

# A Proposed Framework of Campus-Oriented Online Text Messaging System

<https://doi.org/10.3991/ijim.v14i16.11454>

Yan Watequlis Syaifudin (✉), Yoppy Yunhasnawa, Yushintia Pramitarini,  
Awan Setiawan, Erfan Rohadi, Pramana Yoga Saputra  
State Polytechnic of Malang, Malang, Indonesia  
qulis@polinema.ac.id

**Abstract**—Nowadays, the rapid advancement of mobile software development has triggered to emerge of more online text messaging applications. In the environment of higher education institutions, it is also widely utilized to support the communication process between campus members. This research proposed a system of campus-specific online text messaging that is expected to improve academic activities. As a case study, the system was implemented using React Native framework and Push Notification Service for use in State Polytechnic of Malang, Indonesia. The system consists of three main subsystems that work together, including Client Application, Server Application, and Push Notification Manager. To evaluate the proposed system, we conducted a survey. The result of the user satisfaction test shows that most users consisting of teachers, staff, management representative, and students felt more helped in the communication process in the campus environment.

**Keywords**—Text messaging; mobile application, push notification, higher education.

## 1 Introduction

Nowadays, the development of IT implementation has been accelerated at an unprecedented level [1]. Along with it, communication systems and technologies are also improving a lot. New standards like 4G LTE, as well as increasingly advanced mobile operating systems especially Android and iOS, have become widely adopted [2]. Moreover, in the software development area, currently, there are many new technologies and frameworks emerge. These new technologies enabling software developers to be much faster and easier in developing as well as maintaining their application [3]. The mobile device has various features such as personalized interfaces, real-time access to information, context sensitivity, instant interaction and reaction [4].

Currently, text messaging is widely used and is an important tool for student to communicate each other [5]. Many generic text messaging can work very well, fast, easy to use, multiplatform, and have a lot of features. However, besides all their superior capabilities, those generic online text messaging apps were not designed specifically for campus environment-related usage. On the other hand, the educational sector especially

higher educations have also improved, and students can enhance their learning abilities with applying information technology [6]. This naturally forces universities and other types of higher education institutions to utilize these online messaging apps due to the need for faster and more efficient communication. Yet, majority of generic online text messaging apps unable to solve specific problems in campus. In addition, the society will not have any problem with texted language [7].

In this paper, we proposed a framework that can be used to implement online text messaging that is specifically used for campus purposes. A campus has specific and continuous activities, such as teaching activities carried out by teachers to students, and research discussion activities between teachers and students [8]. Moreover, this framework can be integrated with existing campus systems, so that the required information can be shared with each other. To meet these needs, we developed a system consisting of three sub-systems, namely Client App, Server App, and Push Notifications Manager. Client App subsystem or client application is an application that runs on users' mobile devices and implemented using React Native technology. Server App or server application is a subsystem that acts as a bridge between one client application and other client applications. The 3rd subsystem, Push Notification Manager (PNM) is an application that runs on the same machine where the server application submodule is runs and supported by Google GCM [9].

To evaluate the effectiveness of this system, we implemented this system for educational activities at the State Polytechnic of Malang. Then we conducted a survey of users of this system which included teachers, students and administrative staffs. The survey is divided into two, namely questionnaire of functional requirements and questionnaire of user acceptance. The survey results showed that 83 percent of users agreed and strongly agreed to the first questionnaire, then 87 percent of users gave a positive statement for the second questionnaire. Thus, it is confirmed that the system is useful and helpful for educational activities in higher education institution.

This paper contains 8 sections. Section 2 provide some related research that was related to online messaging, mobile application, and implementation of mobile system in higher education. In section 3, we discuss about the concept of text messaging and the characteristic for campus implementation. Section 4 explain about the method to develop proposed system. Section 5 present the architecture of the system. In section 6, we show implementation process of the system. Evaluation about system implementation is explained in section 7. Finally, we resume the conclusion about this research and provide the future works.

## **2 Related Works**

In 2009, Jones et al conducted a study about the impact of SMS communication to enhance learning environment [10] and the result showed that the communication medium represented a powerful tool for enhancing learning activities.

In 2010, McClean et al developed a SMS based text-messaging to provide communication and voting for students of higher education [11].

In 2013, Premadasa et al [12] proposed a short messaging system that integrated into the Moodle LMS. It can inform the assessment and teaching schedules to students and the resume of teaching activities.

In 2013, Awad et al proposed an Academic Instant Messaging application for student [13]. It can achieve better communication between students to share knowledge and experiences between them.

In 2017, Shi et al reported an application of WeChat teaching platform that had multiple functions and supported to wireless network technology [14]. This new application is feasible and effective in facilitating teaching activities and in developing the students' competencies.

In 2018, Glowacki et al developed HealthyhornsTXT, a text-messaging to promote health and wellness for college students [15]. It focused more on promoting healthy behaviors and preventive strategies, rather than support learning activities.

### **3 Campus-Oriented Messaging**

Currently, text messaging is widely used and is an important tool for student to communicate each other [5]. Many generic text messaging can work very well, fast, easy to use, multiplatform, and have a lot of features. However, besides all of their superior capabilities, those generic online text messaging apps were not designed specifically for campus environment-related usage. In this section, we would like to explain the intention of campus-oriented messaging.

#### **3.1 The messaging application**

Text messaging is the form of composing, sending, and receiving digital messages in the text data type that consist of alphabetic and numeric characters. Text messaging can create a conversation between two or more users using electronic devices, including mobile devices, PCs, or other type of compatible digital computer. The examples of implementation of text messaging are SMS, chat board, and many popular instant messaging. The utilization of text messaging in academic area has earned reputation in recent years including in higher education [11]. The communication between teacher, staff, students using messaging has the potential to enhance the support given to students by academic departments during the academic period in university.

Text messaging is the most effective way to reach students, an important entity in an education institution. In specific way, students need a real-time media that can distribute any information and news related to their study interactively. For this purpose, a campus-oriented text messaging has some features related to the exclusively of higher education system, including user, education system, and organization. Implementing the campus-oriented text message intervention was feasible and useful for students but was also met with some challenges [15]. The system has to meet with the organization rule and all user must be adaptive to the new system.

### **3.2 Characteristic of campus-oriented messaging**

There are a number of the academic institution problems that are possible to solve by building a custom text messenger app, some of the most important ones are described as follows.

### **3.3 Homogenous type of users**

In typical generic text messaging app, generally, there is no exception in the treatment of all of their users that anyone who uses the app is just the same, a user. In those kinds of application, there is no specific differentiation in terms of rules and privileges. It is true that in some cases, the user of generic text messaging app is separated but, it is just based on the context of business and/or advertising. As an example, sending of broadcasts to promote some products is a good purpose, but it might be not very useful for campus environment due to the fact that majority of the higher educations are not a profit organization.

For generic text messaging apps, commercial users and regular users are not so much different. In contrast, there are many real differences between student, teacher, and staff on campus. For a simple instance, a teacher might usually have a different topic to tell whenever he/she send a message to a student and another teacher. By differentiating and highlighting the types of user, we can help each user to instantly grasp the context of the conversation even in any non-ideal situations like whenever a user received a message from a new contact or at the time when they are in a low focus level conditions.

### **3.4 The messages**

Majority of generic text messaging apps handle broadcast messages and individual messages in same way. Broadcast messages are just plain messages, the difference is just the number of recipients. Both ways are having a same risk: broadcast information loss. Because of vast number of messages received every day, there are always chances that a teacher will misinterpret an incoming broadcast as a regular chat from a group or a student and at a maximum extent. Also, the subsequent consequence will be missing an importance broadcast either from management or superiors.

Majority of university teachers, especially in Indonesia, have a lot of daily work. Their job is not only teaching. To get the job fulfilled, they have to communicate with a lot of persons every day, either their subordinates, superiors, or of course their students. Each message has a degree of importance. Messages that came from a different type of user has a different level of importance. Misinterpreting the source of the message will also potential in causing some kinds of incidents like belated reading of an important message from a superior. By dividing messages into broadcast and non-broadcast, as well as adding some insights to the user about from whom the message came, the new messaging app can help users to avoid such problems.

### **3.5 Academic elements integration**

Student supervision to some extent can be executed through some simple additions in a messenger app without having a dedicated complex management software. A messaging app is one of some most prominent apps on the campus. In a case of State Polytechnic of Malang, most of the teachers are using a popular text-messaging much more than the official academic information system application. They use the popular one to do various tasks related to the classes they teach. The collaborative learning itself can extend the teaching outside the classroom and delivers specific individual learning [16]. Starting from a simple task like giving announcements about tasks, giving notification to a class leader if they are going to come late, until an important thing like managing their last-year supervised students.

That kind of way can be solved in any messaging apps, but it will easier and more intuitive by integrating academic-specific features in a messaging app. For example, if the registration system can be connected to campus' information system for validation, the teachers will not be required to create a new contact every time any new student sending a chat to them. This simple integration also makes any of the users to rest assured that there will be no fake account or scam.

## **4 The Proposed System**

To overcome the mentioned problems, we hypothesized some strategies in the form of functional requirement list that is possible to be implemented in a new online text messaging platform. The proposed strategies are corresponded to each problem mentioned earlier. There are three aspects to be defined first to make solution for this system, including identify user classification, describe chat organization, and define the functional requirements.

### **4.1 Identification of user classification**

The first step is dividing the type of users and differentiating the treatment of each of the types of user. This differentiation is based on each user's role on the campus. Generally, there are 4 main types of roles in a university campus, namely student, teacher, staff, and management. Each of the parts is related to each other, but at the same time, they have different working environment along with the different business. That is why in the application context, they must be differentiated.

In the chat application that is created, the contact list is designed in such a way to enable the user in understanding the type of contact automatically, where any students name automatically appears without manually register their name first on the contact form, as well as the name of the teacher, staff, and management. By implementing it, the users will be guaranteed to be aware of the person whom they are going to chat, whether he/she is a student, teacher, staff, or management.

#### 4.2 Chat organization

Secondly, the application must be able to organize the messages based on its sender's account type. The users must be aware that any message they received is coming from students or any other account types so that they could be able to accurately decide the degree of importance of the messages. Users also have to be understood clearly about to whom a message will be sent. Furthermore, users must also be well-informed, from all messages that are listed in their phone, which one is a regular chat, and which one is a broadcast message. All of this will only be possible to implement whenever the users have been correctly differentiated, enabling the app to modify anything that can be seen, known, and done by each of user types based on their needs.

#### 4.3 Functional requirements

There must be some meaningful integration of academic elements in the messaging app. The integrated elements are not necessarily all and complex. They can be very simple, but they must have some considerable degree of overall benefits for the campus, and it is suggested to be in the form of academic managerial capabilities. In this research work, it is decided to pick two features related to the solution.

The system would be able to organize all active students into an automatically formed group, enabling every teacher to send messages to a class member, precisely. Another feature is the capability for teacher user type to set the members of his final project students that they have to advise on the current academic year.

In summaries, all the specific functional requirements that must be implemented in the campus-oriented online messaging app can be seen on the Table 1.

**Table 1.** Campus-oriented Messaging App Functional Requirements

Special Functional Requirements		Related Features	
No.	Requirements	No.	Features
1	User types differentiation	1.1	Different login types
		1.2	Automatic contact name and types recognition
		1.3	Contact list grouping based on user's type
2	User type-aware Messages organization	2.1	The division of chat type into broadcast message and regular message
		2.2	The chat list grouping based on user's type
3	Academic elements integration	3.1	Student's class settings and auto group formation.
		3.2	Teacher's supervision members group.

### 5 System Architecture

In this research work, we developed an application that runs on mobile devices. The application has an ability to send and receive text messages, in which a message sent from one sender device to one or more corresponding destination device(s) based on the ID of its user. From the overall perspective the complete system that was developed,

consists of three main subsystems, namely Client App, Server App, and Push Notifications Manager as shown in Fig. 1.

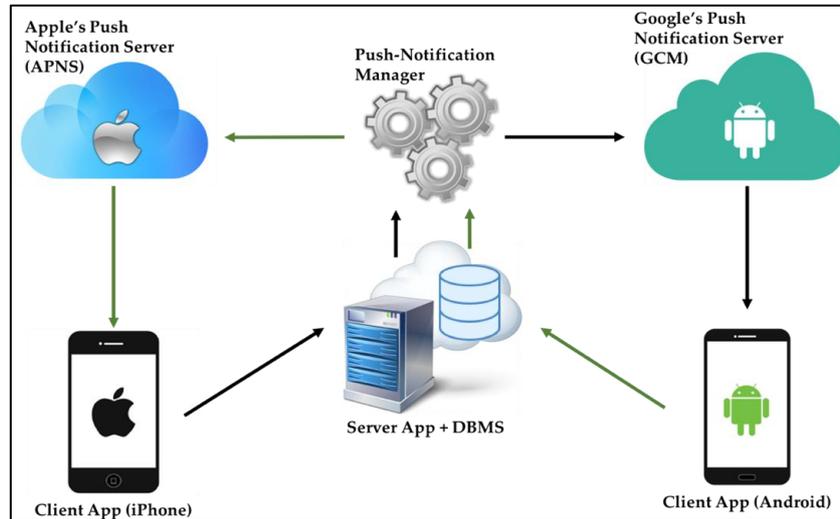


Fig. 1. System architecture

### 5.1 Client application

Client App subsystem is an application that runs on users' mobile devices. By using the Client App, users would be able to send a message to one or more destinations whether it is to the same mobile operating system or different and this also applies to receive a message. This subsystem contains most of the important features that are implemented. The most important one is the message sending and receiving feature. This feature works on standard web client-server communication mechanism.

When a user sends a message, the application engine converts the text into a plain string in JSON format. Then, this JSON string is sent to the Application Server subsystem via the HTTP protocol with the POST method to a specified URL where the Server Application is deployed. Every transmitted JSON string will always have two types of data namely API Key and payload. The API Key is a string that is encrypted specifically for the validation purpose of any data being sent to the server. While the payload is the actual data sent by the user. It is possible to be either the path to the service contained in the server application or the string containing a chat message.

Finally, this client application subsystem is implemented using React Native, a framework for developing JavaScript-based mobile applications [17]. Brito et al showed that React Native produces the best results in all aspects than other frameworks [18].

## **5.2 Server application**

Server Application is a subsystem that functions as a bridge between one client application and other client applications. Certain components communicate with a database server and the Push Notification Manager subsystem. It is created using PHP programming language and connected to MySQL database server instance. The main task of this subsystem is to receive the message sent by the client application, validate it, and then forward it to the push notification manager subsystem.

When there is a message sent by one of the users using the Client Application in the form of a HTTP request, this subsystem will check the API key contained in the request that was sent. This API key serves as a tool to ensure that the request being sent is a genuine request that comes from a registered user who owns an account on the system. Validation is performed by comparing the key sent by the client application with another special key generated periodically by the server app subsystem with a specific mechanism. If the API key is successfully validated, meaning the two keys are identical, the message will then be extracted from the JSON request's payload through the decryption process to be temporarily stored in the database.

In the request's payload sent by the user through the Client Application, there is also information about the user ID that is being targeted by the sender. The Server application will then query the user table to find a unique device id from those intended users. After the user id and corresponding UDID are found, the Application Server will compile the message and UDID in a special format and then forward it to the push notification manager subsystem.

## **5.3 Push notifications manager**

This subsystem is an application that runs on the same machine where the Server Application is runs. It is also written in PHP language in order to make the process of future maintenance easier. The main function of this subsystem is to forward the message received by the Server Application to the correct Push Notification Servers owned the corresponding operating system vendor of the mobile device [19] to which the chat message is sent. These vendor servers work by sending text in certain internet packets aimed at a special port resides on each user's mobile device [20]. If the recipient uses a mobile phone with the Android operating system, this subsystem will forward the message to Google's GCM server [21] [22]. On the other hand, if the intended user uses iOS mobile device, the message will be sent to Apple's APNS server [23]. Finally, the process of pushing messages to each destination device will be handled by each of those vendor servers themselves dynamically [24].

## **6 System Implementation**

In this section, we describe the implementation process of system architecture and then evaluate it to all users.

## **6.1 Development process**

The development process of the online messaging system in this study was conducted with the SDLC Waterfall model method where the stages of software development are carried out sequentially starting from the requirements planning process which is carried out first and then followed by the design process, execution, testing, and release [25]. All components in the entire system are written using the object-oriented programming paradigm.

## **6.2 System design**

In the client application, a special design pattern is used whose main purpose is to decouple the business logic and the user interface for the sake of better software design [26]. While the business logic is implemented in a class in a file with the extension “\*.js” which is a derivative of the React Native's 'Component' class. Because of the nature of JSX that can be used mixed with JavaScript, the source code of the user interface tends to be merged with the business logic code in a common file called 'Screen'. Therefore, it is necessary to make a strict separation between the two, the business process script is placed in a separate file, as well as the JSX code that is also written on a different file.

In the design pattern, as shown in Fig. 2, the business logic is still placed on Screen files as it is commonly done in any other React Native applications. The file ends with \*Screen.js, where in one screen file there must be only one class which is the representation of each single page shown to the user (it is known as 'Activity' on Android or 'ViewController' on iOS) in the application. The difference is that in the 'render ()' function section, where there are usually JSX codes, it is replaced with a global function that is imported from the appropriate view file. This global function returns JSX to the Screen class which is finally displayed to the user as the application interface. In each instance of Screen class there will always be a property called 'State'. This property can be associated with any entity classes which in this submodule is marked with a file name ending in '\*Data.js'. If there is an interaction by the user that results in changes to the data, the Screen class will update the corresponding entity object contained in the property state. The changes will then be applied through the implementation of the 'setState ()' function which automatically sends a message to the React Native engine. The engine will further adjust the UI in the component hierarchy called 'virtual DOM' [27], so that the proper UI with updated contents can be displayed to the user.

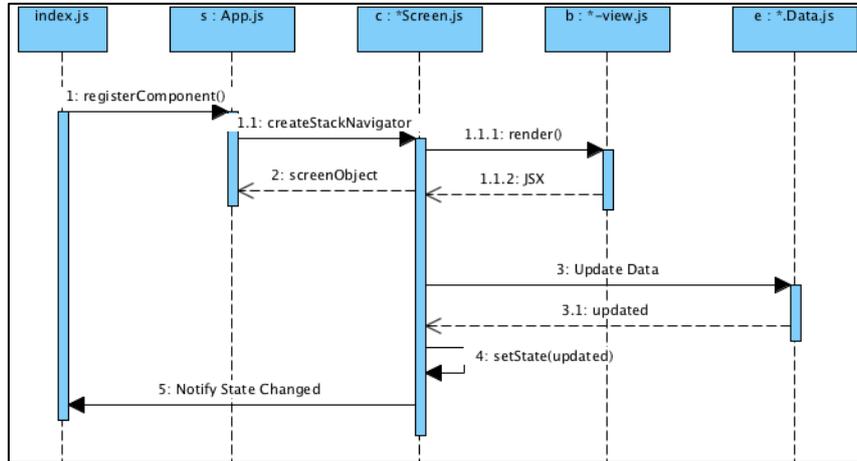


Fig. 2. Client App subsystem

### 6.3 Implementation result

At this stage, three subsystems have been built, namely Client Application, Server Application, Push Notifications Manager. Client application can be run on mobile devices and has several features such as user login, user registration, displaying messages, displaying announcements, displaying profiles, and chatting features as show in Fig. 3.

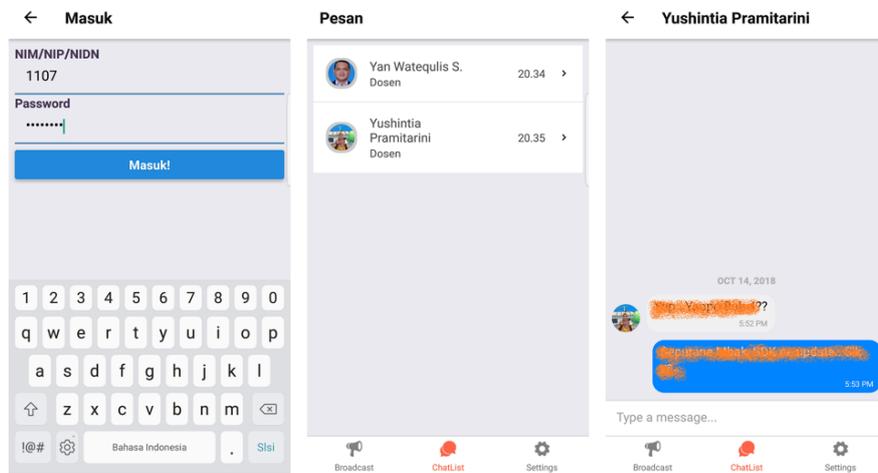


Fig. 3. Client Application features (login, message, and chatting)

For Server Applications and Push Notification Managers that use PHP 5.3, by default the object-oriented programming is supported [28]. While for the client application, the React Native engine has adopted the JavaScript ECMAScript 6 (ES6) [29] standard that supports object-oriented syntax. However, because JavaScript is a programming

language with the functional programming paradigm [30], it must be ensured that the ES6 syntax is always used. For instance, to declare a class, the keyword 'class' must be used instead of 'function' as it is commonly used in JavaScript 5 dialect.

## 7 Evaluation

After the entire system was completed, the compiled mobile app installer of the Client App subsystem distributed to several testers consisting of students, teachers, and staff to be installed on their phones. To face the user requirements, we implement the evaluation to 50 students, 10 staffs, 20 teachers, and 10 management representatives. The questionnaire provided to the respondents is about to which extent a tester agreed or disagreed with a list of statements where each statement symbolizes the importance of a corresponding feature contained in each hypothesized special functional requirement. The list of statements and their relation to each hypothesized feature are presented in the Table 2.

**Table 2.** Questionnaire of Functional Requirements Hypotheses

No.	Questioned Statements	Feature No.
Q1	Somehow, the way I perform communication to a (student/teacher/staff) is different.	1.1
Q2	I am lazy to manually save a (student/teacher/staff)'s name and phone number to my contact book.	1.2
Q3	I need to know precisely which person is a (student/teacher/staff) by seeing my contact list at a glance.	1.3
Q4	I experienced difficulties in finding important broadcast message quite often.	2.1
Q5	I have some kind of priority level when reading incoming messages from any of the (student/teacher/staff).	2.2
Q6	I often need to know who the members of a year/class are.	3.1
Q7	I experienced difficulties in keeping track of (students under my supervision/who are my supervisors) during the final project.	3.2

Those list of statements in the questionnaire has been adjusted for each type of application users. That adjustment is in the form of choice (student/teacher/staff) that is available in almost all the text of the statements. While the 'Feature No.' column in the above table shows the relationship between each question and its corresponding feature that is mentioned in Table 1. To complete the questionnaire, respondents must provide answers that represent what they feel after reading each of the questions. The answer to each question statement can only be either 'Disagree', 'Agree', or 'Strongly Agree'. A summary of the results of the respondents' answers can be seen in Fig. 4.

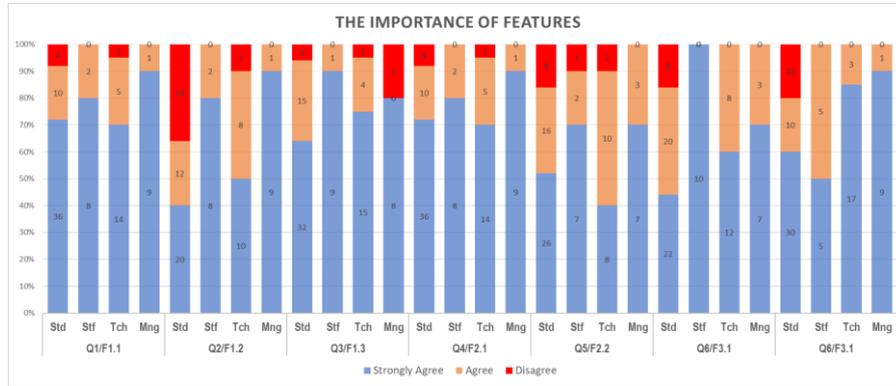


Fig. 4. 1<sup>st</sup> Questionnaire Result

As it can be observed from the chart, all hypothesis points always get the 'Strongly Agree' answers more than the 'Disagree' answer. The average answer of 'Strongly Agree' is 57.7% for student, 78.6% for staff, 64.3% for teacher, and 82.9% for management. Otherwise, the average answer to 'Disagree' is only 15.7% for student, 1.4% for staff, 5% for teacher, and 2.9% for management. Based on the statistics, it can be concluded that the overall features hypothesized were indeed considered important by the users and needed to be implemented in the system.

After they filled out the questionnaire, the respondents were then requested to utilize the Client App subsystem in their phone for a certain period. Respondents were asked to send messages to each other, regardless of their type. After the application trial period is over, all the testers are again requested to fill out a second questionnaire which contains questions about how far the hypothesized features implemented help them in communicating at the campus. The list of questions is summarized in the Table 3.

Table 3. Features Acceptance Test Questionnaire

No.	Questioned Statements	Feature No.
Q1	It is become easier for me to differentiate the way I perform communication to a (student/teacher/staff).	1.1
Q2	I can access (student/teacher/staff's) name and phone number in my contact book without manually saving it first.	1.2
Q3	I am able to know precisely which person is a (student/teacher/staff) by seeing my contact list at a glance.	1.3
Q4	It is become easier for me to find important broadcast messages.	2.1
Q5	It is become easier to me to determine proper priority level when reading incoming messages from (student/teacher/staff)	2.2
Q6	It is become easier to me to know who the members of a year/class are.	3.1
Q7	It is become easier to me in keeping track of (students under my supervision/who are my supervisors) during the final project.	3.2

The list of questions given to the testers in the second questionnaire was also made in a similar format as the questions in the first questionnaire. Each of these questions also relates to the list of feature hypotheses. However, in this questionnaire, respondents

were only asked to choose binary answers consisting of 'Yes' and 'No' option. If the tester answers 'Yes', then it can be interpreted that the related features have succeeded in carrying out their functions. Otherwise, if the tester answers 'No', then it can be interpreted the opposite. A summary of the results of this second questionnaire can be seen in Fig. 5.

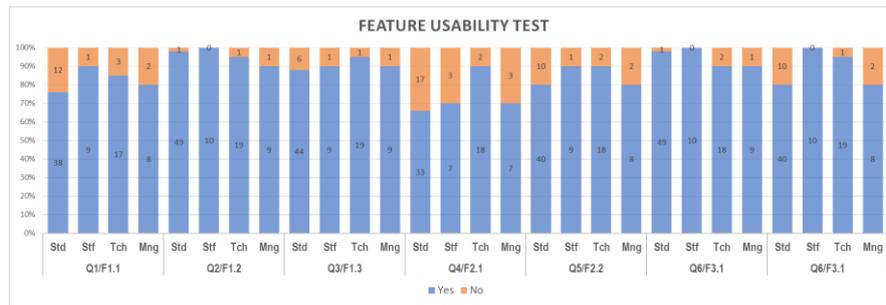


Fig. 5. 2<sup>nd</sup> Questionnaire Result

From the results, all of the features represented by each question in the questionnaire get "Yes" answer much more than "No" answer. There is only one feature that gets "Yes" answer less than 70%, the no. 2.1 feature for student and staff, which features the separation of broadcast messages and regular messages. Some testers answered 'No', probably because, even though the broadcast messages had been separated from regular messages and grouped into one place, some of the testers had a very large number of broadcast messages so they still found it somewhat difficult to find the important ones. Therefore, this point is possible to be improved in future research.

## 8 Conclusion

The rapid development of information technology has caused many online messaging applications to emerge and some of these messaging applications are very popular among mobile device users. Some issues were considered as the most important and discussed by this research, including all types of users who are considered the same regardless of whether they are teachers, students, or staff, equal treatment of all types of chat, and lack of essential academic features integrated into the application.

The system created consists of three subsystems, the Client App subsystem in the form of a mobile application, Server App subsystem, and Push Notification Manager in the form of applications that run on a server computer. The results of the survey show the majority of users agree that the features applied are necessary to improve the quality of communication on campus. Whereas from the results of the application testing by the tester, it was found out that most of the tester felt that the developed application, could solve the problems that were hypothesized earlier.

In the future, we will continue to develop several features that will complement this framework, such as the recognition of important messages, language translation [31], security aspects [32] [33], and implementation in cloud system [34]. In addition, we

will also develop the design of this framework and spread it to the teaching environment [35], it can be more open for integration with existing academic systems.

## 9 References

- [1] Buhalis, D., & Law, R. (2008). Progress in information technology and tourism management: 20 years on and 10 years after the Internet-The state of eTourism research. *Progress in Tourism Management*, 24(4), 609-623. <https://doi.org/10.1016/j.tourman.2008.01.005>
- [2] Tilson, D., Sorensen, C., & Lyytinen, K. (2012). Change and Control Paradoxes in Mobile Infrastructure Innovation: The Android and iOS Mobile Operating Systems Cases. 45th Hawaii International Conference on System Sciences. 1324-1333. Maui, HI, USA: IEEE. <https://doi.org/10.1109/HICSS.2012.149>
- [3] Martinez, M., & Lecomte, S. (2017). Towards the Quality Improvement of Cross-Platform Mobile Applications. *IEEE/ACM 4th International Conference on Mobile Software Engineering and Systems (MOBILESoft)* (pp. 184-188). Buenos Aires, Argentina: IEEE. <https://doi.org/10.1109/MOBILESoft.2017.30>
- [4] Papadakis, S. (2018). Evaluating pre-service teachers' acceptance of mobile devices with regards to their age and gender: A case study in Greece, *International Journal of Mobile Learning and Organisation*, vol. 12, no. 4. <https://doi.org/10.1504/IJMLO.2018.10013372>
- [5] Boştină-Bratu, S. (2015). Text Messaging Vs Academic Writing - A Case Study. *The 21st International Conference Knowledge-Based Organization*, 21, 545-550. Sibiu, Romania. <https://doi.org/10.1515/kbo-2015-0093>
- [6] Papadakis, S., Kalogiannakis, M., Sifaki, E., & Vidakis, N. (2018). Evaluating Moodle use via Smart Mobile Phones. A case study in a Greek University, *EAI Endorsed Transactions on Creative Technologies*, vol. 18, no. 16, pp. 1-9. <https://doi.org/10.4108/eai.10-4-2018.156382>
- [7] Shafie, L. A., Darus, N. A., & Osman, N. (2010). SMS Language and College Writing: The languages of the College Texters. *International Journal of Emerging Technologies in Learning*, 5(1), 26-31. <https://doi.org/10.3991/ijet.v5i1.1010>
- [8] Syaifudin, Y., W., Funabiki, N., Kuribayashi, M., & Kao, W., C. (Accepted/In press). A proposal of Android Programming Learning Assistant System with implementation of basic application learning. *International Journal of Web Information System*.
- [9] Ding, J., Song, W., & Zhang, D. (2014). An Approach for Modeling and Analyzing Mobile Push Notification Services. *2014 IEEE International Conference on Services Computing* (pp. 725-732). Anchorage, AK, USA: IEEE. <https://doi.org/10.1109/SCC.2014.99>
- [10] Jones, G., Edwards, G., & Reid, A. (2009, November). How can mobile SMS communication support and enhance a first-year undergraduate learning environment? *Research in Learning Technology*, 17(3), 201-218. <https://doi.org/10.1080/09687760903247625>
- [11] McClean, S., Hagan, P., & Morgan, J. (2010). Text Messaging for Student Communication and Voting. *Bioscience Education*, 16(1), 1-12. <https://doi.org/10.3108/beej.16.4>
- [12] Premadasa, H., & Meegama, R. (2013). Implementation of concatenated short messaging service in a campus environment. *8th International Conference on Computer Science & Education* (pp. 207-212). Colombo, Sri Lanka: IEEE. <https://doi.org/10.1109/ICCSE.2013.6553911>
- [13] Awad, M., Kettaneh, A., Qazbak, F., & Haddad, M. (2013). Academic Instant Messaging Engine (AIME): Using technology to facilitate students' interaction. *2013 International Conference on Current Trends in Information Technology (CTIT)* (pp. 18-22). Dubai, United Arab Emirates: IEEE. <https://doi.org/10.1109/CTIT.2013.6749471>
- [14] Shi, Z., & Luo, G. (2017). Application of WeChat Teaching Platform in Interactive Translation Teaching Application of instant message system in cooperative learning. *International*

- Journal of Emerging Technologies in Learning, 11(9), 71-75. <https://doi.org/10.3991/ijet.v11i09.6113>
- [15] Glowacki, E. M., Kirtz, S., Wagner, J. H., Cance, J. D., Barrera, D., & Bernhardt, J. M. (2018, January). HealthyhornsTXT: A Text-Messaging Program to Promote College Student Health and Wellness. *Health Promotion Practice*, 19(6), 844-855. <https://doi.org/10.1177/1524839917754089>
- [16] Papachristos, D., Alafodimos, N., Arvanitis, K., G., Kalogiannakis, M., Kikilias, P., & Papadakis, S. (2010). An Educational Model for Asynchronous E-Learning. A Case Study in a Higher Technology Education, *International Journal of Advanced Corporate Learning*, vol. 3, no. 1, pp. 32-36, 2010. <https://doi.org/10.3991/ijac.v3i1.987>
- [17] Chinnathambi, K. (2018). *Learning React: A Hands-On Guide to Building Web Applications Using React and Redux*. Addison-Wesley Professional.
- [18] Brito, H., Gomes, A., & Bernardino, Á. S. (2019). JavaScript in mobile applications: React native vs ionic vs NativeScript vs native development. 14th Iberian Conference on Information Systems and Technologies (CISTI). Coimbra, Portugal, Portugal: IEEE. <https://doi.org/10.23919/CISTI.2018.8399283>
- [19] Stohy, R. A., Ghetany, N. e., & Gharib, H. A. (2016). A Proposed System for Push Messaging on Android. *International Journal of Interactive Mobile Technologies*, 10(3), 29-34. <https://doi.org/10.3991/ijim.v10i3.5567>
- [20] Bell, K. M., Bleau, D. N., & Davey, J. T. (2011). USA Patent No. U.S. Patent 8 064 896.
- [21] Yilmaz, Y. S., Aydin, B. I., & Demirbas, M. (2014). Google Cloud Messaging (GCM): An Evaluation. 2014 IEEE Global Comm. Conf. 2847-2852. Austin, TX, USA: IEEE. <https://doi.org/10.1109/GLOCOM.2014.7037233>
- [22] Vallina-Rodriguez, N., Aucinas, A., Almeida, M., Grunenberger, Y., & Crowcroft, J. (2013). RILAnalyzer: A Comprehensive 3G Monitoronn Your Phone. The 2013 Conference on Internet Measurement Conference (pp. 257-264). Barcelona, Spain. <https://doi.org/10.1145/2504730.2504764>
- [23] Rothman, B. S., Dexter, F., & Epstein, R. H. (2013, August). Communication Latencies of Apple Push Notification Messages Relevant for Delivery of Time-Critical Information to Anesthesia Providers. *Anesthesia & Analgesia*, 117(2), 398-404. <https://doi.org/10.1213/ANE.0b013e318299a7f7>
- [24] Liu, X., Cheng, B., Zhai, Z., & Chen, J. (2016). Design and Implement of Push Notification Server in Mobile IoT Environment. 2016 IEEE International Conference on Mobile Services (MS) (pp. 208-209). San Francisco, CA, USA: IEEE. <https://doi.org/10.1109/Mob-Serv.2016.40>
- [25] Balaji, S., & Murugaiyan, M. S. (2012, June). Waterfall vs V-Model Vs Agile: A Comparative Study on SDLC. *International Journal of Information Technology and Business Management*, 2(1), 26-30.
- [26] Leff, A., & Rayfield, J. T. (2001). *Web-Application Development Using the Model/View/Controller Design Pattern*. 5th IEEE International Conference on Enterprise Distributed Object Computing (p. 118). Seattle, Washington, USA: IEEE.
- [27] Danielsson, W. (2016). *React Native application development - A comparison between native Android and React Native*. Linköping, Sweden: Linköpings universitet.
- [28] Gutmans, A., Bakken, S. S., & Rethans, D. (2004). *PHP 5 Power Programming*. Hagerstown, Maryland: Prentice Hall.
- [29] Prusty, N. (2015). *Learning ECMAScript 6*. Packt Publishing.
- [30] Silva, L. H., Ramos, M., Valente, M. T., Bergel, A., & Anquetil, N. (2015). Does JavaScript software embrace classes? IEEE 22nd International Conference on Software Analysis, Evolution, and Reengineering (SANER) (pp. 73-82). Montreal, QC, Canada: IEEE. <https://doi.org/10.1109/SANER.2015.7081817>

- [31] Rutkas, A. (2008). Language translation module for instant messaging systems. 2008 International Conference on "Modern Problems of Radio Engineering, Telecommunications and Computer Science" (TCSET). Lviv-Slavsko, Ukraine.
- [32] Demiroglu, D., Das, R., & Tuna, G. (2017). An android application to secure text messages. 2017 International Artificial Intelligence and Data Processing Symposium (IDAP). Malatya, Turkey: IEEE. <https://doi.org/10.1109/IDAP.2017.8090330>
- [33] Aggarwal, P. K., Grover, P. S., & Ahuja, L. (2018). Security Aspect in Instant Mobile Messaging Applications. 2018 Recent Advances on Engineering, Technology and Computational Sciences (RAETCS). Allahabad, India: IEEE. <https://doi.org/10.1109/RAETCS.2018.8443844>
- [34] Ji, Z., Ganchev, I., O'Droma, M., & Zhao, Q. (2014). A Push-Notification Service for Use in the UCWW. 2014 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (pp. 318-322). Shanghai, China: IEEE. <https://doi.org/10.1109/CyberC.2014.88>
- [35] Syaifudin, Y., W., Rozi, I., F., & Asri, A., N. (2017). Web-Based Communication Model Between Teacher and Student in Foreign Language Software. Malaysian Construction Research Journal, 1(Special Issue), 231-240.

## 10 Authors

**Yan Watequlis Syaifudin** is member of IEEE and an associate professor at State Polytechnic of Malang, Indonesia. He received the bachelor's degree in informatics from Bandung Institute of Technology, Indonesia, in 2003, and the master's degree in information technology from Sepuluh Nopember Institute of Technology, Indonesia, in 2011, respectively. His research interests include Educational Technology, Information Systems, and Database Systems.

**Yoppy Yunhasnawa** is a lecturer at State Polytechnic of Malang, Indonesia. He received the master's degree from Chang Gung University, Taiwan. His research mainly focused on Artificial Intelligence, Information Systems, and Software Engineering. Email: [yunhasnawa@polinema.ac.id](mailto:yunhasnawa@polinema.ac.id).

**Yushintia Pramitarini** is a lecturer at State Polytechnic of Malang, Indonesia. She obtained the master's degree at Sepuluh Nopember Institute of Technology, Surabaya, Indonesia. Currently, her research interests include Information System, Machine Learning, and Software Engineering. Email: [yushintia@polinema.ac.id](mailto:yushintia@polinema.ac.id)

**Awan Setiawan** is an associate professor at State Polytechnic of Malang, Indonesia. His research mainly focused on Management Information Systems, Internet of Things, and Power Systems. Email: [awan@polinema.ac.id](mailto:awan@polinema.ac.id).

**Erfan Rohadi** is a senior lecturer at State Polytechnic of Malang, Indonesia. He received the PhD degree from Nagasaki University, Japan. His research mainly focused on Antenna Engineering, Information Systems, and Geographic Information Systems. Email: [erfanr@polinema.ac.id](mailto:erfanr@polinema.ac.id).

**Pramana Yoga Saputra** is a lecturer at State Polytechnic of Malang, Indonesia. He received the master's degree from Sepuluh Nopember Institute of Technology, Indonesia in 2014. His research mainly focused on Artificial Intelligence, Information Systems, and Software Engineering. Email: [pramanay@gmail.com](mailto:pramanay@gmail.com).

Article submitted 2019-08-06. Resubmitted 2020-03-14. Final acceptance 2020-03-14. Final version published as submitted by the authors.