KnE Social Sciences

3rd ICTVET 2018
3rd UNJ International Conference on
Technical and Vocational Education and Training 2018
Volume 2019

Conference Paper

Design and Development of Student Admission System in the State University of Jakarta

Hanifa Fissalma and Hamidillah Ajie

Technical Implementation Unit of Information and Communication Technology, State University of Jakarta, Jakarta 13220, Indonesia

Abstract

Student admission system is a must in every university, and so in the State University of Jakarta. Web applications are more widely used in the development of information systems because of the ease in distributing information to many users. By using the concept of re-used oriented, student admission system built in 2015 is developing by adding, editing, and deleting a requirement based on policy that enforce today. Moreover, adding web service as an intermediary between system and database as an extra feature in new system. The research was conducted using the Feature Driven Development (FDD) which is one of the Agile Method that does not have fixed rule in traceability model between requirement and feature, where this research went through several stages, namely needs to develop an overall model, build a feature list, plan by feature, design and build by feature. This study produces several products, there are Software Requirement Specification (SRS) and system that are built with React-Redux for its front-end and Laravel as its back-end.

Keywords: design and development, requirement, re-used oriented, feature driven development, web service

Corresponding Author:
Hanifa Fissalma
hanifa@unj.ac.id

Received: 11 January 2019
Accepted: 14 February 2019
Published: 25 March 2019

Publishing services provided by Knowledge E

Selection and Peer-review under the responsibility of the 3rd ICTVET 2018 Conference Committee.

1. Introduction

Student admission system is a mechanism for selecting a new student in higher education, including at State University of Jakarta. The mechanism of student admission system in State University of Jakarta includes a sequence of processes starting from registration, determination of tuition fees, until students who are accepted to pay the first tuition that has been determined. Student admission system is expected to be able to find out the number and distribution of applicants for new student registrants, which will have an impact on university accreditation and the increase of new students who have good quality [1]. So far, State University of Jakarta has an information about each admission. So that there is an inequality data between one department to another for total number of new student. In addition, there is still little information presented to

How to cite this article: Hanifa Fissalma and Hamidillah Ajie, (2019), “Design and Development of Student Admission System in the State University of Jakarta” in *3rd UNJ International Conference on Technical and Vocational Education and Training 2018*, KnE Social Science, pages 92–101. DOI 10.18502/kss.v3i12.4076

prospective students who are accepted at State University of Jakarta regarding campus life, study fees, scholarships and so on.

One of the information systems that must be well designed for State University of Jakarta is student admission system. State University of Jakarta has started PENMABA since 2012 but until now it has always undergone changes and improvements from year to year, both due to changes in regulations and changes in system requirements. Based on interviews with head of Student Admission System, Mr. Ifan Iskandar, the obstacles faced by Student Admission System 2015 version are:

1. There is still a peak load to access student admissions, at the time after the announcement of the SBMPTN and at the end of the closing of the registration PENMABA, which resulted in the system being difficult to access.
2. There is no system that integrates student admissions for the entire entrance.

In addition to technical problems, there are other obstacles including:

1. There is a change of policy from year to year regarding the Student Admissions System
2. Slow distribution of information from higher level of management to the development team which results in a delay in the system development work schedule
3. So far building an information system at State University of Jakarta has not begun with analysis and design, but directly into implementation. So that the requirements often missed and the bug fixes are hampered.

2. Literature Review

2.1. Reuse oriented software engineering

Software reuse is the process of implementing or updating a software system using existing software components. The process increased productivity, quality, and reliability, and decreases costs and implementation time. In short, the development of reuse processes and repositories results in a knowledge base that improves quality after reuse, minimizes the amount of development work needed for future projects, and ultimately reduces the risk of new projects based on repository knowledge [2]. In most software projects, there are several software reuses. This often happens when people who work on projects know that a design or code is similar to what is needed.

They search, change as needed, and combine them into their systems [6]. The type of Software Reuse are:

1. Application System Reuse, recombines all applications by combining one application into another (COTS reuse) or creating an application family such as MS Office.
2. Component Reuse, the component (subsystem) of one application is reused into another application.
3. Function Reuse, reuse software components that implement a well-defined function.

2.2. Feature driven development (FDD)

Feature Driven Development (FDD) is a process that is designed and implemented to present the results of work repeatedly in a certain time and can be measured. FDD is an approach that refers to making systems using methods that are easy to understand and implement, problem solving techniques, and reporting that are easily understood and controlled by stakeholders [5].

An important element in FDD is the feature. Palmer and Felsing (2002) define features as: **"<action> <result> <object>"** where <**object**> can refer to a person, place or thing. FDD focuses on continuous processes, integration and small releases, which reduce conflicting changes and shorten the feedback cycle by providing fast feedback and improvements.

FDD consists of five series of processes in designing and building a system. The iterative part of the FDD process (Design by Feature and Build by Feature) supports the development of Agile with a fast adaptation process to change requirements and business needs.

The following of FDD processes according to Palmer and Felsing (2002):

1. Develop an Overall Model

   In this phase the overall problem domain description is compiled. After gaining an understanding of the functions of business needs, domain expertise, and the overall scope of the project, the Domain Expert starts the entire design model. Documented requirements are governed by appropriate cases and technical specifications.

2. Build a Feature List

Feature list is what the client sees for the validity and completeness of the system. The language used is as simple as possible so that the client understands. In the next stage after determining the entire system set, developers must now identify what features can be made in the list of each module produced. The following pattern defines a feature according to Coad (1999):

<action> the <result> <by | for | of | to> a (n) <object>

Where an <**object**> is a person, place, or something.

3. Plan by Feature

   This phase is an initial activity that produces a development (Design and Build) High-Level Plan. This High-Level Plan contains all features, based on their priorities and dependencies.

   The Project Manager, Development Manager, Chief Programmer and other Actors act together to prepare sequential lists of all the features listed in phase 2, and sort features based on their priorities. Identifying interdependencies between features before implementation is an important part of this phase.

4. Design by Feature and Build by Feature

   After getting the feature set listed with priority, the Class Owner helps form their feature team.

   Each iteration is generally scheduled for 2 days to 2 weeks. In this phase, the system goes through sequential processes of product development: Designing, Development, Testing, Integration and Testing Systems. After a successful iteration, the finished feature is driven to the main build, while the next design and build iteration starts with a new set of features.

3. Result

3.1. Specification and analysis of old system

Actors in the old system are as follows:

The next step is to find a use case. Use cases can be seen through the old system source code and in the Software Requirement Specification (SRS). The list of old system use cases are:

In the old system database, there are 24 tables. Based on the existing source code, from all tables in the old system database, there are 8 tables that are not used. Based

DOI 10.18502/kss.v3i12.4076

Table 1: (a) List of Old System's Actor.

No.	Actor	Description
1	Public User	A visitor who accesses the system without logging in first. This visitor has access to view information about Penmaba, a list of study programs, a list of locations and so on.
2	Candidate	Users registered with the User level as registrants but have not made the payment process.
3	Participant	Users registered with the User level as registrants and have made payment processing.
4	Super Admin	Users registered with User level as super admin, who have the authority to add User, location, room and monitor the results of the monitoring panel.
5	Helpdesk	Users registered with the User level as helpdesk, who have the authority to view location, room and monitor the results of the monitoring panel
6	Chancellor and Student and General Unit (BAKHUM)	Users registered with the User level as chancellor and BAKHUM, who have the authority to monitor the results panel

Table 2: (b) List of Old System's Use Case.

No.	Use Case	Description
1	View Information	A process for viewing an information about admission system
2	Register	A Candidate who will do the admission process. First, need to register by username to go to the next step
3	Login	Process to enter username and password for all actor, exclude public user
4	Choose a major	Process after being registered as User, the Candidate is required to choose the type of exam and the destination study program to be processed how much the costs must be incurred
5	Print an invoice	Process after selecting the type of exam and the study program, the Candidate prints the bill to be paid to the Bank.
6	Upload a photo	Process of uploading photos whose terms and conditions are adjusted
7	Fill personal form	Process to fill in the participant's biodata that will be used to print participant cards.
8	Print a participant card	Process after filling in personal form, participants can print the participant card and cannot replace entries
9	Monitoring panel	Process used by admin in the admin panel to monitor recapitulation.
10	Upload room and location	Process used by admin to add room and location if the existing room and location are exhausted or nearing an end.
11	Add user	Process used by admin to add users with a certain user level.
12	Print album and presence	process to print an album when one room is fully charged.

on interviews with interviewees and seeing from old system source code, there were several problems experienced when the old system took place, including:

1. There is a tendency to create multiple users.
2. Many of tables in the database that are not used.

3. Chancellor and BAKHUM actors have never used their accounts, relying only on printed reports obtained from super admin or helpdesk.

3.2. Modification system

Modification is done by adding, changing or removing the requirements, both functional requirements and non-functional requirements to improve the quality and fulfillment of functional requirements. List of system modifications as follows:

3.3. Designing new system

3.3.1. Determination of main feature

The first stage in making a requirement list is determining the main features. The main features between the old and new system have not changed much, only added recapitulation of the remaining room and seats, recapitulation of those who have already paid and recapitulation of skills tests that have been paid, while announcement management is omitted due to the system being unable to use in old system. The main features at new system of Penmaba Mandiri are:

3.3.2. Functional requirement and functional area list

The next stage is to make a list of functional requirements. Requirements in the old system are developed with added, modified or deleted requirements based on the current policies. Then, functional requirements are collected based on the level of the user (Registrants, Participant, Admin, Helpdesk, Committee) and grouped based on the similarity of features and dimensions called functional area. After that, coding and labeling are given on the functional requirements list starting from RPM001. RPM which means ”Requirements Penmaba Mandiri” and 001 is freely chosen.

3.3.3. Generalization

Generalization is a stage of functional requirement efficiency in order to divide the work area domain. Members of the developer team will work on the project based on the division of the area domain that has been received. Area domains are generalizations of requirements and domain as a whole.

TABLE 3: (c) List of Modification System.

No.	Feature Name	Action	Reason
1	Developing PENMABA portal	Adding system	There is a need to accommodate students who are accepted from all entrance, not only those from Penmaba. Because students who are accepted through the SNMPTN and SBMPTN still use manual methods to collect data
2	Developing web service	Adding feature	Making web services is done so that later all functions needed by new system can be taken from the web service. In addition, following the development of the era that currently uses a lot of web services
3	Actor changes on the Admin system, both in the PENMABA portal and in new system	Changing subject	Initially the user admin numbered 5 actors, namely: Super admin, Helpdesk, BAKHUM, and Chancellor. Changed to only 3 actors, namely Super Admin, Helpdesk and Committee. This is because the BAKHUM and Chancellor actors have never accessed the system and only asked for a printed recapitulation to the super admin or helpdesk.
4	Adding recapitulation per type of exam	Adding feature	There is a need to know the number of participants based on the type of exam
5	Adding recapitulation of remaining room and seat	Adding feature	There is a need to know the remaining room and seat used by the exam
6	Adding recapitulation of skill test	Adding feature	There is a need to know the number of participants who take the skills test as well as who have paid for the skills test
7	Transaction log	Adding feature	There is a need to know the activities carried out by the admin so that if there is a change in data, it can be known who is responsible for changing the data
8	Graphic	Adding feature	There is a need to see a graphic of the number of participants in general based on time
9	Non-functional requirement	Detailed explanation	1. The new system uses bootstrap and build in php, minimizing design by referring to the flow of registration by maximizing JavaScript to access web services to minimize taking overall data. 2. The system is displayed in Bahasa Indonesia 3. The system must be available 24 hours and 360 days a year for registration needs 4. The system cannot lose data in the database 5. Can support 1000 activities every day without interruption 6. The login process is only in less than 5 seconds

3.3.4. Build a feature list

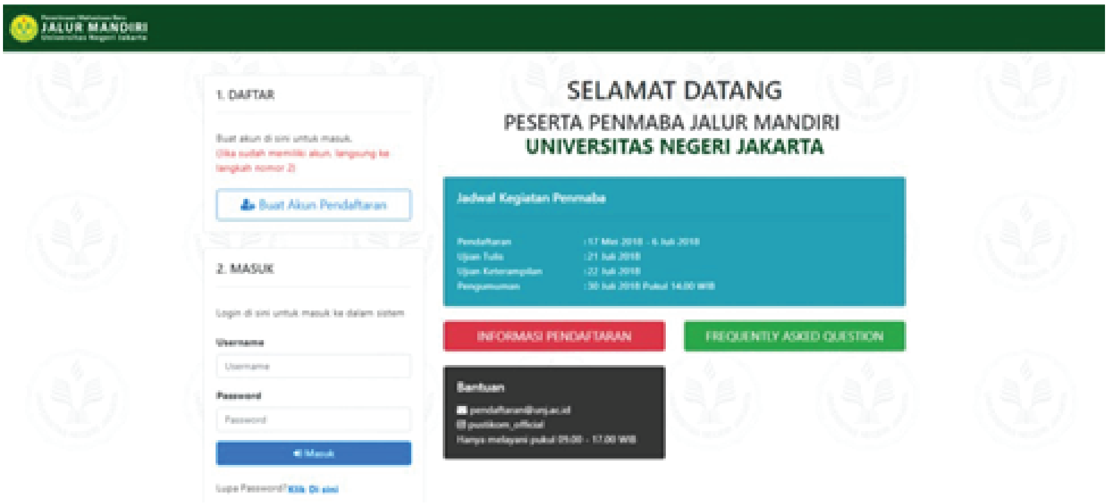
After creating tables resulting from generalization requirements, then the requirements are made into feature sentences. The following is a partial table of feature rule changes based on generalizing requirements:

3.3.5. Build a feature set

To facilitate tracking the process between features, a coding is given for each feature and also includes the active subject or actor who uses the function. Checklist marks indicate the relationship between features and subject. For more details, there are partial results from the following set features:

3.4. Development system

After requirements is made based on FDD rules, the last step is to build the system. Based on the modifications that must be made, the display results of the student admission system at State University of Jakarta are as follows:



SELAMAT DATANG

PESERTA PENMABA JALUR MANDIRI

UNIVERSITAS NEGERI JAKARTA

1. DAFTAR

Buat akun di sini untuk masuk.
(Jika sudah memiliki akun, langsung ke langkah nomor 2)

2. MASUK

Login di sini untuk masuk ke dalam sistem

Username: Username

Password: Password

Lupa Password? Klik Di sini

Jadwal Kegiatan Penmaba

Pendaftaran	: 17 Mei 2018 - 6 Juli 2018
Ujian Tulis	: 21 Juli 2018
Ujian Keterampilan	: 22 Juli 2018
Pengumuman	: 30 Juli 2018 Pukul 14.00 WIB

Bantuan

pendaftaran@unj.ac.id
pustikom_official
Hanya melayani pukul 09.00 - 17.00 WIB

Figure 1: (a) Example of Interface of Student Admission System at State University of Jakarta.

4. Discussion

After analyzing Penmaba 2015 version system and designing it on 2018 version, can see a differences in total of functional requirement. In the Penmaba 2015 version there are 48 user functional requirements and 77 admin functional requirements. With five actors, namely Admin, helpdesk, vice of rector, BAKHUM, registrants and participants, also added web service features and several features additional in the requirements of 2018 version.

Redesigned the database to tidy up relationship between data, addition of reference tables and addition of tables due to additions feature. After it was designed, it became 24 tables in 2018 version. In 2018 version design there are additional features, namely recapitulation of remaining location, recapitulation for skills test, and transaction log. In addition, there are changes in the actor, in 2015 version there are six actors, there are, participants, admin, helpdesk, vice of rector, and BAKHUM, while in 2018 version becomes only three actors there are participants, admin, and helpdesk. Vice of rector and BAKHUM were removed because in 2015 version, they are never opened the system and only rely on printed recapitulation provided by the admin or helpdesk. In addition, the addition of web services on this system to reduce load on the system itself. So when the system is accessed by many users it does not reduce the performance of the system itself.

5. Conclusion

Reuse-oriented can be applied in the development of an information system and be able to shorten the time of system development, and the application one of the Agile Methods, Feature Driven Development (FDD) can also be done with the aim of simplifying searching if there is an error in testing the final application and can be used as reference to further project development at State University of Jakarta. In the future, this system will through usability testing to measure the effectiveness of the user interface based on input and complaints from users in 2018

Funding

This work was supported by the admission of new student office (Panitia Seleksi PENMABA) and the vice of rector 1 for students affair.

Acknowledgment

The authors would like to thank Mr. Muchlis R.Luddin as vice of rector 1, Mr. Ifan Iskandar as the head of admission of new student and Mr. Agung Premono as the secretary of admission of new student for their comment and suggest regarding development of new system. We also would like to thank all the member of Technical Implementation Unit of Information and Communication Technology for their support and advice about this research

Conflict of Interest

The authors have no conflict of interest to declare.

References

[1] 2015. Software Requirement Specification Situs Penmaba Universitas Negeri Jakarta tahun 2015, Pustikom Universitas Negeri Jakarta.

[2] Ravi A. & Nirmala, D. R. K. (2015). Software Reusability: A Framework Using Software Components and Reusable Assets. [Proceeding] *Journal of Theoretical and Applied Information Technology*, vol. 72, no. 3, pp. 431.

[3] Dennis, A., Wixom, B. H., & Tegarden, D. (2009). *System Analysis Design UML Version 2.0 3rd Edition*. New York: John Wiley & Sons, Inc.

[4] Kendall J.E. & Kendall K.E. (2011). *System Analyst and Design 8th Edition*. New Jersey: Prentice Hall.

[5] Palmer, S.R. & Felsing, J. M. (2002). *A Practical Guide to Feature-Driven Development*. Upper Saddle River, New Jersey: Prentice-Hall.

[6] Sommerville, I. (2009). *Software Engineering 9th Edition*. Boston: Addison-Wesley.

[7] Tripathy, P. & Naik, K. 2014. *Software Evolution and Maintenance: A Practitioner's Approach.* New York: John Wiley &Sons, Inc.