsive to mobile action by default. Therefore, students can log in to the portal from any device with ease. This project aims at developing a mobile-responsive Student Information Portal that shall enable university students to browse grades, course schedules, and personal information while on the go.

**Proposed Approach**

Guided by insights from these related works, the proposed Student Information Portal shall be developed using Microsoft Power Apps. There are several advantages accruable to this:

**• Low-Code Development:** Unlike traditional systems that require great programming and customization, Power Apps features rapid application development with less need for coding in order to reduce both time and cost.

**• Integration with Microsoft Ecosystem**: The best part about Power Apps is that it closely works together with other Microsoft offerings like SharePoint, Power Automate, and Power BI for better data sharing, workflow automation, and analytics.

• **Scalability and Flexibility:** The system will be designed to easily scale and adapt to changing institutional needs with respect to future expansions or additions of features.

**• Mobile Optimization:** A mobile-responsive design will ensure that students can access their information from any device, which moves in step with the growing trend toward mobile-first systems.

Whereas developments into student information systems are currently at an advanced stage, most of the existing platforms are either highly expensive, complicated to maintain, or lacking in certain modern features such as mobile optimization and integration with cloud-based tools. This project utilized low-code innovation through the implementation of Power Apps in developing a rather affordable, flexible, user-friendly Student Information Portal. This portal has been designed to fill the gaps in legacy and cloud-hosted systems to provide something more modern, integrated, and scalable.

**Experiment Design**

The key objective of this experiment is to design, develop, and test a functional Student Information Portal using Microsoft Power Apps. This shall involve designs in managing students' data, course registration, keeping track of grades, and smoothing communication between students and administrators. This experiment shall fall into several phases, namely requirement gathering, data collection, system development, and testing. Each of these phases shall entail data, tools, and methodologies that shall ensure that the deployment of the portal is well realized.

**1. Objectives**

•Development of working prototype of the Student Information Portal

• Functionality, usability and performance of the portal will be tested with predefined set of test cases

• User satisfaction, identification of areas to improve after receipt of feedback from students, faculty and administrator.

**2. Data Sources**

Following are the datasets which will be utilized in the experiment:

Student Information Dataset: The data set includes sample records of students, personal details such as name, student ID, contact information; academic records include course enrollment, grades by students earned, and attendance. In the event of unavailability of such a dataset in realistic grounds, separate mocked-up test datasets will be created for testing. Other Requirements: Data Fields: Personal Information - Name, Student ID, Date of Birth, Address, Email. Academic Records- Course Enrollment, GPA, Grades, Credit Hours Earned. Attendance - No of Absent days, Attendance Rate of Particular Courses.

**Course Catalog Dataset:** This dataset shall provide information regarding available courses, including course codes, names, descriptions, and prerequisites, among others, along with instructor details.

**Administrative Dataset:** This dataset shall contain user roles and permissions to be given to faculty and administrative users. This shall define what access each user role has to the student data and administrative functionalities inside the system.

**3. Tools and Platforms**

Following is the list of tools and platforms that shall be used in the design, development, and testing of the Student Information Portal:

• **Microsoft Power Apps:** This shall be the key tool to be utilized in developing the user interface and application logic for the portal. Power Apps will have the advantage of a low-code platform that is easy to customize and scale.

• **Microsoft Dataverse**: Microsoft Dataverse shall be utilized for data storage, managing structured data of students and courses. Dataverse seamlessly integrates with Power Apps and provides an enterprise-grade security and data management system.

**• Microsoft Power Automate**: For workflow automation, the system will integrate Power Automate within the portal. This would be applied to the automation of email notification tasks, confirmation of course enrollments, and grade updates.

• Microsoft Power BI: For data visualization and reporting, Power BI will be used to create dashboards that represent KPIs on student attendance rates, academic performance, and trends in course registration.

• SharePoint: SharePoint will be used for document management, especially for administrative documents related to the enrollment of students, academic reports, and records about faculty.

**4. Experimental Design Phases**

**Phase 1: Requirement Gathering**

• **Objective**: Identify major requirements from significant stakeholders like students, faculty members, and administrators.

• **Research Method:** Interviews or questionnaires are to be conducted to realize the needs and preferences of users. Requirements are to be documented to align with project goals.

• **Expected Output:** Clearly defined functional and non-functional requirements, to assist in guiding the development process.

**Phase 2: Data Collection and Setup**

• **Objective:** Create datasets needed during development and testing of the portal. This could include either collecting real data or creating simulated data based on institutional needs.

**• Approach:** Sample student information, course details, and administrative data should be created in Microsoft Excel or Dataverse.

**• Deliverable:** To provide a fully populated dataset in order to test portal functionalities, such as course registration, grade tracking, and profile management.

**Phase 3: System Development**

**Objective**: Development of Student Information Portal on Microsoft Power Apps

**Steps:**

**Interface Design:** It will be done with Power Apps, the designing of an intuitive interface for students, faculties, and administrative staff. The UI should comprise dashboards for displaying lists of courses, enrollment forms, grade displays, and profile management features.

**Data Integration**: Data stored in Microsoft Dataverse has to be integrated with the portal so that the data about student and academic records can be accessed in real time using Power Apps.

o **Automation Setup**: Use Power Automate for automation of workflows around enrollment notifications, grade updates, or other attendance alerts.

o **Reporting Integration:** Allow Power BI integration to visualize data reports based on the students' performance and trends at the institution.

• **Expected Output:** A functional prototype of the Student Information Portal integrating all key functionalities and data.

**Testing Phase**

**Objective:** The objective is to test the functionalities of the portal for usability, performance, and so on. The system will be required to meet the defined requirements and function reliably under any circumstance.

**Methodology:**

**a. Unit Testing**: Every small unit in the system must be tested, such as data entry forms, course registration workflows, and grade tracking.

**b. Integration Testing:** Test the integration of each and every system component, which includes data from Dataverse, workflows from Power Automate, and visualizations from Power BI.

**User Acceptance Testing (UAT):** Engage a set of users-students, faculty, and administrators-interacting with the portal, thereby getting feedback related to usability, performance, and overall satisfaction. The tools for soliciting feedback would be either questionnaires or interviews.

**Security Testing:** Ensure data security so that data is available only to authorized users through role-based access controls.

• Expected Output: A usable, tested Student Information Portal that will meet the requirements of the project at hand and be very user-friendly, dependable, and reliable.

**Phase 5:** Evaluation and Feedback

Objective: To garner feedback on the system to realize its worth and area in its operations that might need adjustment.

**Methodology:**

**Surveys:** Get user feedback regarding ease of use, functionality, and overall user satisfaction with the system.

**Data Analysis**: Use Power BI to analyze trends in system use, hence highlighting some areas for possible improvement.

**• Anticipated Output**: A report outlining user feedback, system performance, and identification of areas that may need further refinement or features.

**5. Provided Code and Tools**

While the development shall be based mostly on Microsoft Power Apps and related tools, the following shall also be integrated into the project:

• Microsoft Power Apps have templates on educational applications that one can use to start building. Templates can be taken and adapted according to the needs.

**• Power Automate workflow template:** Prebuilt workflow automation templates like email notifications, approval for data entries, among others, will cut down the development time.

**• Power BI Dashboards**: It would be easy to utilize and adapt pre-built Power BI dashboards to track key metrics and trends in student data.

**6. Challenges**

• **Data Migration:** If there is any migration of existing student data from some legacy system, it may be a challenge to ascertain that the data are correct and intact during the migration process.

**• User Training:** Training may need to be provided to faculty, students, and administration because they might be new to this system, particularly those not used to the interfaces of Power Apps.

**• Scalability:** The same can be scaled up, though performance testing under heavy data sets or multiple users will be important to ensure smooth operation.

**Conclusion**

This experimental design forms the foundation of the action research approach, which systematically involves a process of designing, testing, and evaluating a Student Information Portal with Power Apps. This project, therefore, aspires to design a scalable, cost-effective, user-friendly system that leverages Microsoft's low-code platform to enhance the management of student data. The solution for overall management of students' information shall involve real or simulated datasets, automatic workflows, and data visualization tools.

Project Plan-Timeline & Milestones

Project Duration: 3 Months

The Student Information Portal Using Power Apps development will fall into discrete phases, each having detailed milestones to assure that the entire project is kept on course. This includes the planning phase, design phase, development phase, test phase, and deployment phase. Below is an itemized list of milestones and the corresponding timeline for completing the project.

**Project Plan-Timeline and Milestones**

Duration of the Project: 3 Months

The development of the Student Information Portal Using Power Apps will be divided intointo clear phases with events that keep the project on course. The phases are: planning, design, development, testing, and deployment. A milestone and timeline to complete the project in items is as shown below.

**Month 1: Planning & Design**

**Milestone 1: Requirement Gathering and Analysis (Week 1-2)**

* **Tasks**:
  + Meet with stakeholders (students, faculty, administrators) to identify key requirements for the portal.
  + Document functional and non-functional requirements, including features like course registration, grade tracking, and data security.
* **Outcome**: Detailed requirement specification document, approved by stakeholders.

**Milestone 2: Data Collection & Setup (Week 2-3)**

* **Tasks**:
  + Identify or simulate datasets (student information, course catalog, and administrative data).
  + Set up data storage using Microsoft Dataverse or SharePoint, ensuring proper organization of student records and course details.
* **Outcome**: Complete dataset ready for integration into the portal.

**Milestone 3: Design of User Interface (UI) and System Architecture (Week 3-4)**

* **Tasks**:
  + Design mock-ups and wireframes for the student, faculty, and administrator dashboards.
  + Plan the system architecture, detailing how data will flow between Power Apps, Dataverse, Power Automate, and Power BI.
* **Outcome**: Finalized UI design and system architecture diagram.

**Month 2: Development**

**Milestone 4: Development of Core Features (Week 5-7)**

* **Tasks**:
  + Build the core components of the Student Information Portal in Power Apps, including:
    - **Student Dashboard**: Display grades, course schedules, and personal information.
    - **Course Registration**: Implement functionality for students to enroll in courses.
    - **Grade Tracking**: Provide real-time access to academic performance.
  + Integrate data from Microsoft Dataverse to populate these features.
* **Outcome**: Core features of the portal built and integrated with sample data.

**Milestone 5: Workflow Automation with Power Automate (Week 7-8)**

* **Tasks**:
  + Set up automated workflows for common administrative tasks:
    - **Enrollment Notifications**: Automatically notify students and faculty of course registrations.
    - **Grade Updates**: Trigger alerts for updated grades or academic reports.
  + Ensure seamless integration with Power Apps for real-time updates.
* **Outcome**: Functional automated workflows for student registration and grade notifications.

**Milestone 6: Integration of Power BI for Analytics (Week 8)**

* **Tasks**:
  + Create Power BI dashboards to track student performance, attendance, and course registration trends.
  + Set up administrative access for generating and exporting reports.
* **Outcome**: Fully integrated Power BI reports available in the portal for data analysis.

**Month 3: Testing & Deployment**

**Milestone 7: Unit and Integration Testing (Week 9)**

* **Tasks**:
  + Perform unit testing of individual components, such as course registration forms and grade tracking.
  + Conduct integration testing to ensure smooth data flow between Power Apps, Power Automate, and Power BI.
  + Resolve any bugs or issues discovered during testing.
* **Outcome**: Verified functionality of all portal components, with no critical issues remaining.

**Milestone 8: User Acceptance Testing (UAT) (Week 10)**

* **Tasks**:
  + Invite a group of students, faculty, and administrators to test the portal and provide feedback on usability and functionality.
  + Collect feedback on ease of use, system performance, and overall satisfaction.
  + Make necessary adjustments based on feedback.
* **Outcome**: A tested, user-approved version of the portal, ready for deployment.

**Milestone 9: Final Deployment and User Training (Week 11-12)**

* **Tasks**:
  + Deploy the Student Information Portal to the live environment, ensuring that all data is migrated successfully, and the system performs as expected.
  + Conduct training sessions for students, faculty, and administrators to familiarize them with the portal’s features and functionalities.
  + Provide user documentation and support for future reference.
* **Outcome**: Fully deployed and operational Student Information Portal, with trained users and available support materials.

**Milestone 10: Project Review and Handover (Week 12)**

* **Tasks**:
  + Review project outcomes against the initial objectives and requirements.
  + Provide a final report documenting the development process, challenges faced, and solutions implemented.
  + Official handover of the portal to the institution’s IT department for ongoing maintenance and updates.
* **Outcome**: Project completion, with the portal fully handed over and project closure documentation submitted.

**Summary of Milestones and Timeline**

| **Milestone** | **Task** | **Duration** | **Expected Completion** |
| --- | --- | --- | --- |
| **Milestone 1**: Requirement Gathering | Gather user requirements and finalize project specs | Week 1-2 | End of Week 2 |
| **Milestone 2**: Data Collection | Prepare datasets and set up data storage | Week 2-3 | End of Week 3 |
| **Milestone 3**: UI/UX Design | Create wireframes and system architecture | Week 3-4 | End of Week 4 |
| **Milestone 4**: Core Features | Develop key portal functionalities | Week 5-7 | End of Week 7 |
| **Milestone 5**: Workflow Automation | Automate workflows using Power Automate | Week 7-8 | End of Week 8 |
| **Milestone 6**: Power BI Integration | Set up analytics and reports using Power BI | Week 8 | End of Week 8 |
| **Milestone 7**: Testing | Unit and integration testing | Week 9 | End of Week 9 |
| **Milestone 8**: UAT | Perform user acceptance testing | Week 10 | End of Week 10 |
| **Milestone 9**: Deployment & Training | Deploy portal and train users | Week 11-12 | End of Week 12 |
| **Milestone 10**: Project Review | Final project review and handover | Week 12 | End of Week 12 |

The project plan for the **Student Information Portal Using Power Apps** is structured over a 3-month period, with clearly defined milestones to ensure that the project progresses smoothly and on schedule. Each phase builds on the previous one, from gathering requirements and designing the system to development, testing, and final deployment. With regular reviews and user feedback, the project will be iteratively improved to deliver a high-quality portal that meets the needs of students, faculty, and administrators.

**References**

1. Zornada, L., & Velkavrh, T. (2005). Implementing ERP systems in higher education institutions. International Journal of Computing and Information Sciences, 3(1), 43-49.
   * This article discusses the challenges faced by educational institutions in managing legacy systems and implementing ERP systems, emphasizing the limitations of traditional SIS platforms.
2. Bhatti, T., Ali, A., & Mohammed, I. (2018). Cloud-based student information systems: An empirical analysis. Journal of Education and Learning Systems, 7(2), 102-115.
   * This study highlights the benefits of cloud-hosted student information systems in terms of scalability, remote access, and cost savings, which serve as a comparison point for Power Apps.
3. Ebert, C., & Duarte, C. (2018). Low-code development platforms: New era of software development. IEEE Software, 35(5), 31-36.
   * This paper reviews low-code platforms, including Power Apps, and discusses their potential to reduce development time and empower non-programmers to build applications in various sectors, including education.
4. Khatri, M., & Soni, A. (2020). A comprehensive study on Microsoft tools for managing educational data. International Journal of Computer Applications, 176(4), 15-21.
   * This study explores the use of Microsoft tools like Power BI, Power Automate, and SharePoint for educational data management, with a focus on the advantages of integration within the Microsoft ecosystem.
5. Alharbi, A., Hameed, T., & Khan, F. (2019). Mobile-first design in student portals: A usability evaluation. Journal of Educational Technology Systems, 48(2), 174-190.
   * This paper evaluates the importance of mobile-friendly designs for student information systems, with findings that are used to justify the mobile-optimized design of the Power Apps-based portal.

6. Niagara college student Portal: https://login.niagaracollegetoronto.ca/oauth/v2/auth\_login