A User-Centered Design based Collaborative System for Jum'a Preacher Scheduling

Hariandi Maulid [1], Indra Azimi [2], and Amir Hasanudin Fauzi [3]

[1,2,3,4] Department of Informatics Engineering, School of Applied Science, Telkom University, Indonesia
[1]hariandimaulid@tass.telkomuniversity.ac.id, [2]indraazimi@tass.telkomuniversity.ac.id, [3]amir_hf@tass.telkomuniversity.ac.id

Abstract
Majlis Tabligh Muhammadiyyah of Bandung Regency of Indonesia is currently managing 21 Mosques in its area. With only 75 preachers, handling 1.092 slots for Jum'a preacher scheduling every year is extremely difficult while it managed by the traditional mean. Other than collision, special request and constraints proposed by Imam (preacher) or Mosque manager make the scheduling process worst. This paper proposes a collaborative based system to address scheduling system with special requirements. Scheduling compilation will be divided into three stages. Then it uploaded into the system. As a result, Imam and Mosque managers are able to offer slots as their best choice, while Majlis tabligh administrator can easily manage Jum'a preacher scheduling.

***Keywords*:** collaborative scheduling system, Jum'a preacher, user-centered design

1. Introduction

Indonesia is the largest Moslem population in the world with 87,2 percent of all Indonesian are Moslems (International Religious Freedom Report for 2016). One of Moslem obligation is Jum'a prayer (Friday prayer). Jum'a prayer must be held in a Mosque congregationally together with other Moslems and preceded by Jum'a sermon delivered by a preacher. The one who leads the prayer and delivers the sermon is usually the same person called Imam (Padela et al, 2011). The Mosque manager is responsible for scheduling and assigning Imam for each Friday throughout the year. In Masjid Al-Akhbar Surabaya, East Java, Indonesia, the Imam selection process becoming the most important part before scheduling. The Imam must meet criteria set by the high priest or Mosque manager and get recommended by Moslem Council of Indonesia (MUI). They need 8 months to complete a fixed timetable (Wirawan, 2017). Most of the Mosques in Indonesia like Al-Akbar in Semarang, West Java, create a special task force for managing Friday Prayer activities from Imam selection to Imam scheduling (Sofwan, 2013; Kamaluddin, 2016).

Majlis Tabligh Muhammadiyyah (Muhammadiyyah Council for Propaganda) of Bandung Regency in West Java, Indonesia is managing more than 21 Mosques in southern Bandung area. For Jum'a prayer, they only have 75 Imams to lead the prayer and deliver the sermon at all Mosques. Accordingly, there are 1.092 slots that must be scheduled by Majlis Tabligh for every year. Every October, Majlis Tabligh will conduct a plenary meeting with all Imams and Mosque managers to decide a fixed timetable for next year. It takes around three months for Majlis Tabligh to compile such agreed timetable.

The main problem of managing that big number slots with a limited resource is high collision probability. Aside collision problem, Imam or Mosque manager may propose special requirements such as the time and place the Imam assigned or how many times an Imam can be assigned at the same Mosque in a month. After a fixed timetable is compiled, another problem may possibly occur. Since the timetable distributed via printed media, the Imam or Mosque manager may not aware about the schedule assigned to them.

This paper proposes a collaborative based system to address scheduling problem with special requirements. Timetable compilation period will be conducted in three stages. At the first stage, Imam and Mosque manager propose their special requirements through the mobile application interface. The second stage, Majlis Tabligh administrator compiles timetable draft noticing to special requirements using a specially programmed sheet. After compilation, Majlis Tabligh will discuss timetable draft in plenary session and upload a fixed timetable into a system which is the last stage of timetable compilation.

The rest of this paper is organized as follows. Section 2 discusses a brief overview of the collaborative system and user-centered design approach. Section 3 details our proposed system complemented by some existing researches from previous work. Section 4 describes the implementation of the system. Next section details performance evaluation of the proposed system using usability testing and closed by conclusion section.

2. Rudimentary and Methodology

This section will explain a brief overview of methodology and approach used in this research.

2.1 Information Security

The term of collaboration usually refers to an activity that allows a group of people to work together on common task (Bafautsou et al, 2001). A collaborative system is a set of system or groupware or multi-user system that designed to allow high-level interaction between the user such as cooperation and communication on particular task (Tolone et al, 2005), (Zhang et al, 2008). There are four (4) categories for such collaborative support system, namely: (1) group document handling; (2) real-time conferencing; (3) non-real-time conferencing, and (4) electronic meeting system (EMS) (Bafautsou et al, 2001). For the purpose of this paper, a non- real-time conferencing collaborative system is used as a framework for developing a Jum’a preacher timetable.

2.2 User-Centered Design

User-Centered Design (UCD) is a design philosophy that put and involve users as a center of system development process. In other words, UCD allows users to influence how a design takes shape. There are three (3) types of user that must be taken into account for successful design of product: (1) primary users: users who actually use the system regularly; (2) secondary users: users who occasionally use the system; and (3) tertiary users: users who will be affected by the use of system (Abras et al, 2004).

3. Human Intentions Based on the Sharia

There are some researches were conducted in the area of collaborative and user-centered design. Smaradottir et al (2015) developed a collaborative system for inter-municipal dementia utilizing a user-centered design approach. Yao et al (2015) proposed a new approach which unifies collaborative and content-based web service recommendation. They tested their approach with a real-world web services.

Another user-centered design project was conducted at the Europian Union and Southern Norway which develop a collaborative telemedicine system for remote monitoring of chronic obstructive pulmonary disease (Smaradotir et al, 2016). While Esteves and Pereira (2015) utilized user-centered design approach for mobile hospitality application , Jankowski et al

(2017) used it in development of rehabilitation devices after stroke. More research on this approach also comes from nursing and deaf field as detailed in (Al-Masslawiet al, 2017) and (Anindhita et al, 2016).

Figure 1 depicts all stages in designing and developing collaboration system for Jum'a preacher. It contains five activities: collecting data user need and expectation, preliminary design, evaluation of design and alternative, prototypes, and final artifact. These activities are iterated until the users are satisfied with the overall system, especially at preliminary design, evaluation of design and prototyping.

3.1 Requirements Analysis

The first stage is collecting data representing user need and expectations. To obtain them, we use some collecting data techniques including interview, observation, and questionnaires. There are 25 respondents in total who participated in this stage. Some respondents involved only in interview and some of them involved in both interview and questionnaires. Table 1 details the number of users that participated in user requirement analysis.

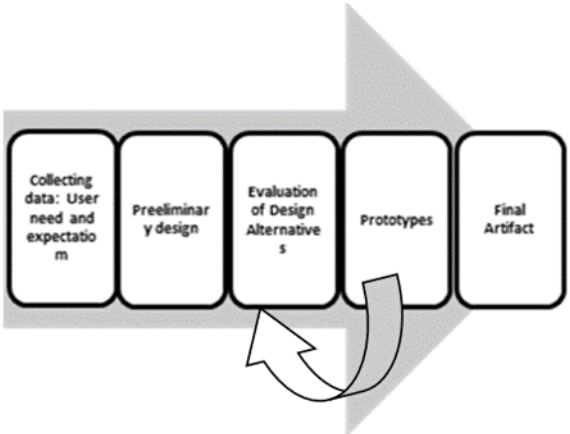


Figure 1: User-Centered Design of Proposed System

Table 1: User requirement analysis respondents

User Target	Interview	Questionnaires
Preacher/Imam	5	10
Mosque Manager	3	5
Majlis Tabligh	2	2

From this stage, we found that the main problem of compiling schedule is timetable collision that is caused by special timetable requirements proposed by either Imam/preacher or Mosque manager. These special requirements, at the end, become informal but inclusive constraints. According to the users, they require a system that (1) manages to provide an automatic collision detection while compiling Jum'a preacher timetable (2) facilitates all users to collaborate each other while proposing their expected timetable (3) views all previous and upcoming timetables specific to the user, and (4) provides a facility for the user to notify timetable administrator while he is unable to conduct the upcoming preaching schedule.

3.2 Preliminary Design

The previous stage has described that the proposed system should provide a solution from user problems in compiling Jum'a preacher timetable. The user final goal is allowing them to manage preacher timetable easily. In order to achieve it, some tasks must be taken including: (1) user input and update all related data including: preacher, Mosque, and constraint, (2) users need to propose their expected timetables, (3) users can identify timetable collision, (4) users can view all preacher timetable specific to the user (5) users can notify administrator of the system in accordance with his inability to perform his preacher task.

From these user goals and tasks, we can design a general system flow as depicted in figure 2. The flow started by input data such as preachers, Mosque, and constraints data not to mention current year setting. It then opens a request period for preacher or Mosque manager to propose their special requirement throughout the year. After that, the System Administrator can compile the final timetable and upload it to the server. Users now can view their previous and upcoming timetable and also request for absence via one button. The timetable request represents a collaboration between users, especially between users and administrator of the system. In the current system, preachers and Mosque managers are requesting expected timetable in a plenary meeting. This meeting can last hours and hours and even need four or five more meetings to complete.

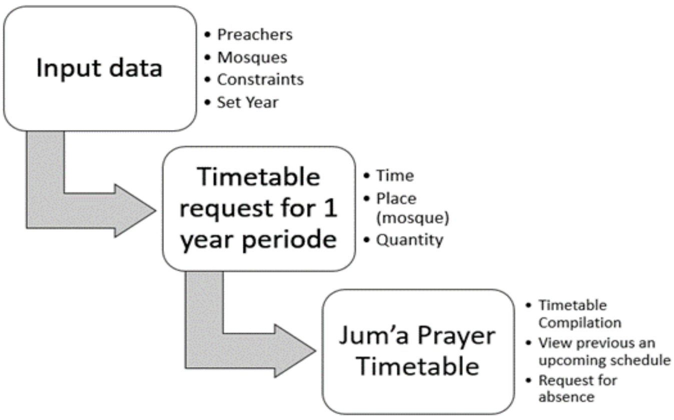


Figure 2: Proposed system flow

4. System Implementation

This section discusses the implementation of a collaborative system for Jum'a preacher scheduling with three iterations according to user-centered design approach. Firstly, we sketch a low fidelity prototype by involving end-user candidate. This sketch leads to high fidelity prototype including its interfaces. Next, the prototype tested and evaluated. These processes iterated until the final artifact agreed by end users. All these processes conducted based on the proposed system flow described in figure 2. With this system, Imam and Mosques managers can collaborate with Majlis Tabligh administrator to compile timetable by proposing special requests or constraints. Based on these requests, administrator completes all timetable slots and uploads it into system servers. In the end, all users can view their timetables and access other features provided by the system.

4.1 Firs Iteration -- Web-based Environment

During the first iteration, low and high-fidelity prototypes are discussed and refined. A timetable compilation scenario is completed by Imam, Mosque manager, and Majlis Tabligh member with the help of low fidelity prototype. Low fidelity prototype is a sketch of the flow of system and is done by drawing on the paper using a pencil. They can straightforwardly modify the prototype by drawing or making a note on low fidelity prototype paper based on their roles to make it easy to use and useful. At the end of this scenario, we can determine each user rules and functionalities.

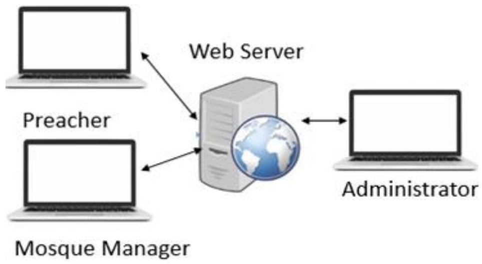


Figure 3: System Architecture (web-based)

We set three different types of user for the application along with their functionalities as detailed in figure 3. The first user is preacher/Imam. This user has eight functionalities including logging in and logging out of the system, updating profile, requesting special timetable, viewing the previous and upcoming schedule, viewing Mosques information along with their managers, contacting Mosque manager via phone call or SMS, and notifying System Administrator when he unable to perform his preaching task. The second user is Mosque manager. This type of user has close similar functionalities to the previous user. Except he has no right to notify System Administrator concerning the absence of assigned Imam. Instead of contacting the System Administrator, Mosque manager has functionality to contact Imam via phone call or SMS. The last user is member of Majlis Tabligh as System Administrator. His special functionalities including viewing and contacting detail information of Imam, Mosques and their managers, compiling timetable, uploading timetable, sending broadcast message, and also getting notification from Imam who unable to preach on his schedule.

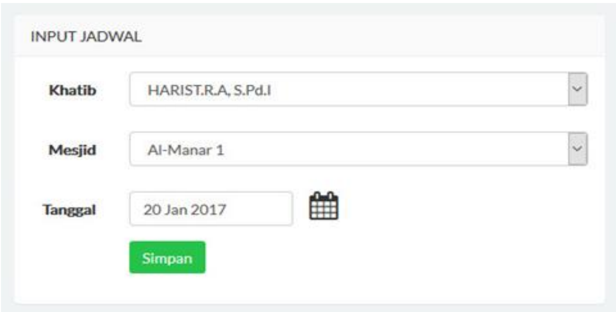


Figure 4: Web Interface for timetable input

After low fidelity prototype along with user management is set, we implement high fidelity prototype using web technology as a development tool and user interface. All user can access, request, and view timetables via a browser. We ask all types of users to experience the prototype. Most of the users are familiar with the flow of the system as they had been involved in the low fidelity prototype design phase. The problems arise when users come to fill special request form. Users must repeat the submission process as many as the number of expected requirement they input. For example, if an Imam wishes to propose three different timetable requests, he must fill the form three times and submit it also three times (one for each request). The System Administrator user has an even worst problem. Since he has to manage more than thousands of slots in a one-year period. The more slots to set manually the more clicks to do. As depicted in Figure 4, to input one timetable slot, the administrator has to select an Imam (*khatib*) via a drop-down menu and choose a Mosque (*masjid*) followed by determining date (*tanggal*). Then click submission (*simpan*) button to complete the process. He must perform this action all over again until all slots are fully assigned. Another problem is the fact that not everyone having access to a computer or laptop. All of users use the cell phone instead. They find that accessing browser via cell phone monitor is stressful.

4.2 Second Iteration – Mobile-based Environment

To deal with the problems found in the first iteration, we decided to switch the environment from web-based to mobile-based as a first step in respond to web access problem from the cell phone. Figure 5 captures this change in system architecture. However, despite the limitation of the special request form and timetable slots management, we kept their design unchanged to find out more whether or not the environment takes a significant impact on systems' ease of use and access. An access system for the System Administrator is slightly different from other users. We kept the web-based interface while adding a mobile-based access system for some reasons. First, System Administrator manages the big number of timetable data. He needs more spaces to go vertically or horizontally for extracting data, and this done by a wider screen than a cell phone monitor. Second, the System Administrator also has access to system web server including Cpanel and database server. In doing so, web access is more preferred.

All users are asked to use a redesigned high-fidelity prototype. They fully impressed and appreciated with the redesigning system into a mobile-based application. The new prototype considered more convenient to use than the previous one. Nevertheless, special requirement submission form still becomes trouble as the users must click multiple submissions for multiple timetables requirements as depicted in figure 6.(b). The users then suggested to the team to provide a form that facilitates multiple requests in just one submission. Similarly to the problem faced by Imam and Mosque managers, the System Administrator has to repeat submission for every unrequested slot. Suppose there are only 20 Imams out of 75 requesting for example 25% slots of 1092, there still 75% slots or 819 slots to assign. Assigning via the web or even mobile system submission form requires multiple clicks and submissions to complete. It is important to find an alternative method for handling a lot of data to input.

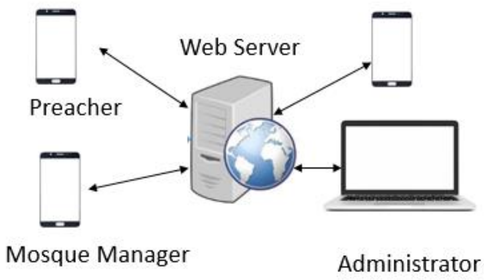
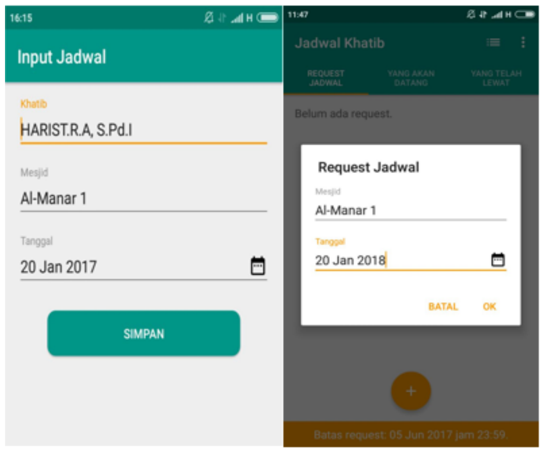


Figure 5: System Architecture



(a) (b)

Figure 6: (a) Mobile interface for timetable input form (b) Mobile interface for special requests

Nonetheless, System Administrator user considers that the redesigned timetable input form is better than the previous one. Figure 6.(a) represents a timetable input form in a mobile version. To select an Imam (khatib), the user can type one or two characters in the khatib text box and the system will provide the suggestion in alphabetic order. Moreover, we find another new problem during this second iteration phase. Logging in process, which requires users to pass their usernames and passwords, considered as a bit hard process since the users find it difficult to remember usernames and passwords. Unfortunately, we cannot omit this logging in process because of its urgency in identifying users and linking them with session and database.

4.3 Third Iteration – Final Prototype Refinement

Recall some problems discovered during the second iteration phase ranging from multiple submissions issue to username and password matter, the team must provide other ways to satisfy users. In order to solve the multiple submissions process either for Imam or System Administrator, we try to redesign and simplify the business process of proposing special requirement and timetable compilation. In the previous design, Imam or Mosque managers have to choose some fields provided by the system such as date and Mosque to visit. Each field has its own type and character and is stored into a timetable database. These fields relate each other by a series of composite keys to automatically detect timetable collision. However, this automatic collision detection leads to comparison process between fields causing the user to submit one requirement in one submission. It can be inferred that when a user proposes multiple requirements, he must commit multiple submissions.

After some deep discussion with users, we decided to eliminate automatic collision detection and let users propose their special requirements via a string or text box memo. This requires an additional table in the database to store request data. Users can simply type sentences like "Please schedule me only at Masjid Al-Azhar on first Friday of every month. Please do not assign me at Masjid Al-Furqon". Users might be able to send multiple requests by typing in the text field provided in just one submission. This affects a timetable compilation procedure at System Administrator side. Previously, special requests uploaded by users will automatically be included into timetable database if no collision occurred. Now, the System Administrator must look up them in request table.

D	E	F	G
AL-MANAR 3	AL-MANAR 4	AL-IKHLAS	NURUL IMAN
KOMARUDIN, S.Pd.I	AJANG. D	NAFIS THOHANI	DADANG NURDIN
PENDI, L.C	DADANG NURDIN	PENDI, L.C	AMAN
GUNGUN,S.Ag	H. AMAT. R	E. K. ISKANDAR	AEP S,S.Pd.I
ITA SASMITA	AEP S,S.Pd.I	Drs, RAGANIE E	H. AMAT. R
DONI C. A	H. SUHANA	FAHMI I. H, M. MPd	KOMARUDIN, S.Pd
HARIS R.A, S.Pd.I	AWAN. F	GUNGUN,S.Ag	UNDANG T. A
AGUS S, M.M,Pd	KOMARUDIN, S.Pd	H. AMAT. R	ARIS SY, S.S
FAHMI I. H, M. MPd	H. ABAS TOYYIB	AGUS S, M.M,Pd	Drs, RAGANIE E
UNDANG T. A	FAHMI I. H, M. MPd	GUNGUN,S.Ag	AGUS S, M.M,Pd
KOMARUDIN, S.Pd.I	AJANG. D	Drs,H. IWAN S	DADANG NURDIN

Figure 7: Special Programmed Excel Sheet

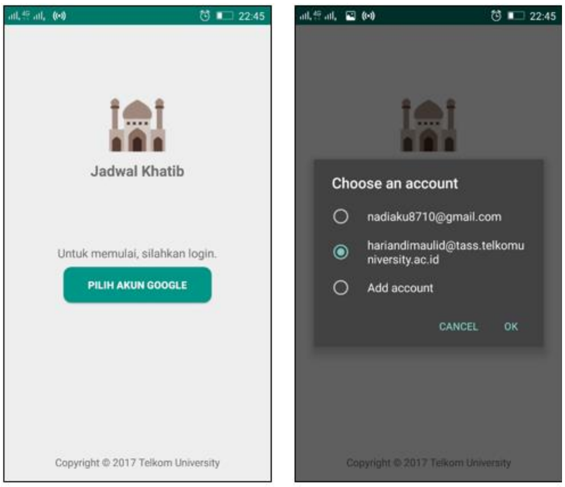


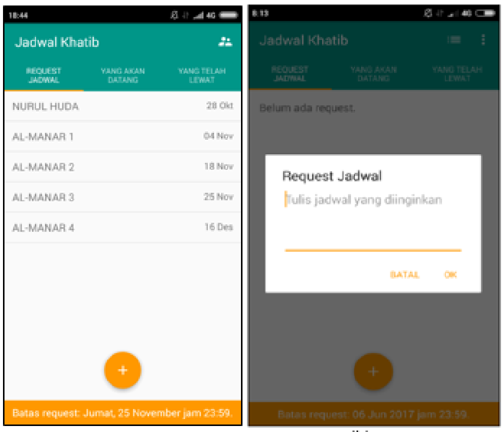
Figure 8: Logging in with email

Since automatic collision detection is omitted, the System Administrator must manually input these requests into timetable slots. Some may think that the new procedure does not solve the previous problem but increase the administrator load. However, this assumption is not true. We provide a special programmed excel sheet allowing an administrator to work offline. This excel sheet is supplemented by logical constraints and rules agreed by Majlis Tabligh and Mosque managers. Simply click the appropriate column, the administrator can arrange timetable easily. The sheet will automatically detect a collision when a rule or constraint is violated. In figure 7 the collision represented in red and orange color. Once a timetable compilation is complete, the administrator can upload the file into the server.

The last uncovered problem in second iteration phase is the login process. Users need a simpler and easier way instead of remembering username and password. We offer a solution to this problem by making use of users' email account as username as well as password. This inspired by Google Play Store in identifying users before allowing them to use application. As all user's cell phones run on Android Operating System, they at least have one email account inside it to activate Google Play Store access. Users can use this email and register it to the System Administrator to be inserted into users' record database. After registration, the user can pick that email upon logging into the system as seen in figure 8.

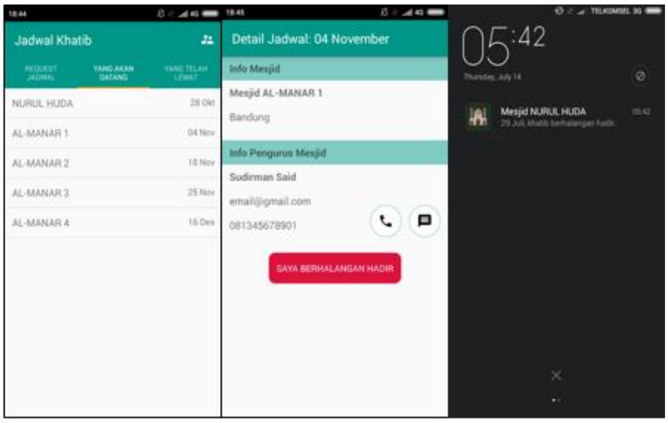
5. Usability Testing

The main usability goal of designing Jum'a preacher scheduling is Usefulness and Ease of Use (Al-Masslawi et al, 2017), (Holden and Karsh, 2010). To achieve the goals, we conduct a usability testing to measure a quality which represents the ease of use of the human-computer interface (Esteves and Pereira, 2015). Nielsen (1994), in Heuristic Evaluation: Usability Inspection Methods, stated that to identify the most important problem in usability design we only need fewer people rather than a big number of them. Accordingly, the test is conducted to 10 users which are divided into two groups in equal number. Each group comprises all types of user: 2 Imams, 2 Mosque managers, and a System Administrator.



(a) (b)

Figure 9: (a) Special request interface (b) Special request form, users can type and submit multiple requests in the text box.



(a) (b) (c)

Figure 10: (a) View upcoming timetable (b) Notification of absence (Imam view) (c) Notification of absence (admin view)

The test is conducted along with the third iteration phase. We put one observer in each group to record what the users do during the test. We created a test scenario in three parts. (1) *Prototype Exploration*: we ask users to explore the final high-fidelity prototype created in the third iteration in a 3-minute period. (2) *Tasks Completion*: We create different task completion scenario for each type of user according to their roles. For the Imams, we instruct them to log in, create special requirements, view schedules, and notify the administrator about his absence. For Mosque managers, we ask them to log in, create special requirements, view schedule of their Mosques, and contact Imam for schedule and task confirmation. For the last type of user, System Administrators, we give them a scenario in completing users account registration, timetable compilation, notification of absence. See figure 9 and figure

10 for detail high fidelity prototype. (3) *Interview and Questionnaire*: We conduct an interview and questionnaire to obtain user feedback about usefulness and ease of use of the targeted application.

We conduct the interview process in a 30 minute period for each group with 8 questionnaires to complete. This given questionnaire is formed in 7-point Likert scale, 7 for extremely likely while 1 for the opposite (Al-Masslawi et al, 2017). According to observers' report, users in each group do not have any difficulties while performing prototype exploration and task completion part. This because they have been involved in during the first iteration until the last phase. However, one of the observer reported that there are two out of five users, in group two, having difficulties in recognizing button or application sign. After discussion, we found that those users are new to the Android-based smartphone. In the final stage, we align the rsult of questionnaires with the feedback obtained from interview session. Users tend to consider that the usability goals are achieved. This indicated by the positive scores given by usefulness and ease of use. The first group contributes better interpretation in ease of use with 5,95 score. While they share equal average in usefulness as illustrated in Table 2.

Table 2: Usability Testing Results

		Mean	
Usefulness		**Group 1**	**Group 2**
1	It saves time	6	5,8
2	It increase productivity	5	5,4
3	It makes job easier	5,8	5,8
4	It is useful to the job	5,6	5,4
Ease of Use			
1	It is easy to complete tasks	6,2	6,4
2	It is easy to acquire skills	6	5,8
3	It is easy to learn	6,2	5,8
4	It is clear and understandable	5,4	4,6

6. Conclusion

The paper has discussed steps in designing a collaborative system for Jum'a preacher scheduling based on user-centered design approach. We have learned many things such as users' perspective and expectation as the iteration passed. Users play a major role in deciding user interfaces, procedures, and even the method. We changed the real-time conferencing collaborative method into non-real-time collaborative method based on user requests. This does not only lead to automatic collision detection removal but also interface and procedure refinement. Nevertheless, we admit that we missed medium fidelity prototyping during the prototyping phase. In the future, we would like to find out more about implementing real-time conferencing collaborative system for Jum'a preacher scheduling with a new algorithm for scheduling with constraint and more user-friendly interfaces.

7. Acknowledgements

This research is funded by The Government of Republic of Indonesia, through "Penelitian Produk Terapan" scheme in year 2017. We would like to thank the Government for financial support of the research and School of Applied Science of Telkom University which provides testing ground for the prototype implementation.

8. References

Abras C., Maloney-Krichmar D., and Preece J., (2004). User-centered design, Encyclopedia of Human Computer Interaction. Thousand Oaks, Sage Publication, 2004.

Al-Masslawi D., Handfield S., Fels S., Lea R., and Currie L. M., (2017). User-Centered Mapping of Nurses' Workaround to Design Principles for Interactive Systems in Home Wound Care.Proceeding IEEE International Conference on Healthcare Informatics, pp 314-323, 2017.

Anindhita V., Lestari D. P., (2016). Designing Interaction for Deaf Youths by using User Centered Design Approach. Proceeding of International Conference on Advanced Informatics: Concepts, Theory and Application (ICAICTA), IEEE Conference Publication, 2016.

Bafautsou G. and G. Mentzas, (2001). A Comparative Analysis of Web Based Collaborative System. Workshop on Web Based Collaboration, Munich, 3-7 September 2001.

Bureau of Democracy, Human Rights and Labor, (2016). International Religious Freedom Report for 2016, Indonesia. U.S. Department of State.

Esteves M. and Pereira A., (2015). Y.S.Y.D.- You Stay You Demand: User-Centered Design Approach for Mobile Hospitality Application. Proceeding of International Conference on Interactive Mobile Communication Technology andLearning (IMCL), pp 318-322, 2015.

Holden R. J. and Karsh B. -T., (2010). The Technology Acceptance Model: Its past and future in health care. Journal of Biomed Inform, vol. 43, no. 1, pp. 159-172, February, 2010.

Jankowski N., Schonijahn L., Kreitlow A., Gotze E., and Wahl M., (2017). A user-centered design approach in development of rehabilitation devicesafter stroke. Proceeding of InternationalConference on Rehabilitation Robotics (UCORR), pp 965-970, 2017.

Kamaluddin, (2016). Tata Kelola Mesjid Sebagai Pusat Dakwah dan Pemberdayaan Masyarakat Islam. Hikmah: Jurnal Ilmu Dakwah dan Komunikasi Islam, Vol.10 No. 1, 2016

Niesen, J., (1994). Heuristic Evaluation. In Nielsen, J., and Mack, R.L. (Eds). Usability Inspection Methods.John Willwy & Son, New York, NY.

Padela A. I., Killawi A., Heisler M., Demonner S., and Fetters M. D., (2011). The role of Imams in American Muslim Health: Perspectives of Muslim Community Leaders in Southeast Michigan. Journal of Religion and Health, vol. 50, issue 2, pp 359-373, June 2011.

Smaradotir B. F., Martinez S., Rabbersvik E. H., and Fensli R., (2015). eHealth-extended Care Coordination: Development of a Collaborative System for Inter-municipal Dementia Teams. Proceeding of International Conference on Computational Science and Compiutational Intelligence (ICSCI), pp 749-753, IEEE Conference Publication, 2015.

Smaradotir B. F., (2016). The Steps of User Centered Design in Health Information Technology Development: Recommendations from a PhD Research Study. Proceeding of International Conference on Computational Science and Compiutational Intelligence (ICSCI), pp 116-121, IEEE Conference Publication, 2016.

Sofwan, R., (2013). Penguatan Manajemen Pemberdayaan Fungsi MasjidAl-Fattah di Kelurahan Krapyak Semarang. Jurnal Pemikiran Agama untukPemberdayaan, Vo.13 No. 2. Universitas Islam Negeri Walisongo.

Tolone W., Ahn G. J., Pai T., and Hong S. P., (2005). Access Control In Collaborative Systems. ACM Computing Surveys (CSUR), 37, 1 March 2005.

Wirawan, W., (2017). Manajemen Operasional Khotbah Jum;at Masjid Nasional Al-Akhbar Surabaya. Available online at digilib.uinsby.ac.id

Yao L., Sheng Q. Z., Ngu A. H. H., Yu J., Segev A., (2015). Unified Collaborative and Content-Based Web Service Recommendation. IEEE Transaction on Services Computing, vol. 8, issue 3, pp 453-466, IEEE Journals & Magazines, 2015.
Zhang X., Nakae M., Covington M. J., and Sandhu R., (2008). Towards a Usage-Based Security Framework for Collaborative Computing Systems. ACM Transaction, 2008

Biodata

	Hariandi Maulid was obtained his first degree in Informatics Engineering from STMIK Jabar, Bandung in 2003. He further his study to University of Bristol, UK and accomplished his Master of Science on Internet Technology with Security at Department of Computer Science, Faculty of Engineering in 2011. Currently he serves as a senior lecturer at Informatics Engineering Study Program, School of Applied Science, Telkom University.
	Indra Azimi graduated from Informatics Engineering Study Program, Tanjungpura University in 2010. He obtained his master degree in informatics from Bandung Institute of Technology in 2014. Currently he is a full-time lecturer at Informatics Engineering Study Program, School of Applied Science,Telkom University.
	Amir Hasanudin Fauzi graduated from Electronic Engineering Department, Mataram University in 2011. He obtained his master degree in Game and Digital Media Engineering from Bandung Institute of Technology in 2013. Currently he is a full-time lecturer at Informatics Engineering Study Program, School of Applied Science,Telkom University.