An open cloud-based platform for the creation and delivery of smart applications and services

Heba M. Fadhil¹, Mohamed I. Younis²*, AbdulRahman A. Alsewari¹, Mohammed Abd-Alrazzaq¹, Ameer Toran Mahdi¹, Alaa A. Alomoush²

¹Information and Communication Department, University of Baghdad, Baghdad, Iraq
²Computer Engineering Department, University of Baghdad, Baghdad, Iraq
³Faculty of Computing, Universiti Malaysia Pahang, Pahang, Malaysia

*younismi@coeng.uobaghdad.edu.iq

Abstract. With the increasing integration of computers and smartphones into our daily lives, in addition to the numerous benefits it offers over traditional paper-based methods of conducting affairs, it has become necessary to incorporate one of the most essential facilities into this integration; namely: colleges. The traditional approach for conducting affairs in colleges is mostly paper-based, which only increases time and workload and is relatively decentralized. This project provides educational and management services for the university environment, targeting the staff, the student body, and the lecturers, on two of the most used platforms: smartphones and reliable web applications by cloud hosting. The services include project management, attendance marking, various notifications and alerts, files and resources, a grading system, and an assignment management system. This paper also aims to ultimately digitalize most of the information inside the college which is a great challenge. The proposed system defines five actors, namely: an admin; who is the person with the highest authority within the system, the head of the department, the department coordinator, the project coordinator, the lecturers, and the students. Each of those actors has its privileges and the interaction among participants via the cloud. The mobile platform targets the students exclusively and interacts with the web platform. Finally, this paper stresses the benefits of the proposed system and gives the avenues for future development.

Keywords. Smart Campus, Web Portal, Mobile Application, Cloud, IoT.

1. Introduction

Ever since computers and the World Wide Web have been introduced into organizations, they have offered exponential growth and realized possibilities which were thought to be extremely difficult before. Management systems have especially benefited from this. The traditional paper-based methods consume too much time; reports need to be drafted and delivered by hand, meetings need everyone’s physical presence, information and news need to be distributed manually, and simple operations on data need much time to go through all of that paperwork [1].

Online management systems provide an environment in which all of the above-mentioned work is reduced to simple operations, and make some of the operations completely automated, with the bonus of that they can be done anywhere as long as the person is connected to the network [2].
Colleges are organizations which can especially benefit from these advantages, especially in Iraq. The amount of paperwork and waiting involved in doing even the simplest of tasks is immense. Moreover, there is relatively poor communication between the faculty and the students which suffer greatly from human error. Furthermore, the existing members are reluctant to adapt to more technologically advanced methods, and still prefer the traditional one, partly because most existing systems are never implemented completely and fail to address the needs of the faculties and students directly, or are too complicated for them to use [3].

Various College Management Systems have existed since the ‘90s; each providing a customized solution for the college it is used in. In existing systems, the majority of the college, maintaining the records and other information is the annual process. Taking an existing system keen on deliberation, concise on the necessities they anticipate and so on. All of these require more work and more time [4]. The project focuses mainly on colleges and the synchronization of all the sparse and diverse information on the regular college schedule. Usually, students face problems getting the correct notifications at the right time, sometimes important notifications such as campus interviews, training and placement events, holidays and special announcements. Smart Campus attempts to mind the gap between students, teachers, and college administrators. Therefore, in the environment, like a college campus, the information (e.g., oral communication, notices, files) can be reached for communication and made available for the participants through the android devices. The use of architectural Model-View-Controller (MVC) makes both the development maintenance and corrective maintenance easier because of the separation of concerns in the development of an application (e.g., data management, GUI) from and web service which requires fast and efficient maintenance of the application.

Implementing a cloud-based smart campus system, which is tailored for the needs of the currently existing faculty and students, is a major step towards solving the above-mentioned issues and guaranteeing a future that is committed to technology and its benefits.

The objective of this paper is to provide a central database for all of the information within the campus, and provide a fast and reliable communication method between the student body and the faculty using smartphones, both Android and IOS, and the web portal, with the automation of some of the tasks within the campus.

2. Literature Survey

The smart campus is an emerging field of study which attracts researcher. Many related works try to design and implement smart campus system from different points of view such as infrastructure, services, and usability.

Moreno et al [5] provided a series of services implemented on intelligent campus for energy efficiency. Services implemented include internal settlement assessment, energy consumption forecasting, convenience, and energy conservation. Arshad et al [6] designed a mixed label system to label the contents and devices of the campus environment to become smart internet-based objects. In a plan developed, features contain scalability and security for campus contents and Internet objects. Xiang et al [7] proposed a campus information dissemination target to university. A model of intelligent campus been proposed by Atif and Mathew [8], which integrate learning resources in the real world within a social network on the campus level. Liu et al [9] introduced a smart digital campus combining both internet of things and cloud computing technologies and illustrated the complications that can be detected after the creation of intelligence application platforms. Younis et al [10] proposed a secure graduation certificates issuing system (GCIS). They attempt to provide the graduation certificates in a secure server based on digital signature. However, the proposed system is used inside the examination committee of a department and do not scale with college and university. They stress the need of reliable web portal through a cloud hosting.

3. College Software

The college software is a cloud-based and an enterprise resource planning (ERP) software which provides an online solution for computerization of administrative activities of a college campus. This
software handles academic and non-academic activities. The education landscape is facing continuously changes for academic and non-academic sectors as a result of digitalization era. To facilitate this, institutions must shift the gear toward the state-of-the-art innovations and technologies. The software enables colleges to modernize the education process easier and provide a unified framework where learning and teaching can thrive.

Cloud-based college ERP provides a total online solution for computerization of all administrative activities, which includes admission, student administration, attendance, grading, etc. This software has many modules to execute multiple operations on the college campus. Each module can be tailored according to the requirements of the college.

Following are the benefits of college management system software:

- reduces manual workload
- increased productivity
- effective communication between teachers & students
- reliable backup & recovery system
- minimal data redundancy
- cost-effective

The college management software is a vital requirement to help participants to reach effective management for complex operations. While implementing an ERP it's very important that it should fulfill institute requirement and generate expected results. Implement a well suitable, flexible and innovative education ERP. The system proposed in this project explores the following tasks:

1. Project management: This includes the process of assigning teams to the projects and managing the project tasks until completion.
2. Attendance marking: This includes marking the student's attendance performance and sending alerts when required.
3. News and notifications: This includes the college newsletter, mentioning important events within the campus and sending notifications when needed.
4. Files and resources: This includes the various textbooks and study materials that the students need.
5. Grading system: This includes grading the students until the end of the course and delivering their grades on time.
6. Assignment management system: This includes placing assignments for the students and getting their answers for grading.

3.1. MVC

The MVC is an architectural pattern that provides the separation of concerns principals in developing a GUI application. In MVC design an application consists of three components: the model, the view, and the controller as seen in figure.1 MVC is an industrial-standard framework that enables the developers to provide extensible and expandable products [11].

![Figure 1. MVC components interaction.](image-url)
3.1.1. Model
The Model represents the logical way that the user interacts with the data. The Model is responsible for the data movement between the View and Controller components. For instance, a Customer object retrieves information from a table in a database, manipulates or renders it for displaying or updates the records.

3.1.2. Controller
Controllers process the interaction logic between the Model and the View components. For instance, a Customer controller takes the inputs from the Customer View and updates the database through the Customer model. Alternatively, the controller is re-used to view the Customer data. As such, the controller provides two-way interfaces.

3.1.3. View
The View component includes all UI components that are physically the end-user interacts with them. Examples include radio-buttons, drop-down menus, text areas, etc.

3.2. Technologies and Frameworks
The project's web portal portion was developed using Laravel for its backend and Vuejs for its frontend. The mobile application portion was developed using React Native. Some features required a Nodejs server.

3.2.1. React Native
React Native is an open-source tool released by Facebook community in 2015 [12]. The React Native is widely used by the developers for developing mobile applications target to iOS and Android. It uses the same declarative approach to constructing user interfaces as React for the web. React Native also aims to reduce many native development inefficiencies. Rather than deal with the slow process of write, compile, deploy and debug, which can cripple development on larger native apps, React Native allows you to simply refresh the app after making a change without the slow compile and deploy steps, just like on the Web! This makes for a much-improved developer experience and, unlike normal native development, React Native allows you to share far more code across platforms. However, Facebook points out that React Native is not intended to be a write once, run anywhere solution. They acknowledge that each platform has its own look, feel, and capabilities. Instead, React Native allows you to leverage common technologies to build across multiple platforms. They call this learn once, write anywhere [12].

A React Native application runs the low-level mobile platform code through a JavaScript “bridge” layer to interact with the phone for accessing all its functionality and presenting a UI. There is a JavaScript engine interface that exists on both iOS and Android. This is a low-level layer that then interacts with the OS for the phone to do the same things that other platform languages on the phones do.

3.2.2. Vuejs
Vue is a popular library with rich features that has easy adoption with official extension libraries. The core library of Vue (e.g., React) manipulates the view layer. On the other hand, the extension libraries like Vuex and Router, responsible for data management and routing respectively [13] [14] [15]. The key design principles of Vue are as follows:

The key design principles of Vue are as follows:
- Focus: Vue has opted for a small, focused API, and its sole purpose is the creation of UIs
- Simplicity: Vue's syntax is terse and easy to follow
• Compactness: The core library script is ~25 KB minified, making it smaller than React and even jQuery
• Speed: Rendering benchmarks beat many of the main frameworks, including React
• Versatility: Vue works well for small jobs where you might normally use jQuery, but can scale up as a legitimate SPA solution.

3.2.3. **Node.js**
Node is a run-time environment for JavaScript with platform-independent and open-source functionalities. Node.js is widely used in programming command-line tools outside a browser, unifying both the client-side as well as server-side into the industrial standard JavaScript programming. Besides, it is used for deploying networked services. JavaScript is dominant the web programming, when combined with the Node’s simplicity, facilitate the integration of the on-the-shelf third-party modules [14].

3.2.4. **Laravel**
Laravel is a PHP open-source framework based on MVC. It is used for building robust web applications with elegant syntax and rich features. The current version of Laravel is 5.5. Examples of famous web applications and services written by Laravel involve, but not limited, customer portal, image cropper, social network, and RESTful APIs [13] [16].

3.2.5. **Socket.io**
Socket.IO is a JavaScript package for real-time enables the developers to write bi-directional streams to handle the communication between client-sides and server-sides [17].

3.2.6. **SQL**
A database is a set of structural tables that can be accessed or updated without knowing the details of the tables, by using industrial standard application programming interface (API). Structured Query Language (SQL) is dominant query language in a relational database development [18]. SQL statements enable the developers to write queries for gathering data and reports generation.

3.3. **Design Methodology**
Throwaway Prototyping Model is a powerful framework to identify the requirements for a project when the needs are poor and ambiguous. Prototyping is considered as proof by doing. The end-user enters a loop of satisfaction to complete the project (i.e., Close-Ended Prototyping as shown in figure 2). Thus, Throwaway Prototyping Model methodology is adopted for the following reasons:
1. Users participate in the development cycle.
2. The users get experience about the system under the development.
3. Bugs can be detected at the early stages.
4. Faster feedback from users to assist in better developments.
5. Missing features will be identified smoothly.
6. Confusing can be resolved
7. Difficult functions can be refined through the step-wise-refinement process.
8. Provides fast implementation for an incomplete working application.
9. Simplify Verification and Validation (V&V) for software testing.
4. Implementation and Results
The following subsection discuss the system design, platforms, services and describe the mobile applications.

4.1 System Design
In the proposed system, the students can access both the web portal and the mobile application. The Admin, as well as other faculties, can access the web portal only. The participants need to log in to use the applications. The applications have privileged to access the database. The database is implemented in the cloud and shared through the internet as shown in figure 3. Various user privileges’ types are enlightened as seen in figure 4.

1. **Student**: After the students’ login, they can access various services and applications like: voting, read the notice, read the blogs published by the staff, read the manuals, etc.
2. **Lecturer**: After the lecturers’ login, they can write the post as well as notes on the blog, view blogs, making an election for voting, upload grades, and absents of students, resources, and other services.
3. **Department head**: the department head manages all department options; he can add courses, subjects and assign them to lecturers, add students and manage their accounts.
4. **Admin**: The admin can manage all departments of the college in every detail. In addition, he/she has permission to add the department head and all departments’ accounts and control their management.
5. **Projects committee**: it a group of lecturers has permissions to assign projects to students after they get votes of the project from students. In addition, the department head can create these groups.

4.2 System Platforms
- **Mobile Application**: Only students use mobile applications. There are many applications for student involves projects tasks, assignments, resources, attendance, and grades.
- **Web Portal**: All actors can access to the web portal. The services inside the web portal are depending on the privilege of the actors as shown in figure 5.
4.3 System Services

4.3.1 Projects Management

This module manages projects of all stages and departments of the college, so department head can upload the projects and assign lecturers for them firstly, then students can vote on projects with a partner for each one. After that, the projects committee can give students their projects based on their votes as shown in figure 6. So each project has its supervisors and student, supervisors can upload tasks of projects to the student, when students complete their tasks return feedback to the supervisor, see figure 7.

4.3.2 Grades

Lecturer can upload marks of exams and student get their marks directly, when all exams have completed, lecturer can calculate the midterm mark automatically by clicking on "publish midterms" button, an algorithm will run to collect all exams and calculate then midterm grade, after that the Department head get a notification to approve the midterms of the stage students as shown in figures 8 and 9.

4.3.3 Attendance

Lecturer uploads the hours of absents of his subject for each student, and then alerts will be calculated automatically depending on hours of the subject see figures 10 and 11.

4.3.4 Files and Resources

Lecturers and staff upload files on cloud and student can download them. The files include the lectures, sheets, and books of subjects as shown in figure 12.

4.3.5 Assignments

Students upload homework's and assignments as pdf or images for lecturers, each lecturer can see only his assignments as shown in figure 1).

4.3.6 Blogs and news:

The staff from lecturers and department heads even admins can post news or events. Admin can post for one department or all and one stage or all stages because he has full permissions for post, department head and lecturer can post for only his department to one stage or all stages, Students just see the posts for their departments, so they cannot see the posts for other departments as shown in figure 14.
4.4 Mobile Application
The cross-platform mobile application was developed using React Native and relies on Expo Notifications for notification services. It consists of a login page and 5 swappable tabs, as shown below in figure 15.

4.4.1 Login
The student provides his email and password to log in. There is also a link to the web portal where the user can register a new account. The users can choose whether to stay logged-in the next time they use the app. In case the users forget their respective passwords, they are redirected to the web portal to acquire a new password via email see figure 16.

4.4.2 Login Main page
The main page is shown in figure 17 consists of a search function and a button to the notifications list in the header, and the five mentioned tabs.

4.4.3 Study
The study contains four basic services:
1. Grades: The users will receive their grades of a subject once they are uploaded to the web portal by the lecturer, along with a notification.
2. Assignments: Once an assignment is assigned to the students via the web portal, they will receive a notification. The application will show the description of the assignment along with the deadline. The students can choose to submit their answer from the application, which is only possible before the assigned deadline as shown in figure 18.
3. Project Tasks: The students will receive the tasks assigned by the project supervisor with its description, and have the option to submit a prompt when the task is completed as shown in figure 19.
4. Resources: The resources show the files (Textbooks, PDFs, Images) of the subjects of the course the student is currently enrolled in, which can be downloaded.

4.4.4 Profile
The profile tab shows the profile of the student including his information as shown in figure 20.

4.4.5 Bus Order
The bus order system connects to Socket.io running on a Node.js server. Each student can submit a single order when he or she wants and the order increments the total number of orders as shown in figure 21. The transportation department then can inform the students that a bus is on the way, resetting the number of orders and still keeping the students from ordering. After the bus makes a lap around the campus, the transportation department cancels, letting the students order again as shown in figure 22.
Figure 5. Web portal homepage.

Figure 6. Projects management.

Figure 7. A project with tasks, each task has deadline.

Figure 8. Grades management.

Figure 9. Grades of students with the approval of admin on it.

Figure 10. Attendance management.
Figure 11. Attendance of students with the approval of admin on it.

Figure 12. Files upload from lecturer to students.

Figure 13. Assignments module.

Figure 14. Blog and news module.

Figure 15. Mobile application pages.

Figure 16. Login page.
Figure 17. The main page.

Figure 18. Uploading an assignment.

Figure 19. Project tasks.

Figure 20. Profile.
5. Conclusions

This paper has been discussed the development of an open, cross, cloud-based platform for the creation and delivery of smart applications and services. The proposed system has the following advantages when compared to the literature review. First, it provides cross-platform functionality and follows the industrial design strictly. Second, the proposed reduces the time and physical resources required to reach information and thus, facilitates the working beyond both place and time barriers. Third, the proposed system provides an easy way to maintain the database records based on the reliability of the cloud, as a result, the system is considered a reliable system. Fourth, provides an easy interaction among the participants. Besides, the system is open, that can be extended horizontally by providing other services and applications and can be expanded vertically by adding hierarchies hosting. Finally, and more to the points, all participants improve their familiarities with the institution roles significantly and the quality of the institution will be improved as a result. From this perspective, there are multiple avenues for the future-works. One avenue is to add healthcare services and applications. Another avenue is to add payment and security-aware applications. Finally, the database could be available for machine learning researchers to perform statistics and machine learning methodologies to improve the skills and the rank of the University.

Acknowledgments

This research is funded by, UMP (RDU190334): A Novel Hybrid Harmony Search Algorithm with Nomadic People Optimizer Algorithm for Global Optimization and Feature Selection, and (FRGS/1/2018/ICT05/UMP/02/1) (RDU190102): A Novel Hybrid Kidney-Inspired Algorithm for Global Optimization Enhance Kidney Algorithm for IoT Combinatorial Testing Problem.

References

[1] Magno-Tan, M.J., Crisostomo, A.V., Villaflor, B. and Faller, J.C. “Cloud-Based College Management Information System for Universities”. International Journal of Information and Education Technology, 4(6), p.508, 2014.
[2] Bilawane, Shilpa, Jambhulkar, Pranali and Naidu, Kokila “Information System Based On College Campus”. *International Journal of Engineering and Computer Science*, 4, pp. 2319-7242, 2015.

[3] Rajesh Shah, Makhan Kumbhkar, “Cloud-Based College Management Information System for Autonomous Institute” *International Journal of Application or Innovation in Engineering & Management (IJAEM)*, 5 (5), May 2015.

[4] Sagar Rajebhosale, Shashank Choudhari, Sachin Patil, Akshay Vyavahare, Sanket Khabiya, “SMARTCAMPUS – An Academic Web Portal with Android Application”, International Research Journal of Engineering and Technology, 3 (4), 2016.

[5] M. V. Moreno et al.” Applicability of Big Data Techniques to Smart Cities Deployments”, IEEE Transactions on Industrial Informatics, 13(2):800809, Apr. 2017.

[6] S. Arshad, M. A. Azam, S. H. Ahmed, and J. Loo,” Towards Information Centric Networking (ICN) Naming for Internet of Things (IoT): The Case of Smart Campus”. International Conference on Future Networks and Distributed Systems (ICFNDS), July 2017.

[7] Y. Xiang , D. Chang ,B. Chen ,” A smart university campus information dissemination framework based on wechat platform”. *In: LISS*, vol 2013, pp 927–932,2015.

[8] Y. Atif, S. Mathew and A. Lakas, "Building a smart campus to support ubiquitous learning", *Journal of Ambient Intelligence and Humanized Computing*, 6 (2), pp. 223-238, 2014.

[9] Y. Liu, W. Zhang and P. Dong, "Research on the Construction of Smart Campus Based on the Internet of Things and Cloud Computing", *Applied Mechanics and Materials*, pp. 3213-3217, 2014.

[10] Mohammed Issam Younis, Hayder Faez Abdulkareem, Hamid Mohammed Ali, “Construction of Graduation Certificate Issuing System Based on Digital Signature Technique”. *Journal of Engineering*, 21(6), pp. 15-36, 2015.

[11] MVC hierarchy, https://www.tutorialspoint.com/mvc_framework.

[12] Eric Masiello, Jacob Fricmann, Mastering React Native, Packt, 2017.

[13] Anthony Gore, Full-Stack Vue.js 2 and Laravel 5, Packt Publishing, 2017.

[14] Ethan Brown, Web Development with Node and Express, Orielly Publishing, 2014.

[15] VueJS Framework Documentation https://vuejs.org/v2/guide/.

[16] Laravel Framework Documentation https://laravel.com/docs/5.7.

[17] Mohammed Issam Yonis, Taif Fadhl Hussein, “Design and Implementation of a Contactless Smart House Network System”. *International Journal of Electrical and Computer Engineering* 8(6), pp. 4663-4672, 2018.

[18] Aniqa Rehman, Saba Latif, Nazir Ahmad Zafar, "Non-Deterministic Formal Modeling of Registration System towards Smart Campus", 12th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS), pp. 1-6, 2018.