



Mobile Class Attendance System (MobCAS)

Nurul Hafizah Zainal Abidin

Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Perak Branch Tapah Campus, Malaysia

nurul354@uitm.edu.my

Muhammad Ridhwan Muhamidon

Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Perak Branch Tapah Campus, Malaysia

2012142547@isiswa.uitm.edu.my

Bushra Abdul Halim

Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA Melaka Branch, Alor Gajah Campus, Malaysia

bushra270@uitm.edu.my

Roslah Arsad

Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Perak Branch Tapah Campus, Malaysia

rosla280@uitm.edu.my

Article Info

Article history:

Received June 12, 2021

Revised Sept 15, 2021

Accepted Oct 15, 2021

Keywords:

Mobile Application
Student Attendance
University Students
phpMyAdmin
MySQL Database

ABSTRACT

People nowadays choose to use mobile applications in their daily tasks. Due to COVID19 pandemic, students and lecturers faced a new challenge in learning and teaching, which have to be conducted online. One critical issue in online and distance learning is to monitor the students attendance. Google meets has been widely used but few problems emerged such as limited in notifying absenteeism. Therefore, this study aims to develop a smartphone application called MobCAS that can be used to record student attendance and to calculate the percentage of attendance automatically for reporting the absenteeism. This mobile application was created on the Android platform and communicates with a webserver using phpMyAdmin as the database platform. This paper provides the fundamental design and implementation of MobCAS. The results from users survey indicated that the system is easy to operate, effective and acceptable to use in the future.

Corresponding Author:

Nurul Hafizah Zainal Abidin,
Faculty of Computer and Mathematical Sciences,
Universiti Teknologi MARA, Perak Branch, Tapah Campus
Perak, Malaysia
Email: nurul354@uitm.edu.my

1. Introduction

Educators encountered new challenges when classes were transformed from face-to-face into online and distance learning. One of the challenges is to ensure that all students to attend the class on the schedule time. The best approach to get full attendance in class is by using a mechanism that can notify the student absenteeism. In most of universities in Malaysia, monitoring student absenteeism is a compulsory to avoid the students being barred of their final examination.

Lecturers can utilize many software tools to support the online teaching and learning implementation. As for examples, many of them are commonly used Google Classroom[1], Google Meet[2] and Microsoft Team[3]. Nevertheless all of these tools are merely depending on email account without any connection to the registered student database. It created a difficulty for the



lecturer to manage the students attendance when the username of the email is different with the full name of the student. Although an online learning management system[4] is provided by the university to be used by the lecturers and students, this system is commonly abandoned with a lot of system functions that mostly have a performance and low connection issues[5]. Therefore, a very light weight system that particularly customized for the university requirement of attendance management system such as a mobile attendance system is useful for the lecturers and students.

This paper provides the report of study that focused on the design and development of a class attendance system based on mobile application technology that can be deployed in an Android device. Named as MobCAS, the system was customly designed in accordance to the Universiti Teknologi MARA(UiTM) rule regarding to the final examination barred if the students were not attending at least 80% of the classes for each subject.

There are two main objectives of this study, which are to to develop a mobile application for students class attendance and to calculate the percentage of student absenteeism in the mobile application. The proposed design and development introduced in this paper is useful to be replicated by the inexpert system developers who intend to produce a mobile application mainly related with class attendance and management.

2. Literature Review

Development of mobile application (mobile apps) technologies is a significant project due to the high demanding needs. Mobile apps typically available through several application distribution platforms or application store and operated by the owner of the mobile operating system such as App Store distributed by iTunes, Google Play distributed by Google and BlackBerry App World distributed by BlackBerry. A mobile platform or mobile operating system (OS) is a set of programs that runs on a computer or mobile devices. Mobile OS are seen in smartphones powered by the Apple's iOS, Google's Android, BlackBerry OS, Microsoft's Windows Phone, and many more[6]. IOS was developed by Apple Inc and this operating system distributed exclusively for Apple hardware such as iPhone, iPod and iPad. Apple products gives a good feeling and exclusivity and not become sluggish in performance even after installing apps but most of the mobile apps for IOS are not free. Meanwhile, Windows Mobile OS developed by Microsoft Corporation used for PCs, mobile devices, and Xbox. The advantages of Windows Mobile OS is its convenience to be used by many type of users. Blackberry OS is an operating system developed by BlackBerry Ltd and only support for Blackberry devices. The advantages of Blackberry are easy typing because most blackberry still use QWERTY keyboard, very fast and snappy. However, Blackberry has major problem with the battery life, application support and associated with complex development platform for mobile applications[7]. Application development on the Android OS, IOS and blackberry requires different approaches, programming languages and has different methods of app publication[8]. However, the most flexible approaches and widely acceptances by mobile apps users is Android OS[9].

Recently, a variation of attendance system have been introduced with different kinds of attractive features and technologies. Researchers in [10], presents the design and architecture of mobile application for student attendance that was integrated with mark management system. The system need to be integrated with marking management as the attendance percentages need to be calculated with the students final grading. Different in the mobile attendance system proposed by [11], the researchers incorporated image recognition technology to automate the attendance records by recognizing the students' face. Similarly, based on image processing technology, bio-metric fingerprint authentication for Android based attendance vigilance system has been developed in [12]. As to get parents involvement in their child's school attendance, mobile application with SMS gateway has been developed by researchers in [13]. Furthermore, by using a combination of QR code and GPS technology, a location-aware of event attendance system has been implemented for speeding up the process of taking students' attendances in the Universiti Teknikal Malaysia (UTEM)[14]. Firebase database in cloud was used to manage the attendance information of this system. There are several database platforms suitable for mobile apps can be used other than Firebase such as SQLite, Oracle Database Lite and MySQL. Firebase is more powerful for updating data in realtime but MySQL is more flexible to works on many programming languages including PHP, PERL, C, C++ and JAVA as well as useful and efficient to handle a large databases[15].

3. Methodology

The research process is separated into four phases: data collection, design and development, implementation, and evaluation. The first phase was to collect the data set of degree students in UiTM Tapah, detail of classes and lectures.

The second phase involved two important tasks namely the database design and the application development. MySQL Database has been used as the database platform as it is compatible with Android Studio for the mobile apps platform. Next development was the mobile application interfaces. A function to calculate the percentage of student absenteeism has been incorporated in the application.

After completing the application development, three important tasks were implementation and functional test. The first testing was to ensure that the function to record the students' attendance is correctly stored in the mobile application and the second testing is verify the correctness of the percentage of student absenteeism. Figure 1 presents the flowchart of the MobCAS that show each important process.

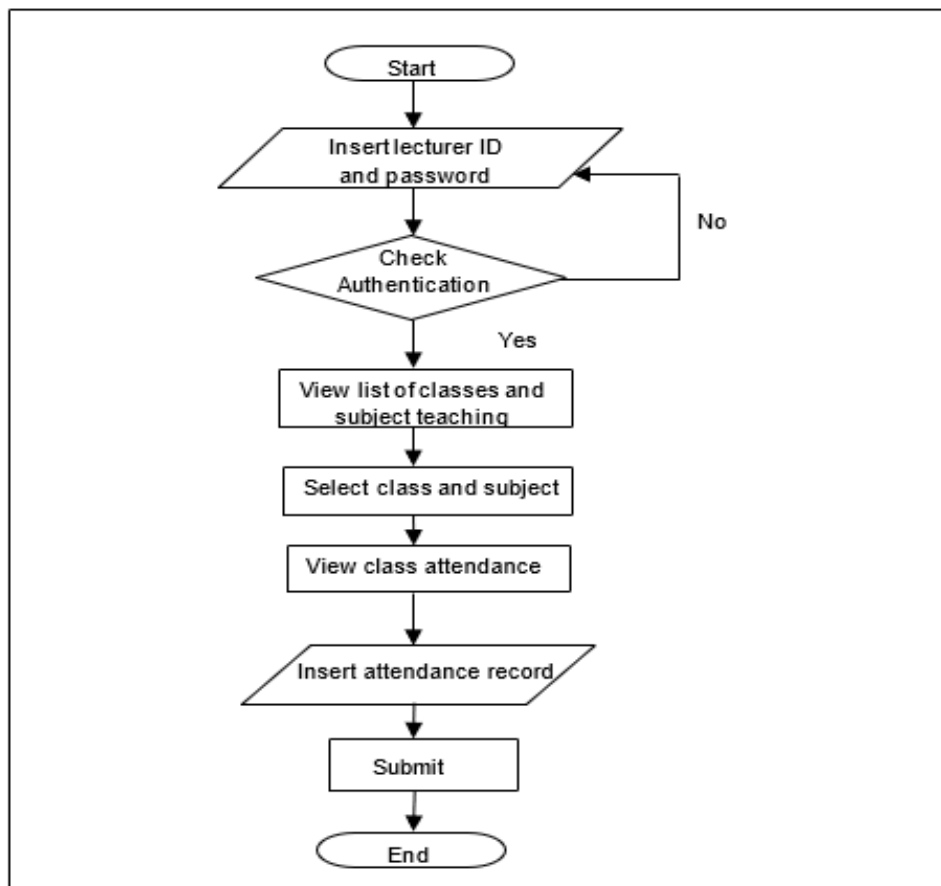


Figure 1. The MOBCAS flowchart

To use this system, users need to log in their lecturer ID and password for authentication. If the authentication process failed, the system is unable to open the next interface. Then, next interface will appear to allow users to select class group and subject. The users can turn-in their attendance for the allocation time slot and view the attendance list. At the same time, the system will calculate the percentages of absenteeism, which also can be viewed by the users. Figure 2 shows the proses of viewing absenteeism results.

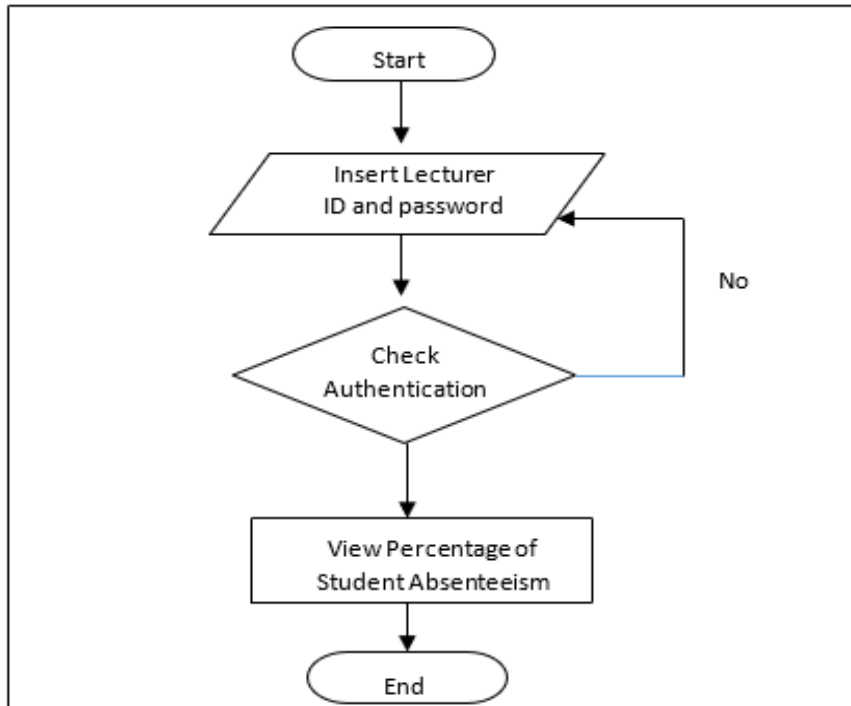


Figure 2. Percentage of Students' Flow Chart

The MobCAS system was evaluated by conducting a survey on users who were degree students and a few of lecturers as to measure the system usability and to verify the accuracy of absenteeism function.

3.1 The Development

This part describes the application development based on two development phase, namely the development of database and development of the mobile application system.

3.2 The Database

The database structure of MobCAS is shown in Figure 3, developed by using phpMyAdmin. It has seven tables that connecting *student*, *course*, *lecturer*, *student_register*, *groups*, *attendance*, *present_absent* and *timelist*. A student can have more than one registration of subject and more than one time of absent. A lecturer can have create different attendances for the specific subject along a whole semester. The attendance table is a bridge between *student*, *group*, *course*, *timelist* and *present_absent* tables. Lecturers are separated into two categories: system administrators, who have full access to the system, and lecturers who have limited access to the project. Examples of data when stored in the database for *student* and *lecturer* tables can be depicted in Figure 4 and Figure 5.

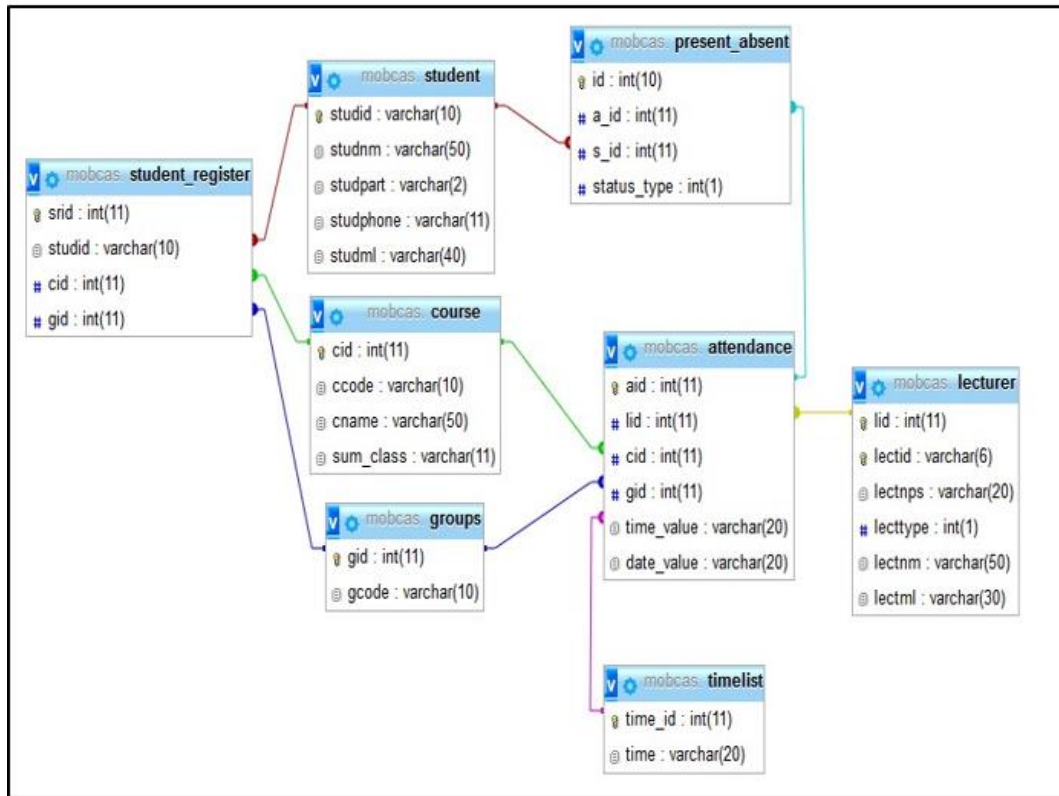


Figure 3. Database structure of MobCAS

studid	studnm	studpart	studphone	studml
2012142547	MUHAMMAD RIDHWAN BIN MUHAMIDON	06	0134764815	ridhwanr91@gmail.com
2012170679	MAHFUZAH BINTI RAMLAN	06	0124551251	fuzahramlan@gmail.com
2012508915	NOOR SAIDATUL MUNIRAH BINTI SAAIDUN	06	0134080278	saidatulmunirah02@yahoo.com
2012560845	NUR MUHAMAD FIRDAUS BIN ABDULLAH	06	0125150411	dukeon92@gmail.com
2012770601	NOOR KAMILAH BINTI MOHAMED AZHAR	06	0194471029	noorkamilahazhar@gmail.com
2012905665	SITI NAZEHA BINTI JAAFAR	06	0126909967	nazehajaafar@gmail.com
2012953573	MUHAMMAD FARIDUDDIN BIN MOHD YASIN	06	0134848665	fariduddin.my@gmail.com
2012989317	NUR MAISARAH HARUN	06	0129893172	bobybo_yankee@yahoo.com
2012989537	MOHD FAZRIN B CHEK WIRAMAN	06	0134162199	mohdfazrinwiraman@gmail.com

Figure 4. Example of data in *student* table

lid	lectid	lectnps	lecttype	lectnm	lectml
1	123456	mobcas	1	MASURAH BINTI MOHAMED	masura@perak.uitm.edu.my
2	234567	1234	2	SURAYA BINTI MASROM	suraya23@perak.uitm.edu.my

Figure 5. Example of data in *lecturer* table

The *course* Table store information about all subjects registered taken by the degree students in UiTM Tapah. This table consists course code and course name as shown in Figure 6. The *timelist* table lists all of the available learning durations based on the learning schedule as shown in Figure 7.

cid	ccode	cname
1	CSP650	PROJECT
2	ENT600	TECHNOLOGY ENTREPRENEURSHIP
3	CSC662	COMPUTER SECURITY
4	ITS610	ETHICAL, SOCIAL AND PROFESSIONAL ISSUES
5	PHI454	HISTORY AND PHILOSOPHY OF SCIENCE
6	ELC550	ENGLISH FOR ACADEMIC WRITING
7	CSC580	PARALLEL PROCESSING
8	CSC577	SOFTWARE ENGINEERING: THEORY AND PRINCIPLES
9	CSC569	PRINCIPLES OF COMPILERS
10	CSC563	PROGRAMMING PARADIGMS
11	TMC451	MANDARIN II

Figure 6. Example of data in *course* table

time_id	time
1	8:00 - 8:50
2	9:00 - 9:50
3	10:00 - 10:50
4	11:00 - 11:50
5	12:00 - 12:50
6	2:15 - 3:05
7	3:15 - 4:05
8	4:15 - 5:05
9	5:15 - 6:05
10	6:15 - 7:00

Figure 7. Example of data in *timelist* table

The *student_register* table (refer Figure 8) stores information about students, courses, and groups. Each student can enrolled more than one subjects, each with its own group or class. The table provides the information required by the standard attendance form, such as the date and time, lecturer information, and information on absent or present students. The list of pupils, however, cannot be displayed in this table due to the (many to many) relationship. A bridge table was created to address this issue. The table of *present_absent* as in Figure 9 used to store information about students that are either absent (2) or present(1).

srid	studid	cid	gid
1	2012142547	1	2
2	2012142547	2	2
3	2012142547	3	2
4	2012142547	4	2
5	2012142547	5	2
6	2012560845	1	2
7	2012560845	2	2
8	2012560845	3	2
9	2012560845	4	2
10	2012560845	5	2
11	2012770601	1	2

Figure 8. Example of data in *student_register* table

id	a_id	s_id	status_type
54	62	2012170679	1
55	62	2012170679	2
56	64	2012170679	2
57	66	2012170679	1
58	66	2012170679	1
59	64	2012170679	2
60	66	2012170679	1
61	66	2012989537	1
62	67	2012170679	1
63	67	2012989537	1

Figure 9. Example of data in *present_absent* table

3.3 The Interfaces

The main interfaces of MobCAS application are depicted in Figure 10. If the device is not connected to an internet, a pop-up message will appear to inform the message error. Users must enter their staff id and password to be verified before the next page is displayed, as shown in Figure 11. The administrator had to create a unique ID and password for the lecturers. The names and email addresses of the lecturers will be displayed on the next windows after the login is successfully verified with the correct username and password. Two buttons can be selected according to its function either fill attendance or view percentage of absenteeism.

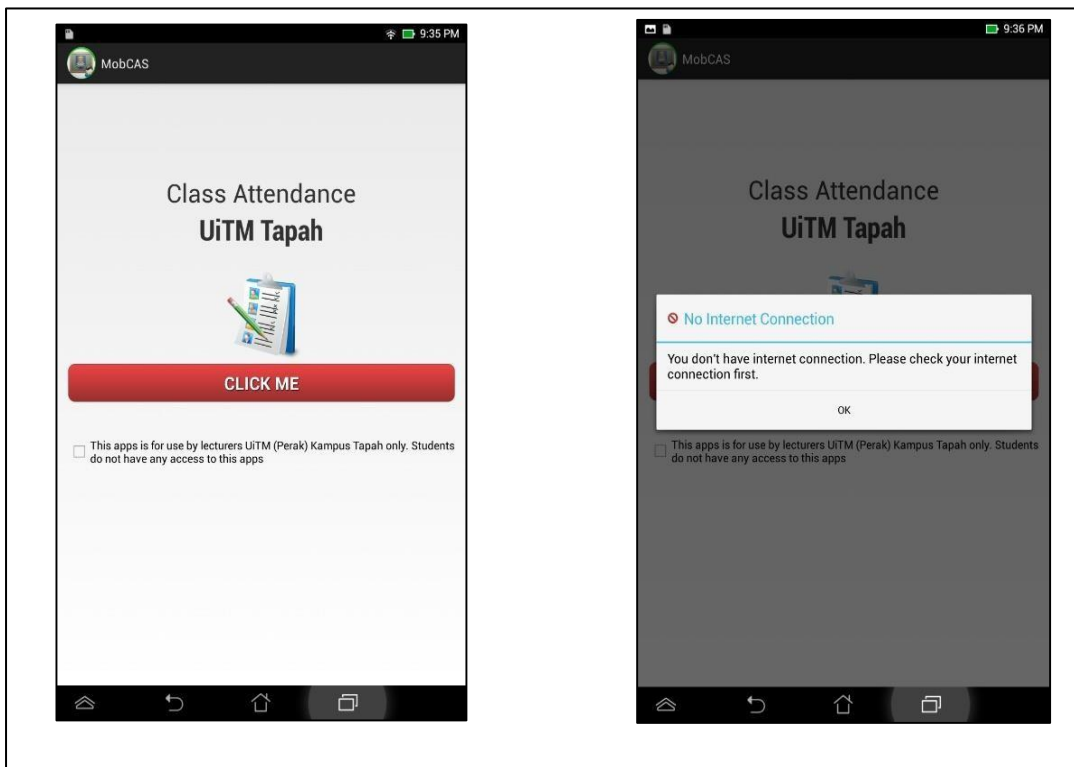


Figure 10. MobCAS main windows

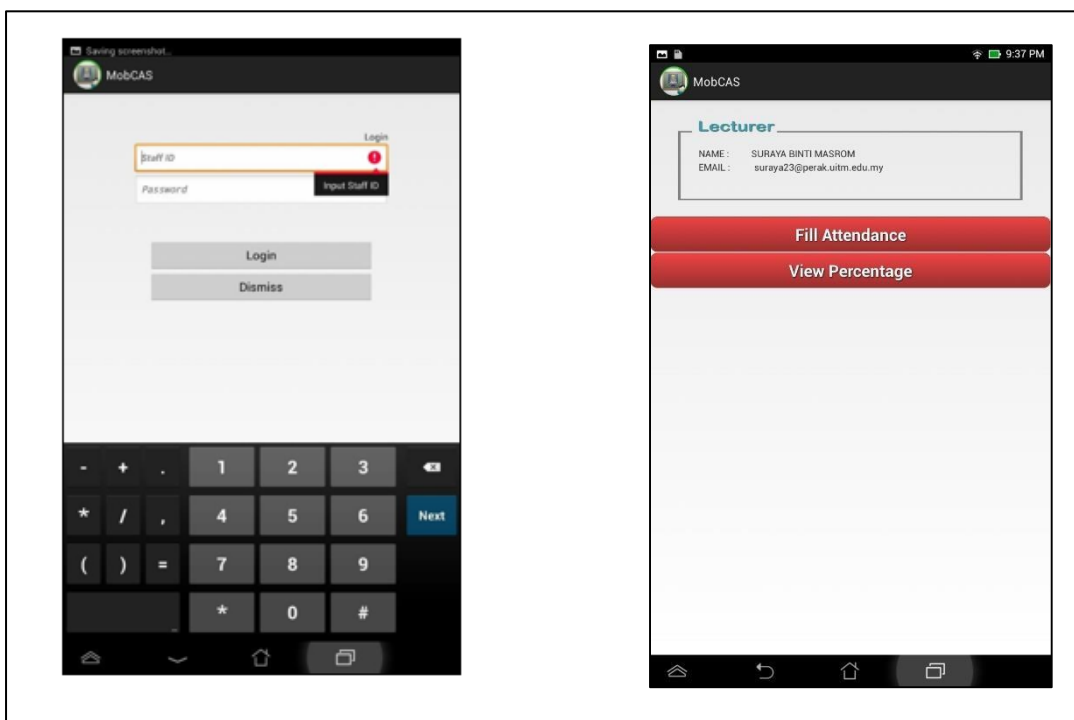


Figure 11. Lecturer windows

If the lecturer chooses Fill Attendance, they need to select the course and group as shown in Figure 12. In order to alter a student's status (present/absent) in the class, the lecturer must insert a probable time, and click one by one until the end of the list of students. Before the next page to be,

the information regarding attendance, such as date, time, and lecturer ID will be stored in the database, indicating that the data for the class had been generated. Meanwhile, if View Percentage was chose, users need to select the group and course. A list of students with their percentage of attendance based on subject can be viewed as shown in Figure 13. After 14 weeks of study, if a student's attendance does not surpass 80%, it is regarded unacceptable, and the student can be warned.

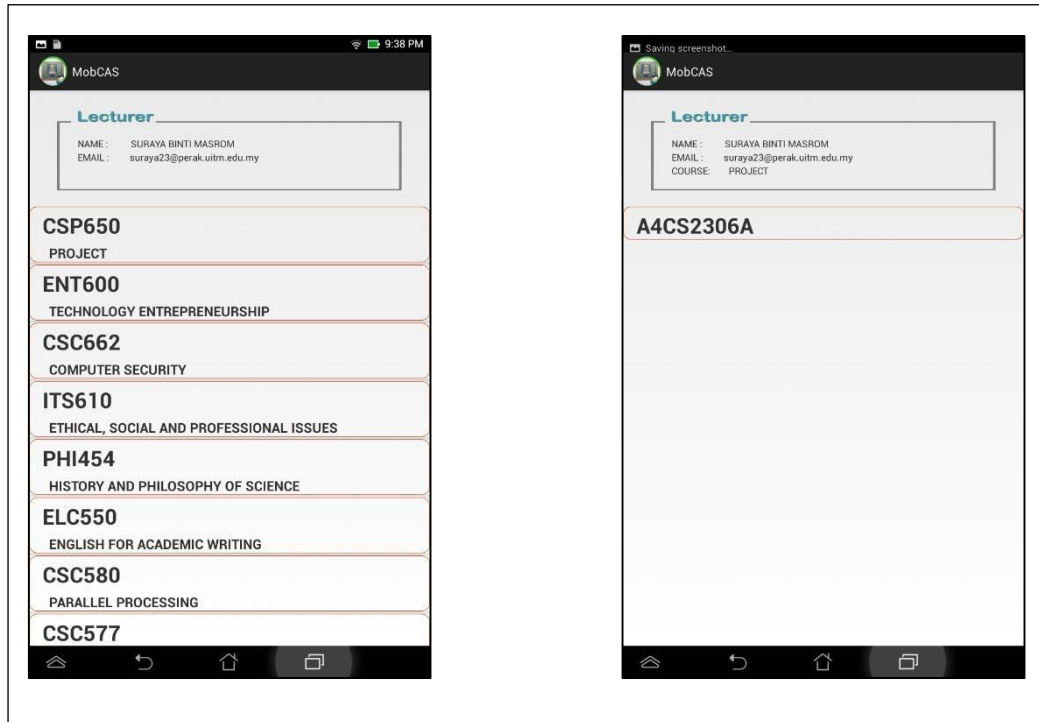


Figure 12. Select course and group windows

2012170679	MAHFUZAH BINTI RAMLAN	56%
part : 06	0124551251	
2012989537	MOHD FAZRIN B CHEK WIRAMAN	33%
part : 06	0134162199	
2012953573	MUHAMMAD FARIDUDDIN BIN MOHD YASIN	11%
part : 06	0134848665	
2012142547	MUHAMMAD RIDHWAN BIN MUHAMIDON	0%
part : 06	0134764815	
2012770601	NOOR KAMILAH BINTI MOHAMED AZHAR	0%
part : 06	0194471029	
2012508915	NOOR SAIDATUL MUNIRAH BINTI SAAIDUN	11%
part : 06	0134080278	
2012989317	NUR MAISARAH HARUN	11%
part : 06	0129893172	
2012560845	NUR MUHAMAD FIRDAUS BIN ABDULLAH	11%
part : 06	0125150411	
2012905665	SITI NAZEHA BINTI JAAFAR	11%
part : 06	0126909967	

Figure 13. List Name of Student and Their Percentage

3.4 Connection between Database and MobCAS

A connection between the external database and MobCAS is important to ensure that the system can be deployed. Therefore JSON (JavaScript Object Notation) parser has been used to support the communication between the PHP and JAVA programs. Figure 14 depicts a segment of the JAVA codes in an Android application.

```
/**
 * verify data user enter in apps and data from database
 */
protected String doInBackground(String... args) {
    String username = username_xml.getText().toString();
    String password = password_xml.getText().toString();

    // Building Parameters
    List<NameValuePair> params = new ArrayList<NameValuePair>();
    params.add(new BasicNameValuePair("username", username));
    params.add(new BasicNameValuePair("password", password));

    // getting JSON Object
    // check user identity accepts POST method for secure
    JSONObject json = jsonParser.makeHttpRequest(url_check, "POST", params);

    // url to check data from database
    private static String url_check = "http://mobcas.3eeweb.com/check.php";
}
```

Figure 14. Segment of JAVA Codes to connect with PHP

This codes is to allow the PHP codes (Refer Figure 15) to retrieve data from the phpMyAdmin database. The PHP file uses Structure Query Language (SQL) commands to connect, read and manipulate data in the database. Data from the database will be transmitted to MobCAS using the PHP codes as illustrated in Figure 16. The JAVA codes to accept data from a PHP file and to be displayed in the MobCAS application are presented in Figure 17.

```
!k?php
// check for required fields
if (isset($_POST['username']) && isset($_POST['password'])) {

    $username = $_POST['username'];
    $password = $_POST['password'];

    // include db connect class
    require_once __DIR__ . '/db_connect.php';

    // connecting to db
    $db = new DB_CONNECT();

    // mysql inserting a new row
    $result = mysql_query("select * from lecturer where lectid = '".$username.'" AND lectnps = '".$password.'"");
}
```

Figure 15. PHP codes to connect and retrieve data in the database

```

// mysql inserting a new row
$result = mysql_query("select * from lecturer where lectid = ".$username."
AND lectnps = ".$password."");
$rows = mysql_num_rows($result);
$row = mysql_fetch_array($result);

// check if row inserted or not
if ($rows > 0) {
    // successfully inserted into database
    $response["success"] = 1;
    $response["message"] = "User Found.";
    $response["lid"] = $row['lid'];
    $response["lectnm"] = $row['lectnm'];
    $response["lectml"] = $row['lectml'];

    // echoing JSON response
    echo json_encode($response);
} else {
    // failed to insert row
    $response["success"] = 0;
    $response["message"] = "User Not Found.";

    // echoing JSON response
    echo json_encode($response);
}

```

Figure 16. Information retrieved From the Database Wait for Java Response

```

try {
    int success = json.getInt(TAG_SUCCESS);

    if (success == 1) {
        // successfully created product
        lid = json.getString(TAG_LID);
        name = json.getString(TAG_NAME);
        email = json.getString(TAG_EMAIL);

        Intent i = new Intent(getApplicationContext(), successActivity.class);
        i.putExtra(TAG_ID, lid);
        i.putExtra("name", name);
        i.putExtra("email", email);
        Log.d("Create Response", " PERGI ISI COURSE, BAWA "+name+"+email);
        startActivity(i);

        // closing this screen
        finish();
    } else {

```

Figure 17. JAVA codes

4. Results and Discussions

After implementing the system to be used by a few of lecturers and degree students who registered in the Computer Science program in UiTM, Tapah branch, a survey was conducted to measure the system usability. These users have used the system for two weeks and have to give their perception on the system usability. The scope of usability has been explained to these users, which include system ease of use, fast processing, user friendly and the accuracy of calculating the percentage of absent. The users were categorized as competent Android users and have a basic knowledge on developing a mobile application. Figure 18 is the pie chat to present the result of survey.

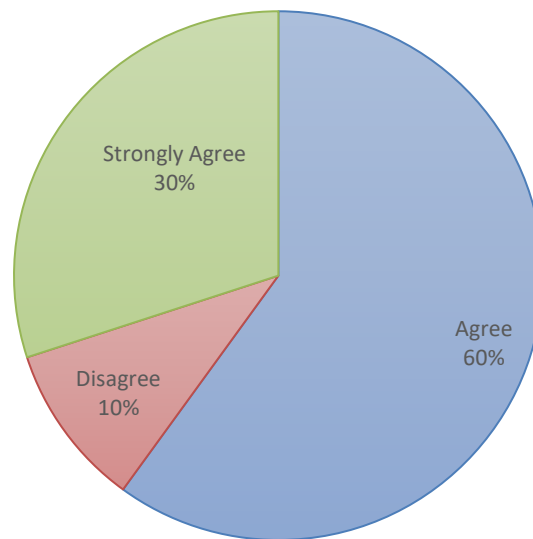


Figure 18. Usability acceptance of users on the system

Based on their experienced on using the MobCAS application, most of the users were not having problem in installing the application on their Android device. All the functions provided by the system have works very good and information can be displayed fast. Therefore, as seen in Figure 18, majority of them found that the system is useful and believed that the system is acceptable.

5. Conclusion

The MobCAS provides a paperless solution in collecting student's attendance which can minimize the misplaced attendance sheet. The automated student presence's percentage also helps the lecturers to observe their students attendance. However, there is a limitation where the lecturer needs to clicks the student's name sequentially to record the attendance. For future work, new functions may be added in the system such as the student details or their academic report. It is also suggested that there is a connection between the MobCAS and the current web portal of academic system in UiTM.

Acknowledgment

The authors gratefully acknowledge the Universiti Teknologi MARA (UiTM), Perak branch for giving the authors the opportunity, support, and facilities to carry out this research project.

References

- [1] I. N. M. Shahrane, J. M. Jamil, and S. S. M. Rodzi, "The application of Google Classroom as a tool for teaching and learning," *J. Telecommun. Electron. Comput. Eng.*, vol. 8, no. 10, pp. 5–8, 2016.
- [2] N. Septantiningtyas, J. Juhji, A. Sutarman, A. Rahman, N. Sa'adah, and others, "Implementation of Google Meet Application in the Learning of Basic Science in the Covid-19 Pandemic Period of Student Learning Interests," in *Journal of Physics: Conference Series*, 2021, vol. 1779, no. 1, p. 12068.
- [3] S. ÇANKAYA and G. DURAK, "Integrated Systems in Emergency Distance Education: The Microsoft Teams.," *Necatibey Fac. Educ. Electron. J. Sci. & Math. Educ.*, vol. 14, no. 2, 2020.
- [4] C. Gautreau, "Motivational factors affecting the integration of a learning management system by faculty.," *J. Educ. Online*, vol. 8, no. 1, p. n1, 2011.
- [5] S. Roy and C. Williamson, "Why is my LMS so slow? A Case Study of D2L Performance Issues," in *Proceedings of ISCA Computers and Their Applications (CATA)*, 2018, pp. 34–39.
- [6] J. Joseph and K. Shinto Kurian, "Mobile OS--Comparative Study," *J. Eng. Comput. & Appl. Sci.*, vol. 2, no. 10, pp. 10–19, 2013.

-
- [7] J. M. Wargo, *PhoneGap essentials: Building cross-platform mobile apps*. Addison-Wesley, 2012.
- [8] K. W. Tracy, "Mobile application development experiences on Apple's iOS and Android OS," *Ieee Potentials*, vol. 31, no. 4, pp. 30–34, 2012.
- [9] R. Gangaiamaran and M. Pasupathi, "Review on use of mobile apps for language learning," *Int. J. Appl. Eng. Res.*, vol. 12, no. 21, pp. 11242–11251, 2017.
- [10] D. Singh, E. J. Leavline, and P. M. Vijayan, "Mobile application for student attendance and mark management system," *Int. J. Comput. Intell. Res.*, vol. 13, no. 3, pp. 425–432, 2017.
- [11] R. Samet and M. Tanriverdi, "Face recognition-based mobile automatic classroom attendance management system," in *2017 International conference on cyberworlds (CW)*, 2017, pp. 253–256.
- [12] H. Adal, N. Promy, S. Srabanti, and M. Rahman, "Android based advanced attendance vigilance system using wireless network with fusion of bio-metric fingerprint authentication," in *2018 20th International Conference on Advanced Communication Technology (ICACT)*, 2018, pp. 217–222.
- [13] M. I. Dacholfany, "Getting parents involved in child's school: using attendance application system based on SMS gateway," *Int. J. Eng. & Technol.*, vol. 7, no. 227, pp. 167–174, 2018.
- [14] Z. Ayop, C. Yee, S. Anawar, E. Hamid, and M. Syahrul, "Location-aware event attendance system using QR code and GPS technology," *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 9, pp. 466–473, 2018.
- [15] T. R. Bhongale, S. S. Dhamnekar, A. K. Sanadi, S. K. Nandgave, A. A. Pawar, and S. A. Pardeshi, "Survey on tools and technologies applicable for mobile application development," *Int. Res. J. Eng. Technol.*, vol. 5, no. 3, pp. 2985–2988, 2018.