CHAPTER 1

Introduction

1.1 Overview

Students enrolled in Bachelor of Engineering and Bachelor of Science Honors are required to undertake Thesis Project before being eligible to graduate. Undergraduate thesis is commonly undertaken during the last 2 sessions of the program. During the thesis project, students are working on the research and development under the supervision of the academic staffs.

Being successfully allocated to the topic that best matches the students' interests is the key towards successful accomplishment of a thesis project. It in turn leads to the benefits of academic staffs whose project is performed by their students. There have been systems built to accommodate the needs of allocation of thesis topics to the students. These systems vary in functionalities from the one that only provides the mechanism of searching thesis topics on the website up to those that support online nominations of topics.

1.2 Problem Definition

Publishing topics online has been a classic mechanism to provide quick access to the available topics offered by supervisors. Ideally, the interface should be built to accommodate users with abilities to find thesis topics. However, many of the website designs ends up with a lot of usability problems that exhibit remarkable dissatisfaction of users.

Having access to the list of topics alone is not enough. Most of the tedious activities remain on the process of matching students with the thesis topic. The

common approach to this is by having students directly contact the academics staff to get permission of a thesis topic and obtain their signature on what so called *thesis nomination form*. However, the use of paper based forms as a means of nominating thesis topics are felt obsolete.

Two new ways of nominating thesis topic had been proposed by the Thesis Coordinator, Albert Nymeyer, namely, *preapproved nomination* and *preference nomination*. Both of them eliminate the use of paper based form as a way of nominating topics and they are currently being used as a mean of allocating students to topics at CSE. Although this appears as a great advance in topic allocation, both of the protocols are complicated and require undergraduate teams' involvement to handle the nomination process manually.

1.3 Objectives

The aim of this thesis is to develop a thesis management system that handles the management of thesis topics and provides mechanisms of handling the process of getting students allocated to the thesis topics. Several other key issues, such as performance, scalability, accessibility, security and usability will also be taken as a consideration in design and construction of this new system.

In order to achieve these objectives, we will look into the existing solutions preferably those that provide the similar functionalities. We will also rely on the end users' view and perspectives to the current system. This will put a great impact on of how this new system should be evolved and how the process could be improved to achieve better user satisfaction. While this approach is being performed, the new system should not deviate greatly from the basic requirements.

1.4 Problem Scope

The scope of my project is to design the user interfaces to provide user with ease of use related to the early stages of thesis activities, namely publishing thesis topic on the website, finding thesis topics, nominating topics and allocating topic to students. The development of user interfaces focuses on how the system facilitates the actions that user need to perform. User centered design was carried out to meet the user preferences and requirements of the system interface. Implementing the 2 protocols for nominating topics is also the main course of this project. The model of the new protocols is aimed for simplicity, understandability and learnability for the users of the system.

1.5 Organization of Thesis

There are 7 chapters in this thesis.

Chapter 1, *Introduction*

Introduces the overview of the problem and the description of the tasks that need to be undertaken in this thesis.

Chapter 2, *Background*

Discusses some of the existing solutions that try to address the problem that is described on Chapter 1. We also discuss the advantages and disadvantages of each solution in order to find out possible import on the features and functionalities while also tackling their weaknesses. We also present the framework of evaluating good solutions in order to have a reference to the design decision made at the development as well as a measurement for assessing our final solution. At the end of the chapter, we will be presenting the high level overview of the solution.

Chapter 3, *Requirements*

Discusses the functionality and features that is expected from the final product of this thesis. These are documented in the form of requirement

document that reflects the problem domain that we are trying to solve.

Chapter 4, *Use Case Specification*

This chapter relates to the functional requirements that is discussed on Chapter 2, and presents them in the form of use case specification. We show the principal actors of the system as well as the interaction between them and use cases. Our discussion is supported by the scenario of each execution of use case.

Chapter 5, *Design*

Presents the entity relationship database modeling for our system and the transformation of that logical design into a more implementation oriented relational schema. We also show the top-down functional decomposition of the system based on the flow of data using DFDs.

Chapter 6, *Implementation*

Discusses the three tier architecture that is used on our application. We also present the overview of the system's user interface in the form of interactive tour and provide description of GUI components.

Chapter 7, *Evaluation*

Evaluates the final product based on the framework to evaluate good solutions that is discussed in Chapter 2. We will inspect whether the final product has delivered the functionalities on the specified requirements and whether it satisfies the expectation and satisfaction of the end users.

Chapter 8, *Conclusion*

Concludes the overall thesis and presents the suggestions of the further improvements that can be made on the system.

CHAPTER 2

Background

2.1 Overview of Related Work

There are currently systems built to support thesis management process. We will look into a couple of systems which offers similar functionalities with our thesis management system. Here we will be analyzing each of them and consider possible import of features to maximize the overall outcome of this project.

2.1.1 CSE Thesis Management System

The old thesis management system used in CSE prior to the year 2004 is the one that was installed on the Disney server. It provides basic functionalities of browsing and searching for thesis topics. At that time, paper based forms were the only solution for students to nominate thesis topics. To get allocated to the topic, students should talk to the supervisor either directly or via email. Once the supervisor agrees to supervise the student, the student must fill in a thesis nomination form that should be signed by the supervisor and assessor. The nomination form is finally submitted to the school office and the information about topic allocation is entered to the database.

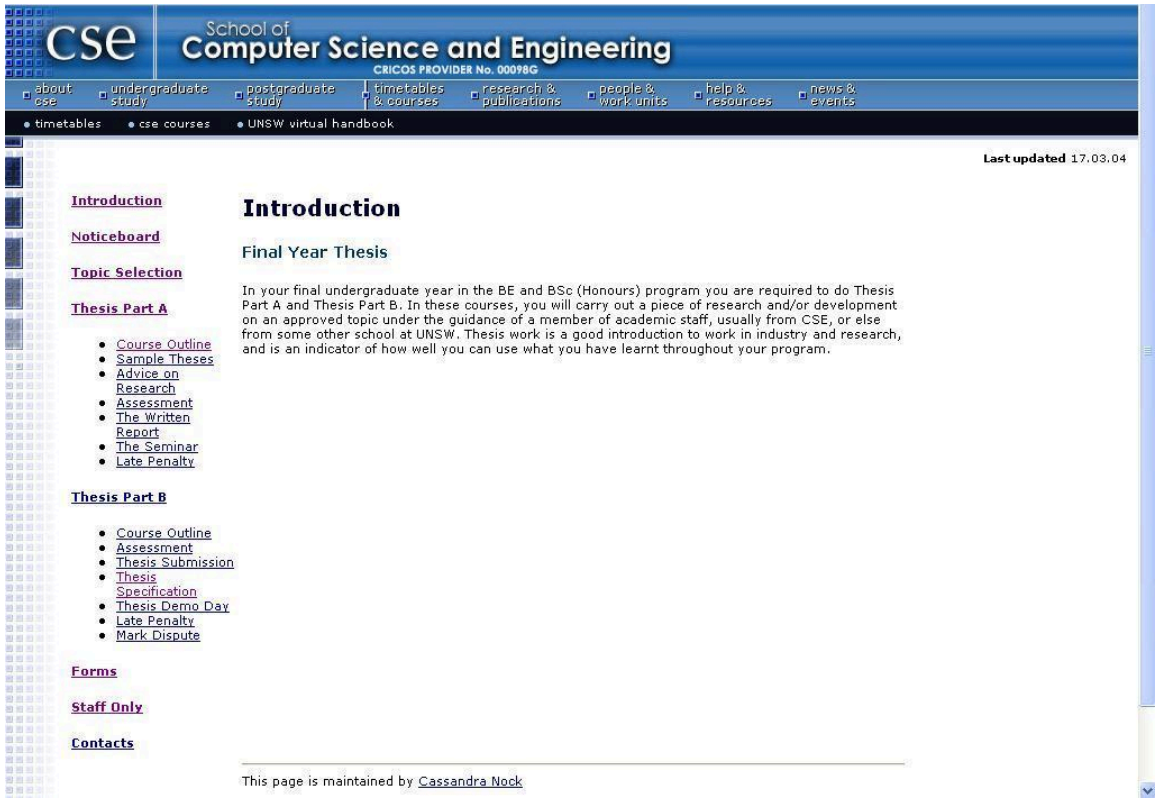


Figure 2.1. CSE Thesis Management System

At the beginning of 2004, a new system was developed by Maria Tzortzis. There have been significant improvements particularly the process of topic nomination itself. With the current system, students can nominate topic via online, i.e. no more paper based forms. The topic nominations are currently done via 2 ways:

- *Preapproved Nomination* in which students approach academic staffs directly to request a thesis topic. Once an academic staff agrees to supervise the student, the student must visit the preapproved nomination form page and enter the approved thesis topic number. Once the form is submitted, an email is sent to the supervisor validating whether the supervisor indeed has approved the students. The supervisor should reply with another mail stating "YES" in the subject header to undergraduate teams. If the supervisor acknowledges the email, one of the undergraduate team will enter the information about topic allocation to the database. If the supervisor denies the approval request or a week has elapsed since the time validation email was

sent, the student will be notified about the failure of the nomination and asked to submit another one.

- *Preference nomination.* In this process, students submit nomination listing 5 thesis topics in the order of preference. From that submission, undergraduate teams will handles the requests by sending email messages to the academic staffs who supervise the topics. There is no insurance that the requests be sent in the same order with reference to the position of the topic listed in nomination form. This is at first performed to provide load balancing among supervisors. However, in session 2 2004, the system changes its procedure to sending the request in exactly the same order as listed by the student. In particular, the request to the first preference topic is sent to the supervisor. If it is approved by the supervisor, the student will be allocated to that topic. If otherwise the supervisor rejects the request or there is no response from the supervisor for a certain period, the second request will be send to next supervisor. This process will continue down the list of the topics until one of the requests is approved by the supervisor. If none of the academic staffs accept the requests, the student will get a notification informing the situation and asked to submit a new nomination.

The good thing about CSE thesis management System is that there is a clear classification for topic types. This is important to notify the students about the type of tasks that would be undertaken and also as a basic guideline for assessment purposes. Also, a new topic type called COG (Course Oriented Group) was also introduced to the system. COG topics are those which are strongly related to the CSE courses and prioritized for pass-level students.

However, the system poses several problems. The problems with the current systems can be classified into 2 groups: usability problems and protocol problems. These problems are based on my own experience and hypothesis that is further supported by the survey that can be found on appendix A. Among the

usability problems are:

- *The long topic list overwhelms the users*. The excessive amount of topics that is displayed on one page will makes the users difficult to scan the topic list. In turn, users will be forced to scroll along the page. Besides, a page that contains so much information will reduce the response time for page download. Based on my experience the current page that consists of 855 topics takes 10 seconds on average to load from a remote network location with an ADSL connection.
- *Students are not familiar with staff initials*. Based on the results of the interview, 65% of students have difficulties on decoding the meaning of the Staff Initials at the first time. This is because Staff initials are derived from the concatenation of the first characters of each words of the staff's full name to avoid collisions between staff initials. However, students typically only know the staff's first name and last name. This lead to confusion when users try to deduce the staff's real name from such initials. For example, students will most probably recognize RB as Richard Buckland instead of Roksana Boreli without looking into the staff initials page. The fact that search form also uses initials to find the topics supervised by a staff makes it worse.
- *Poor search functionalities*. The only search option that the current system provides is a type of “fielded search” that search on keywords appearing specific field. This is good when the ability to conduct more specific searching is desired. However, users searching capabilities becomes very limited. For example, students are unable to search for topics that have the keywords “database” on its topic title and supervised by “John Shepherd”. Moreover, students can only search topics based on the fact that certain keywords appear such the fields. A search query “graphics” will certainly miss “3D Modelling”, although “3D Modelling” have a strong relationship with “graphics”. Moreover, the search and browse pages that live in isolation from each other make students' life difficult to switch between tasks. Many of the students suggested an improvement on the search functionality of the current

system.

- *Poor information presentation*. Based on my hypothesis students can't deduct the actual process of preference nomination given a limited information that is presented on the website. It is further revealed on my survey that 65% of students indeed have difficulties on understanding the nomination process.

Among the Protocol problems are:

- *Problems with login account*. One of the most critical security problems with the current system is the missing login account for students. This is considerably dangerous, since a student can nominate topic for the others. The problem is minimized by sending an email notification confirming nominated topic. However, there is no assurance that the emails are getting read by the students. Additionally, even though there are login accounts provided for staffs, one could easily login as the others by following this steps: login as the user "John Shepherd", then modify the URL argument Initials from "jas" to "aa", and you have successfully logged in as "Adnan Admin".
- *Too much admin involvement for handling nomination requests.* The current system requires a great deal of undergraduate teams' involvement during the nomination process. Humans are error prune, slow, suffers from stress and fatigue, especially when they are dealing with a lot of tasks. Indeed, a considerable amount of rated the current system as not sufficiently fast to support the nomination process. It would be good if some kind of automation is applied on the system as many respondents in the survey have suggested. Another comment by John Shepherd is that admin shouldn't interfere in the preapproved nomination process once a student gets approval from a supervisor.
- *Protocols are difficult to learn*. The current protocols require a considerable amount of learning effort by admin since the nomination process done manually. I also found that it is moderately hard to understand the concept of the protocols at the first time, especially the preapproved nomination protocol.

2.1.2 EE&T Thesis Management System

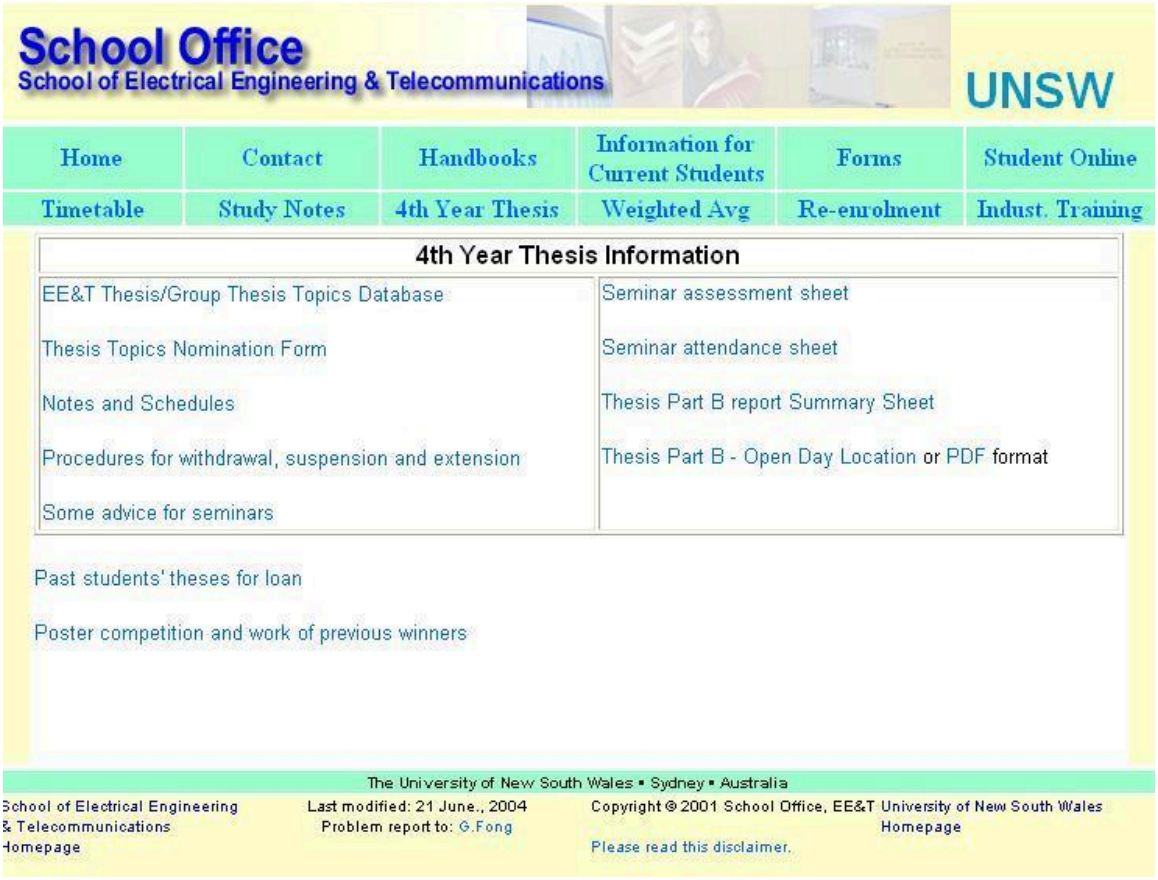


Figure 2.2. EE&T Thesis Management System

The thesis management system that is used on EE&T is similar to the old system that is used on CSE. It provides searching and browsing system that is similar to that of the current thesis management in CSE. The list of topics is also displayed in only single page which poses the same problem of overwhelming. The system uses exactly the same type of search engine as the one that is currently used on CSE, i.e. “fielded search”. Additionally, the system also makes use of Staff Initials as a mean of identifying thesis topics. However, the system provides a search based on the real name of the Staff (Toan Phung instead of TP) which is considered as more useful and meaningful for end users.

Despite of the similarity with the new system used in CSE, the nomination process is conducted in the similar way as the old one. In particular, after student get approval from supervisor, they can download the nomination form from the website and get it signed by supervisor. The form is then submitted to school office and one of the administrators will enter the information about the topic allocation in the database. At the end, a photocopy of the nomination form will be sent to the supervisor for record keeping.

2.1.3 Thesis Management System by Claudine Halim

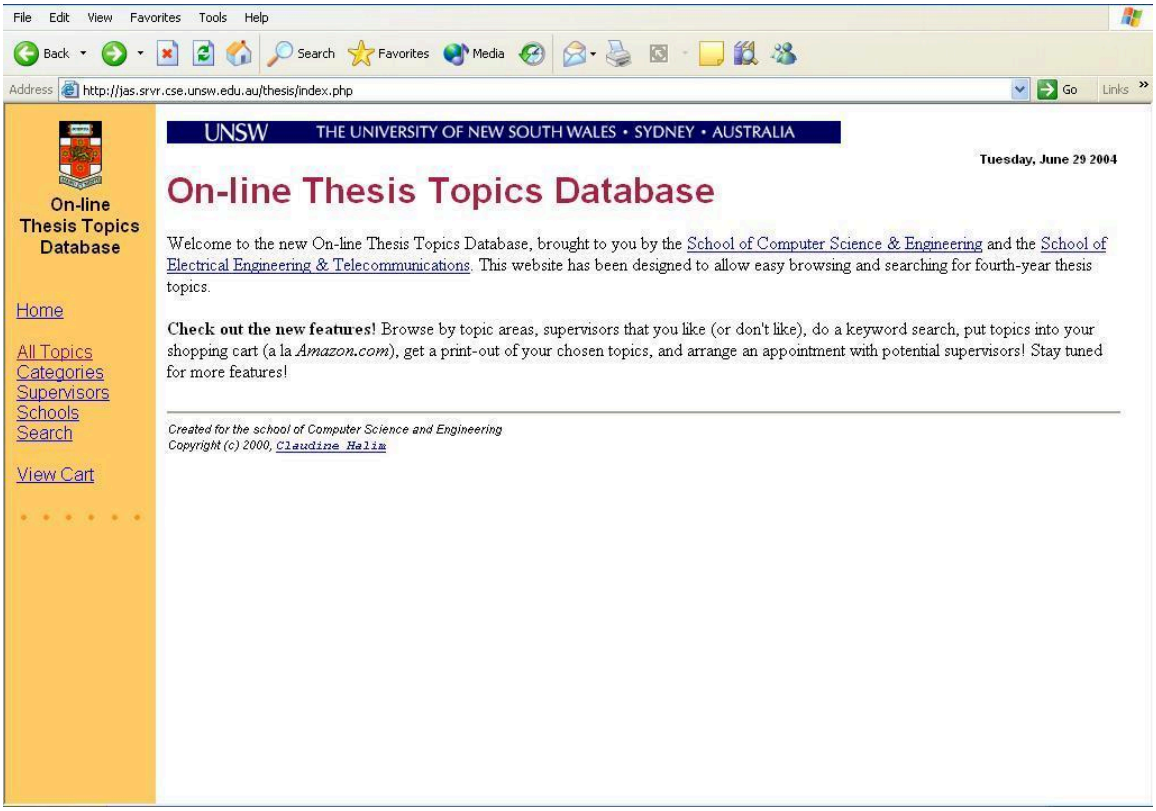


Figure 2.2. Thesis Mangagement System by Claudine Halim

There is already an attempt that is made by a thesis student previously. The system exhibits much better usability and performance than the other 2 system discussed previously. The page containing the list of available topics used page

chunking technique such that only 20 topics are shown in each page. An indexing system of thesis topics based on supervisor names and topic categories has been used to support various mode of searching topics. The system also provides advanced search which includes the ability to formulate complex queries based on multiple criteria, such as search for topics supervised by “John Shepherd” whose title field contains keyword “database”. Additionally, a simple keyword search on thesis topics is also provided.

However, none of the topic nomination process is supported by the system. Instead, the system provides assistance to users with capabilities to put the thesis topics of interests in the shopping cart while browsing the topic list and to obtain the printable version of the topics on the shopping cart. This feature was intended to help user with tedious tasks of memorizing or taking a note of the thesis topics that are to be requested to the supervisors.

2.2 Framework of Evaluating Good Solutions

We should first identify what are the characteristics of a good solution before any attempt is made on developing a solution. Once the characteristics are defined, they will be used as the source of the rationale behind the design decisions throughout the development of the solution.

2.2.1 Functionality

Functionality is an attribute that asses the existence of functions and features that is supported by the software. In particular, users should be able to perform critical tasks and actions given the set of features of the system. This will affect how the users perceive the final product as satisfying their needs and expectations. The desired functionalities should be defined unambiguously in a requirement document to which all the development process will be based on.

2.2.2 Usability

Usability is an attribute that assess how easy a users perform tasks given the user interface of the system. The supported functionality should be understandable and learnable from the point of view of new users who are using the system at the first time. In particular, we shouldn't put reliance on the learning effort of users to use the system. The users use the system to satisfy their needs, not to learn how to use the system. Their learning curve should be kept as shallow as possible. The system should also gives a freedom and choices to users of how to control their actions in the way they comfortable with. In particular, the system status should be clear to users and give sufficient assistance to the users to recover from any errors that take place in the system.

2.2.3 Performance

Performance is a measurement of both speed and time in the fulfillment of particular task executions on the system. The response time that is experienced by the users should give an impression that the system reacts immediately and instantly. User attention should be kept uninterrupted while accomplishing tasks and any delay should not be noticeable while conducting dialogue with the system. Good performance leads to increased frequency of the tasks that can be performed by the users and in turn will improve the usability of the system.

2.2.4 Robustness

Robustness is a measurement of how software to react appropriately even in the case of abnormal condition. The system should be convincing enough to support hundreds of users each of whom will perform a considerable amount of multiple tasks and actions. To ensure the reliability of the system, it should be first validated whether there is possible deviation from the requirements. At the end, the system should also be tested to make sure that it is free from failure for a given time and condition operation. A consideration should also be taken on

possible error generation during the error removal itself.

2.2.5 Maintainability

Maintainability is an attribute in which the functionalities of a system can be easily modified to conform to the requirement changes, to adapt with new environments and to improve other attributes described above. Good maintainability in software in turn leads to a good ease of finding, correcting and recovering from errors. Modularity, self documentation and structured coding are attributes that relate to maintainability that will be expected from the final product.

2.3 Solution Plan

In this subsection, we will be describing the high level features in broad terms that the system is expected to provide. This will be used as a basis for developing the set of detailed requirements that is documented on the following chapter. The views and perspectives of the stakeholders to the system will also be taken into consideration while planning the solution.

Developing a solution that is accessible from anywhere and anytime has been the primary feature of the system to have. Internet appears to provide an important part of the infrastructure of our solution. Geographical distance becomes less and less noticeable as the Internet connections grow significantly in the past few decades. Besides, standardized web browsers that can run on a variety of computers running have now become widely available. It is wise to develop some kind database application that is highly integrated with web. We have already seen that all of the existing solutions that described earlier in this chapter have adopted this attempt.

The system should minimally provide a mean of publishing the thesis topics and provide capabilities for students to browse and search for topics. Usability plays an important part here. The solution should not repeat the mistakes that have been made in the existing systems. The paginating method that has been used by Claudine Halim could be incorporated in order to avoid overwhelming the user and improve the response time of page retrieval. In particular, each page should take no more than 10 seconds to download [Nielson]. Additionally, significant fields should be made exposed in the topics listing page, so that user can derive information of the topics quickly without having to visit the topic description page. Topic Title, Supervisor, and Topic Type are good candidates to be displayed on the topics listing page. The use of staff initial to represent staff members is useless since it leads to confusion and misunderstanding among the users. A

better way of representing staff could be the staff family name which is much more meaningful and sufficiently brief to be displayed on the topics listing page. Also, the system should provide more extensive searching functionalities that could be in a form of advanced search engine. Besides, a simple quick search should also be provided because most of the time users do not need all the full power of advanced search. This quick search should be integrated with the browsing system so that users can easily move tasks between search and browsing [Rosenfeld & Moville].

The system will also incorporate the two nomination protocols that are used on the Thesis Management System used in CSE. Criteria of good protocol include such things as simple, easy to learn, and natural to the users that are executing it. Hence, slight modifications will be made on the two protocols in order to achieve these characteristics. Before any attempt on the design of the protocols is made, the framework itself should be made secure. This is to make sure that the execution of the protocols can be performed properly and appropriately by the right actors. We have seen that the missing login account in the current system on CSE lets unauthorized users to nominate topics for registered students. Login account is one of possible mechanisms that will solve this problem. The need of having a login account has been actually noticed by many of the participants of the survey as shown in Appendix A.

The Preapproved Nomination protocol that is currently used in CSE requires the students to initiate the process, i.e by submitting the topic number of the approved topic. This protocol relies on the absolute credibility of student to submit the topic number only when approval is granted. If students were behaved badly, then supervisor would have a busy day denying fake approvals. A better design of the protocol would be to have supervisor to initiate the process. The reason for this is that we believe more in staffs to perform their task properly rather than students, i.e. academic staffs approving arbitrary students are less

likely to happen than students proclaiming that the approvals have been granted. Another advantage of the protocol design is that it is more natural and suits the familiarity of users about the concept of granting approval. In particular, a supervisor acts as a gate keeper of the thesis topics and only when approval is granted student is given a chance to accept it. In the case of group approval, each of the students should accept the approval in order for the process to be successful. If one of the students denies the approval, then the topic allocation will not be able to proceed. In addition, we wouldn't need an admin work to enter the thesis allocation information to the database. At the end of the process, it should be done automatically. The resulting protocol would be a two way handshake between supervisor and the student that is illustrated on Figure 2.3.

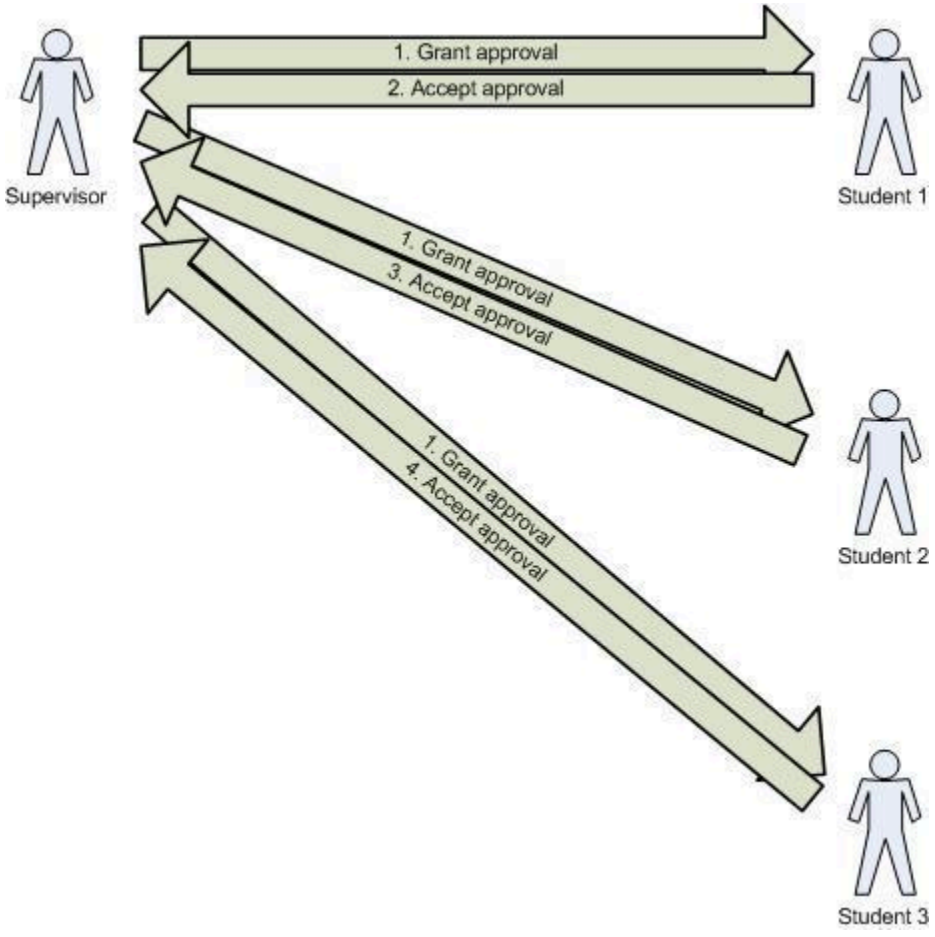


Figure 2.4. Preapproved Nomination Model

The most recent Preference Nomination Protocol that is used on CSE where the topic requests are sent in order of preference is very good in which students are no longer worried about getting allocated to the topic with the lowest preference ranking. However, it lacks of automation in handling the nomination process. In this project, we will move to the similar model with the topic requests directly sent by the student to supervisors in order to improve the performance and efficiency of our system. The difficulty with this approach seems lies on the checking of the expiry of the topic request itself. In fact, this task could be automated by the use of crond daemon, a process that could executes scheduled tasks on a routine basis. The desired preference nomination model is illustrated in Figure 2.5.

Figure 2.5. Preference Nomination Model

For both of the protocol, the system status should be visible to users at all time. In particular, the system should always keep the users informed about which

state of the nomination process they are in through appropriate feedback. Additionally, the users should also be given freedom of how they control their actions in every state of the process. For example supervisors should be able to cancel the approval before the students accept it and students should be able to cancel the nomination at anytime before any of the supervisors accepts the topic request. Last but not least, the system should also provide sufficient information about the how the nomination process is actually conducted so that they can make an informed decision on choosing the protocol process.

CHAPTER 3

Requirements

Capturing the requirements is essential step in the early stage of the system development. In this chapter, we will look into the requirements of the overall system behavior. These requirements are made clear, concise, unambiguous and understandable so that the development of the system shall solve the right problems. The requirements stated here are mainly derived from the solution plan that has been discussed in the previous chapter. It is also affected by the following factors:

1. The results of the survey on current thesis management system used in CSE.
2. Answers to queries from Thesis Coordinator, Albert Nymeyer.
3. Suggestions from the supervisor of this thesis, John Shepherd.
4. Meeting with the Undergraduate Team during commencement of my Thesis A.
5. Interviews with Thesis Database officer, Maria Tzortzis.

These requirements will be partitioned into two categories: Functional Requirements and Non-Functional Requirements. A priority value will be given to indicate the importance of a given requirement item with respect to others. A value of 1 indicates the highest priority.

3.1 Functional Requirements

Identifying the actions that the system should perform and how the system should interact with the users is integral part of performing requirements analysis. In this subsection, each of the functional requirements is addressed and grouped together based on the types of service that the system performs.

Browsing and Searching Topics Requirements

ID	Description	Priority
BST001	The system should allow users to browse topics.	1
BST002	The system should be able to show users the detailed description of the topic.	1
BST003	The system should provide search capabilities for topics.	1

Login Management

ID	Description	Priority
LMG001	The system should provide a mechanism to authenticate users.	1

Topic Management Requirements

ID	Description	Priority
TMG001	The system should allow admin and staffs to add topic.	1
TMG002	The system should allow admin and staffs to update topic information.	1
TMG003	The system should allow admin and staffs to delete topic information.	1

User Management Requirements

ID	Description	Priority
UMG001	The system should allow admin to add student.	1
UMG002	The system should allow admin to update student information.	1

UMG003	The system should allow admin to delete student information.	1
UMG004	The system should allow admin to add staff.	1
UMG005	The system should allow admin to update staff information	1
UMG006	The system should allow admin to delete staff information.	1

Topic Allocation Requirements

ID	Description	Priority
TAL001	The system should allow admin to allocate topic for students manually.	1
TAL002	The system should allow admin to deallocate topic from students.	1
TAL003	The system should be able to notify students and their supervisor about the allocation of topic related to them.	3
TAL004	The system should be able to notify students and their supervisor about the deallocation of topic related to them.	3

Preapproved Nomination Requirements

ID	Description	Priority
PAN001	The system should allow staffs to grant approval of a topic for one or more students.	1
PAN002	The system should allow staffs to cancel the granted approval for students.	2
PAN003	The system should allow students to accept an approval.	1
PAN004	The system should allow students to cancel the accepted approval.	3
PAN005	The system should allow students to reject an approval.	1
PAN006	The system should be able to notify the students when approval is granted.	3
PAN007	The system should be able to notify the students when approval is cancelled.	3
PAN008	The system should be able to notify the supervisor and group members when an approval has already been	3

	accepted by all group members.	
PAN009	The system should be able to notify the supervisor and all other group member when an approval is rejected by one of group member.	3

Preference Nomination Requirements

ID	Description	Priority
PFN001	The system should allow students to nominate topics by preference.	1
PFN002	The system should allow students to nominate group topic by preference.	1
PFN003	The system should send the topic requests in order the order of preference.	1
PFN004	The system should allow students to cancel the preference nomination	1
PFN005	The system should allow staff to accept the topic request.	1
PFN006	The system should allow staff to reject the topic request.	1
PFN007	The system should watch the expiration of topic request.	1
PFN008	The system should notify the supervisor when the topic request is arrived.	3
PFN009	The system should notify the supervisor and the group members when the topic request is canceled.	3
PFN010	The system should notify the students when the topic request is accepted by the supervisor.	3
PFN011	The system should notify the students when the topic request is rejected by the supervisor.	3
PFN012	The system should notify the students when the topic request is expired.	3

3.2 Non-Functional Requirements

In some extent, constraints or restrictions should be considered while developing a system. This helps us to narrow down the selection criteria of a good solution. Many of these non-functional requirements are affected by the suggestions from survey on the current Thesis Management System on CSE, the usability principles by Nielson, Rosenfeld, Morville and suggestions by John Shepherd.

ID	Description	Priority
NFR001	The browse topic page should use page chunking technique so that users are not overwhelmed with the long topic list.	1
NFR002	Important fields should be exposed to users and the staff family name should be used to represent the academic staff in the topic listing page.	1
NFR003	The system should provide extensive search functionality for topics.	1
NFR004	Searching and browsing should be integrated so that users feel comfortable to switch between tasks.	1
NFR005	Searching and browsing system should use the alphabetical and categorical indexing to support difference modes of searching.	1
NFR006	Each page should take no longer than 10 seconds to load during normal system's resources usage.	1
NFR007	The maximum down time is 1 day a week for maintenance.	1
NFR008	The system should provide an encryption for the users' sensitive information, such as password.	1

CHAPTER 4

Use Case Specification

The use case specification will be used as to model the functionalities that are provided by the system based on the functional requirements that are documented in the previous chapter. In this chapter, we will be identifying the principal actors, use cases and their relationships in the system. In addition, scenarios will also be provided as detailed descriptions of the steps that happen during the interaction between each actors and use cases.

4.1 Actors

Figure 4.1 shows the actors and their structural relationships in our thesis management system. As we can see, the user includes all the types of active people that initiate calls to the system to deliver a service to satisfy their needs on performing particular tasks. The generalization is made based on how differently they use the system. There is also an external process that is run on the system to check the validity of the topic requests namely, Cron Daemon.

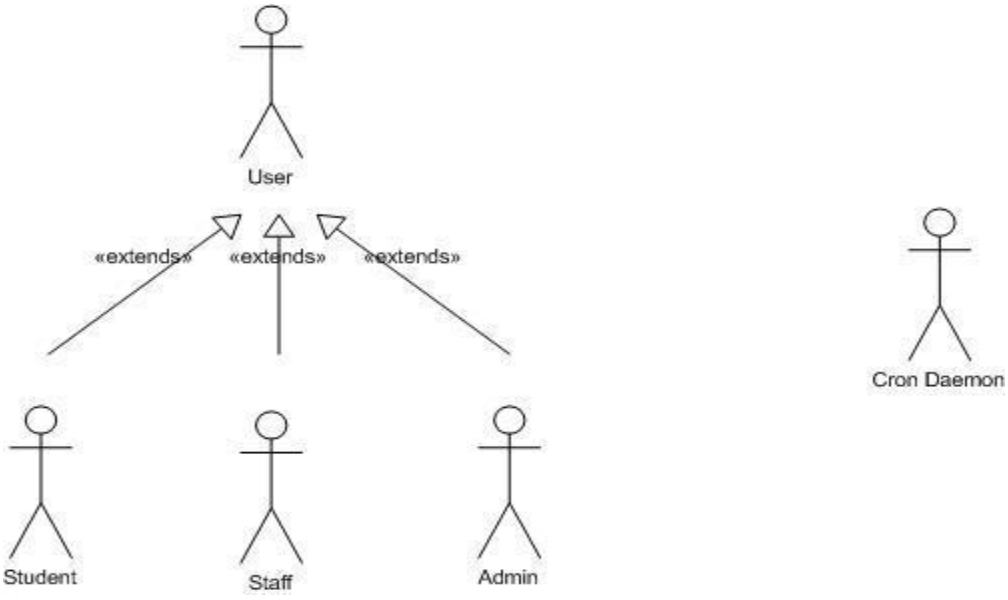


Figure 4.1. Actors and their structural relationships

User

Description	A person who is not registered to the system, but are allowed to take a look at the system and use the browse/search functionalities.
Aliases	General User
Inherits	None
Actor Type	Active Person

Student

Description	A person who is registered to undertake thesis project at the University.
Aliases	Thesis Student
Inherits	User
Actor Type	Active Person

Staff

Description	A person who is registered to supervises the thesis project undertaken by student.
Aliases	Thesis Staff, Academics
Inherits	User
Actor Type	Active Person

Admin

Description	A person who is responsible for the maintenance of WebTMS.
Aliases	Administrator, Undergraduate Team

Inherits	User
Actor Type	Active Person

Cron Daemon

Description	An external system that is responsible to check the topic expiry for every particular period.
Aliases	Administrator, Undergraduate Team
Inherits	User
Actor Type	Passive System

4.2 Use Cases

In this subsection, all of the use cases that represent the dialogues between actors and the thesis management system will be documented. In addition, the detailed description of the use case and the associated flow of events that happens during the invocation of each use case will be presented here. It would have been complicated if all the use cases are presented in one big diagram; hence each of the use cases are grouped into separate figures based on the logical process that occurs on the system.

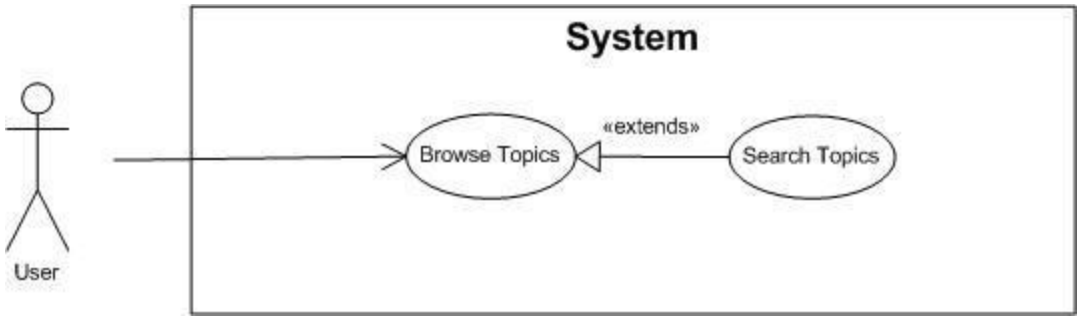


Figure 4.2. Use cases for browsing and searching topics

Browse Topics

Description	The use case provides browsing functionality on the topics that exists on the system.
Actors	User
Pre-conditions	1. The system has already been loaded with the topic data.
Main Flows	1. User Richard opens the WebTMS Website. 2. WebTMS Website is displayed in a window on local machine. 3. Richard clicks on the link “Topics”. 4. The system displays the available topics list. 5. Richard sorts the topic list by supervisor’s last name. 6. The system displays the topics sorted by supervisor’s last name. 7. Richard selects to index “B”. 8. The system displays the topics sorted by supervisor’s last name starting with character “B”. 9. Richard jumps to page 2. 10. The system displays the topics sorted by supervisor’s last name starting with character “B” that is contained on page 2. 11. Richard clicks on the topic title “WWW Room Booking System”. 12. The system displays the topic details on a new page.
Exceptional Flows	1. The system hasn’t been loaded with the topics’ information. The system displays empty topic list.
Post-conditions	None
Requirement Reference	BST001, BST002

Search Topics

Description	The use case provides searching functionality on the topics that exists on the system.
Actors	User
Pre-conditions	1. The system has already been loaded with the topics data.
Main Flows	1. User Richard opens the WebTMS Website. 2. WebTMS Website is displayed in a window on local machine. 3. Richard visits the topic list page and clicks on the “advanced search” link on the page. 4. The system displays the advanced search page. 5. Richard fills in the title field “database” and selects type “Development (DEV)” on the form fields. 6. The system displays all the topics that match the search criteria.
Exceptional Flows	1. The system hasn’t been loaded with the topics’ information. The system displays empty topic list.
Post-conditions	None
Requirement Reference	BST003

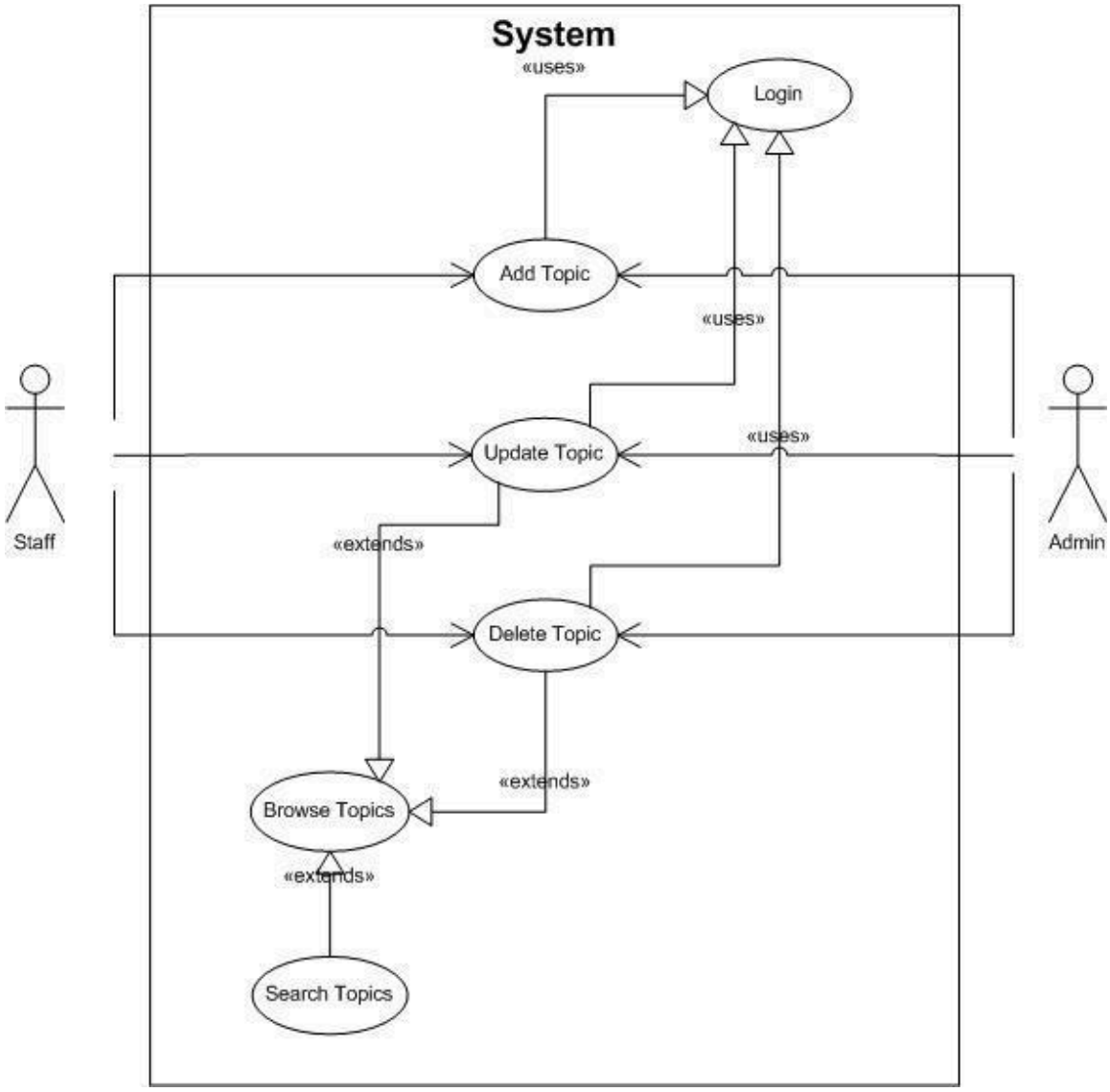


Figure 4.3. Use cases for managing topics

Login

Description	The use case provides authentication for users.
Actors	User
Pre-conditions	1. The system has already been loaded with the user data.
Main Flows	1. User Richard opens the WebTMS Website. 2. WebTMS Website is displayed in a window on local machine. 3. Richard fills in the username “rman744” and password “thesis_db”. 4. Richard is successfully logged in and the available options for students are displayed.
Exceptional Flows	1. Richard enters an invalid password. The system displays an error message and Richard can reenter the account information.
Post-conditions	1. User is successfully logged in and able to perform more advanced operations.
Requirement Reference	LMG001

Add Topic

Description	The use case provides the capability of adding a topic to the database.
Actors	Admin, Staff
Preconditions	1. User has already logged into the system.
Main Flows	1. User John Shepherd logins into the system. 2. The system shows the available options for Staff. 3. John Shepherd clicks to the link “My Topics”. 4. The system displays all of the Topics that is offered by John Shepherd. 5. John Shepherd follows the link to add topic. 6. The system displays the topic entry form on the page. 7. John enters the fields on the forms and submits the form. 8. The system stores the topic information on the database and displays a receipt indicating a successful transaction.
Exceptional Flows	1. John Shepherd forgets to enter the necessary fields on the form. The system pops up an error message when the form is submitted.
Post-conditions	1. The topic information is successfully stored on the database.
Requirement Reference	TMG001

Update Topic

Description	The use case provides the capability of updating a stored topic in the database.
Actors	Admin, Staff
Preconditions	1. User has already logged into the system. 2. The topic to be updated exists on the database.
Main Flows	1. User John Shepherd logins into the system. 2. The system shows the available options for Staff. 3. John Shepherd clicks to the link “My Topics” and click “UPDATE” on the topic “4-th Year Thesis Management System”. 4. The system displays the topic entry form with the fields already filled with the last saved information. 5. John updates the field Prerequisites to “COMP3311” and submits the form. 6. The system updates the topic information on the database and displays a receipt indicating a successful transaction.
Exceptional Flows	1. John Shepherd forgets to enter the necessary fields on the form. The system pops up an error message when the form is submitted.
Post-conditions	1. The topic information is successfully updated.
Requirement Reference	TMG002

Delete Topic

Description	The use case provides the capability of archiving a stored topic in the database.
Actors	Admin, Staff
Preconditions	1. User has already logged into the system. 2. The topic to be deleted exists on the database.
Main Flows	1. User John Shepherd logins into the system. 2. The system shows the available options for Staff. 3. John clicks to the link “My Topics” and clicks “DELETE” on the topic “4-th Year Thesis Management System”. 4. The system displays a confirmation box asking whether to proceed with the action. 5. Maria clicks on the “OK” button. 6. The system archives the topic information on the database.
Exceptional Flows	1. One or more students have been allocated to the topic. The system displays an error page stating the condition. 2. One or more students are requesting the topic to a staff. The system displays an error page stating the condition. 3. Approval of the topic from staff is pending for students’ response. The system displays an error page stating the condition.
Post-conditions	1. The topic entry on the database is successfully archived.
Requirement Reference	TMG003

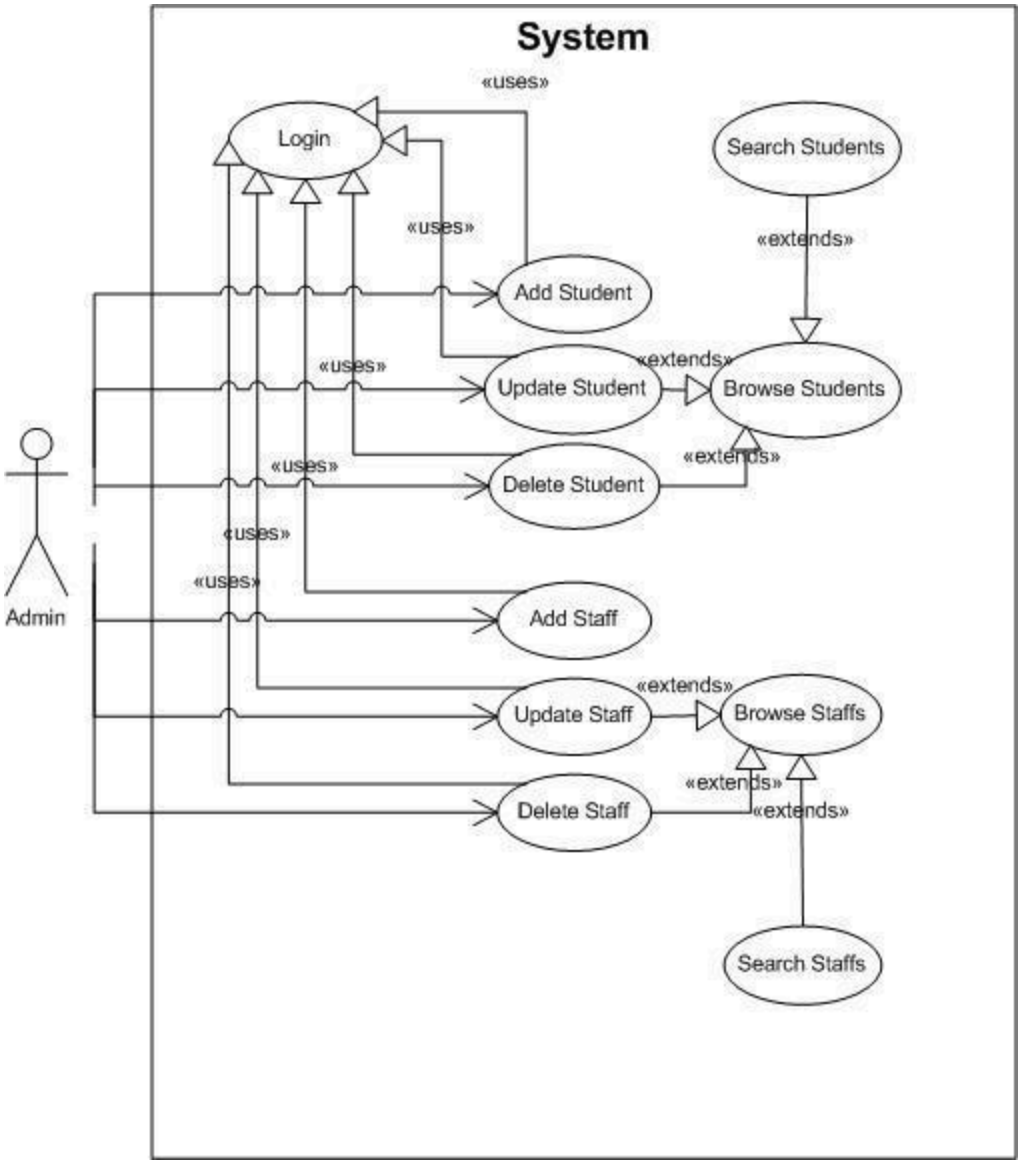


Figure 4.4. Use cases for managing users

Add Student

Description	The use case provides admin with the capability of adding a student to the database.
Actors	Admin
Preconditions	1. User has already logged into the system.
Main Flows	1. User Maria Tzortzis logins into the system. 2. The system shows the available options for Admin. 3. Maria clicks on the link “Students”. 4. The system displays the list of all the students that are registered on the system. 5. Maria follows the link to add new student. 6. The system displays a student entry form on the page. 7. Maria enters the fields on the forms and submits the form. 8. The system stores the student information on the database, displays a receipt indicating a successful transaction and sends email notification about the registration.
Exceptional Flows	1. Maria forgets to enter the necessary fields on the form. The system pops up an error message when the form is submitted.
Post-conditions	1. The student information is successfully stored on the database.
Requirement Reference	UMG001

Update Student

Description	The use case provides the capability of updating stored student information in the database.
Actors	Admin
Preconditions	1. User has already logged into the system. 2. The student to be updated exists on the database.
Main Flows	1. User Maria Tzortzis logins into the system. 2. The system shows the available options for Admin. 3. Maria Tzortzis clicks to the link “Students”. 4. The system displays the list of all the students that are registered on the system. 5. Maria clicks “UPDATE” on the student “Richard Mantik”. 6. The system displays the student entry form with the fields already filled with the last saved information. 7. Maria updates the field Street to “11 Royal Street” and submits the form. 8. The system updates the topic information on the database, displays a receipt indicating a successful transaction and send email notification informing about the update.
Exceptional Flows	1. Maria forgets to enter the necessary fields on the form. The system pops up an error message when the form is submitted.
Post-conditions	1. The student information is successfully updated.
Requirement Reference	UMG002

Delete Student

Description	The use case provides the capability of deleting stored student information from the database.
Actors	Admin
Preconditions	1. User has already logged into the system. 2. The student to be deleted exists on the database.
Main Flows	1. User Maria Tzortzis logins into the system. 2. The system shows the available options for Admin. 3. Maria Tzortzis clicks to the link “Students”. 4. The system displays the list of all the students that are registered on the system. 5. Maria clicks “DELETE” on the student “Richard Mantik”. 6. The system displays a confirmation box asking whether to proceed with the action. 7. Maria clicks on the “OK” button. 8. The system deletes the student information from the database and sends email notification about deregistration.
Exceptional Flows	1. The student has already been allocated to a topic. The system displays an error message stating the condition. 2. The student has already been approved by a staff. The system displays an error message stating the condition. 3. The student has already been nominated by preference. The system displays an error message stating the condition.

	4. The student is included on the preference nomination submitted by other students. The system displays an error message stating the condition.
Post-conditions	1. The student information is successfully deleted from the database.
Requirement Reference	UMG003

Add Staff

Description	The use case provides admin with the capability of adding a staff to the database.
Actors	Admin
Preconditions	1. User has already logged into the system.
Main Flows	1. User Maria Tzortzis logins into the system. 2. The system shows the available options for Admin. 3. Maria clicks on the link “Staffs”. 4. The system displays the list of all the staffs that are registered on the system. 5. Maria follows the link to add new staff. 6. The system displays a staff entry form on the page. 7. Maria enters the fields on the forms and submits the form. 8. The system stores the staff information on the database, displays a receipt indicating a successful transaction and send email notification about the registration.
Exceptional Flows	1. Maria forgets to enter the necessary fields on the form. The system pops up an error message when the form is submitted.
Post-conditions	1. The staff information is successfully stored on the database.
Requirement Reference	UMG004

Update Staff

Description	The use case provides the capability of updating stored staff information in the database.
Actors	Admin
Preconditions	1. User has already logged into the system. 2. The student to be updated exists on the database.
Main Flows	1. User Maria Tzortzis logins into the system. 2. The system shows the available options for Admin. 3. Maria clicks to the link “Staffs”. 4. The system displays the list of all the staffs that are registered on the system. 5. Maria clicks “UPDATE” on the staff “John Shepherd”. 6. The system displays the staff entry form with the fields already filled with the last saved information. 7. Maria updates the field Home Phone to “91234567” and submits the form. 8. The system updates the topic information on the database, displays a receipt indicating a successful transaction, and sends email notification informing about the update.
Exceptional Flows	1. Maria forgets to enter the necessary fields on the form. The system pops up an error message when the form is submitted.
Post-conditions	1. The staff information is successfully updated.
Requirement Reference	UMG005

Delete Staff

Description	The use case provides the capability of deleting stored staff information from the database.
Actors	Admin
Preconditions	1. User has already logged into the system. 2. The student to be deleted exists on the database.
Main Flows	1. User Maria Tzortzis logins into the system. 2. The system shows the available options for Admin. 3. Maria Tzortzis clicks to the link “Staffs”. 4. The system displays the list of all the staffs that are registered on the system. 5. Maria clicks “DELETE” on the staff “Richard Buckland”. 6. The system displays a confirmation box asking whether to proceed with the action. 7. Maria clicks on the “OK” button. 8. The system deletes the staff information from the database and sends an email notification about deregistration.
Exceptional Flows	1. The staff is offering a topic. The system displays an error message stating the condition. 2. The staff is supervising a thesis. The system displays an error message stating the condition.
Post-conditions	1. The staff information is successfully deleted from the database.
Requirement Reference	UMG006

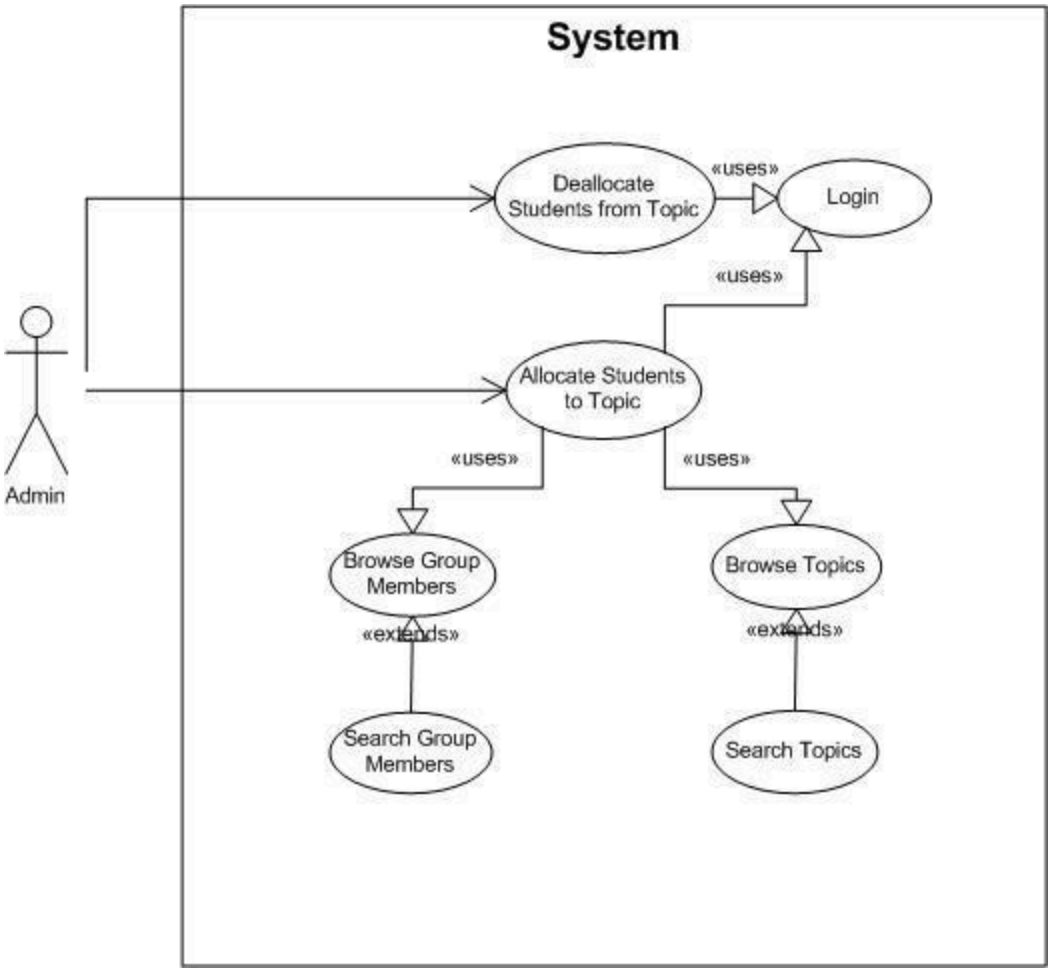


Figure 4.5. Use cases for manually allocating topic for students

Allocate Students to Topic

Description	The use case provides admin with the capability of manually allocating a topic for students.
Actors	Admin
Preconditions	1. User has already logged into the system. 2. The link to be created hasn't already been created between the selected topic and the selected students.
Main Flows	1. User Maria Tzortzis logins into the system. 2. The system shows the available options for Admin. 3. Maria clicks on the link "Topic Allocations". 4. The system displays the page containing all the topic allocations. 5. She follows the link to allocate students to topic. 6. The system displays the group selection page. 7. Maria selects a group member "Richard Mantik" and submits the form. 8. The system displays the topic selection page. 9. Maria selects the topic "4-th Year Thesis Management System. 10. The system links the student "Richard Mantik" to the topic "4-th Year Thesis Management System" and sends an email notification to the student "Richard Mantik" informing the allocation.
Exceptional Flows	1. The selected student Richard have been granted approval from other staff. The system displays an error message stating the condition. 2. The selected students Richard have been proceeding through preference nomination process. The system displays an error message stating the condition.

	3. The selected student Richard has been allocated to a topic. The system displays an error message informing the condition. 4. There is no more quota for the topic. The system displays error message stating the condition. 5. The number of group members doesn't satisfy the criteria for group size for the topic. The system displays error message stating the condition.
Post-conditions	1. The link between the selected topic and the selected students are created.
Requirement Reference	TAL001, TAL003

Deallocate Students from Topic

Description	The use case provides admin with the capability of deallocating a topic from students.
Actors	Admin
Preconditions	1. User has already logged into the system. 2. The link to be created has already been created between the topic and the students.
Main Flows	1. User Maria Tzortzis logins into the system. 2. The system shows the available options for Admin. 3. Maria clicks on the link “Topic Allocations”. 4. The system displays all the current topic allocations on the page. 5. Maria clicks on “DEALLOCATE” on the association between “4-th Year Thesis Management System” and student “Richard Mantik”. 6. The system unlinks the student “Richard Mantik” from the topic “4-th Year Thesis Management System” and sends an email notification to the student informing the de-allocation.
Exceptional Flows	None
Post-conditions	1. The link between the topic and the students are removed.
Requirement Reference	TAL002, TAL004

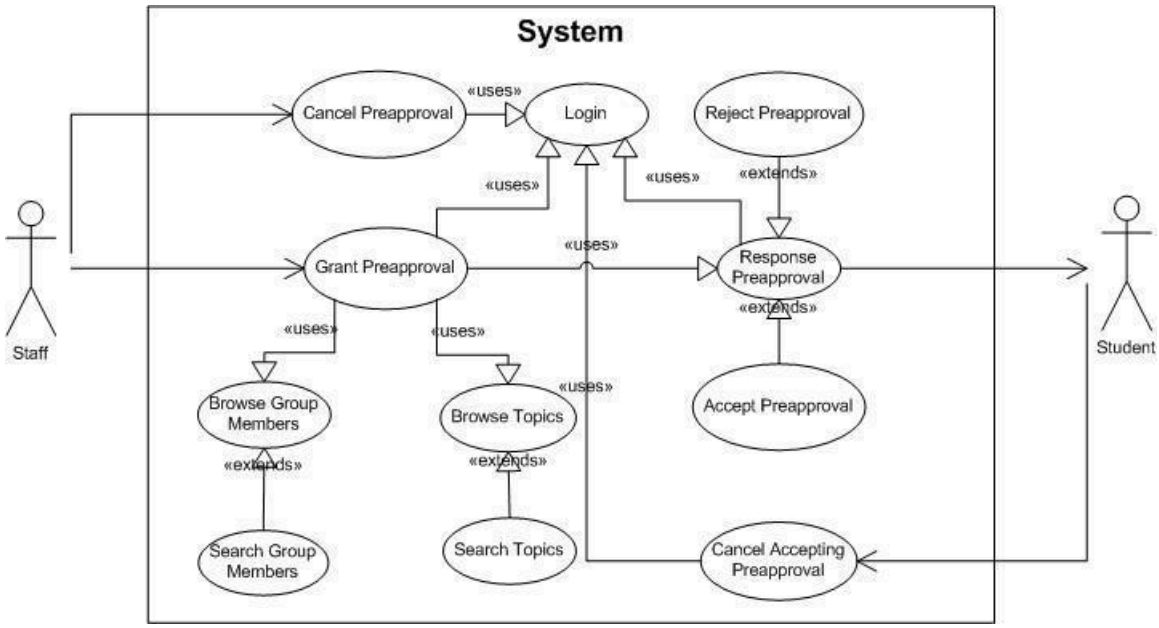


Figure 4.6. Use cases for preapproved nomination

Grant Preapproval

Description	The use case provides staff with the capability to grant approval to students for undertaking thesis project.
Actors	Staff
Preconditions	1. User has already logged into the system. 2. The students haven't been granted approval from any staff member. 3. There is no active preference nomination submitted by the student. 4. The students haven't been allocated to any topic.
Main Flows	1. User John Shepherd logins into the system. 2. The system shows the available options for Staff. 3. John clicks on the link "Preapproved Nomination". 4. The system displays all the current approvals that are pending for students' response on the page. 5. John goes to the link to grant approval to students. 6. The system displays the group selection page. 7. John selects group members "Edmong Yosiardi" and "Nicholas Mario Enrico" and submits the form. 8. The system displays all the topic selection page. 9. John selects the topic "WebCMS". 10. The system stores the approval information for the students and sends an email notification to both students informing the approval.
Exceptional Flows	1. The selected students Edmong and Nicholas have been granted approval from other staff. The system displays an error message stating the condition. 2. The selected students Edmong and Nicholas have been proceeding through preference nomination

	process. The system displays an error message stating the condition. 3. The selected students Edmong and Nicholas have been allocated to a topic. The system displays an error message stating the condition. 4. There is no more quota for the topic. The system displays an error message stating the condition. 5. The number of group members doesn't satisfy the criteria for group size for the topic. The system displays error message stating the condition.
Post-conditions	1. The approval is pending for students' response.
Requirement Reference	PAN001, PAN006

Cancel Preapproval

Description	The use case provides staff with the capability to cancel the approval granted to students.
Actors	Staff
Preconditions	1. User has already logged into the system. 2. The approval is pending for students' response.
Main Flows	1. User John Shepherd logins into the system. 2. The system shows the available options for Staff. 3. John clicks on the link "Preapproved Nomination". 4. The system displays all the current approvals that are pending for students' response on the page. 5. John clicks "CANCEL" on the approval of topic "WebCMS" for the students "Edmong Yosiardi" and "Nicholas Mario Enrico". 6. The system archives the approval information and sends email notifications to students "Edmong Yosiardi" and "Nicholas Mario Enrico" informing the approval has been cancelled.
Exceptional Flows	None
Post-conditions	1. The approval information is archived.
Requirement Reference	PAN002, PAN007

Accept Preapproval

Description	The use case provides student with the capability to accept the approval granted from Staff.
Actors	Student
Preconditions	1. User has already logged into the system. 2. The approval is pending for students' response.
Main Flows	1. User Edmong Yosiardi login into the system. 2. The system shows the available options for Student. 3. Edmong clicks on the link "Preapproved Nomination". 4. The system displays the approval that is pending for group members' response. 5. Edmong clicks on the link "ACCEPT". 6. The system displays a confirmation box asking whether to proceed with the action. 7. Edmong clicks on the "OK" button. 8. The system shows that the status of approval nomination is accepted by Edmong and still pending for Nicholas' response.
Exceptional Flows	1. Edmong hasn't been approved by any staff. The system displays the initial page that contains instructions how to get approval for a topic.
Post-conditions	1. If the approval has been accepted by all group members, each of the group members will be allocated to the topic. Otherwise, the approval will still be pending for other group members' response.
Requirement Reference	PAN003, PAN008

Cancel Accepting Preapproval

Description	The use case provides staff with the capability to cancel the approval that has been accepted by the student.
Actors	Student
Preconditions	1. User has already logged into the system. 2. The approval has been accepted by the student.
Main Flows	1. User Edmong Yosiardi logins into the system. 2. The system shows the available options for Student. 3. Edmong clicks on the link “Preapproved Nomination”. 4. The system displays the Approval that is accepted by Edmong and still pending for other group members’ response. 5. Edmong clicks on the link “CANCEL”. 6. The system shows the approval that is pending for Edmong’s reponse and other group members’.
Exceptional Flows	1. Edmong hasn’t been approved by any staff. The system displays the initial page that contains instructions how to get approval for a topic.
Post-conditions	1. The approval is pending for the student’s response who cancelled it.
Requirement Reference	PAN004

Reject Preapproval

Description	The use case provides student with capability to reject the permission granted to undertake thesis project from Staff.
Actors	Student
Preconditions	1. User has already logged into the system. 2. The approval is pending for the student response.
Main Flows	1. User Edmong Yosiardi logins into the system. 2. The system shows the available options for Student. 3. Edmong clicks on the link “Preapproved Nomination”. 4. The system displays the approval that is pending for Edmong’s response. 5. Edmong clicks on the link “REJECT”. 6. The system displays a confirmation box asking whether to proceed with the action. 7. Edmong clicks on the “OK” button. 8. The system sends message to other group members and the supervisor informing the rejection.
Exceptional Flows	1. Edmong hasn’t been approved by any staff. The system displays the initial page that contains instructions how to get approval for a topic.
Post-conditions	1. The approval information is archived.
Requirement Reference	PAN005, PAN009

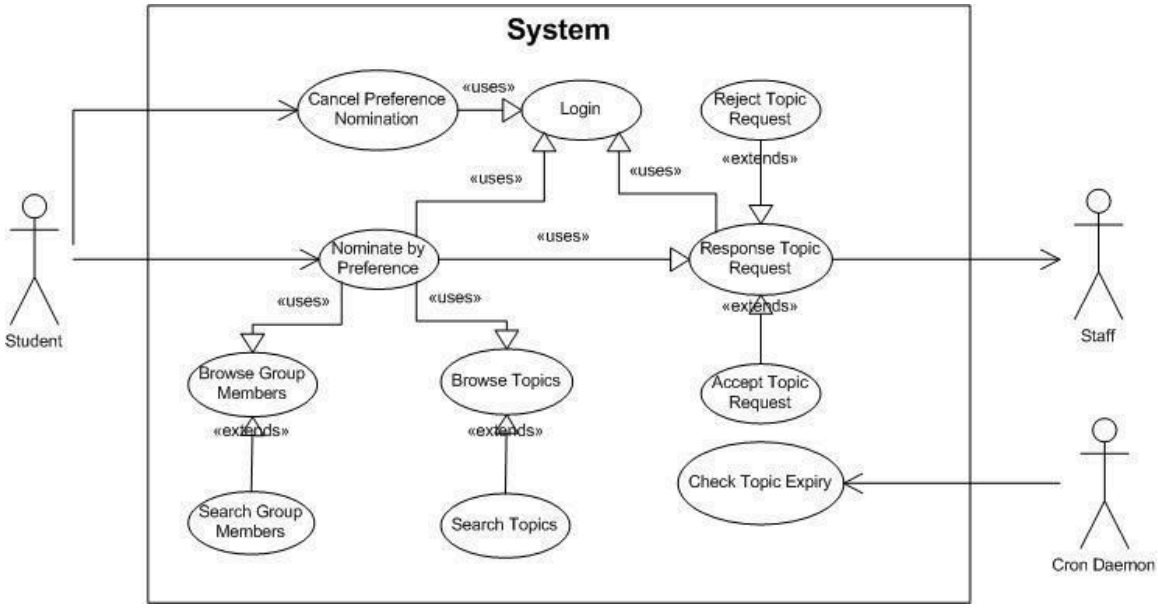


Figure 4.7. Use cases for preference nomination

Nominate by Preference

Description	The use case provides student with the capability to nominate topics by preference.
Actors	Student
Preconditions	1. User has already logged into the system. 2. There is no preference nomination being processed for the student. 3. The students haven't been granted approval. 4. The students haven't been allocated to any topic.
Main Flows	1. User Jennifer Ramli logins into the system. 2. The system shows the available options for Student. 3. Jennifer clicks on the link "Preference Nomination". 4. The system displays the initial page that shows no preference nomination being processed. 5. Jennifer follows the link to nominate topic by preference. 6. The system displays the group selection page. 7. Jennifer includes group member "Josephine Kotjik" and submits the form. 8. The system displays the topic selection page. 9. Jennifer browses the topic selection page and selects the topic "Systematic code generation in compilation". 10. The system redisplays the topic selection page with the topic "Systematic code generation in compilation" put on the topic cart. 11. Jennifer selects the topic "Words Recognition". 12. The system redisplays the topic selection page with the topics "Systematic code generation in compilation" and "Words Recognition" put on the "Topic Cart".

	13. Jeniffer selects the topic “Multimedia Web Diary”. 14. The system redisplays the topic selection page with the topics “Systematic code generation in compilation”, “Words Recognition” and “Multimedia Web Diary” put on the topic cart. 15. Jeniffer submits the topic selection. 16. The system displays the preference nomination whose topic requests are inactivated and also pending for Josephine’s response.
Exceptional Flows	1. There is currently preference nomination for Jennifer being processed. The system displays the current preference nomination being processed. 2. Jennifer has been granted approval from a staff member. The system displays an error message stating the condition when selecting group members. 3. Jennifer has been allocated to a topic. The system displays this condition on the Preference Nomination Page. 4. The selected group member Josephine has been allocated to a topic or pre-approved by a supervisor. The system displays this condition when selecting group members. 5. There is no more quota for the topic. The system displays an error message stating the condition. 6. The number of group members doesn’t satisfy the criteria for group size for the topic. The system displays error message stating the condition.
Post-conditions	1. If all the group members’ preference nominations match with each others, an email notification

	requesting for the topic is sent to the supervisor and the status is being displayed on the page as “waiting for approval”. Otherwise, all the topic requests are deactivated.
Requirement Reference	PFN001, PFN002, PFN008

Cancel Preference Nomination

Description	The use case provides student with the capability to cancel the preference nomination being processed.
Actors	Student
Preconditions	1. User has already logged into the system. 2. There is a preference nomination being processed for the student.
Main Flows	1. User Jennifer Ramli logins into the system. 2. The system shows the available options for Student. 3. Jennifer clicks on the link “Preference Nomination”. 4. The system displays the current preference nomination being processed. 5. Jeniffer clicks on “CANCEL”. 6. The system displays a confirmation box asking whether to proceed with the action. 7. Jennifer clicks on the “OK” button. 8. The system sends email notification to the supervisor about the cancellation and displays the initial page that shows no preference nomination being processed.
Exceptional Flows	None
Post-conditions	1. The preference nomination is cancelled. If the topic has previously been agreed among group members, other group members’ topic requests are also postponed.
Requirement Reference	PFN004, PFN009

Accept Topic Request

Description	The use case provides staff with the capability to accept the student's request for a topic.
Actors	Staff
Preconditions	1. User has already logged into the system. 2. There is a topic requests pending for supervisor' response.
Main Flows	1. User Albert Nymeyer logins into the system. 2. The system shows the available options for Staff. 3. Albert clicks on the link "Preference Nomination". 4. The system displays the topic requests that are waiting for Albert's response. 5. Albert clicks "ACCEPT" on the request for topic "Systematic code generation in compilation" by the students "Jennifer Ramli" and "Josephine Kotjik". 6. The system displays a confirmation box asking whether to proceed with the action. 7. Albert clicks on the "OK" button. 8. The system links the students "Jeniffer Ramli" and "Josephine Kotjik" to the topic "Systematic code generation in compilation", and sends an email notification informing the allocation.
Exceptional Flows	None
Post-conditions	1. The link between the students and the requested topic is created.
Requirement Reference	PFN005, PFN010

Reject Topic Request

Description	The use case provides staff with the capability to reject student's request for a topic.
Actors	Staff
Preconditions	1. User has already logged into the system. 2. There is a topic requests pending for supervisor' response.
Main Flows	1. User Albert Nymeyer logins into the system. 2. The system shows the available options for Staff. 3. Albert clicks on the link "Preference Nomination". 4. The system displays the topic requests that are waiting for Albert's response. 5. Albert clicks "REJECT" on the request for topic "Systematic code generation in compilation" by the students "Jennifer Ramli" and "Josephine Kotjik". 6. The system displays a confirmation box asking whether to proceed with the action. 7. Albert clicks on the "OK" button. 8. The system sends an email notification to "Jennifer Ramli" and "Josephine Kotjik" informing the rejection and sends the next request for the topic "Words Recognition" to Adnan Amin.
Exceptional Flows	None
Post-conditions	1. The rejected topic request is archived.
Requirement Reference	PFN003, PFN006, PFN008, PFN011

Check Topic Expiry

Description	The use case provide cron daemon with the capability to monitor topic requests expiry time.
Actors	Cron Daemon
Preconditions	None
Main Flows	1. Cron Daemon is monitoring for topic request expiry. 2. Cron Daemon founds that the topic requests from the student “Jeniffer Ramli” and “Josephine Kotjik” that is waiting for Adnan Amin’s response is expired. 3. Cron Daemon archives the topic request, send email notifications informing the expiration and send the next request for the topic “Multimedia Web Diary” to Richard Buckland.
Exceptional Flows	None
Post-conditions	1. The expired topic requests are archived.
Requirement Reference	PFN003, PFN007, PFN008, PFN012

CHAPTER 5

Design

Designing the abstraction of the business rules of a system is the essential step towards developing a successful solution. At the end of the process, a model that can be mapped easily into implementation will be produced. In this chapter, we will be looking on the construction of the architecture of WebTMS.

5.1 Database Design

Database is the crucial part of the system, since it provides the storage for all information that is related to the thesis management process. One useful strategy for database design is to perform initial conceptual modeling using ER diagram and transform it to the relational schema that is close to the implementation. In this subsection, we will discuss about these two steps of database design for our thesis management system.

5.1.1 Conceptual Design (ER Diagram)

Entity Relationship (ER) is one of the common modeling paradigms to express the design of database. At this level of abstraction, the database structure is modeled by identifying of real-world objects, their characteristics and interconnection between them. Many of the functional system requirements affect the design decision during the development of the ER Diagram. The overview of the overall design of the Thesis Management System is shown in figure 4-1.

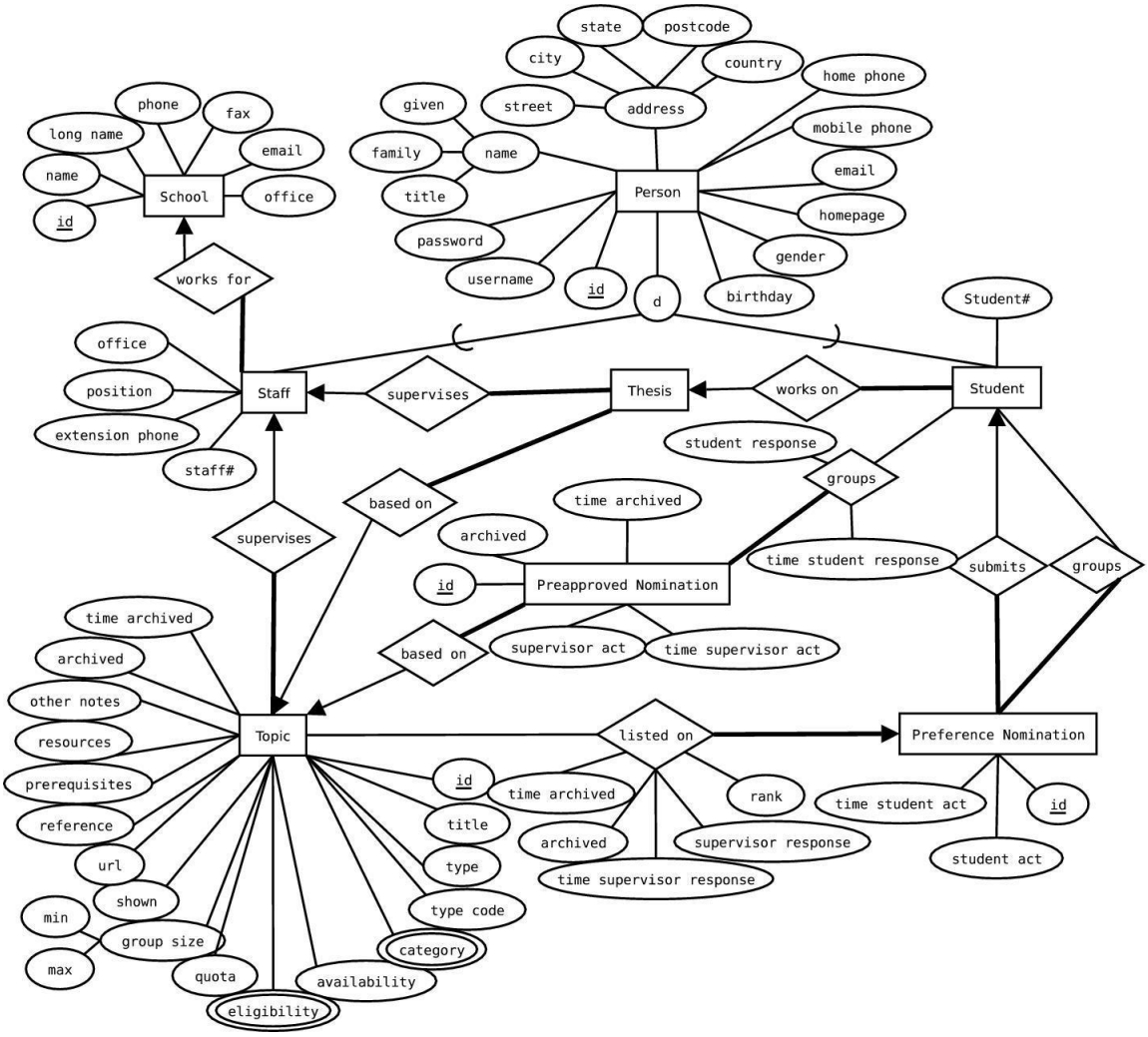


Figure 4-1: The complete ER model for WebTMS

To help you focus on the relationships among the entities, it will be useful to look at ER Diagram at each component. The above complete ER model is segmented into smaller portion and will be discussed in more detail. Using this illustration, it will be much easier to understand the detailed structure of the database and its design decision.

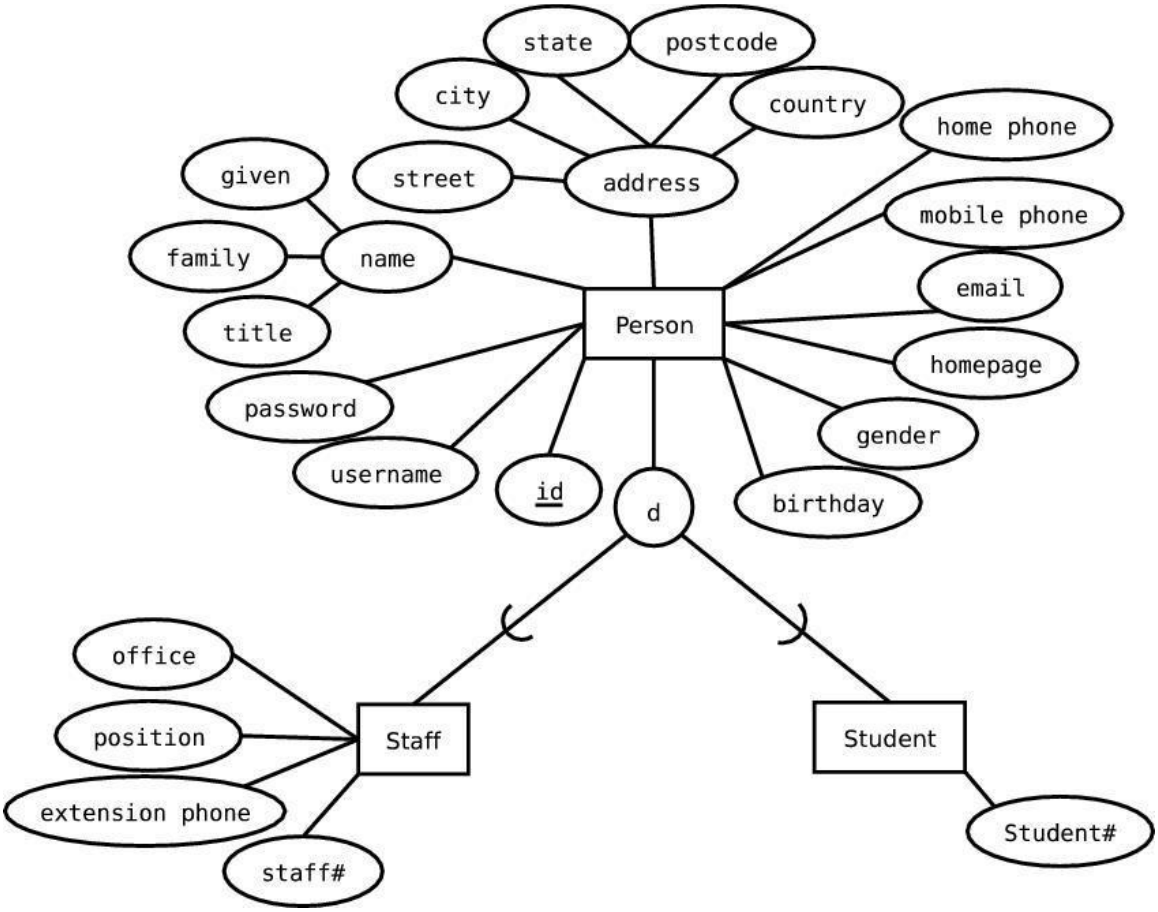


Figure 4-2: The user hierarchy ER segment

Figure 4-2 shows the hierarchy of users of the system. A user can either be a student, a staff or an administrator. An administrator can be represented by *Person* Entity associated with a unique identifier 0, hence there is no need to introduce a special entity that represents Administrator. It can be seen from the diagram that the attributes *username* and *password* are directly assigned to the entity person. The reason why it is designed this way is because it enforces that each person has only one account information. Having multiple accounts for one user reduces the security level of the system, since the more account information a user has, the more likely one of them getting compromised.

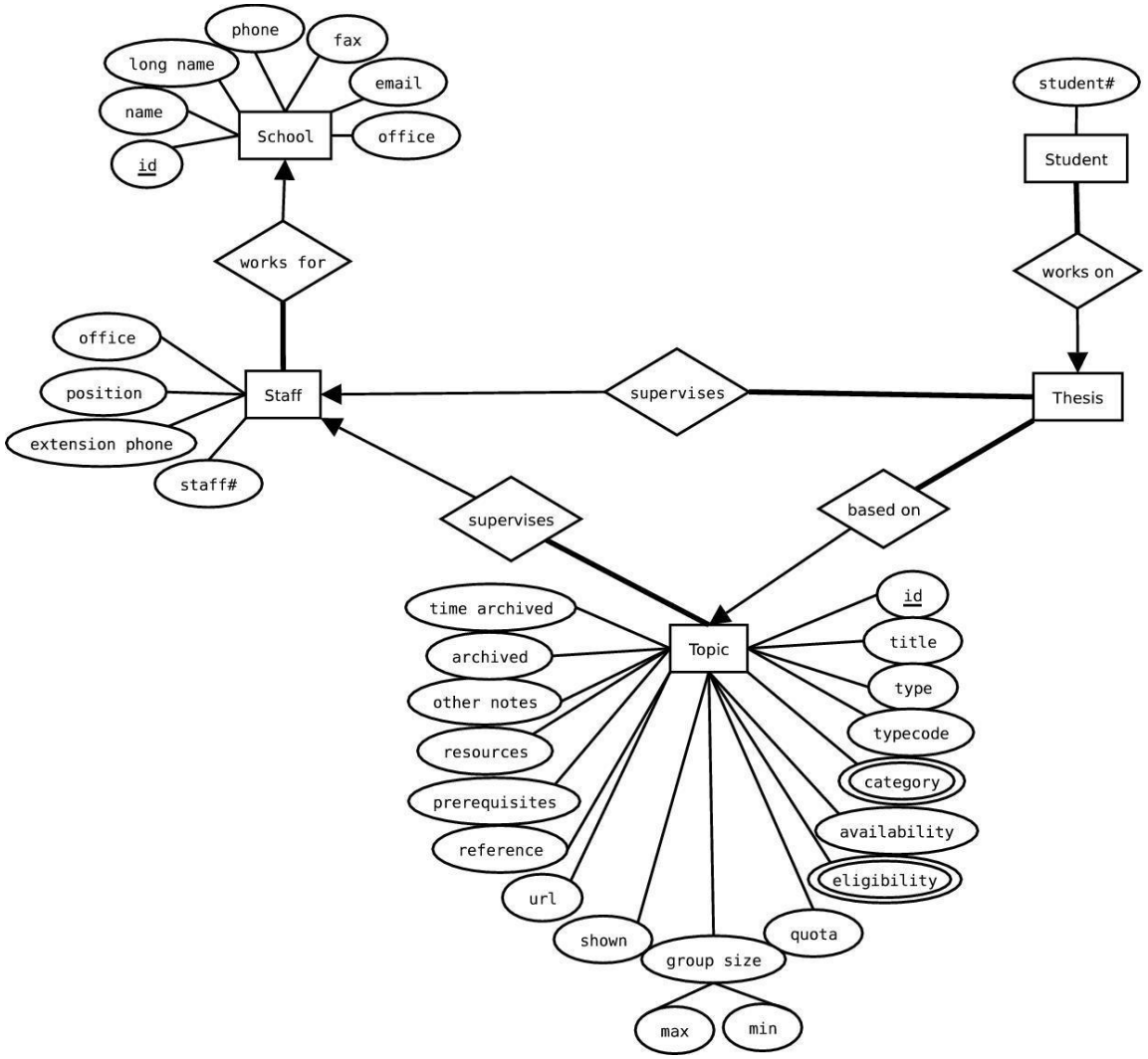


Figure 4-3: Topic Allocation ER segment

Figure 4-3 shows the interaction between topic, supervisor and student. The information about topic allocation is embodied in the *Thesis* Entity. Students working on a team are assigned to the same thesis entity; hence a thesis also serves as group. The attribute *quota* of the Topic entity defines the maximum number of thesis (a group of students) that can be assigned to the topic. The *group size* attribute specifies the minimum and maximum number of students working in a group. A *group size* 1-1 represents an individual topic, while others represents group topic.

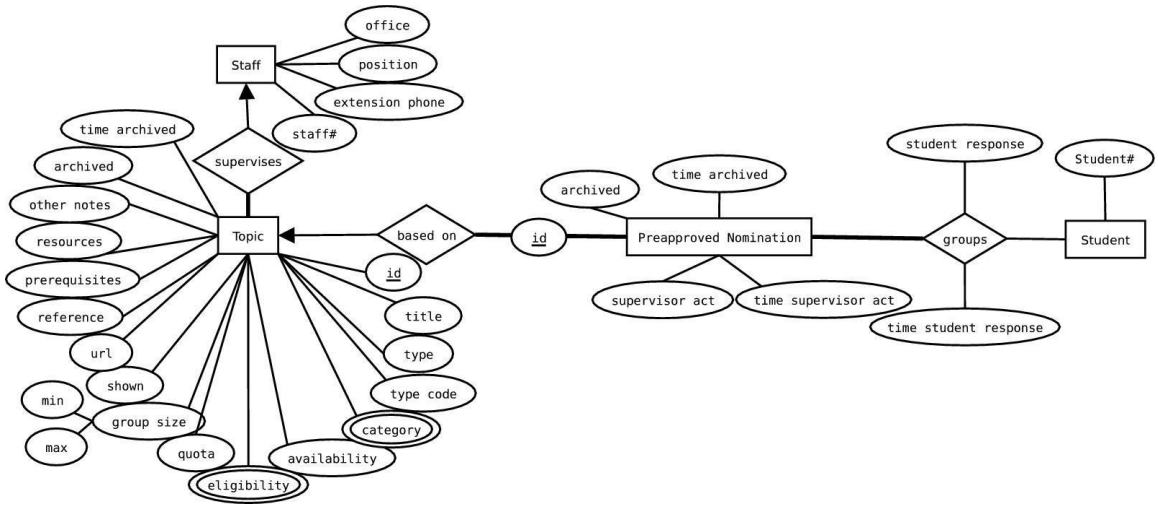


Figure 4.4: Pre-approved Nomination ER Segment

Figure 4.4 shows ER data model for Preapproved Nomination process. An approval granted by Supervisor is represented by the entity *Preapproved Nomination*. The approval is made for a student or a group of students each of which could response to the approval. This is represented by a relationship set groups between *Preapproved Nomination* and *Student* with the attached attributes *student response* and *time student response*.

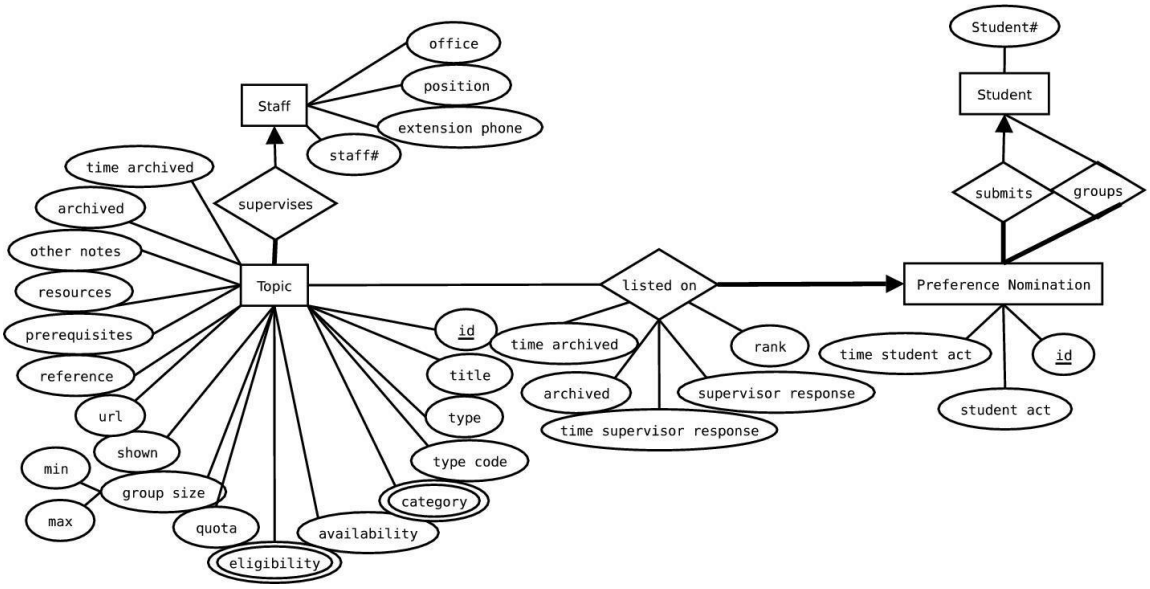


Figure 4.5: Preference Nomination ER Segment

Figure 4.5 shows ER data model for Preference Nomination process. A student submits a preference nomination that lists several topics associated with preference rank. These preference topics will be requested to the supervisors. This information is captured in the relationship between the entities *Preference Nomination* and *Preference Topic* attached with the attributes *rank* and *supervisor response*. A Preference Nomination could also include several other students that are proposed to work on the same topic. This is illustrated by the relationship set groups connecting *Preference Nomination* and *Student*.

5.1.2 Relational Schema

Once the conceptual model is completed, the next step is to transform the conceptual model into relational model where the inter-related tables are defined. The goal of this approach is to produce a data model that maps easily to the selected RDBMS (Relational Database Management System). Each of the entities and relationships are transformed into tables using the formal mapping process. The final result of transformation process is shown in Figure 4.6.

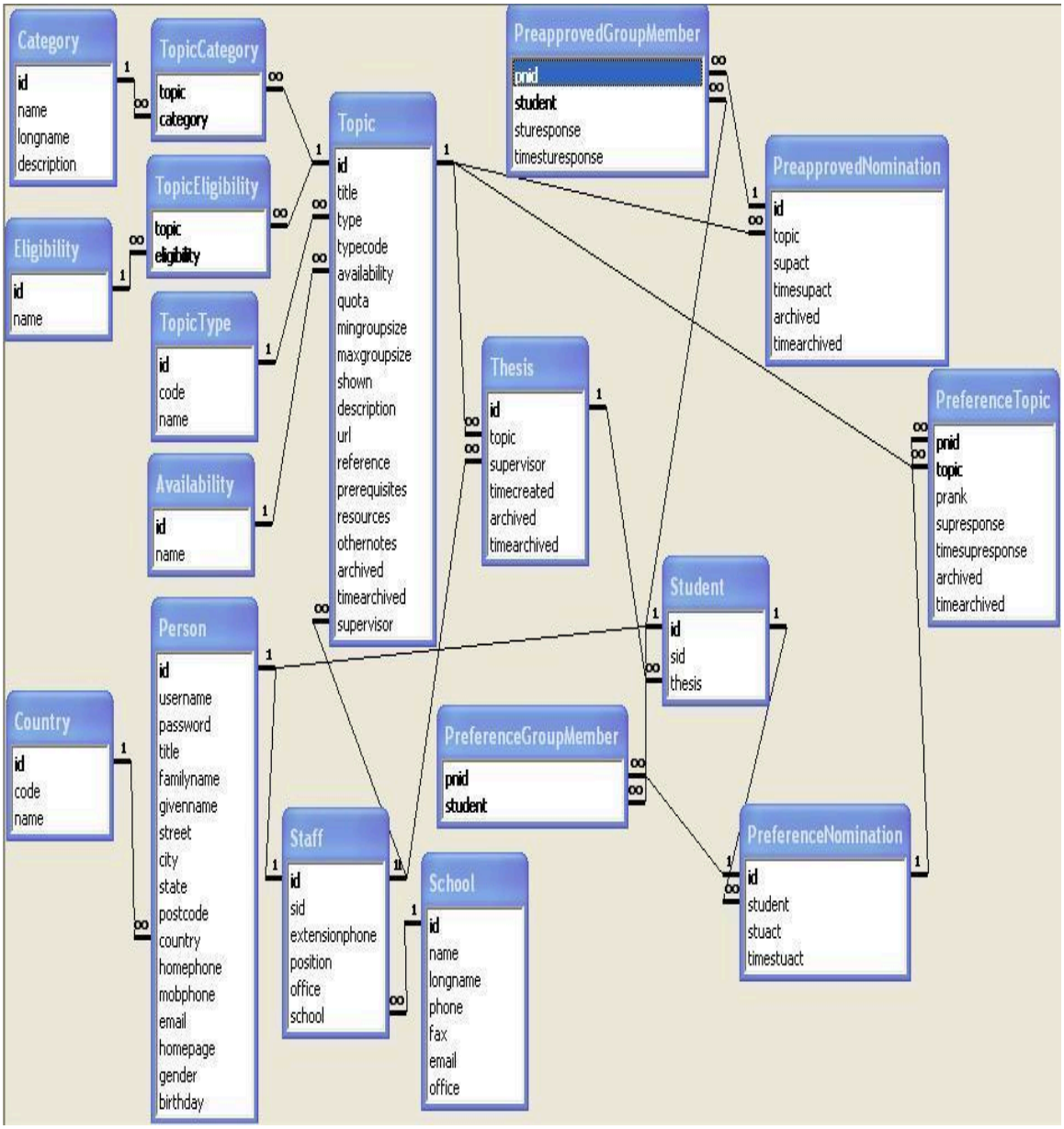


Figure 4.6: Relational Schema for Thesis Management Database

Person

Field	Type	Constraints	References
id (primary key)	serial		
username	varchar (10)	unique, not null	
password	char (15)	not null	
title	varchar (30)		
familyname	varchar (30)	not null	
givenname	varchar (40)	not null	
street	varchar (50)		
city	varchar (50)		
state	varchar (50)		
postcode	varchar (10)		
country	integer		Country (id)
homephone	varchar (20)		
mobphone	varchar (15)		
email	varchar (40)	unique	
homepage	varchar (60)		
gender	varchar (6)	in (‘male’, ‘female’)	
birthday	date		

Users that are registered on the system are stored on the *Person* table. The username is unique to avoid clashes between the login accounts. The *password* that is stored on the table is encrypted using digest encryption algorithm giving constant length irrespective of the password length. The *email* that is stored in must be unique and should conform to RFC-2822 standard. This is critical since mail notifications will be sent to users giving the information about related events.

Country

Field	Type	Constraints	References
id (primary key)	serial		
code	char (3)	not null	
name	varchar (40)	not null	

The *Country* table is used to store the list of all available countries that is part of the address field in the person table. The reason why a separate table is used to store countries is that because it provides finer grained control of which country is valid for an

address.

Staff

Field	Type	Constraints	References
id (primary key)	integer		Person (id)
sid	integer	unique, not null	
extensionphone	varchar (4)		
position	varchar (50)		
office	varchar (50)		
school	integer		School (id)

All the Academic Staffs who are eligible to supervise thesis project is stored on the *Staff* table. The field sid (Staff ID) serves as unique identifier for each Staff Members.

Student

Field	Type	Constraints	References
id (primary key)	integer		Person (id)
sid	integer	unique, not null	
thesis	integer		Thesis (id)

All the Students that are eligible to undertake thesis project is stored on the *Student* table. The field sid (Student ID) serves as unique identifier for each Student.

School

Field	Type	Constraints	References
id (primary key)	serial		
name	varchar (20)	not null	
longname	varchar (60)		
phone	varchar (14)		
fax	varchar (14)		
email	email (40)		
office	varchar (50)		

The *School* table stores all the School that offers the thesis topics. The fact that a topic is offered by a particular school relies on the association between supervisor and school itself, i.e. Staff A working for the School B offering Topic C means that Topic C is

offered by school B.

Topic

Field	Type	Constraints	References
id (primary key)	serial		
title	varchar (150)		
type	integer		TopicType (id)
typecode	char(4)		
supervisor	integer		Staff (id)
availability	integer		Availability (id)
quota	integer		
mingroupsize	integer		
maxgroupsize	integer		
shown	varchar (3)	in ('yes', 'no')	
description	text		
url	text		
references	text		
prerequisites	text		
resources	text		
othernotes	text		
archived	varchar (3)	in ('yes', 'no')	
timearchived	timestamp		

All the available topics in the Thesis Management System are stored on the *Topic* table. Most of the fields such as *title*, *type*, *typecode*, *supervisor*, *availability*, *description*, *url*, *references*, *prerequisites*, *resources*, and *othernotes* in the table serve as information for the topic. Others such as *quota*, *mingroupsize*, and *maxgroupsize* serve as constraints in the topic allocation process. The reason why the field *prerequisites* is not selected as one of the constraints is because it is possible that a supervisor can approve students to take a particular topic despite of the fact that they don't meet the prerequisites. Hence, the field *prerequisites* can just be stored as a text in the table.

TopicType

Field	Type	Constraints	References
id (primary key)	serial		
code	varchar (3)		
name	varchar (30)	not null	

All types of topic are stored on the *TopicType* table. The topic types are also identified by the *code*, i.e. RES for "Research", DEV for "Development", R&D for "Research and Development", and COR for "Course Oriented".

Availability

Field	Type	Constraints	References
id (primary key)	serial		
name	varchar (15)	not null	

All the possible availability values for the topic are stored on the Availability Table. This includes "Session 1", "Session 2", "Either Session", "Unique", and "Not Available".

Eligibility

Field	Type	Constraints	References
id (primary key)	serial		
name	varchar (30)	not null	

TopicEligibility

Field	Type	Constraints	References
topic (primary key)	integer		Topic (id)
eligibility (primary key)	integer		Eligibility (id)

The eligibility information for a topic is stored on 2 tables. The *Eligibility* table stores the all possible eligibility values for topic and the *TopicEligibility* table stores the mapping between topic and eligibility values. In the Thesis Management Database that is used on CSE, the eligibility values are stored as "All Students", "CSE Students", and "BINF Students". This lead to problems when new eligibility value, such as "INFS Students",

comes into place. We must also consider the new combinations "CSE Students and BINF Students", "CSE Students and INFS Students", and "BINF Students and INFS Students". Using the above table structure solves this problem.

Category

Field	Type	Constraints	References
id (primary key)	serial		
name	varchar (20)	not null	
longname	varchar (60)	not null	
description	text		

TopicCategory

Field	Type	Constraints	References
topic (primary key)	integer		Topic (id)
category (primary key)	integer		Category (id)

A topic could belong to one or more categories. The above 2 tables serve this semantic requirement. The *Category* stores all the possible categories for topic in the database, while the *TopicCategory* table stores the information about categorization of topics.

Thesis

Field	Type	Constraints	References
id (primary key)	serial		
topic	integer		Topic (id)
supervisor	integer		Staff (id)
timecreated	timestamp		
archived	varchar (3)	in ('yes', 'no')	
timearchived	timestamp		

The information about thesis project that students undertake is stored on the *Thesis* table. As noted earlier, the *Thesis* table also serves as a grouping of students, i.e. students working on the same team are assigned the same thesis id. The field *supervisor* is provided to handle changes to the supervisor during the commencement of the thesis project.

PreappovedNomination

Field	Type	Constraints	References
id (primary key)	serial		
topic	integer		Topic (id)
supact	varchar (8)	in ('create', 'cancel')	
timesupact	timestamp		
archived	varchar (3)	in ('yes', 'no')	
timearchived	timestamp		

PreapprovedGroupMember

Field	Type	Constraints	References
pnid	integer		Preapproved-Nomination (id)
student	integer		Student (id)
sturesponse	varchar (8)	in ('on hold', 'accept', 'reject')	
timesturesponse	timestamp		

The approvals that are granted from the supervisors are stored on the *PreapprovedNomination* table. Each of the students that are granted the related approval is stored on the *PreapprovedGroupMember* table. An approval has *supact* field flagged with the string 'create' and the *timesupact* set to the time when the preapproval is granted. In addition, all the entered group members entries on the *PreapprovedGroupMember* table have the field *sturesponse* set to 'on hold' and timesturesponse to the time when the preapproval is given. If the approval is cancelled by the supervisor the field *supact* will be set to 'cancel', the field *timesupact* will be set to the time when the supervisor cancel the approval and the approval entry will be archived. Once student gives response to the granted approval, the field *sturesponse* is set to either 'accept' or 'reject', the field timesturesponse is set to appropriately and the associated preapproved nomination entry will be archived.

Preference Nomination

Field	Type	Constraints	References
id (primary key)	serial		
student	integer		Student (id)
stuact	varchar (8)	in (‘create’, ‘cancel’)	
timestuact	timestamp		

Preference Topic

Field	Type	Constraints	References
pnid (primary key)	integer		Preference-Nomination(id)
topic (primary key)	integer		Topic(id)
prank	integer		
supresponse	varchar (30)	in (‘inactive, ‘queuing’, ‘waiting for approval’, ‘accept’, ‘reject’, ‘timeout’)	
timesupresponse	timestamp		
archived	varchar (3)	in (‘yes’, ‘no’)	
timearchived	timestamp		

PreferenceGroupMember

Field	Type	Constraints	References
pnid	integer		Preference-Nomination(id)
student	integer		Student(id)

Each of the preference nomination that is submitted by the student will be stored on the *PreferenceNomination* Table. Once a student submits a preference nomination, an entry with field *stuact* set to ‘create’ and field *timestuact* set to the submission time is entered to the *PreferenceNomination* table. Each of the preference group members will be entered to the *PreferenceGroupMember* Table and each of the preference topics listed will be entered to the database along with the associated rank to the *PreferenceTopic* Table. If the other group members have already nominated in the same way, all of the preference

topics stored on the table will be updated as follows: The *supresponse* field of preference topic with the highest rank will be set to 'waiting for approval' and the others will be set to 'queuing'. Otherwise, all the preference topics will have *supresponse* field set to 'inactive'. The preference topics with the supresponse field set to 'waiting for approval' will be requested to the supervisor of the topic.

Once a supervisor response to the topic request, the *supresponse* field of the preference topic will be updated to either 'accept' or 'reject' and the *timesupresponse* will be set to the time when the response is given. If timeout occurs during the topic request, the *supresponse* field will be set to 'timeout' and the *timesupreponse* will be set to the time when the timeout occurs. In any of this case, the preference topic will be archived. As we can see there is no field *archived* and *timearchived* in the Preference Nomination table since an archived preference nomination is indicated by having all the preference topics archived.

5.2 Data Flow Modeling Diagram

Modeling functional decomposition of the system at the conceptual is important to communicate the idea of complex system. Data Flow Diagram (DFD) provides a mechanism to serve this goal. The advantage of DFD is that it less rigorous than any other object-oriented modeling tools in software design. Hence, the resulting model is a solution that is independent of the implementation choice.

5.2.1 Context Diagram

The Context Diagram representing the highest level view of the entire Thesis Management System is shown in Figure 4.7. Because this is meant to illustrate the high level business activities of the overall Thesis Management System, it only shows the major data flows entering and leaving the system. Most of the output produced in the system is email notifications that are represented by *notification* data flows.

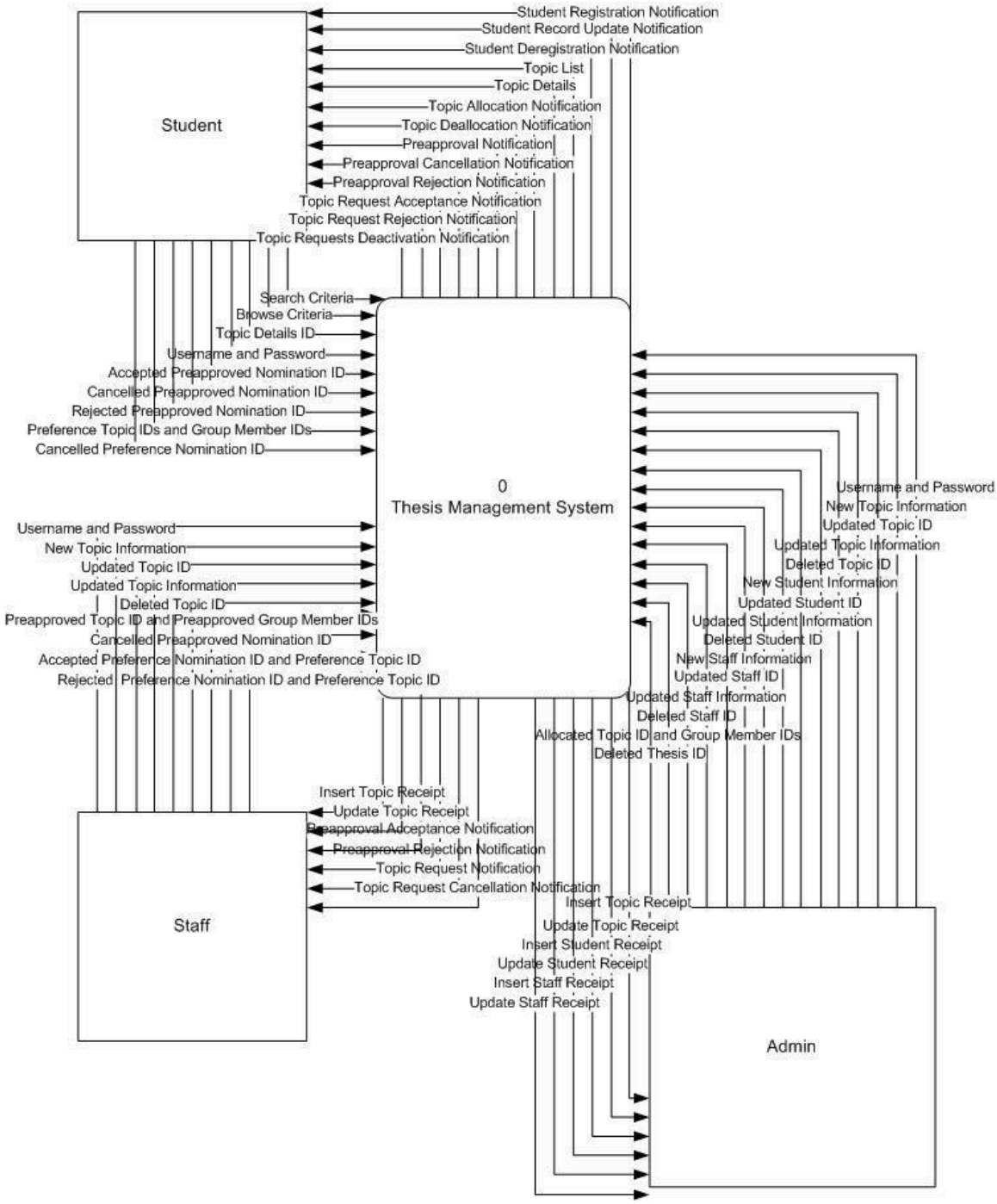


Figure 4.7. Context Diagram

5.2.2 Diagram 0

Diagram 0 is shown in Figure 4.7 and Figure 4.8. It illustrates the major activities involving the major data stores of the system. The data stores itself corresponds to the actual tables that has been described in the previous subsection, the session variables and the cron table that contains pre-scheduled operations. We will import several conventions that are used by Kendall and Kendall for database operations, i.e. an update operation is represented by double headed arrow and a delete operation is represented by dataflow labeled with deleted ID going into the data store. Notice that process 24, *Monitor Topic Request Expiration,* doesn't have corresponding input and output in the Context Diagram. This is because *Monitor Request Expiration* is considered as an internal process to the system.

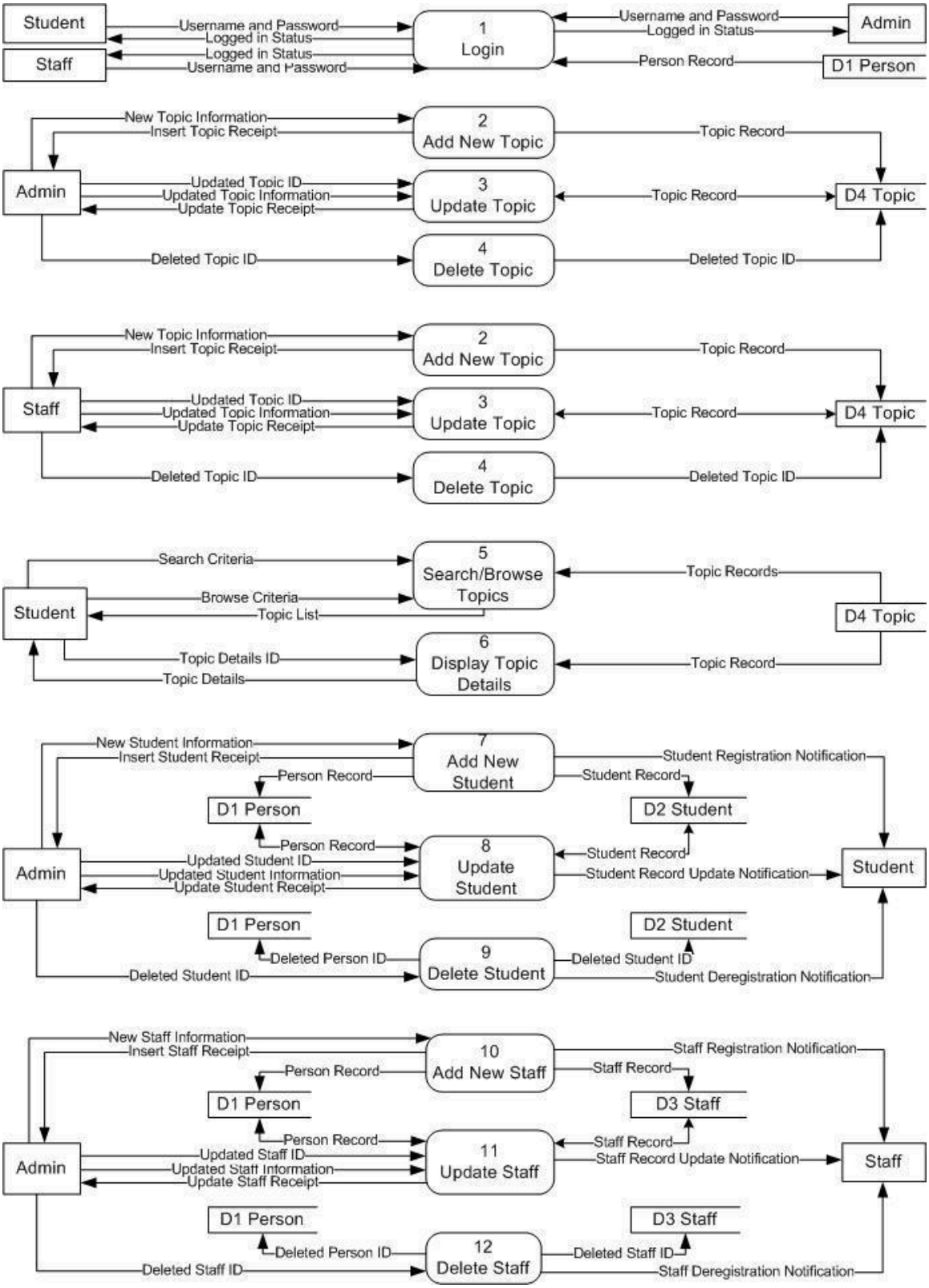


Figure 4.8. Diagram 0 (part 1)

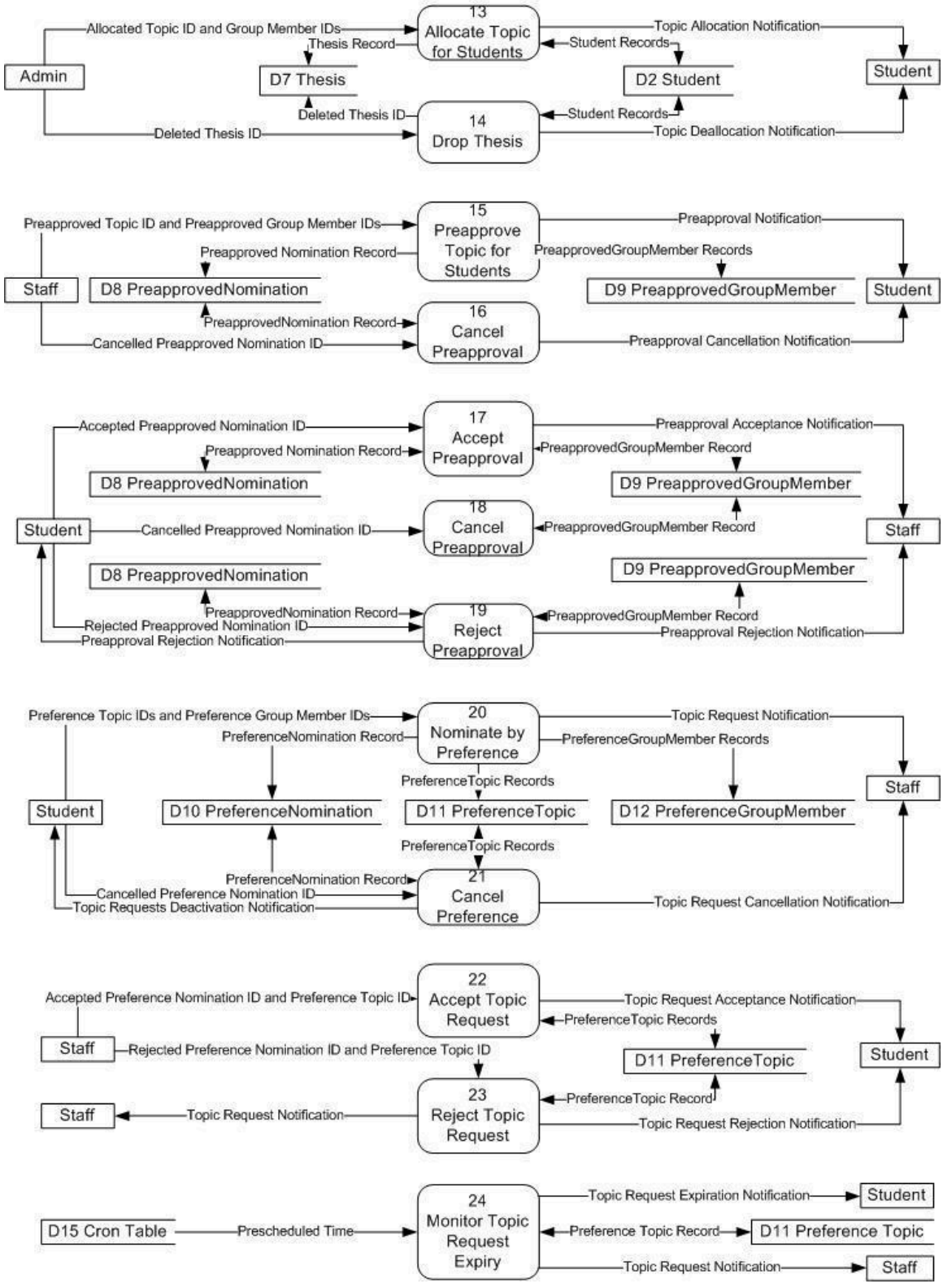


Figure 4.9. Diagram 0 (part 2)

5.2.3 Child Diagrams

Each process on the diagram 0 is exploded to produce more detailed child diagrams. The process in child diagrams use decimal point that is progressing as each process is performed. This convention allows the reader to trace the order of the execution of the processes. You will notice that there is missing diagram 6, this is because process 6 can't be further decomposed into smaller process.

Diagram 1 – Login

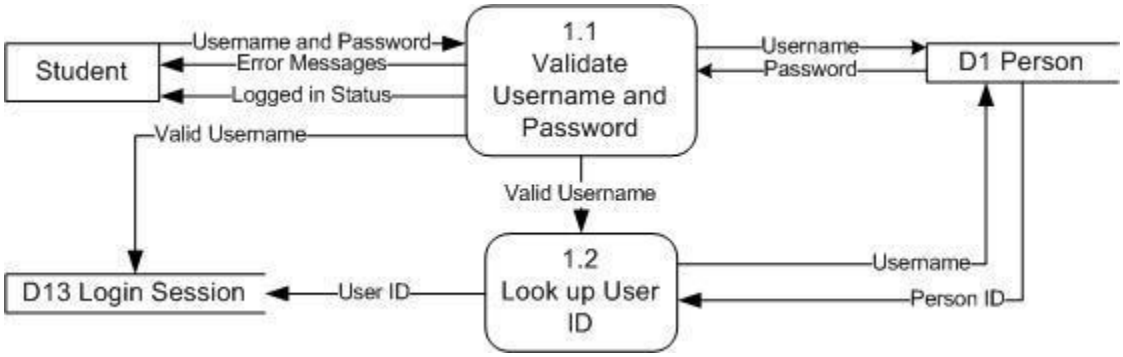


Figure 4.10. Diagram 1 – Login

Diagram 2 – Add New Topic

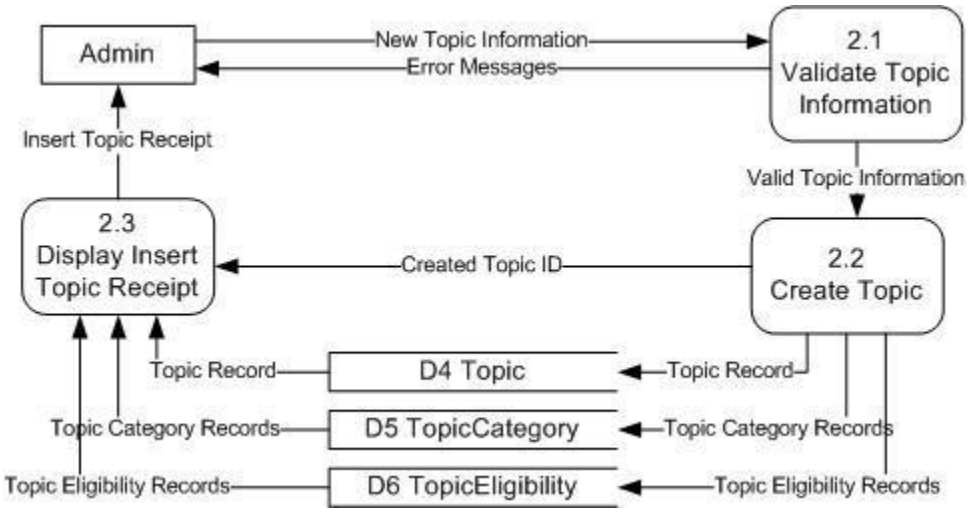


Figure 4.11. Diagram 2 – Add New Topic

Diagram 3 – Update Topic

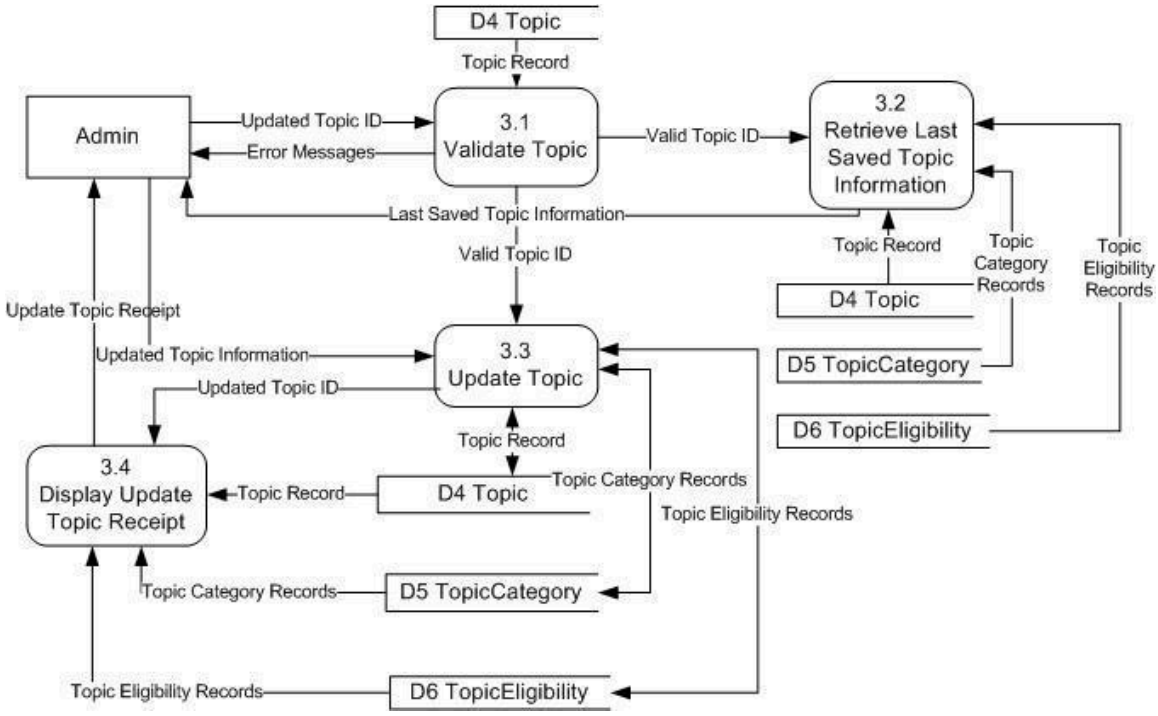


Figure 4.12. Diagram 3 – Update Topic

Diagram 4 – Delete Topic

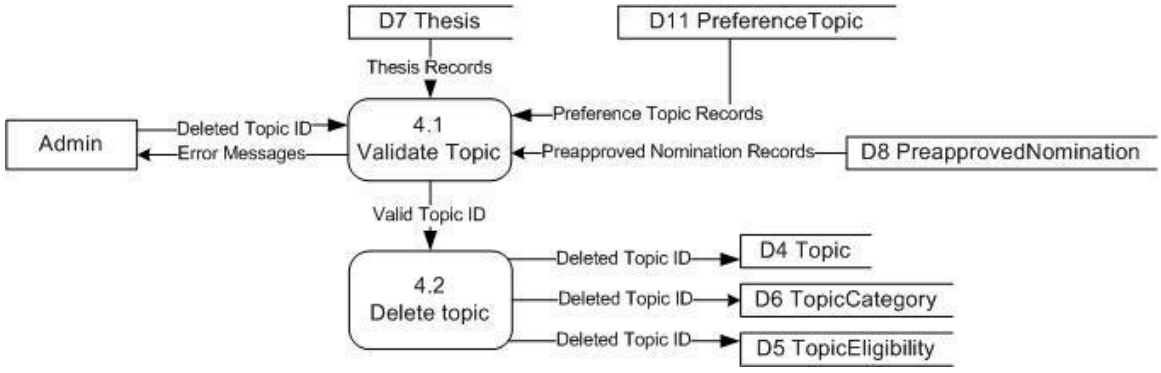


Figure 4.13. Diagram 4 – Update Topic

Diagram 5 – Search/Browse Topics

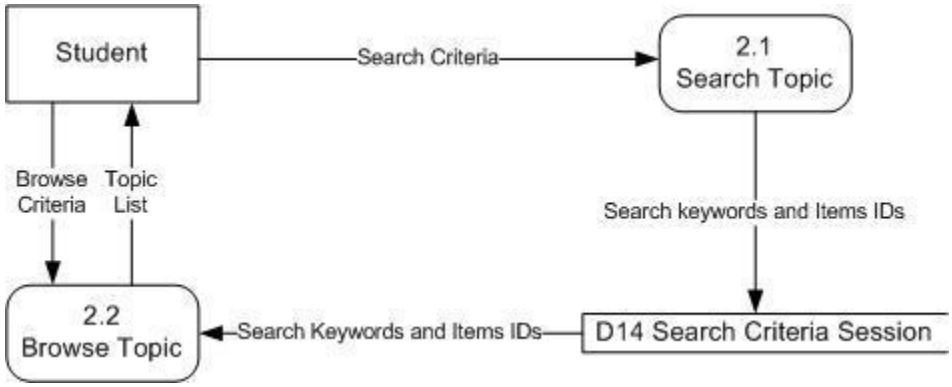


Figure 4.14. Diagram 5 – Search/Browse Topics

Diagram 7 – Add New Student

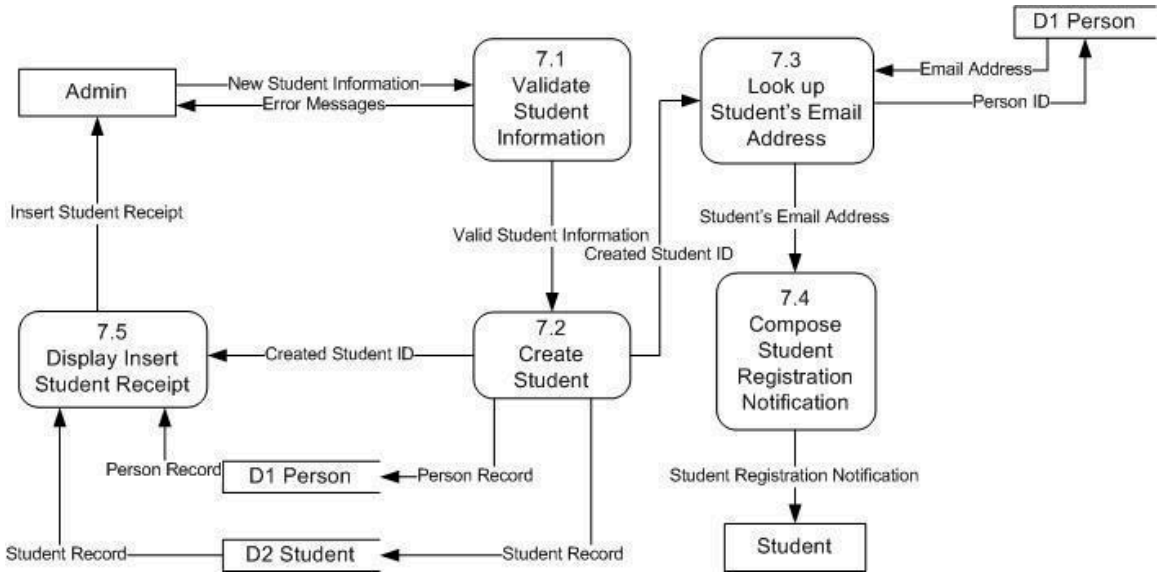


Figure 4.15. Diagram 7 – Add New Student

Diagram 8 – Update Student

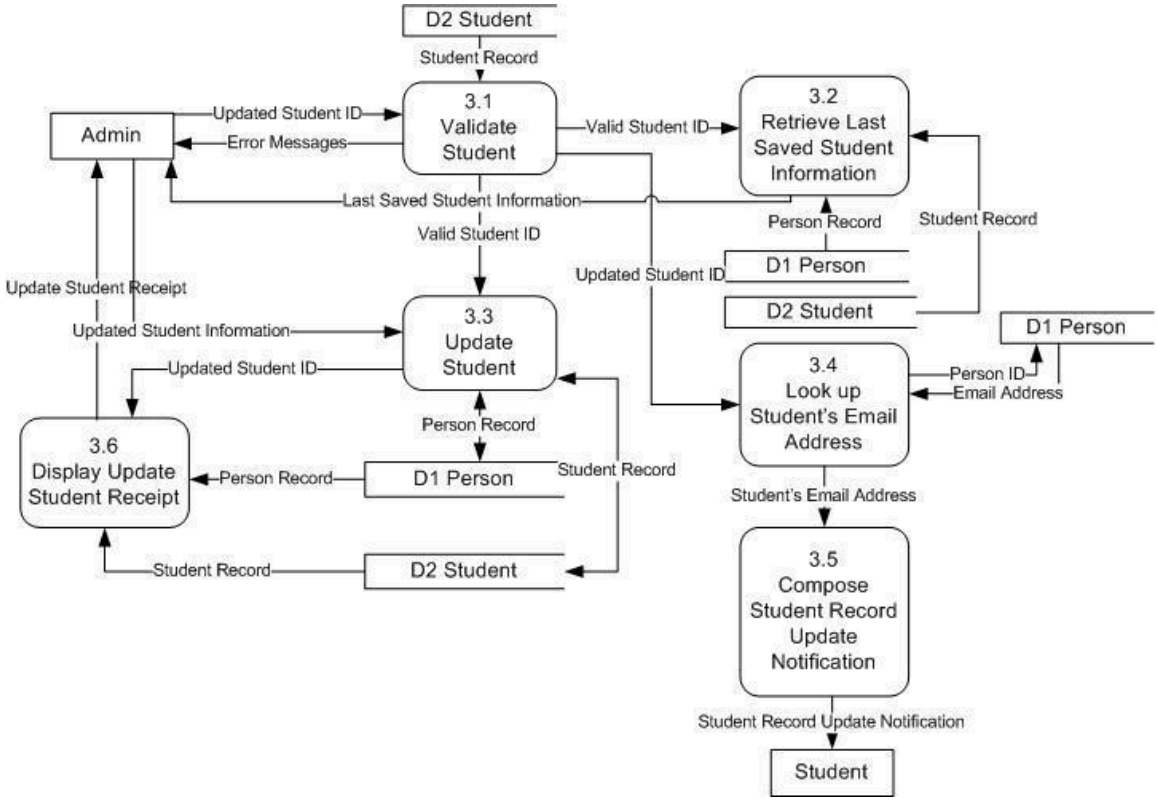


Figure 4.16. Diagram 8 – Update Student

Diagram 9 – Delete Student

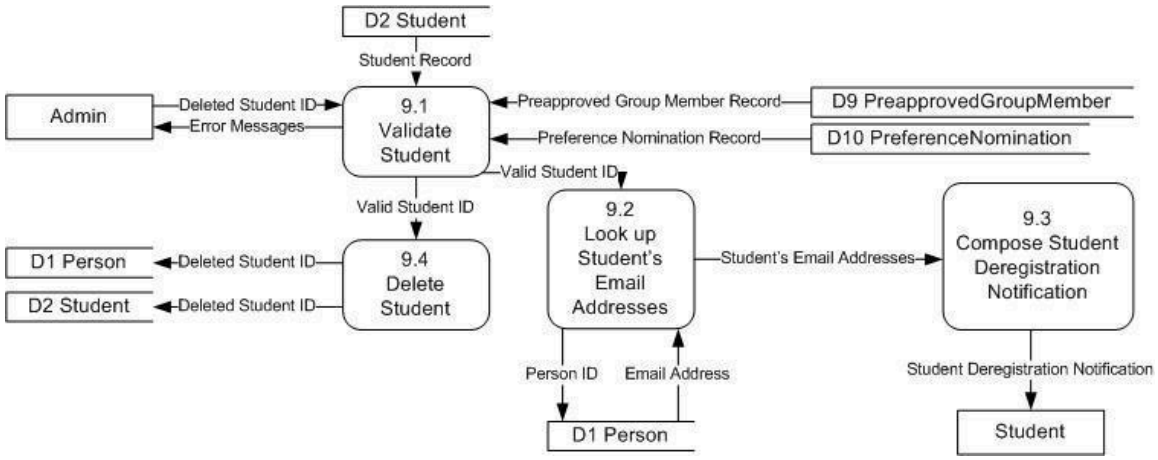


Figure 4.17. Diagram 9 – Delete Student

Diagram 10 – Add New Staff

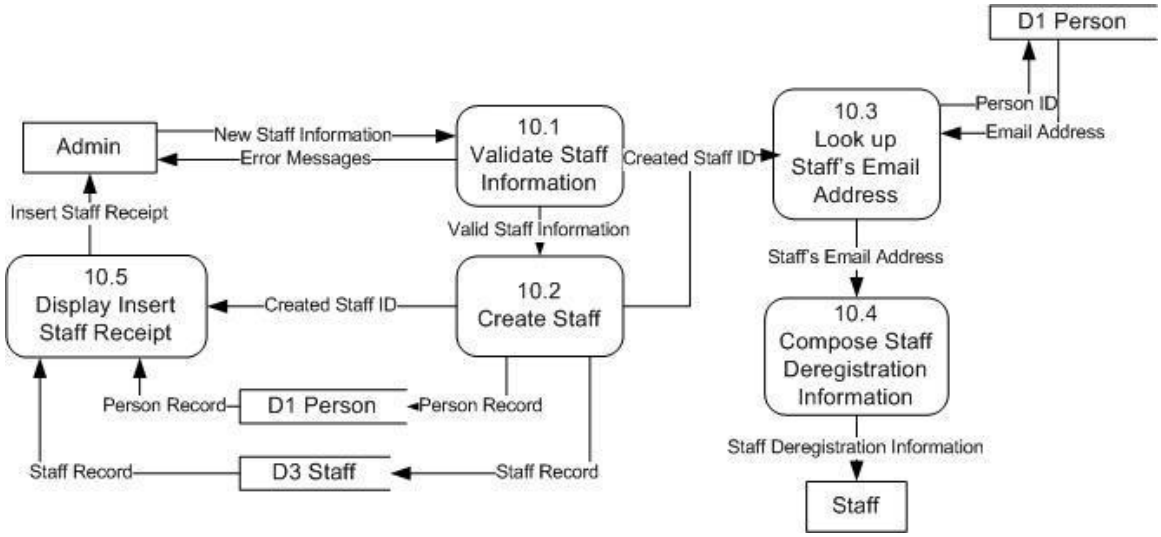


Figure 4.18. Diagram 10 – Add New Staff

Diagram 11 – Update Staff

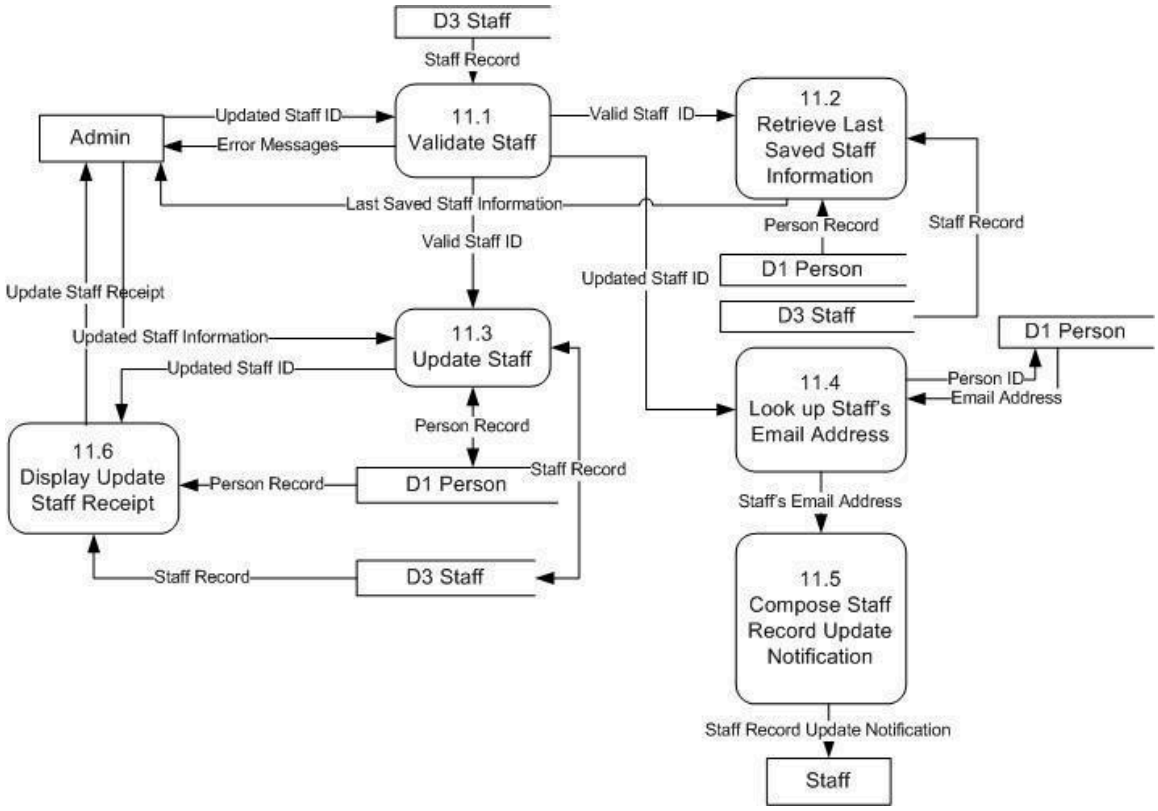


Figure 4.19. Diagram 11 – Update Staff

Diagram 12 – Delete Staff

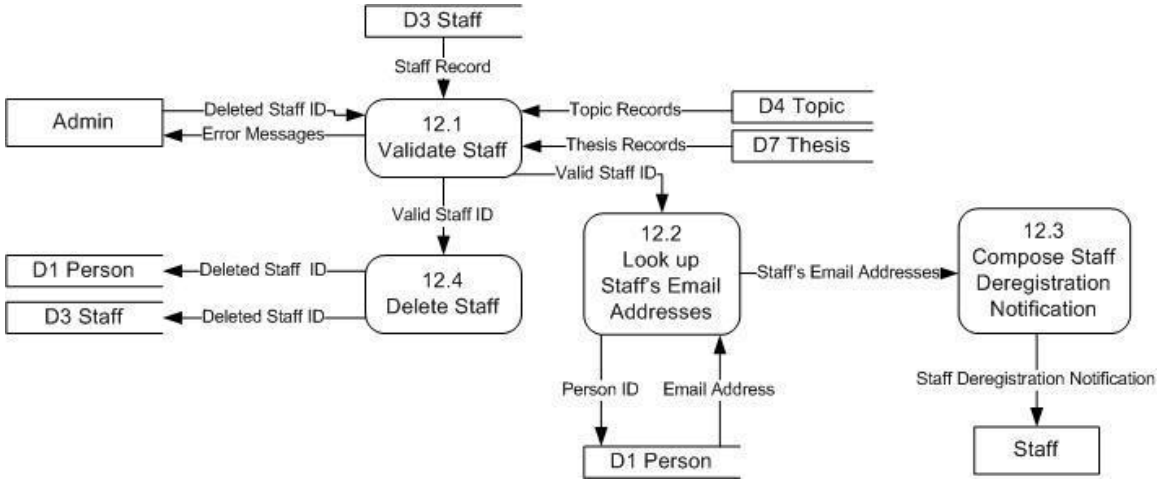


Figure 4.20. Diagram 12 – Delete Staff

Diagram 13 – Allocate Topic for Students

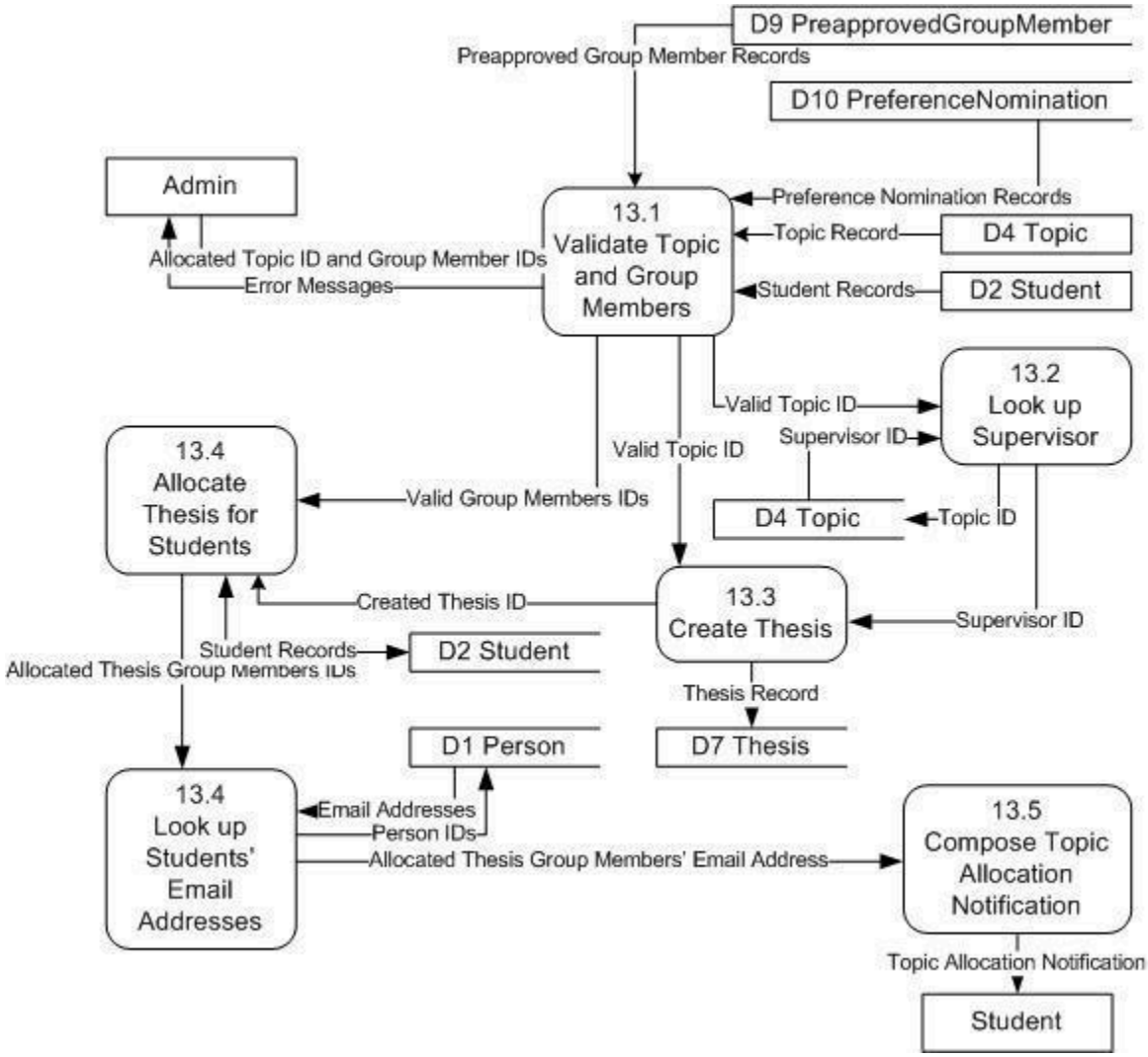


Figure 4.21. Diagram13 – Allocate Topic for Students

Diagram 14 – Drop Thesis

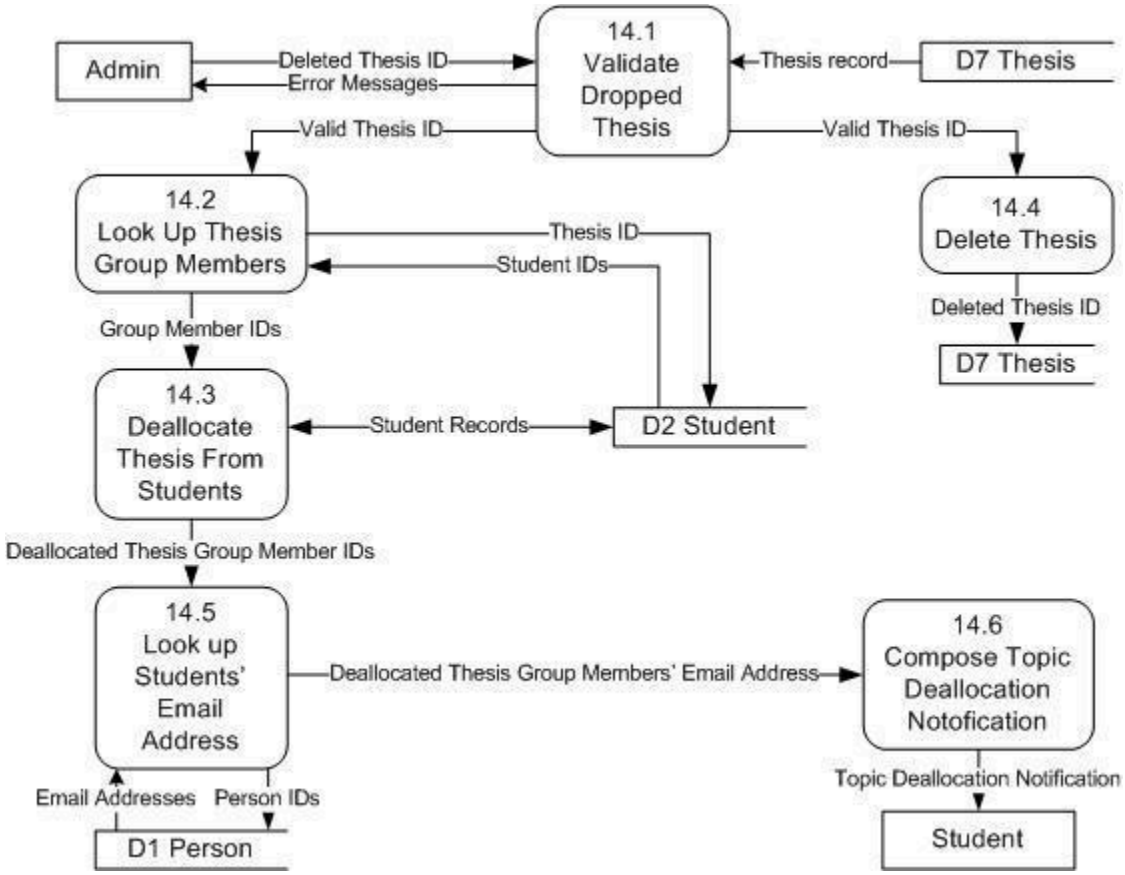


Figure 4.22. Diagram 14 –Drop Thesis

Diagram 15 – Preapprove Topic for Students

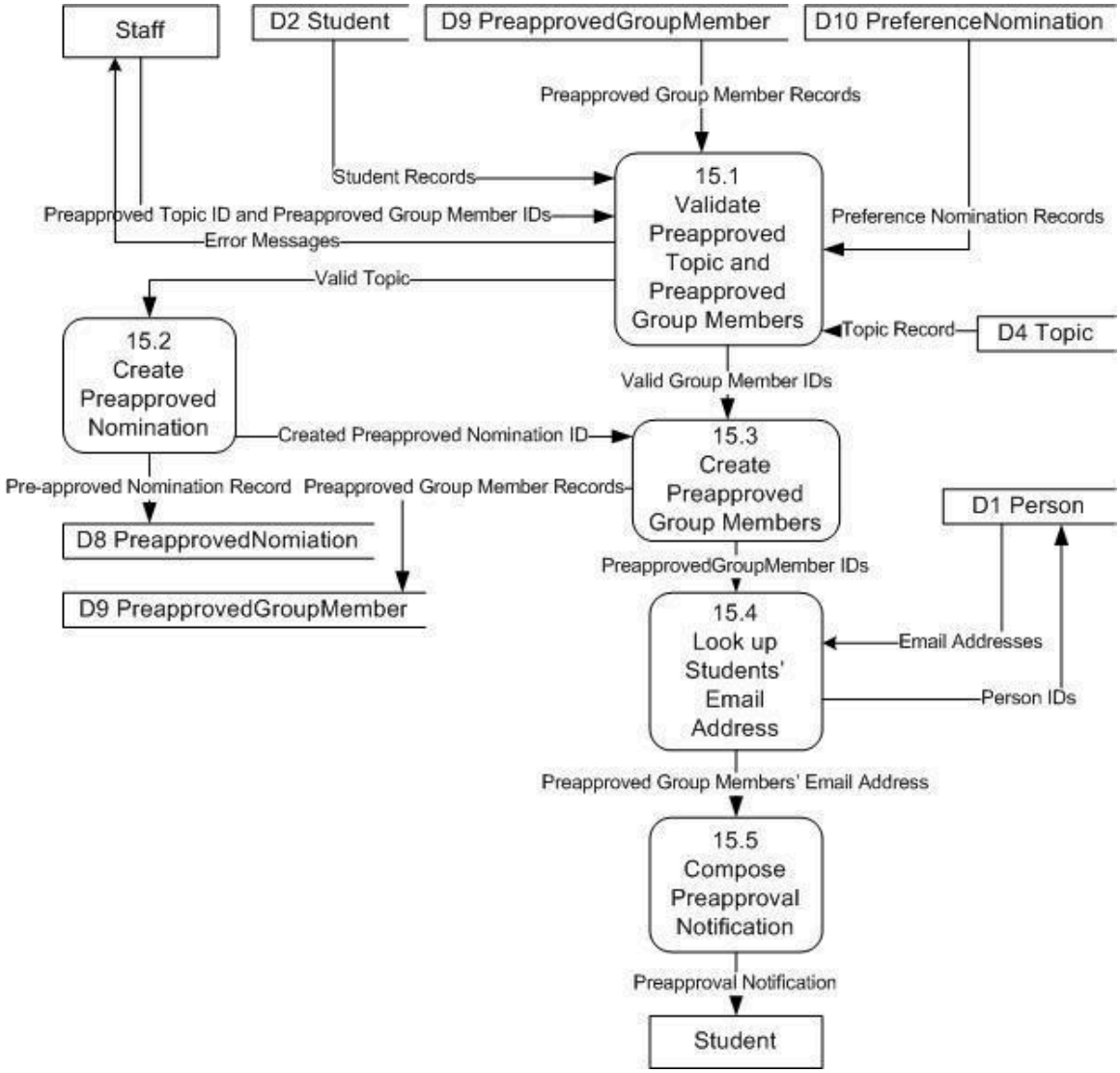


Figure 4.23. Diagram 15- Preapprove Topic for Students

Diagram 16 – Cancel Preapproval

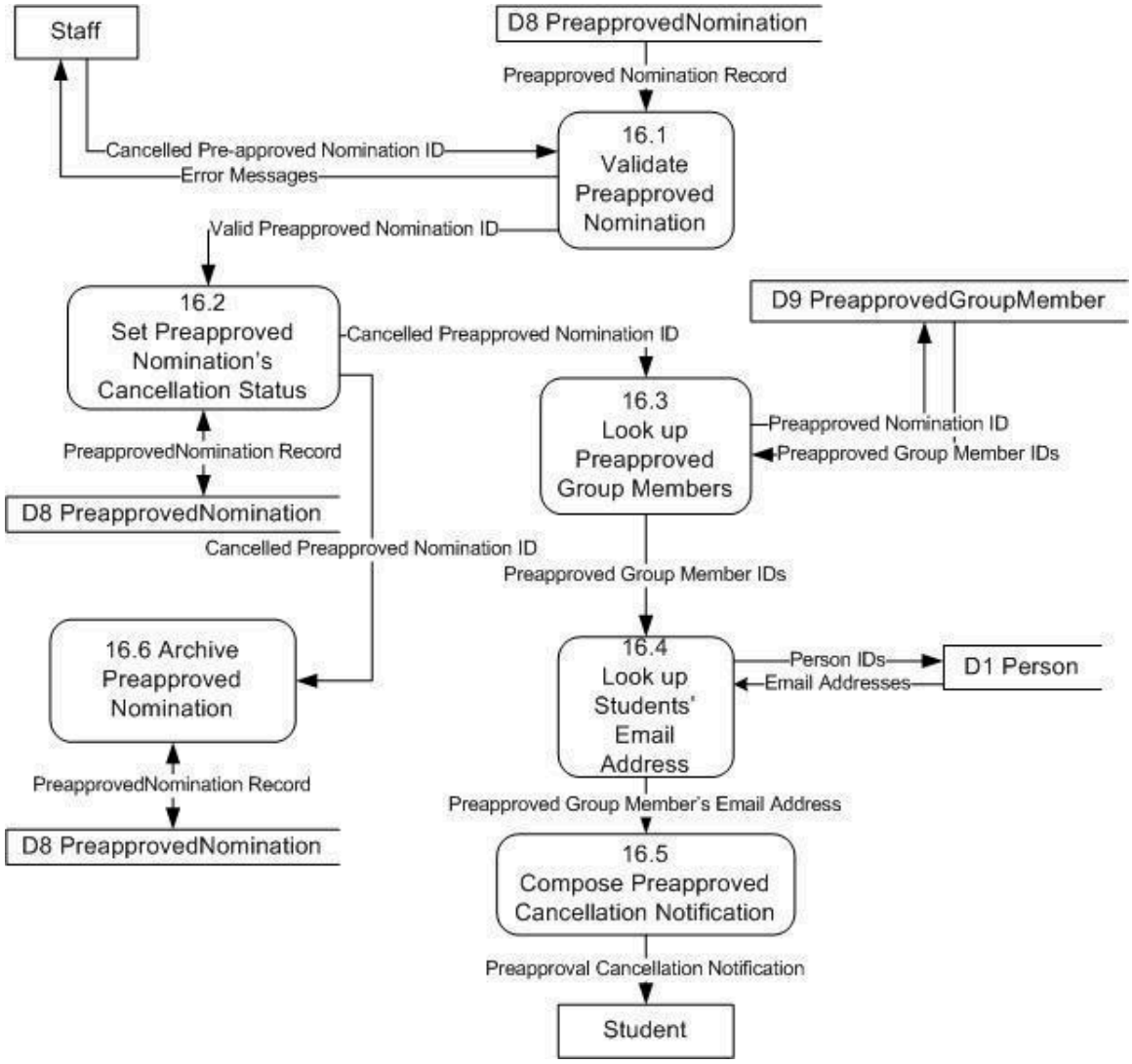


Figure 4.24. Diagram 16 – Cancel Preapproval

Diagram 17 – Accept Preapproval

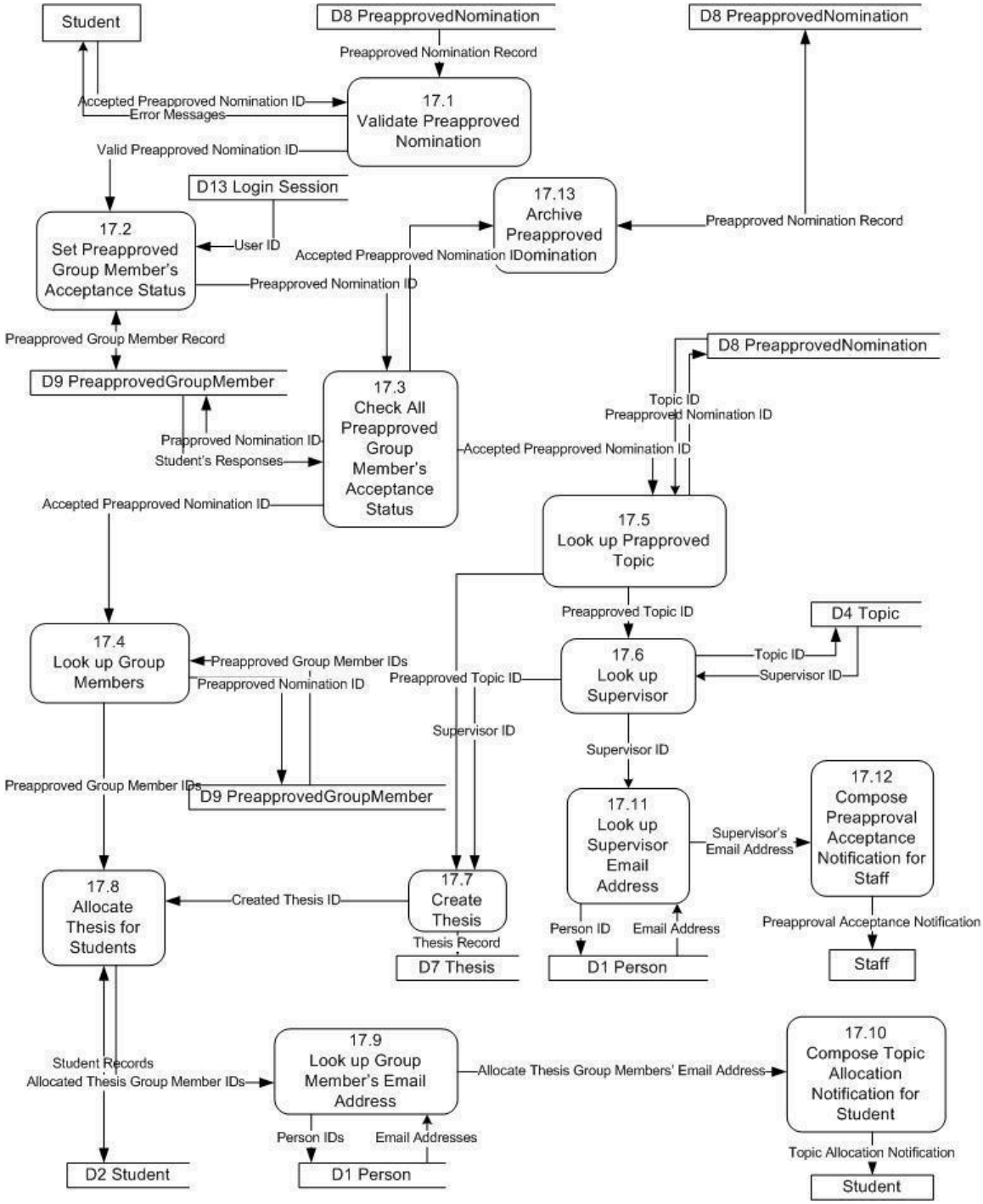


Figure 4.25. Diagram 17 – Accept Preapproval

Diagram 18 – Cancel Accepting Preapproval

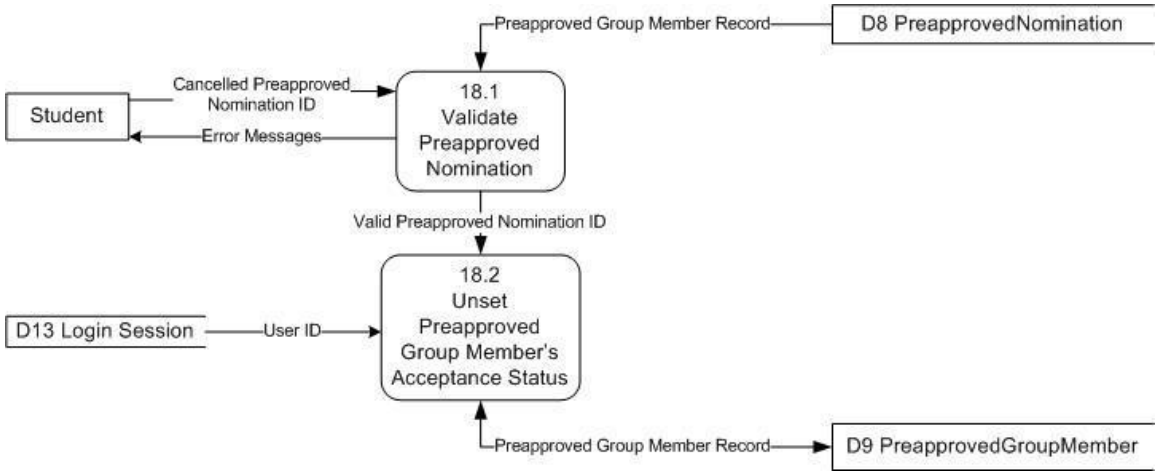


Figure 4.26. Diagram 18 –Cancel Accepting Preapproval

Diagram 19 – Reject Preapproval

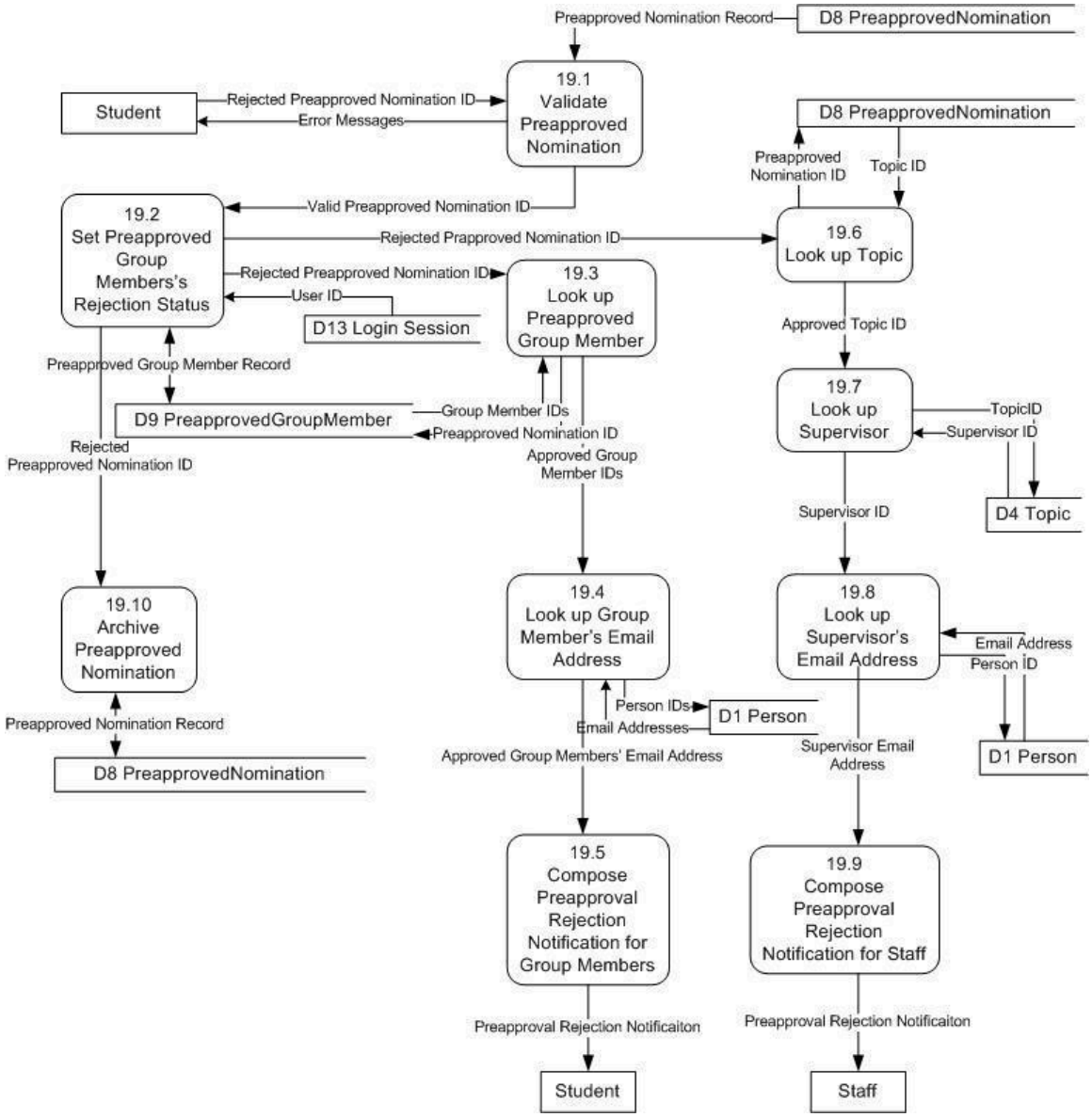


Figure 4.27. Diagram 19 – Reject Preapproval

Diagram 20 – Nominate by Preference

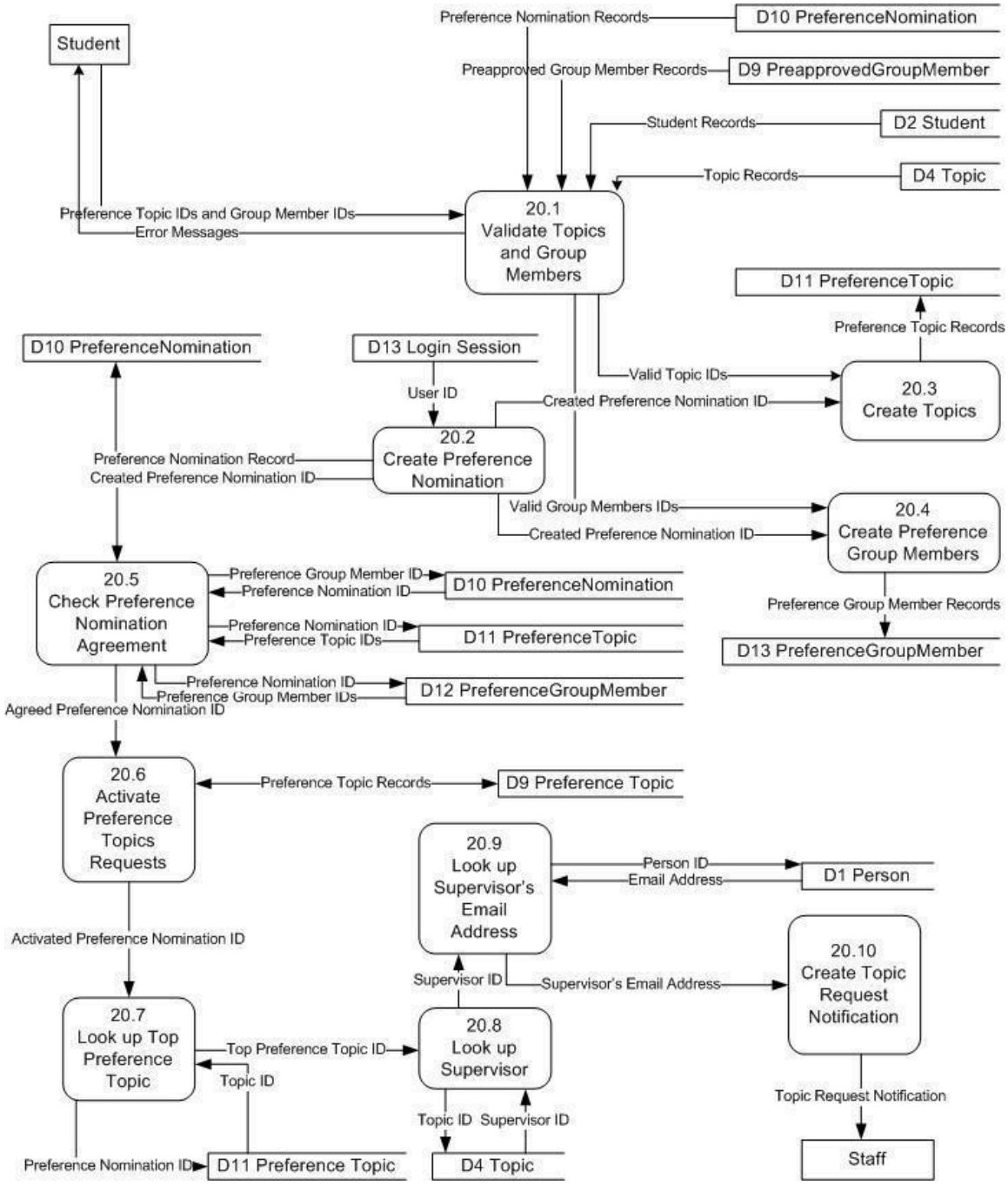


Figure 4.28. Diagram 20 – Nominate by Preference

Diagram 21 – Cancel Preference Nomination

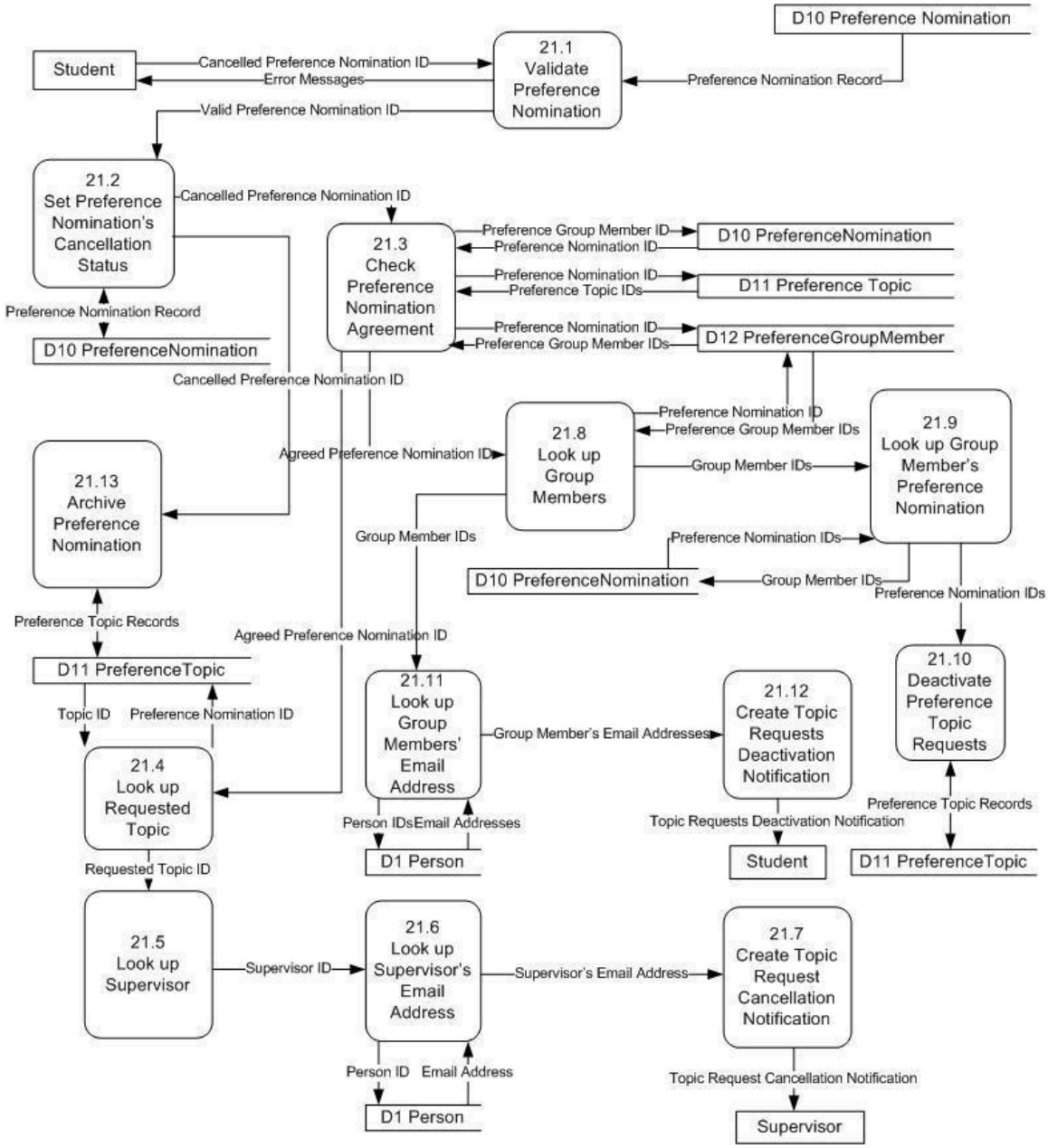


Figure 4.29. Diagram 21 – Cancel Preference Nomination

Diagram 22 – Accept Topic Request

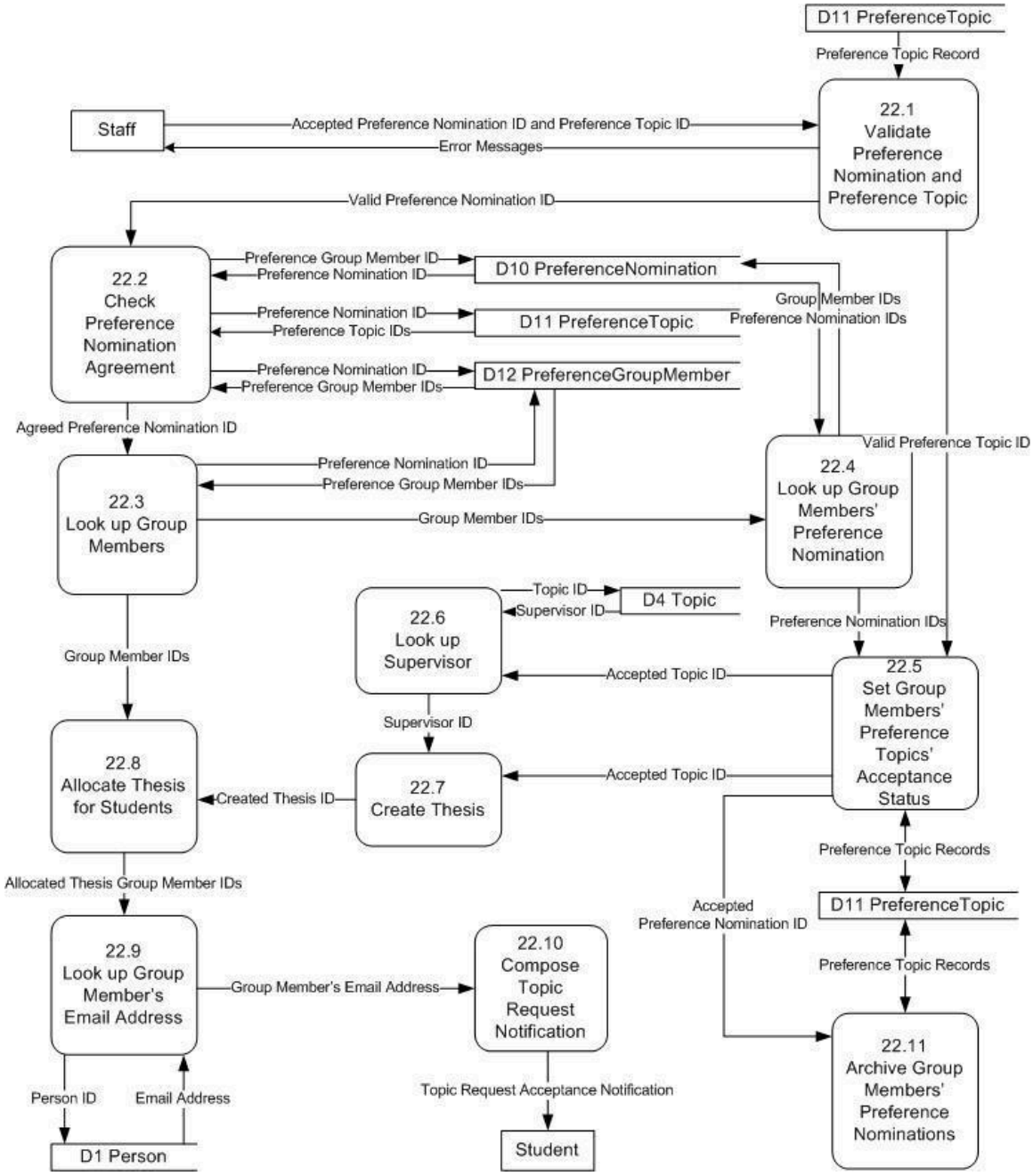


Figure 4.30. Diagram 22 – Accept Topic Request

Diagram 23 – Reject Topic Request

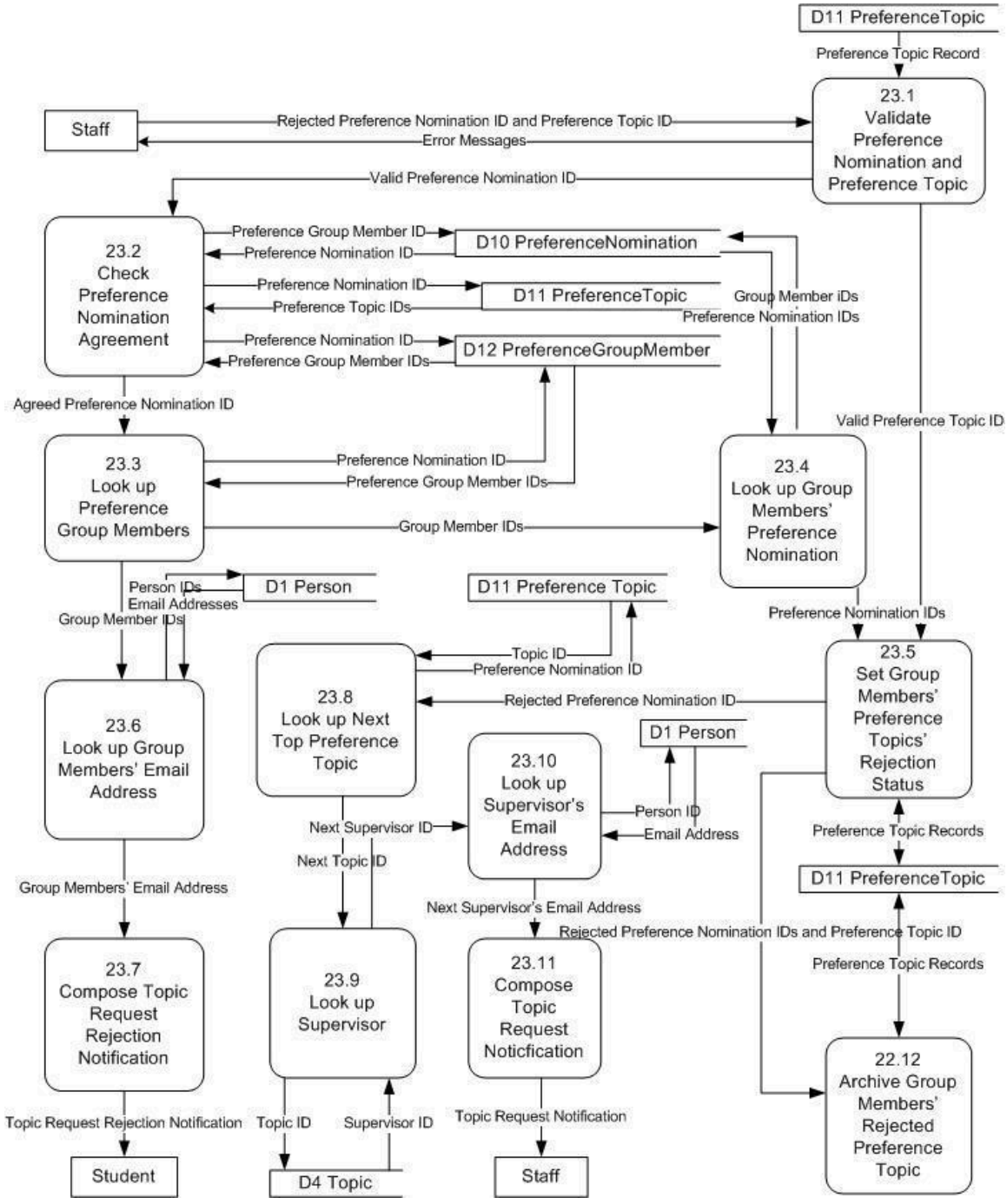


Figure 4.31. Diagram 23 – Reject Topic Request

Diagram 24 – Monitor Topic Request Expiry

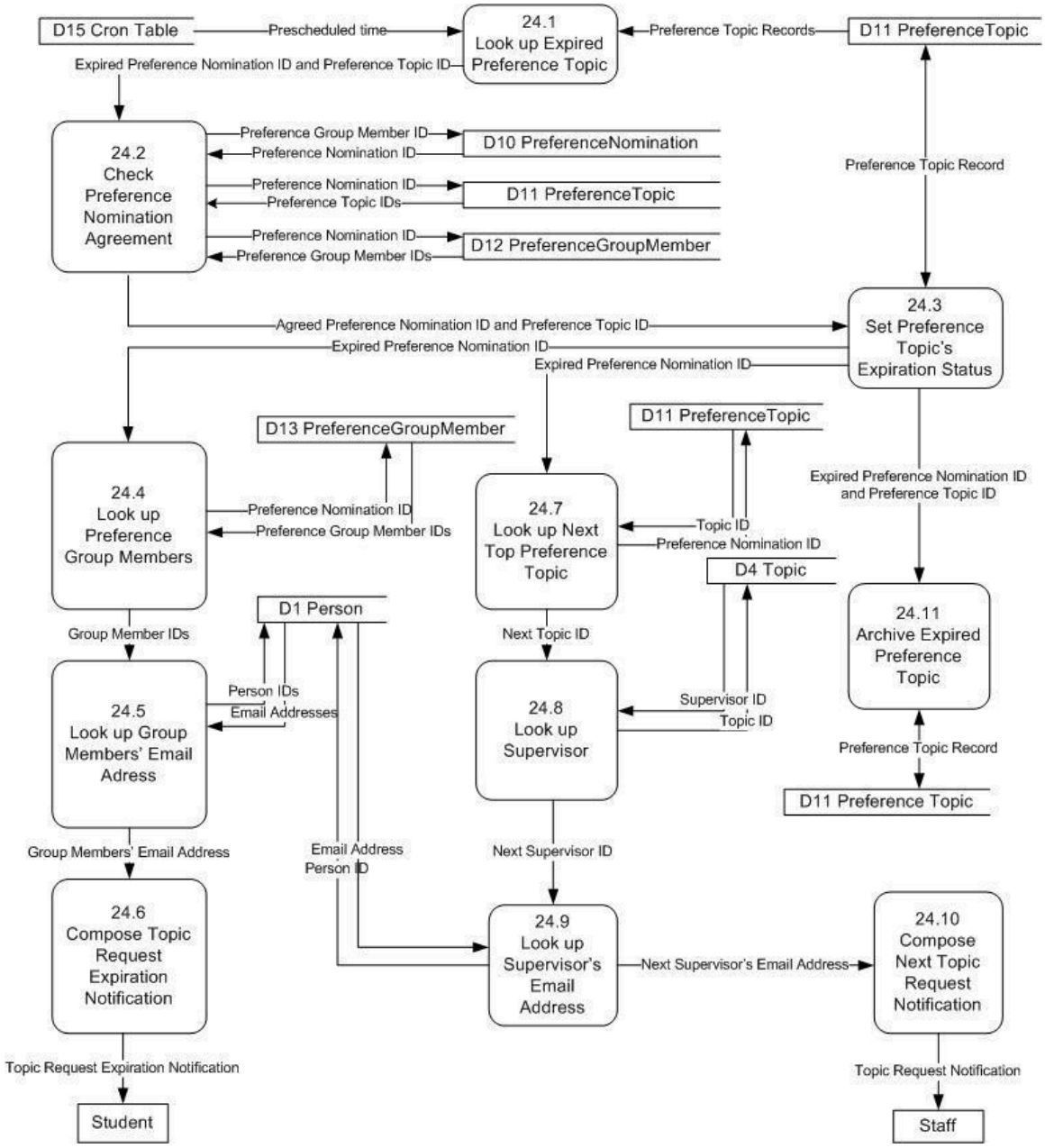


Figure 4.32. Diagram 24 – Monitor Topic Expiry

CHAPTER 6

Implementation

In this chapter, we will be discussing about the implementation of WebTMS. To give a better impression, the user will also be brought into the interface tour of the user interface of the system.

6.1 System Architecture

Our thesis management system is implemented using database applications that are integrated with the Web. It is built on a three tier model as shown in Figure 6.1. At the very end of the system lies the database tier where the all the queries is handled. On top of the database tier is the middle tier that contains most of the business logic of the application and the communication link between the other 2 tiers. On top of the middle tier is the client tier, where the web browsers present the application to the users and handles the interaction between the users and the back end of the application.

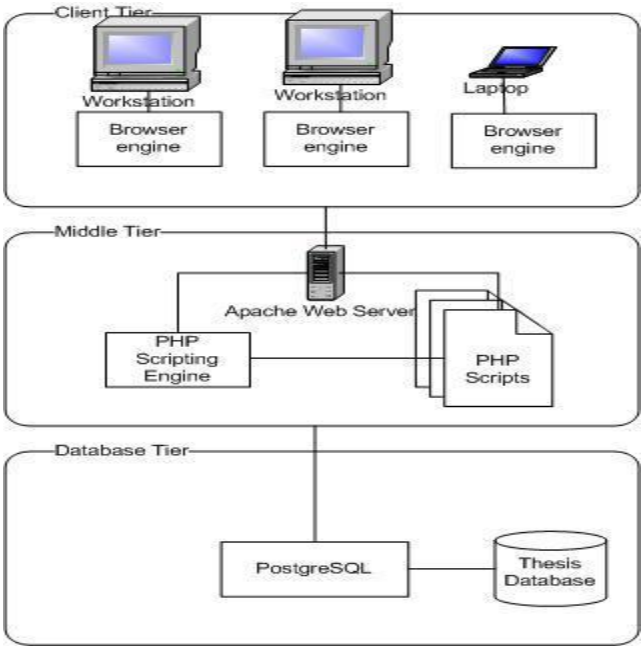


Figure 6.1. System Architecture

In particular, the client tier collects data from the user through the web browser. From HTTP protocol, web browser sends a request for a resource that is located on the web server. Upon receiving this request, the Web server attempts to interpret the PHP scripts by running the scripting engine. This is where the input is being processed and formulated into database queries. The PHP scripting engine then opens a connection to the postgresSQL and sends queries through the opened connection. The postgresSQL process the queries and sends back the results to the scripting engine. The PHP engine structured the results into HMTL resource and sends it back to the web browser. At the end, the web browser displays the HTML resource appropriately to user.

The advantage of this three tier solution is that we have the thin client architecture. That means we don't have to build, install or configure additional software in order to use the system. All the standardized web browsers work fine with the database applications, since most of the application logic is stored on the middle tier. This improves the accessibility of the system for users.

6.2 Interface Tour

6.2.1 The Home Page

Figure 6.2 shows the entry point of WebTMS. This is the first page that the user would see upon visiting the website. The main goal of having this homepage is to answer the questions “Where am I?” and “What does the site do?” The name and logo “Online Thesis Database” in the upper-left corner of the screen serves as visual identity of the website as well as the location of users on the web. It is placed in the upper-left corner so that it is easier to spot because users tend to read left to right, top to bottom. Additionally, the home page also contains straightforward and simple mission statements and brief information that allow users to understand the purpose of the website pretty quickly.

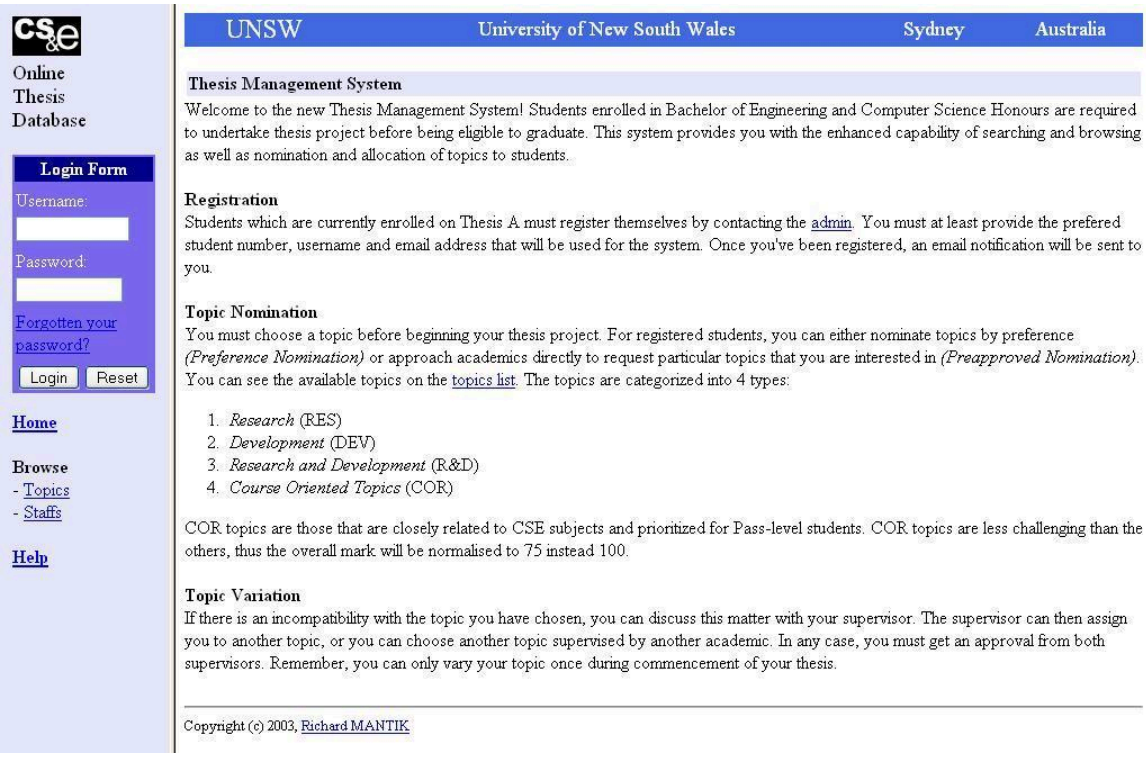


Figure 6.2. Home Page

6.2.2 Browsing and Searching System

The Topics Page shown in Figure 6.3 displays all the available topics on the database. The list can be sorted by clicking on the appropriate column heading

the user wish to sort by. The current sorting criterion is shown visually by highlighting the column and literally by specifying it on the label below the alphabetical index. The alphabetical index is used as *exact organization scheme* that divides the topics into groups that are specified by starting letter. This organization scheme is built for helping users to conduct what so called *known-item searching*, i.e. if a student know the topic title he want to find out is "Words Recognition", he would be able to find the topic title under the index W. The topic records are distributed among pages each displaying a maximum of 20 records to avoid overwhelming users and reduce the page load response time. Users can move between pages by specifying it on the page selection option. Additionally, the current retrieval set along with total number of results is shown below the alphabetical index to give users information about the current location of records. Clicking on the Topic title will link the users to the Topic information page as shown on Figure 6.4.

Figure 6.3. Topics Page

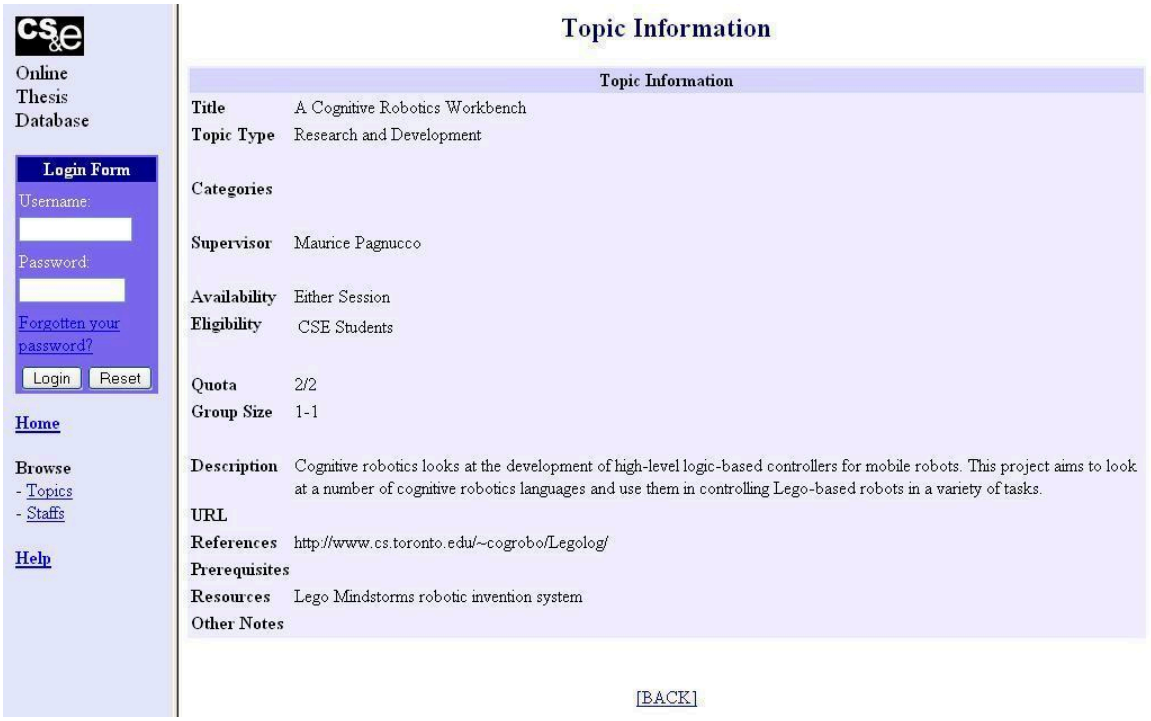
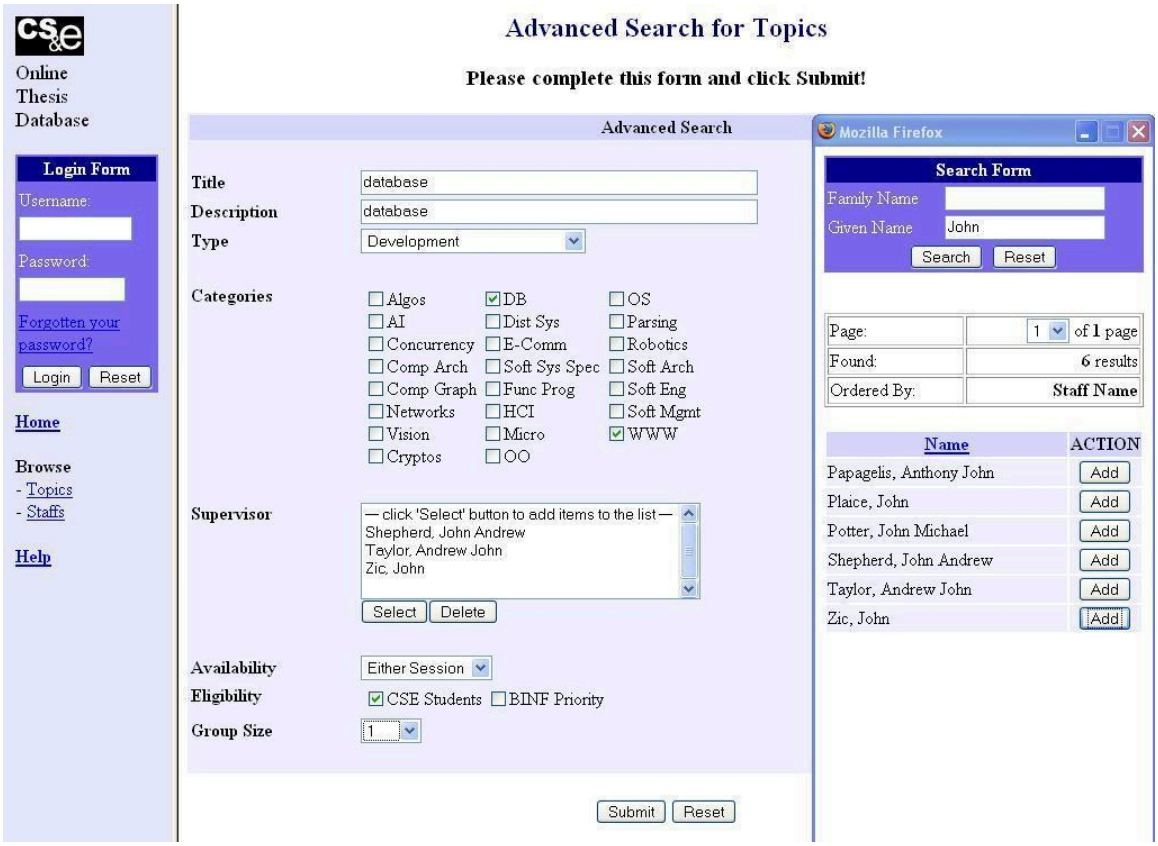


Figure 6.4. Topic Information Page

A simple quick search is provided because most of the time, users do not need all the power of advanced search. Besides, novice users that use the system at the first time would most probably need quick search instead of advanced search. This simple quick search is built in integration with the browsing system to make users' life easier to switch between tasks. This simple quick search is a type of fielded search that allows searching within particular fields. Upon conducting search, the search keyword is cached by the system and will be redisplayed on the search textbox in subsequent steps of browsing. This gives the users feedback what is being searched and also let the users modify query without reentering a new one.

An advanced search is offered on different page that can be found by following the link "Advanced Search" below the simple quick search. This is intended to notify novice users before getting into the page and hurting themselves. The advanced search page is shown in Figure 6.5. This advanced search allows users to conduct more powerful queries that can't be supported by simple quick

search. The category is used as a*mbiguous organization scheme* to allow users to conduct *associative learning.* This organization schemes divides topics into categories that depends strongly on human subjectivity of how to organize them. It is useful for the case where users don't have exact information that they are looking for, for example the topic "4-th Year Thesis Management System" may not have keyword "database" in the title or description field, but may be included as category "database". The advanced search also allows users to specify a set of supervisors in a query. When the "Select" button below the staff list box is clicked, the searching window is popped up. Since the list of supervisors can be quite long it displays a maximum of 20 staff names per page. Additionally, simple search functionality is also provided User can add the supervisors to the search criteria by clicking the button "Add" besides the staff name. The purpose of having this searching window is to enhance the system scalability. Similarly to simple quick search, the searching criteria are cached by the system. If users want to know the current advanced search criteria, they can just revisit the advanced search page. All the text fields, checkboxes and selection boxes will be filled with the current search criteria.



. *Figure 6.5. Advanced Search Page*

Figure 6.6 shows the page displaying all the staff members that are registered on the database. It is meant to allow students to find the information about staff members, such as room or email contact. A simple quick search, sorting functionality, alphabetical indexing and page chunking techniques are also incorporated in the page. In fact, these functionalities will be used throughout all the pages that list records so that they share the same style. Clicking on the Staff name brings the users to the Staff Information page as shown in figure 6.7.

Figure 6.6. Staffs Page

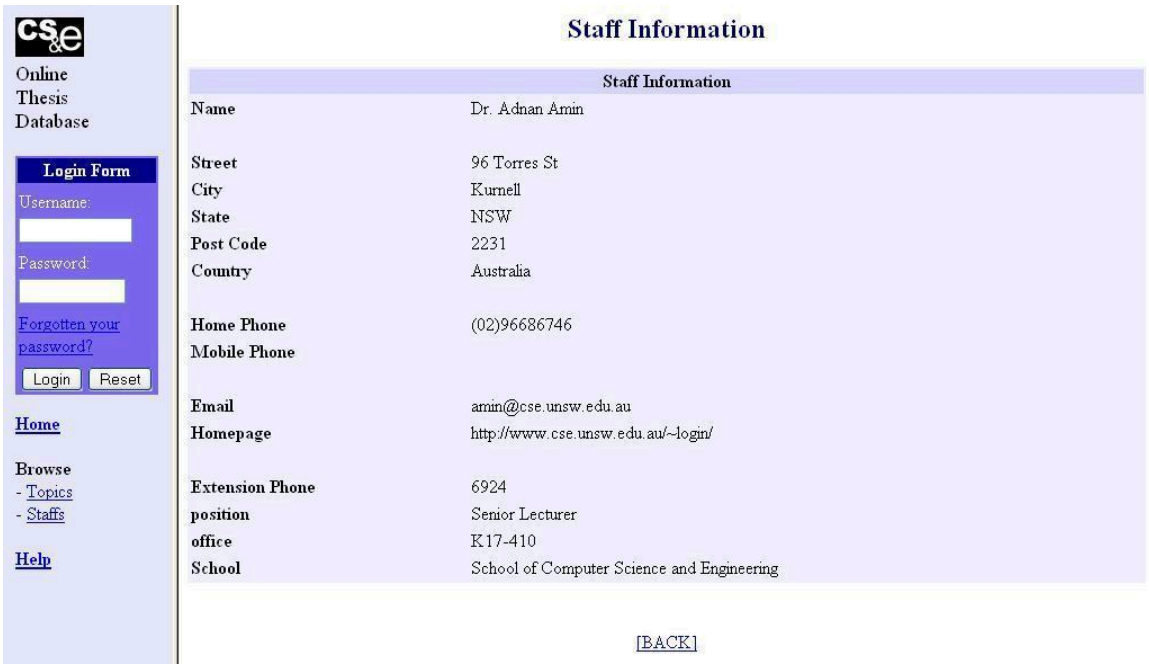


Figure 6.7. Staff Information Page

6.2.3 Administration Pages

Figure 6.8 shows the topic administration page. It is similar to the Topics page with additional functionalities to add, update and delete a topic. Following the link to add topic brings user to the Topic Entry Page as shown in the figure 6.9. Upon submitting the topic entry page, the data will be validated. If validation fails, the topic entry will be redisplayed with error messages in red above appropriate fields as shown in figure 6.10. The two advantages given by the design of the entry page are:

- The user doesn't have to reenter the entire topic entry from scratch when an error is encountered.
- The error messages are placed right above the fields where the errors occur so that the users can recognize them quickly.

Figure 6.8. Topic Administration Page

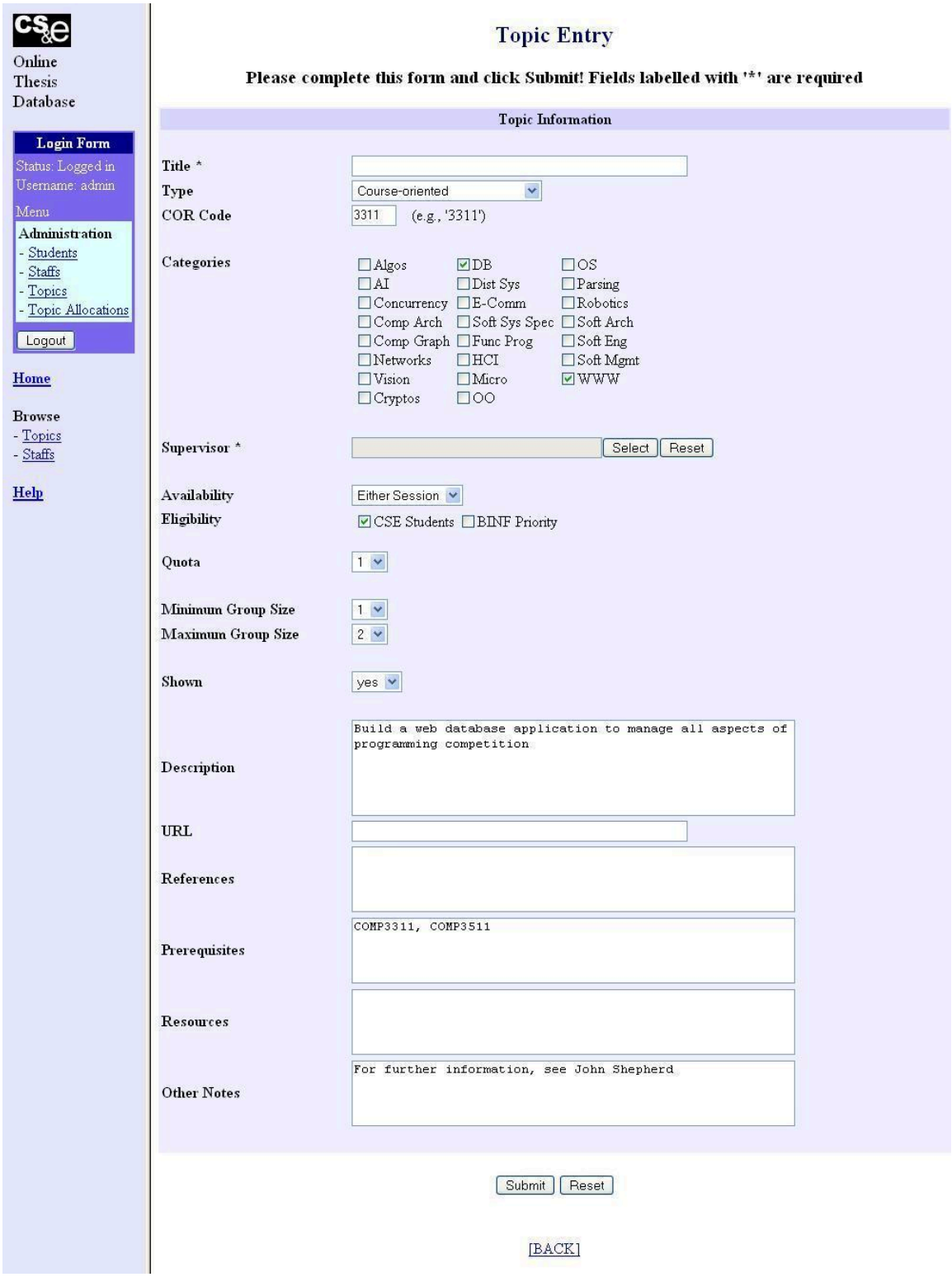


Figure 6.9. Topic Entry Page

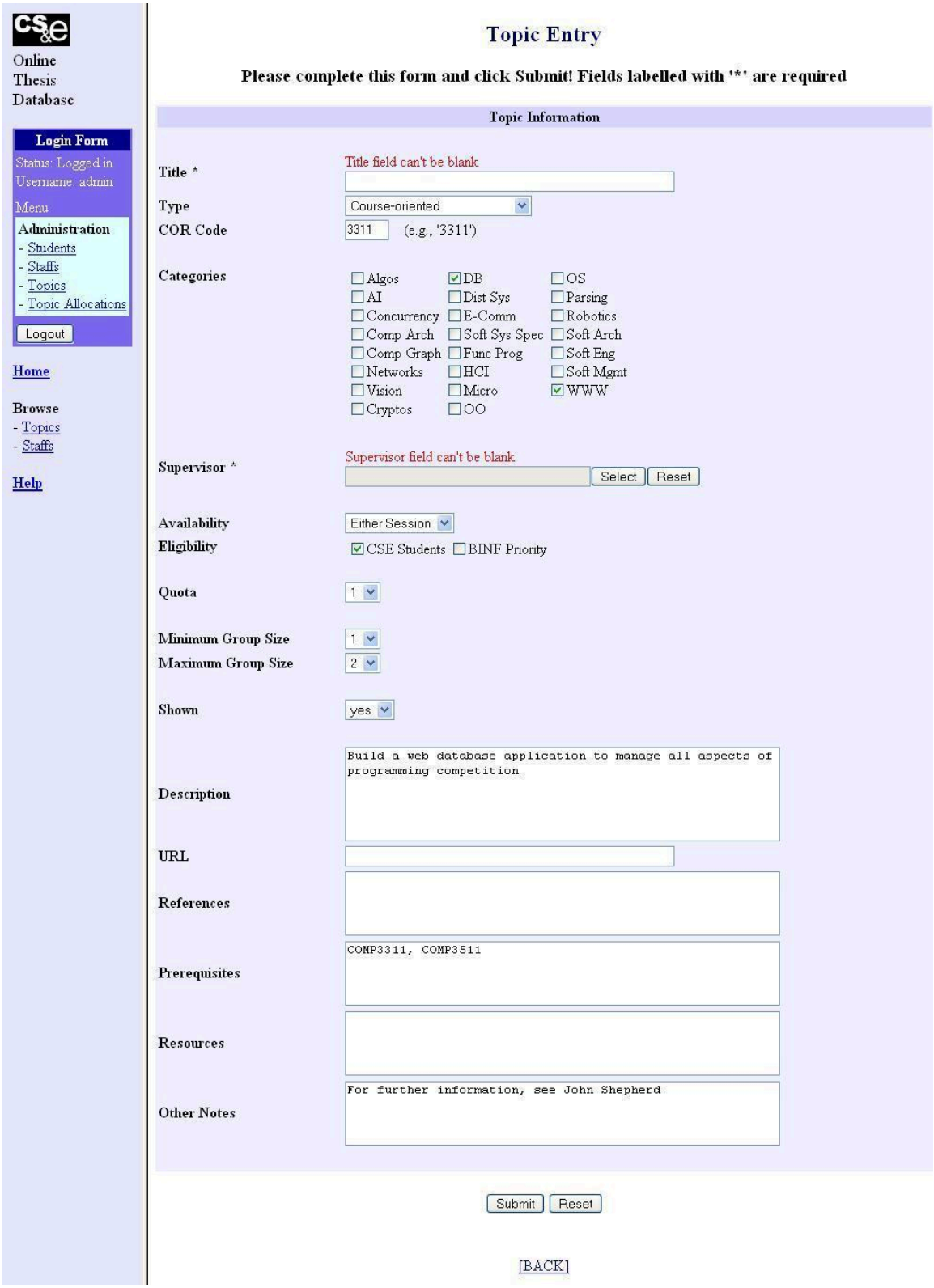


Figure 6.10. Topic Entry Page showing error messages

If there is no error in validation process, the topic receipt page will be displayed to inform the user that the insertion was successful (see Figure 6.11). The insert and update operation share the same interface. The difference is that in update operation, the fields have already been initialized with last saved topic information.

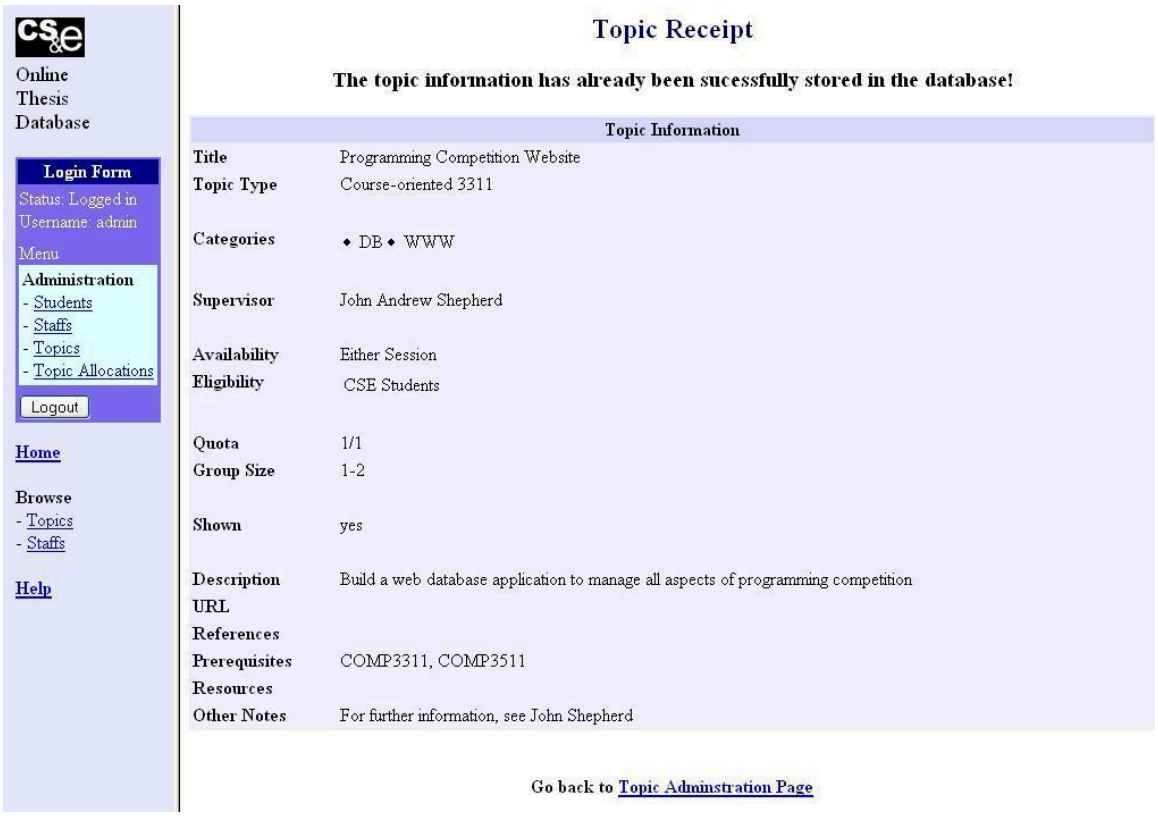


Figure 6.11. Topic Receipt Page

The topic administration page for staff is similar to that for administrator. The difference is that there are no fields “Staff” and “School” in the topic list since they would have been clear for staff members. The topic administration page for staff is titled “My Topics” as shown in figure 6.12.

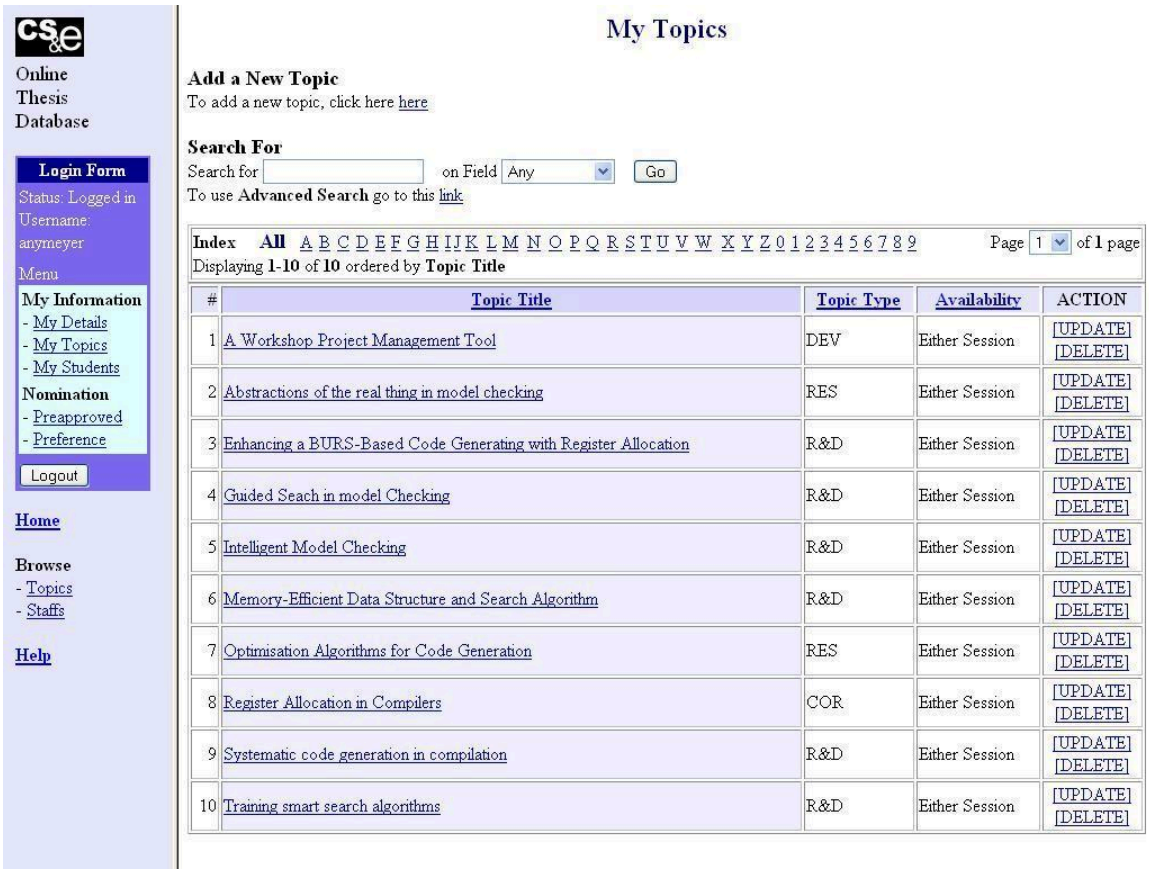


Figure 6.12. Staff’s Topic Page

Figure 6.13 shows Student Administration Page. It is loaded with the list of students that are registered on Thesis Management Database. Similarly to the Topic Administration Page, it also allows user to add, update, and delete students.

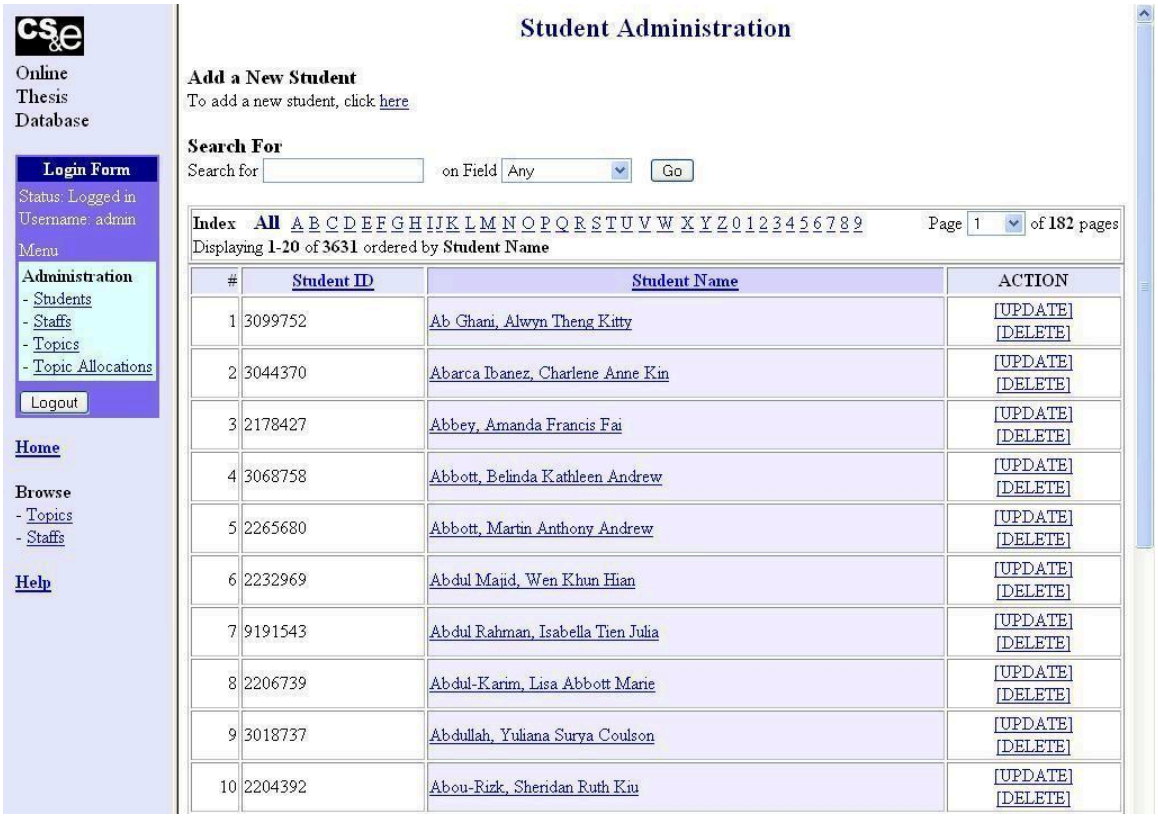


Figure 6.13. Student Administration Page

Figure 6.14 shows the student entry page. The same validation method, i.e. passing back the submitted fields to the entry page when errors are encountered and displaying error messages above the fields, is applied for both of the entry pages.

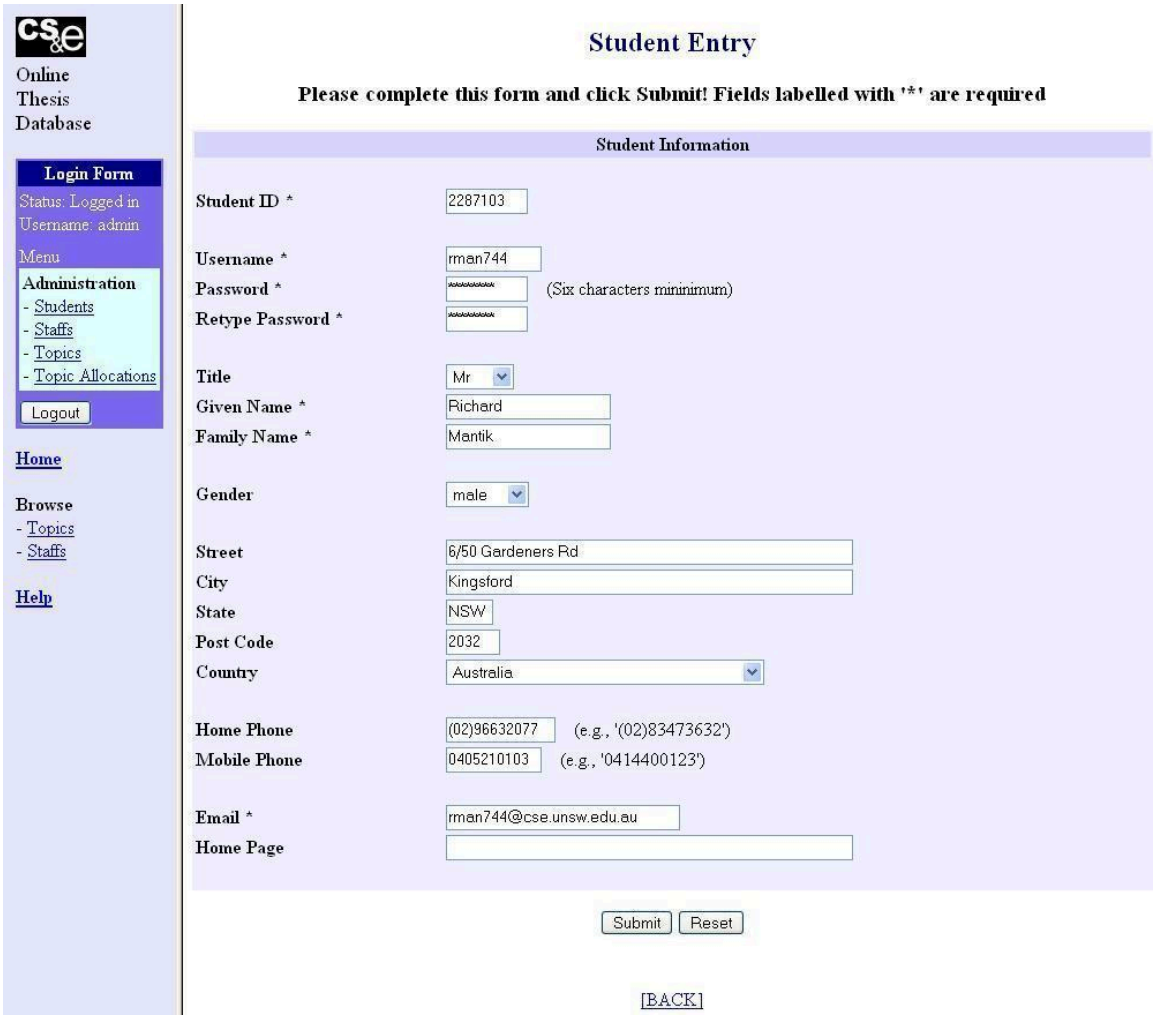


Figure 6.14. Student Entry Page

Figure 6.15 shows the student receipt page that is displayed upon successful form submission.

Figure 6.15. Student Receipt Page

Figure 6.16 shows Staff Administration Page. It is loaded with the list of staffs that are registered on Thesis Management Database. Similarly to the other two administration pages, it also allows the operations add, update and delete.

Figure 6.16. Staff Administration Page

Figure 6.17 shows the staff entry page that is shared for insertion and update of staff information.

Figure 6.17. Staff Entry Page

Figure 6.18 shows the staff receipt page that is displayed upon successful submission of staff information.

Figure 6.18. Staff Receipt Page

6.2.4 Topic Allocation

The topic allocations page as shown in figure 6.19 displays all the topics that have already been allocated for students. In this page, user could add a new allocation or delete existing allocations. To allocate a students to a topic, follow the link below the Label “Allocate Students to a topic”.

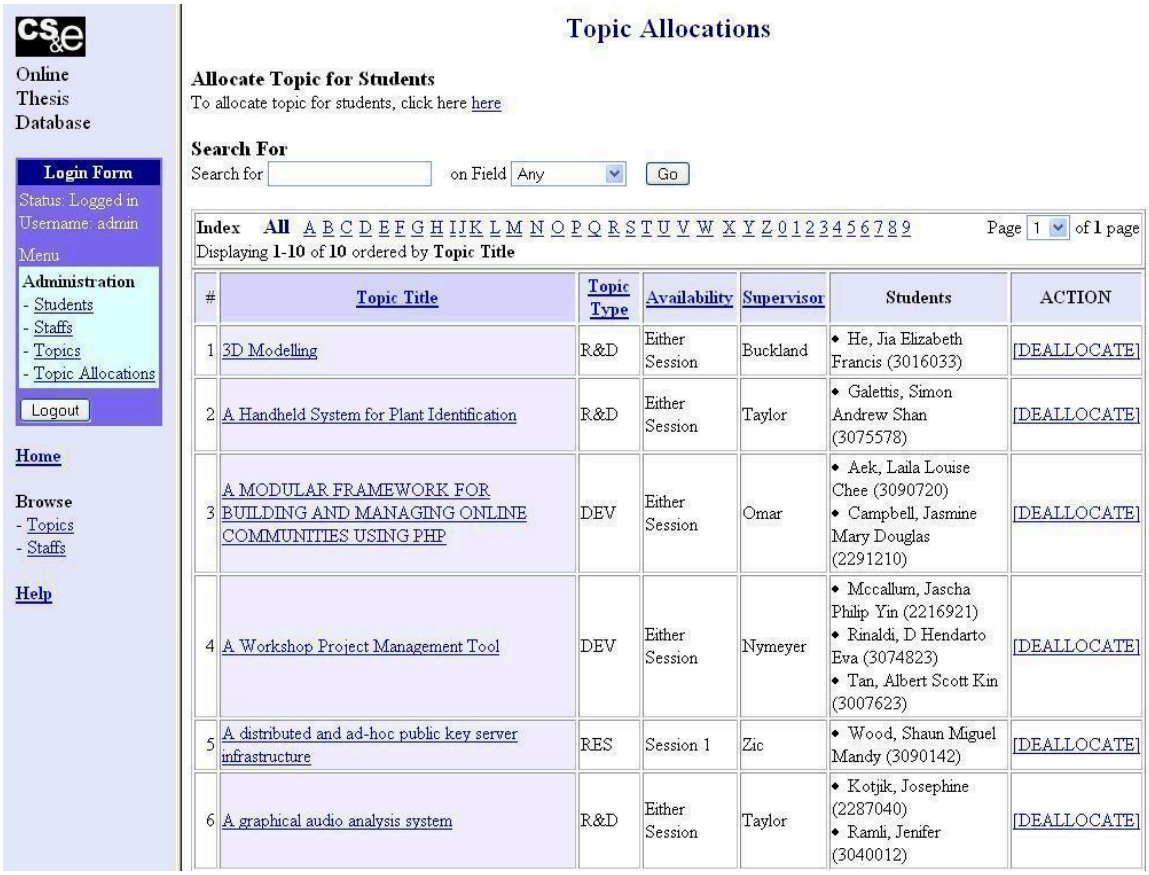


Figure 6.19. Topic Allocations Page

This would take the user to the group member selection page shown in figure 6.20. In this page, user can specify the students that are to be allocated. When the select button below the student list box is pressed, the searching window similar to the one that is used on advanced search page is used. User can add students by clicking the “Add” button besides the student name. When finish, the user can submit the group selection form.

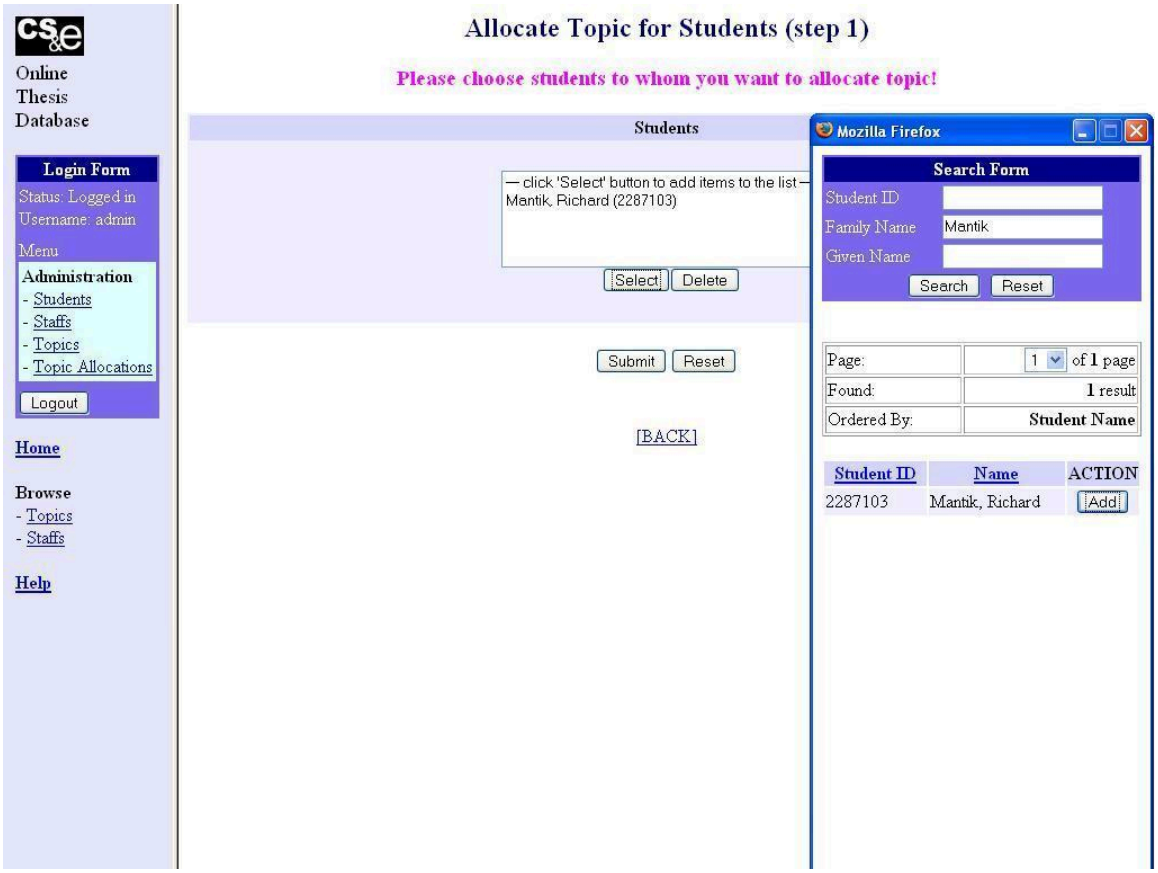


Figure 6.20. Group Members Selection Page for topic allocation

This will bring user to the topic selection page as shown in figure 6.21. Notice that there are 2 additional fields in the topic list, namely quota and group size. The quota fields are in the format “available quota/total quota”. The students selected in the previous page are shown in this page. This is to help users to compare the number of students with the group size criteria of the topic. Selecting the topic will bring the user back to Topic Allocations page with the students already matched with the selected topic (see figure 6.22).

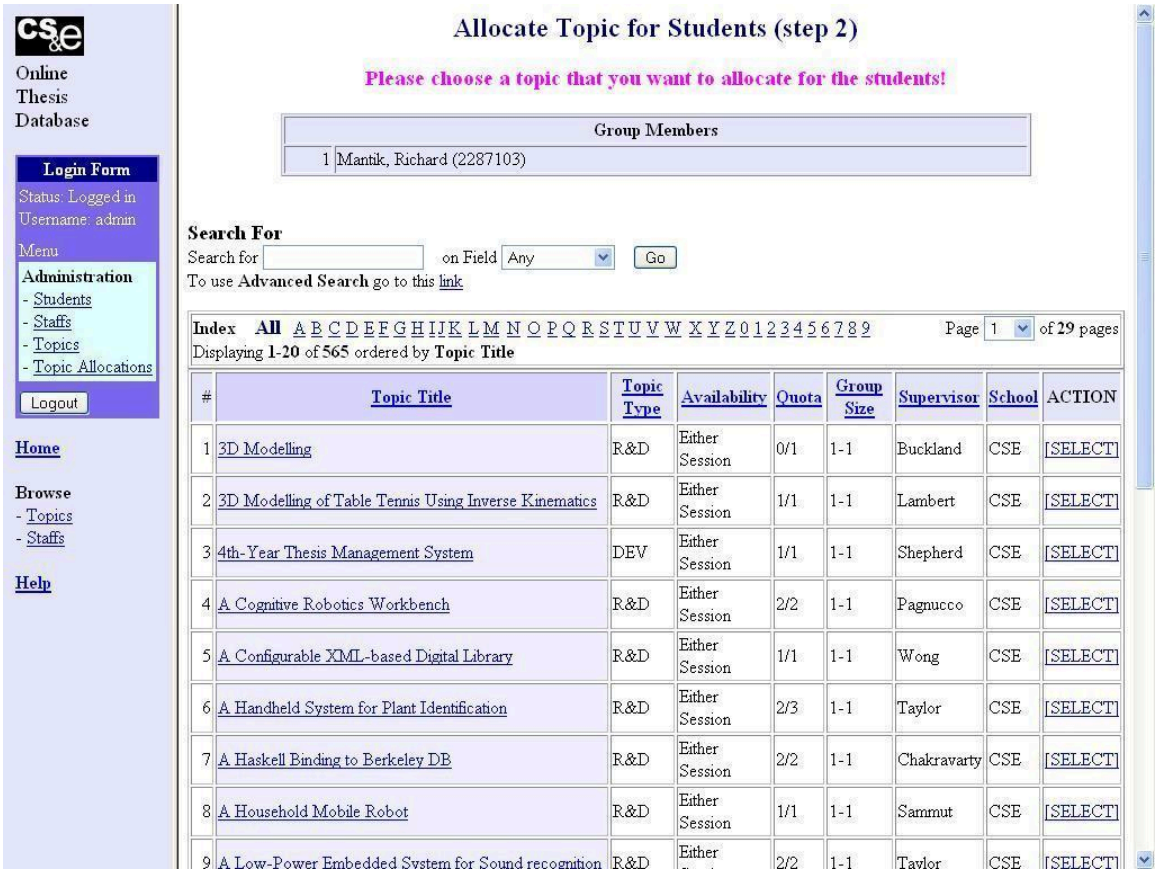


Figure 6.21. Topic Selection Page for topic allocation

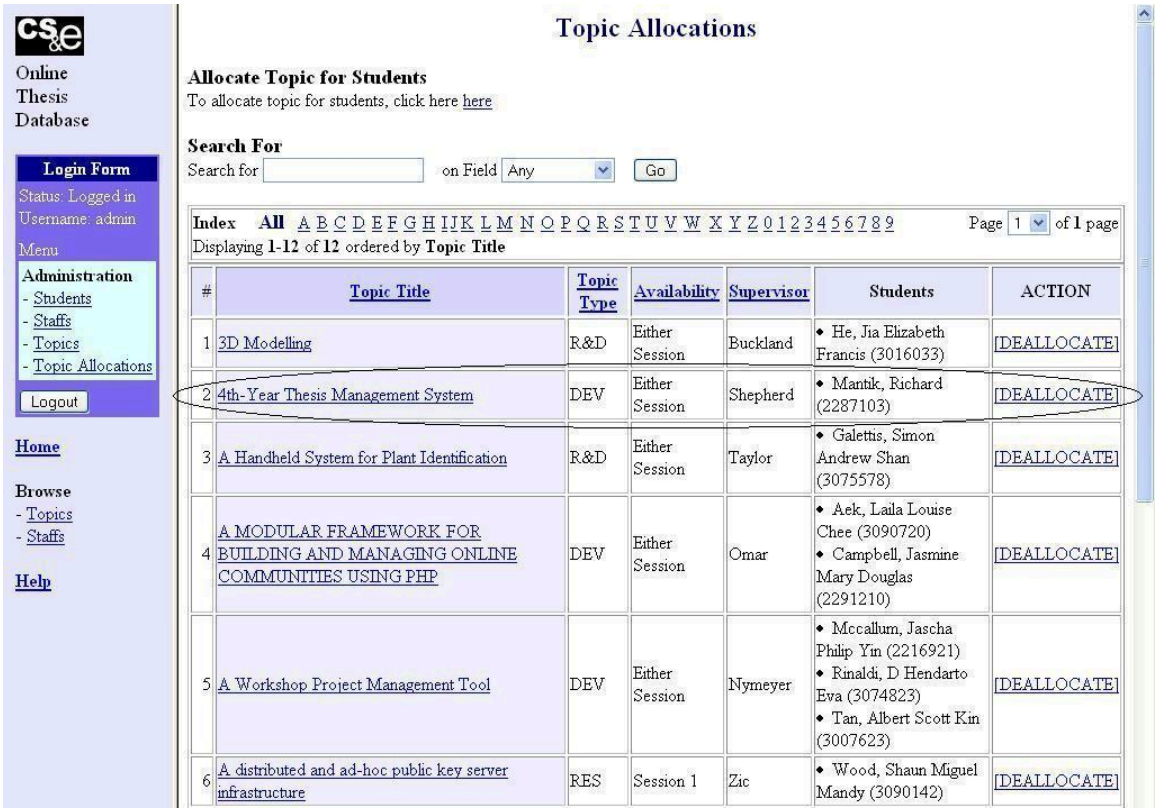


Figure 6.22. Topic Allocations Page after successfully allocating topic

Successful topic allocation is also reflected on the Staff's Students Page as well as Student's Thesis Page. In some sense, Staff's Students Page is similar to the topic allocation page. The difference is that there is no field "Staff" as in the Topic Allocations Page and there is no link to make a new topic allocation, since staffs do not have privilege to do that. The Staff's Students Page and Student's Thesis Page are shown in figure 6.23 and figure 6.24 respectively.

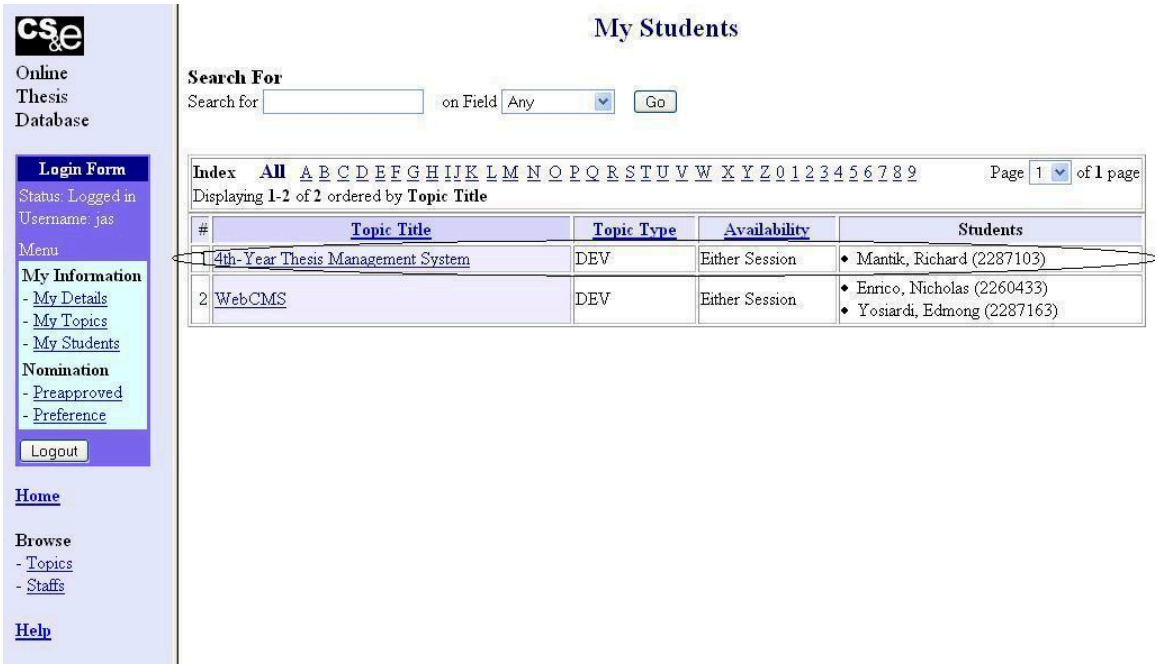


Figure 6.23. Staff's Students Page

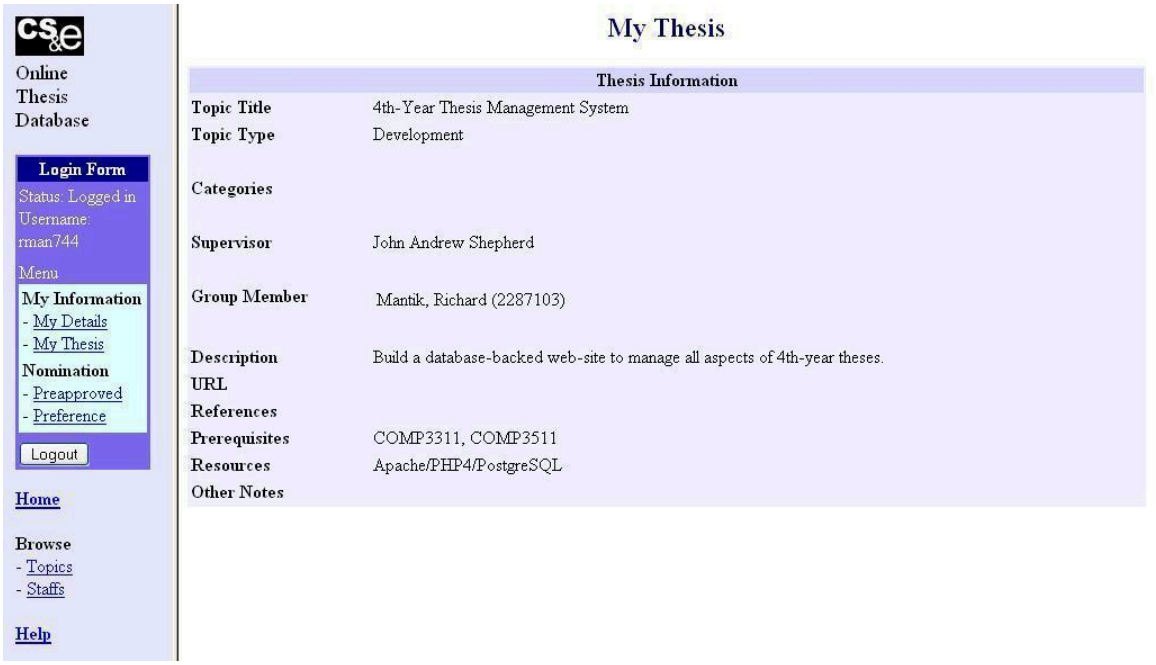


Figure 6.24. Student's Thesis Page

6.2.5 Preapproved Nomination

Figure 6.25 shows the Preapproved Nomination Page for the staff members. To grant approval to students, user can just follow the link below the label “Grant Approval”.

This would bring the user to the Group Members Selection Page (see figure 6.26). Submitting the group selection form will bring the user to the topic selection page (see figure 6.27). These two interfaces are similar to those used in the Topic Allocation.

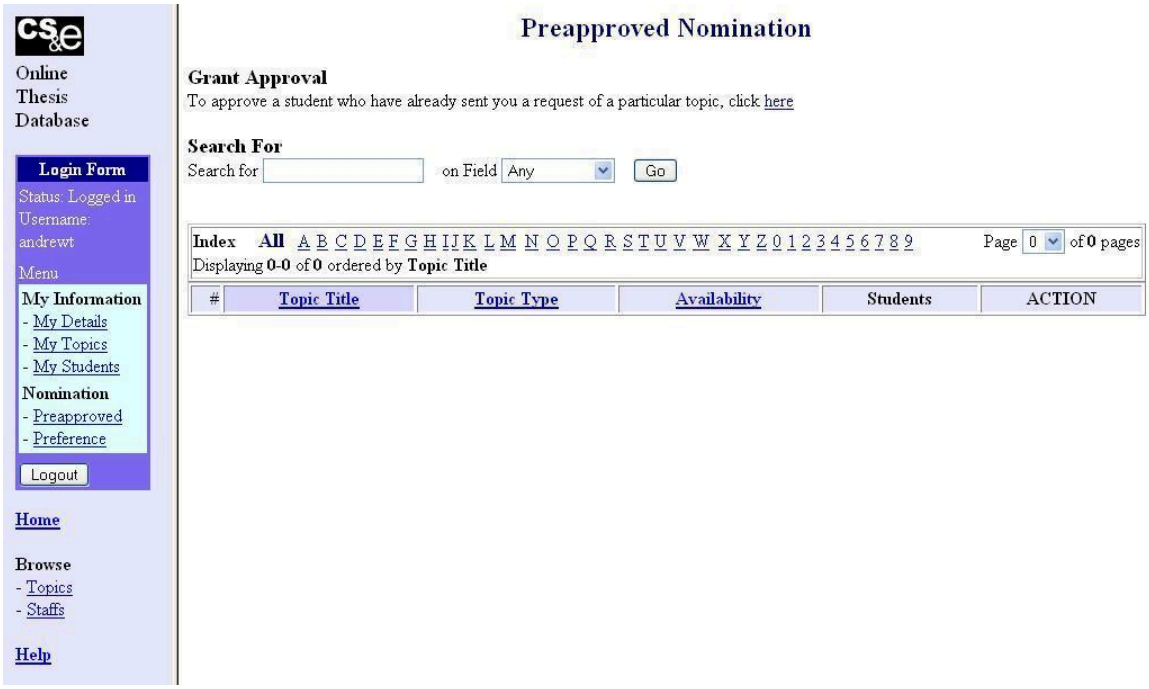


Figure 6.25. Staff’s Preapproved Nomination Page

Figure 6.26. Group Members Selection Page for granting approval

Figure 6.27. Topic Selection Page for granting approval

After successfully granting approvals, the user will be brought to the Preapproved Nomination Page with the approval resides on the list as shown in figure 6.28. For visibility of state of the process, the students' responses are shown at any time in this page. A visual metaphor is adopted in this interface by showing the students' responses in color that maps to the color of traffic lights, i.e. "on hold" is shown in yellow, "accept" is shown in green, and "reject" is shown in red.

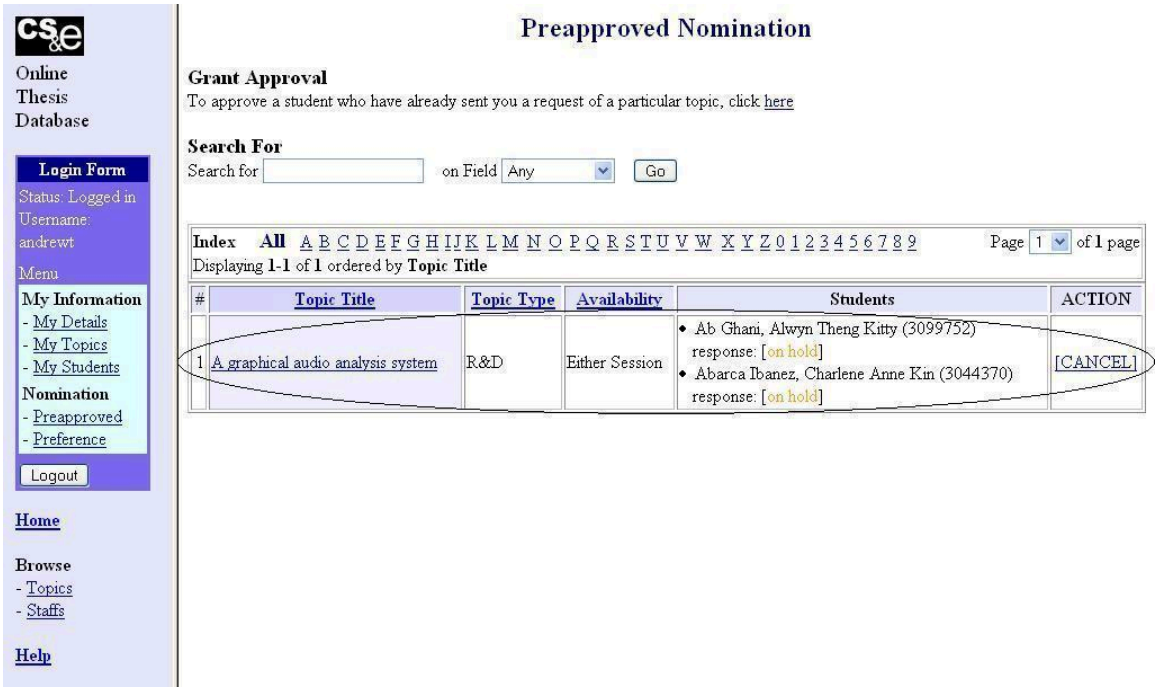


Figure 6.28. Staff's Preapproved Nomination Page after successfully granting approval

The approval is also shown in each of the students' interface to whom the approval is granted. Figure 6.29 shows the Preapproved Nomination Page for one of the students after the approval is granted. Each of the student's response is shown, so that one can remind the others in case they forget to accept the approval.

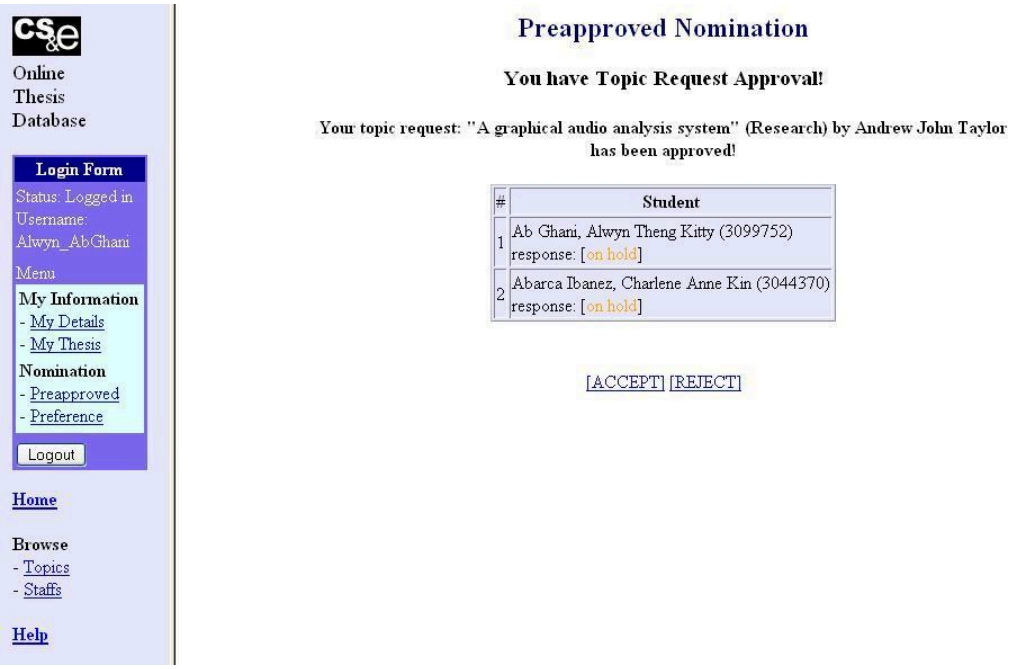


Figure 6.29. Student's Preapproved Nomination Page when approval has been granted

Figure 6.30 shows the Preapproved Nomination Page after the student has accepted the approval. The student who has accepted the approval has a chance to cancel the approval before all the group members accept it. If each of the group members has already accepted the approval, a new thesis group if formed and the Preapproved Nomination Page turns into the one shown in figure 6.31.

Figure 6.30. Student's Preapproved Nomination Page when approval has been accepted

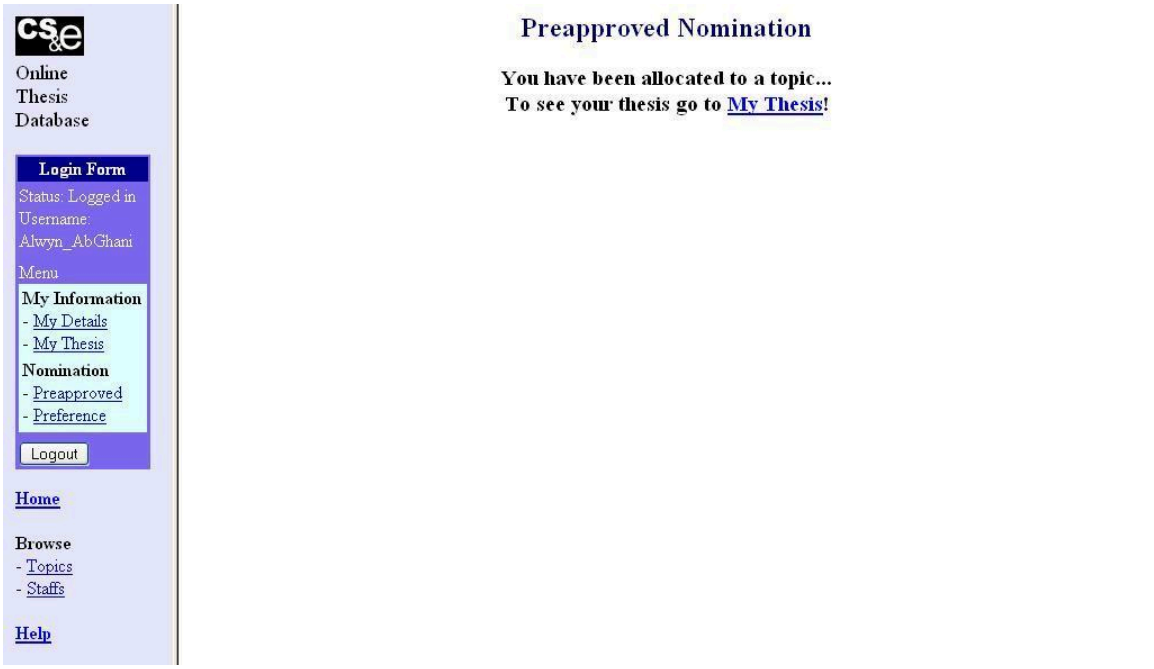


Figure 6.31. Student's Preapproved Nomination Page when the student has been allocated to a topic

6.2.6 Preference Nomination

Figure 6.32 shows the initial Preference Nomination Page for student (before nominating topics or being allocated to a topic). All the information about how nomination process is handled should be exposed in this page so that the user can make informed decision on the right nomination alternative. To nominate topics, user can just follow the link below the label “Nominate by Preference”. This would bring the user to the Group Selection Page (see figure 6.33). Submitting the Group Selection Page will bring the user to the Topic Selection Page (see figure 6.34). A *functional metaphor* “Topic Cart” is used in Topic Selection Page to facilitate user learning by mapping it to the “Shopping Cart” metaphor that is often used on e-commerce websites. User can place topics to the “Topic Cart” one at a time and remove them as long as the page hasn’t been submitted yet.

Figure 6.32. Student’s initial Preference Nomination Page

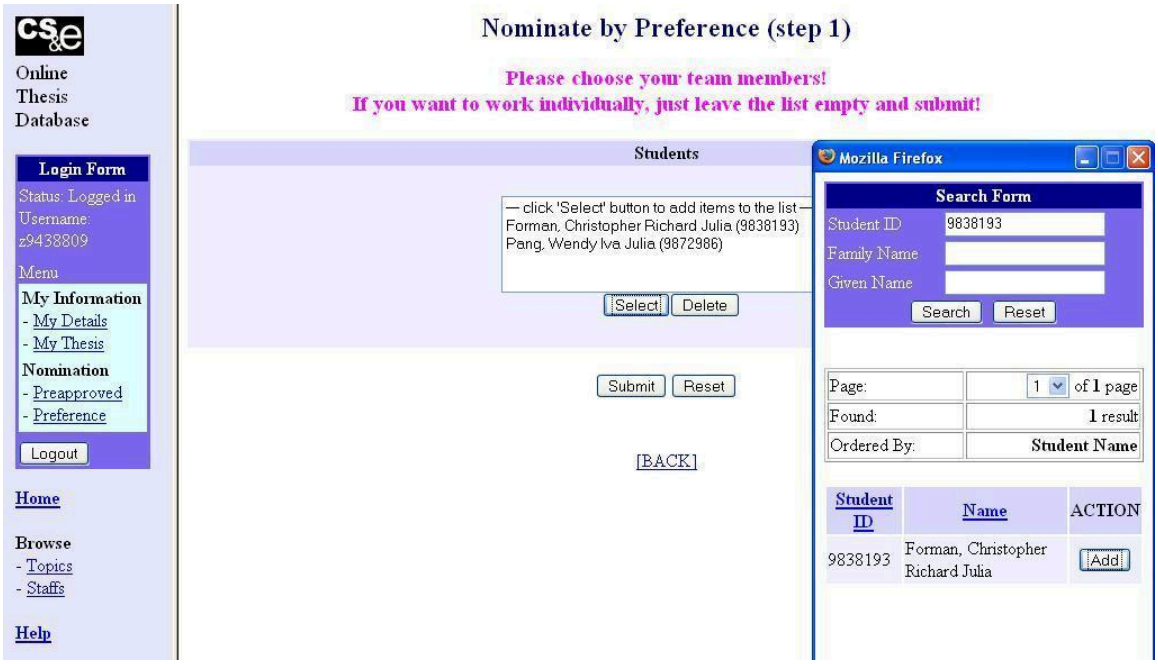


Figure 6.33. Group Members Selection Page for preference nomination

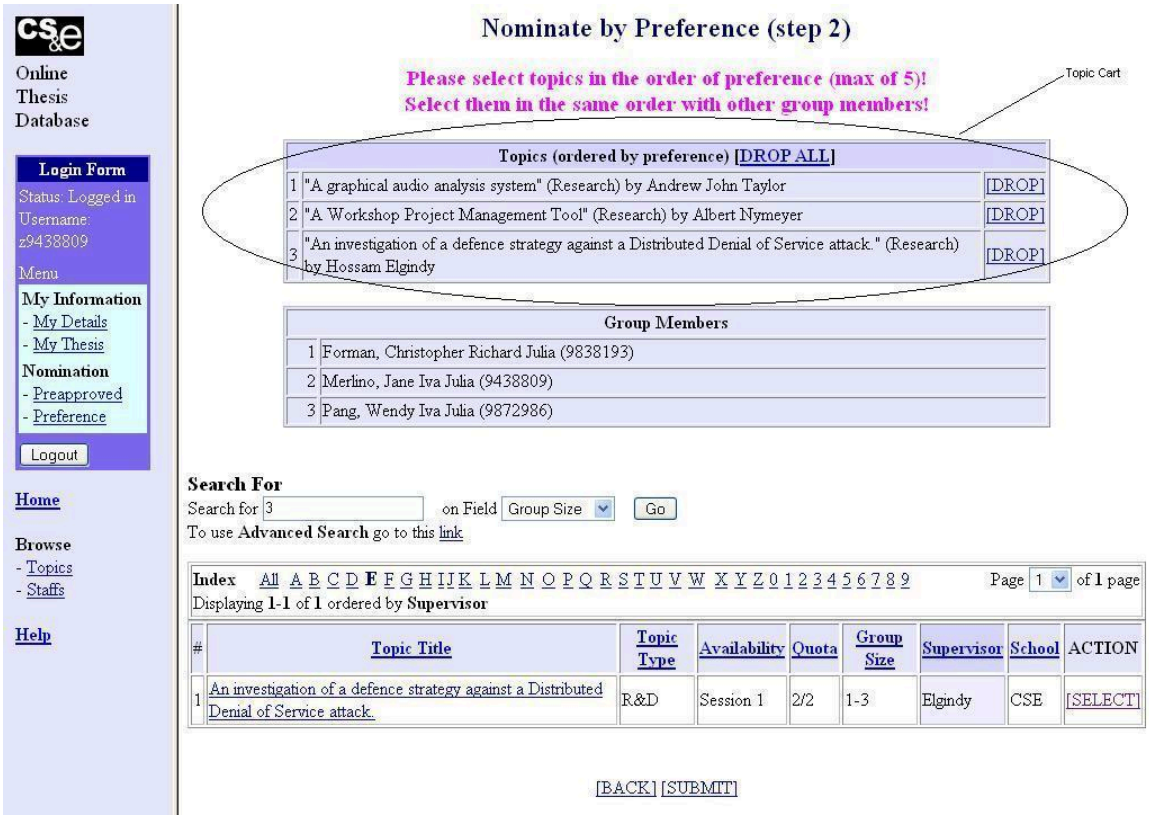


Figure 6.34. Topic Selection Page for preference nomination

Successfully Submitting the Topic Selection Page brings the user back to the Preference Nomination Page as shown in figure 6.35. Displaying the submitted preference topics and group members is important here so that the system state is visible to users. Additionally, the operation cancel is supported in this page so that the user can easily drop the preference nomination at any time before the request is accepted. Another visual metaphor is incorporated in this page by displaying the staffs' response and group members' response in color that have connection with traffic lights, as follows:

- For staff's reponse: "inactive" is shown in red, "waiting for approval" is shown in yellow, "queuing" is shown in grey, "accept" is shown in green, "reject" is shown in red and "timeout" is shown in red.
- For group member's response: "agree" is shown in green, "not agree" is shown in red.

Figure 6.36 shows the preference nomination page after all of the group members have submitted the preference nomination in the same way. Figure 6.37 shows the topic request arrived at staff's Preference Nomination Page.

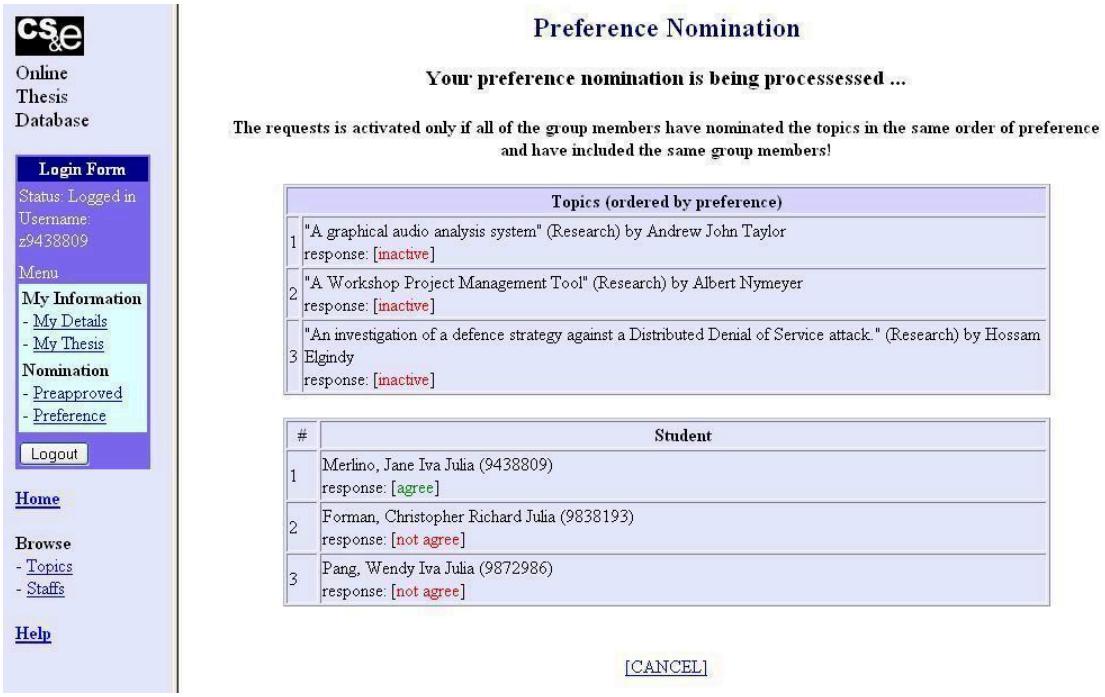


Figure 6.35. Student's Preference Nomination Page after nominating topics

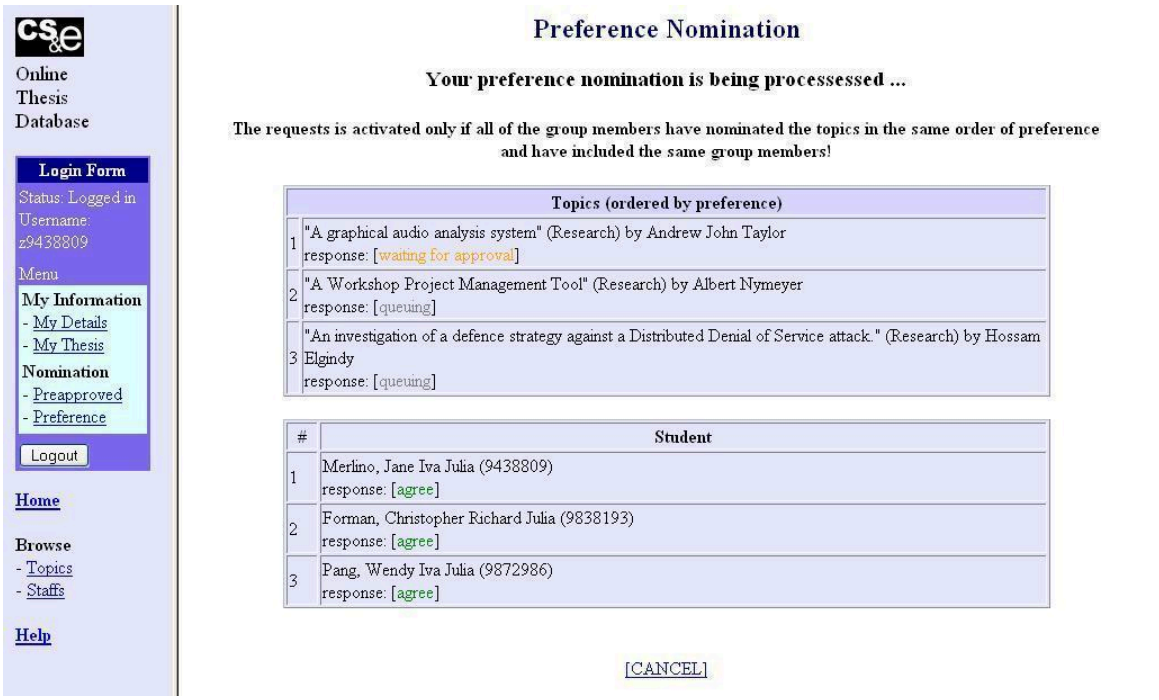


Figure 6.36. Student's Preference Nomination Page after all group members have nominated in the same way

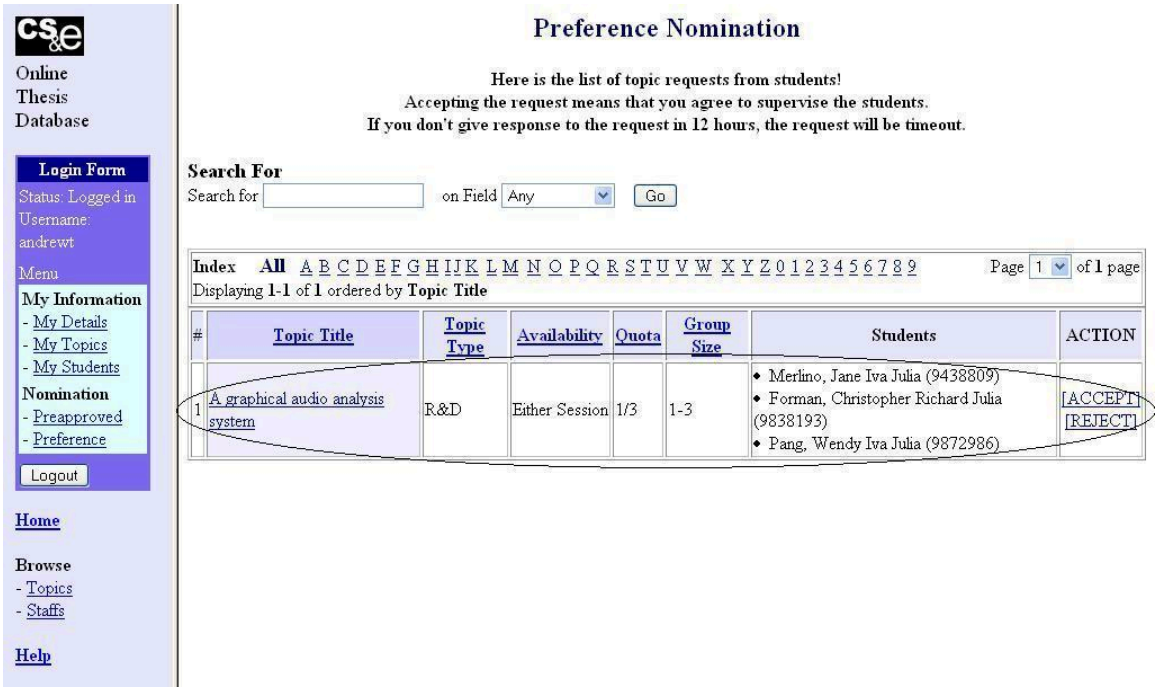


Figure 6.37. Staff's Preference Nomination Page showing the topic request arrived

If a staff has accepted the topic request, a new thesis group if formed and the Preapproved Nomination Page turns into the one shown in figure 6.38.

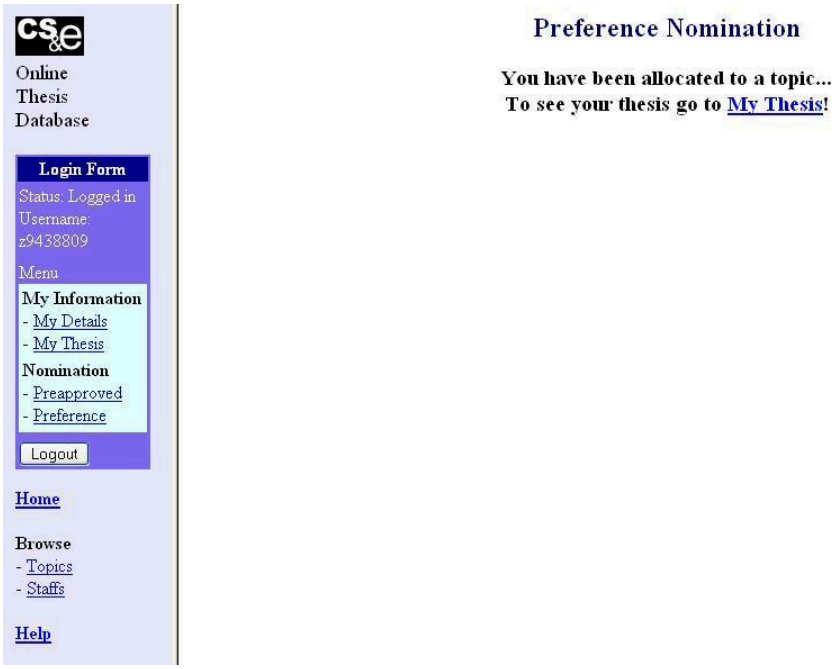


Figure 6.38. Student's Preference Nomination Page when the student has been allocated to a topic

CHAPTER 7

Evaluation

Performing Evaluation is one of the most important post activities in software development to ensure the quality of the product before delivery and to get indication whether it meets the end users expectation and satisfaction. In this section, we will be using the framework of evaluating good solutions that has been discussed in Chapter 2 to perform evaluation to our thesis management system.

7.1 Functionality

There has been a lot of improvement to the functionality since the last thesis management system that is done by Claudine Halim, such as the topic allocation and the 2 online nominations. However, the two applications have been developed based on 2 different set of requirements. To assess functionality more accurately, it is best to trace back to the functional requirements that have been developed during the early stage of development of this project. By comparing the final product with the functional requirement document, it can bee seen that all the functional requirements have been successfully met in our thesis management system.

7.2 Usability

To assess usability of WebTMS, tests were taken where a number of participants were given tasks to perform and asked to give grades to the functionalities and features in terms of ease of use and usefulness. Usability test is important because usability is an element that depends on the users' perspective, i.e. tasks might be easy or straightforward for the developer, but may not always easy for

the users. The overall usability rating of WebTMS is quite high and there are a lot of improvements over the CSE thesis management system. The following table summarizes the usability rating of the functionalities and the system as a whole. Further details of the usability tests could be found in Appendix B.

No	Description	Average Rating (1-5)
1	browsing system	4.2
2	searching system	4.2
3	login system	4.2
4	administration system	4.4
5	topic allocation system	4.2
6	preapproved nomination system	4.2
7	preference nomination system	4.1
8	error handling system	4.1
9	*overall WebTMS*	*4.2*

7.3 Performance

Perhaps performance is the most appealing attribute that WebTMS has. It has been tested by personal observations and usability tests that none of the pages takes more than 1 second to load even from a remote network (1 second is the limit for the user's flow to remain uninterrupted [Nielson]). In addition, a more rigorous performance testing has been performed where the topic list is populated with 10000 dummy records. The result is that there is almost no impact on the response time for the Topics Page to be downloaded. This is because of the page chunking technique that has been incorporated into the browsing system.

7.4 Robustness

The reliability rating of our thesis management system is almost close to 100.

The system has already been tested on several number of cases. All the bugs that are found on the usability tests have already been fixed. Additionally, there have also been tests where 3-5 users accessing the system at the same time, and the system handles concurrent uses of the functions very well. All the tests were conducted on the browsers that are available on the school machines. However, there a bug on the Windows XP Service Pack 2 which relates to the <SELECT> object on JavaScript which makes the searching window doesn't work properly. Hence, to support all of the functionalities of WebTMS, it is advised to use mozilla, conqueror or firebird, since all the tests have been performed on these browsers.

7.5 Maintainability

Our thesis management system also has advantages in this category. The implementation of WebTMS uses Object Oriented technique for most of the business logic. However, several areas including presentation are not implemented in Object Oriented to improve performance. Besides, 80 percent of the codes are produced by generic code (code that produces code) such that the amount of effort to write code is minimized. The total number of code line has been reduced from about 10000 to about 5000 lines when both of these techniques are incorporated in several steps of refactoring. Additionally, the code is structured and commented so that it can be easily read and understood.

CHAPTER 8

Conclusion

Despite of the fact that WebTMS win at many aspects of usability and performance, it couldn't support many of the post thesis activities. However, all of the original requirements have been successfully supported by the current system. Additionally, this thesis management system provides much better service at handling the nomination process. This has been supported by the results of usability tests that have been performed on a number of volunteers.

Admittedly, there are number of features that can be added to improve this thesis management system. Among them are:

- *Undo operation and topic selection reordering.* Undo operation for some of the operations such as topic deallocation, cancel approval and cancel nomination could be supported, so that users could easily rollback to the state before these operations are performed. In addition, topic selection reordering could improve the usability in nominating topics. These two features were suggested by the participants in the usability tests.
- *Topic recommendation.* Based on the students' academic standing, the system could give a recommendation about the topic that is most appropriate for the students. This feature will be very helpful for students to find a topic that closely matches their strength and interests.
- *Prerequisites checking.* Only the requests that are originated from students who fulfill the criteria for the thesis project can be valid. This is a useful feature that helps staffs examining the eligibility of students to undertake the thesis project. In this thesis, such feature is not supported because it will make it impossible for students who are willing to learn a new area of study even if they don't meet the prerequisites.

- *Automatic assessor allocation.* Based on the staff member's backgrounds, the system could match the thesis to the supervisor who is compatible with the research area of the topic. This feature will definitely helps the undergraduate teams to improve the efficiency of their tasks.
- *Automatic seminar timetabling.* Based on the availability of the staff members and the students, the system could find the appropriate time slots for which the number of allocated thesis could be maximized. To be able to implement such feature, it will require an advanced algorithm and it should be mathematically proven. Seminar Timetabling is currently supported by a separate system in CSE and it is infeasible to be incorporated to this project given this limited time period.

Overall, because of the time limitation, not all aspect of thesis management can be supported by this system. However, the system is almost ready to support many of the features required at the beginning of the thesis process. It is hoped that in the future there is such thesis management system that supports both the beginning and the end of the thesis process.

APPENDIX A

Survey Results on Thesis Management System Used in CSE

The results presented here are based on the survey run at the period 16 May – 15 June 2004 on the users of the CSE thesis management system. There have been several surveys before, but are omitted because the system is still under development to support Preapproved Nomination. The survey is taken on different types of users: student, staff and admin that are using the system. There are 20 students, 5 staff, and 3 undergraduate teams responded on this survey each of which is selected on a random basis. Students, staffs, and undergraduate teams are given different types of questionnaire that are chosen based on their perspectives and views to the system.

Summary of findings:

The information presented here is summarized from survey to different types of respondents: student, staff and administrator. Respondents answered questions about their perspectives and views to the current system and their expectation of how the current system can be evolved and improved.

Finding from student respondents:

1. The majority view of the student respondents is that the current thesis management system is a moderately useful tool to find a topic.
2. Most of the student respondents encountered the limitation of the search engine functionality. Also, a considerable amount of respondents feels overwhelmed with the list of topics that are displayed on one page. Other minority of respondents have problems with the navigational system.

3. 65% of the student respondents are unfamiliar with the use of Staff initials at the first time they use the system.
4. 80% of the student respondents prefer undergoing nomination process by approaching the Thesis Staff directly in order to identify the details of the topic and get the best topic that matches their interests. 20% of the respondents prefer nominating topics by preference to avoid hassles because it is more simple and efficient.
5. 87% of student respondents avoid preference nomination because of they are afraid of getting matched with the topic they like least.
6. 65% of student respondents can't deduct the actual process of Preference Nomination with the information presented on the website.
7. A great majority of student respondents prioritize the importance of getting allocated to the topic they like most based on the reason of being motivated while working on the project.
8. 60% of student respondents view the current process of handling nomination process as fast enough.
9. 75% of student respondents voted that the manual handling should be eliminated from current nomination process.
10. 90% of student respondents think of the importance of having login account to prevent other students nominating topics for others.

Finding from staff respondents:

1 The majority of the staff respondents rate the topic management feature as moderately useful.
2. A great majority of staff respondents give average grade of how the system helps them getting matched with students.
3. 80% of the staff respondents find it strange to deal with the topic entry form in the first time.
4. 80% of the staff respondents prefer student approaching them directly when requesting a topic because they'd like to understand the students' interests

and to make sure that the students understand the problem they will be solving.

5. 60% of the staff respondents rate the current nomination process as fast enough.
6. 80% of the staff respondents suggest automation for the task of handling nomination process.
7. 80% of the staff respondents recommend that the system should provide password as an authentication mechanism.

Findings from undergraduate teams:

1. From the admin respondents' point of view the current system is considerably good for managing thesis topics.
2. According to admin respondents, the current system helps them handling thesis topic allocation considerably well.
3. 100% of the admin respondents prefer handling the preapproved nomination process because it is simple and involve less work.
4. None of the respondents feels that it is exhausting or tedious handling the requests sent to the supervisors.
5. 66% of the admin respondents think that the system is quick enough on handling the nomination process.
6. 66% of the admin respondents vote for an automation for nomination process.
7. 66% of the admin respondents vote password to be incorporated in the system.

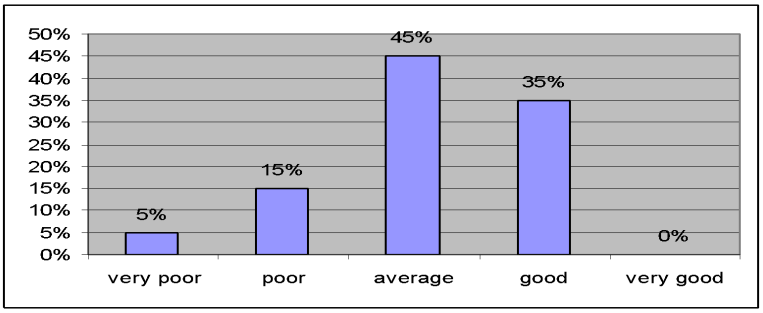
The results of the survey on respondents suggest an improvement to the current system, especially on the browsing and searching functionalities. In addition, it could also be further improved by replacing Staff Initials by any other means to represent Supervisors. Another improvement should also be made on the usability of the topic entry interface for staffs.

When confronted to the question of nomination choices, most students prefer Preapproved Nomination for the reason that they don't feel like being accidentally matched with the topic that they like least. Students typically choose getting the best topic that matches their interests, even though it would take longer time to find such a topic. From the staff respondent's point of view, they prefer students going directly to request topic because they get to know the students interests better and they'd like to be guaranteed that the student know the project they are taking. From the admin respondents perspectives, preapproved nomination is preferable. This indicates the importance of having Preapproved nomination incorporated to the system over the preference nomination.

The overall majority of respondents view to the current system performance on handling the nomination process as reasonably fast. However, it could also be further improved by automating the process of nomination as the majority of respondents suggested. Most respondents see the major security flaws of not having the login account to the system.

Students' Responses:

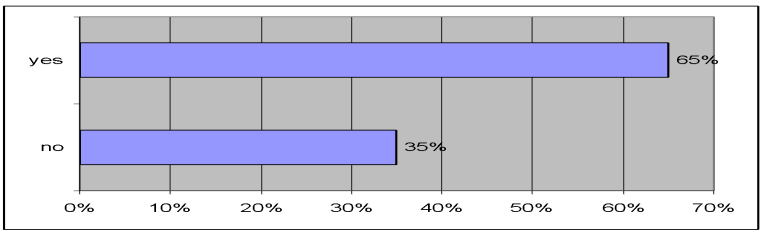
1. Give a grade of how the current thesis management system helps you with finding thesis topic!



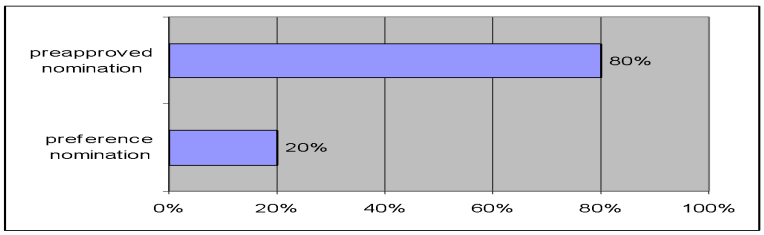
2. Check one or more of the following options that you think give you problems with the following system
   - The excessive amount of topics displayed on only 1 page (9 responses)

- The navigation system is poor (5 responses)
- The search functionalities are limited (12 responses)

3. Did you feel strange encountering Staff initials (RJB for Richard James Buckland and RB for Roksana Boreli) at the first time?



4. Which nomination process do you prefer?



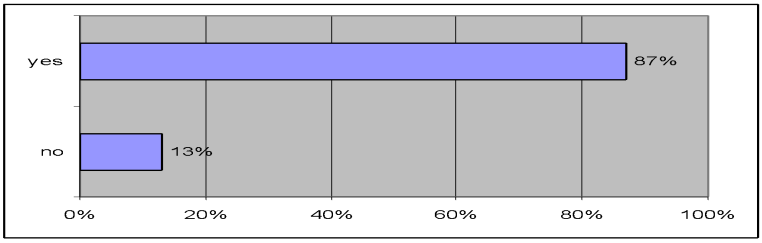
Note:

- preapproved nomination: You approach the academics directly to request a topic.
- preference nomination: You provide a list of 5 topics, and try to getting admin allocate you the topic you like most.

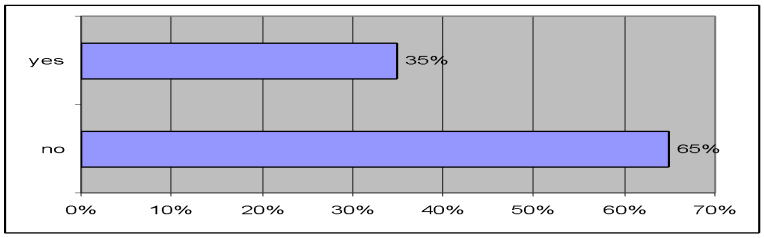
Why?

Most of the students chose preapproved nomination in order to get the topic they like most and to have a better understanding of the details of the topic. Most students chose preference nomination because it is more efficient way of getting allocated to the topic.

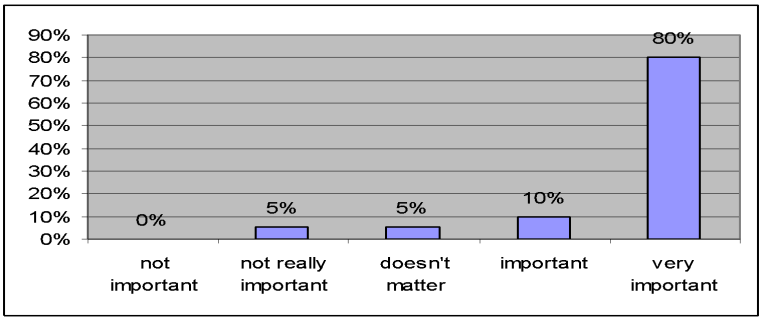
5. If you choose preapproved nomination for previous question, it is because you are scared of getting accidentally allocated to the topic you like least?



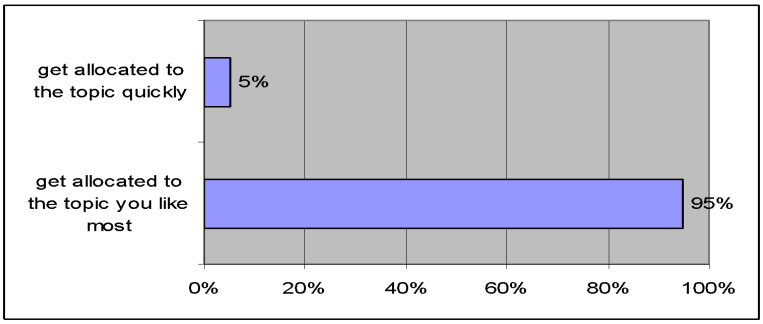
6. Could you actually deduct how the current preference nomination is handled given the information on the website?



7. How much important to you to get allocated to the topic you like most?



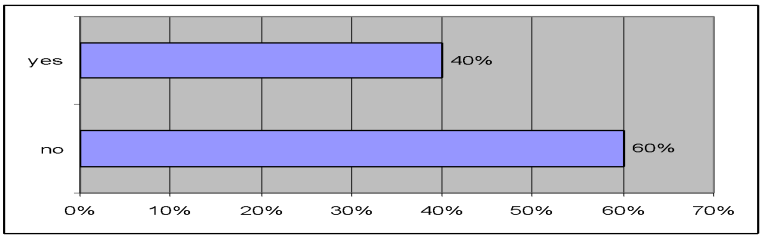
8. Which one is more important to you?



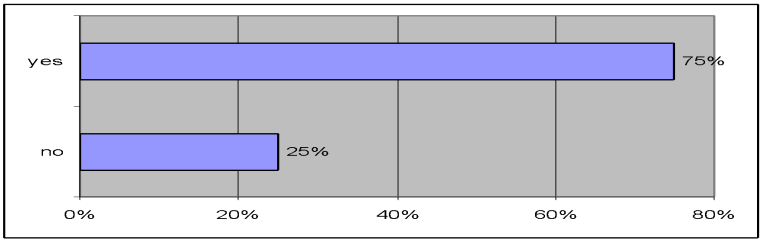
Why?

Most students choose the second option because they feel being motivated if they are put on the project they like.

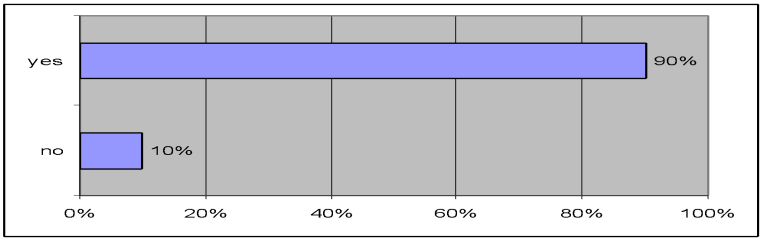
9. Do you think the current system is quick enough on handling nomination process?



10. The current system involves admin work for both preapproved nomination and preference nomination during the topic allocation. Do you think it would be a great idea to have a system that bypasses admin work?

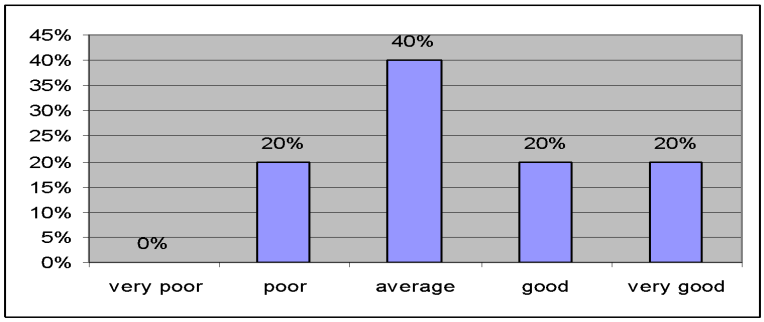


11. Do you think the current system should provide login account for students to prevent submitting nomination for others?

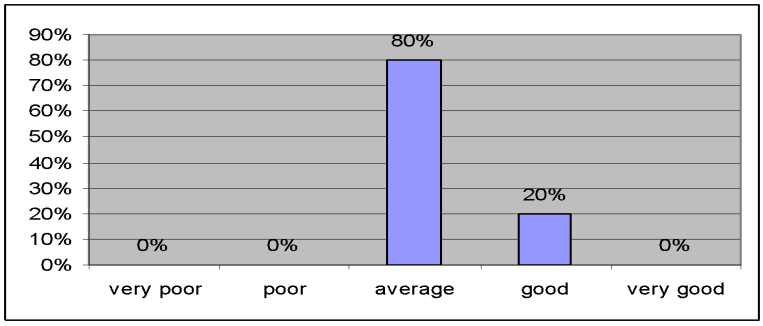


Staffs' Responses:

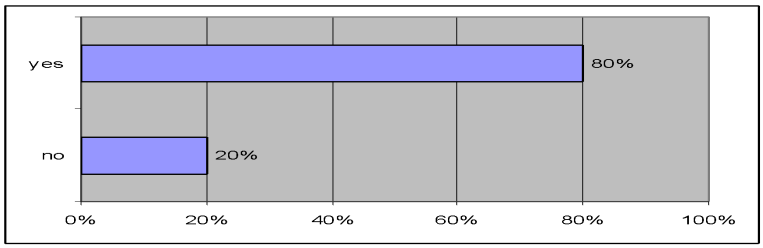
1. Give a grade of how the current thesis management system helps you with managing topics!



2. Give a grade of how the current thesis management system helps you finding thesis students?



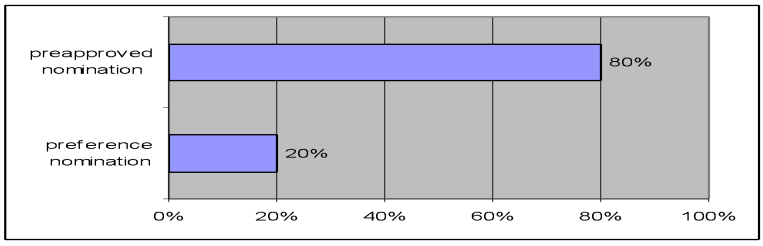
3. Do you feel unfamiliar filling the topic entry form at the first time?



Why?

Staff respondents say some of the fields on the topic entry form are strange.

4. Which nomination process do you prefer?



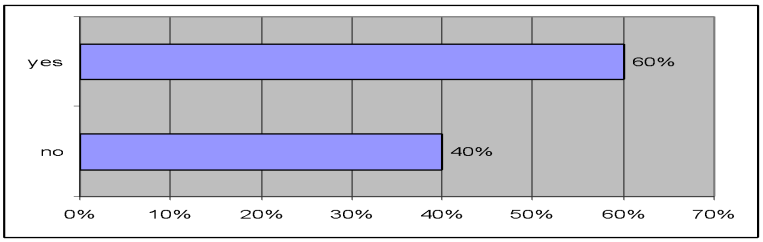
Note:

- preapproved nomination: Students requesting topics by approaching you.
- preference nomination: Students nominate topics and based on those nomination topic requests are sent to you. You are going to response the requests without having to meet in contact with the students.

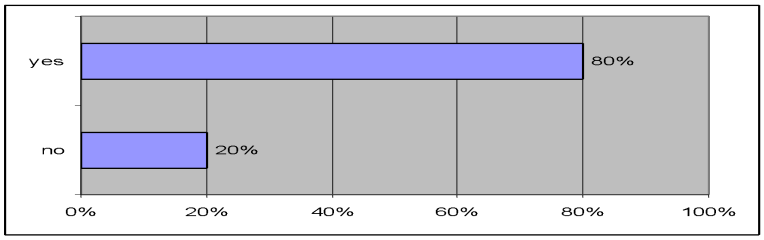
Why?

Most staff respondents prefer students approaching them directly to know the students' background and their interests. In addition, some of them also want to ensure that the students to understand the topic they requested.

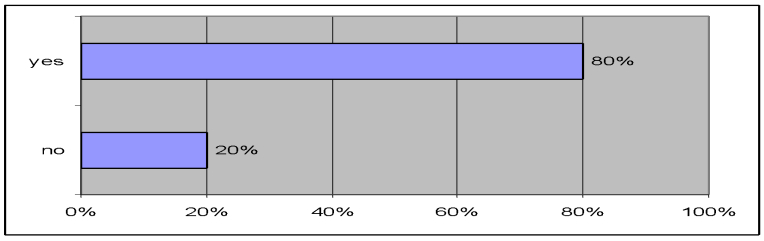
5. Do you think the current system is quick enough on handling nomination process?



6. The current system involves admin work for both preapproved nomination and preference nomination during the topic allocation. Do you think it would be a great idea to have a system that bypasses admin work?

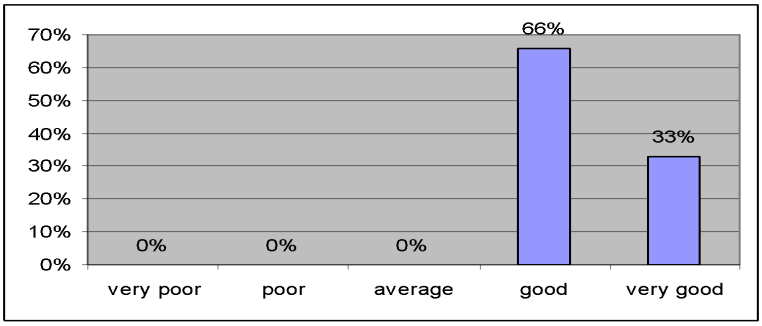


7. Do you think the current system should provide login account for students to prevent submitting nomination for others?

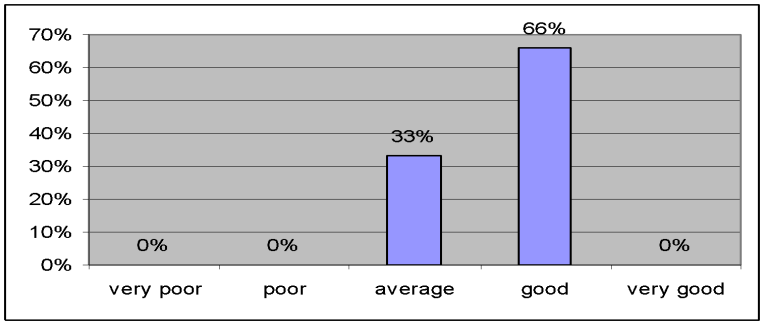


Undergraduate Teams' responses:

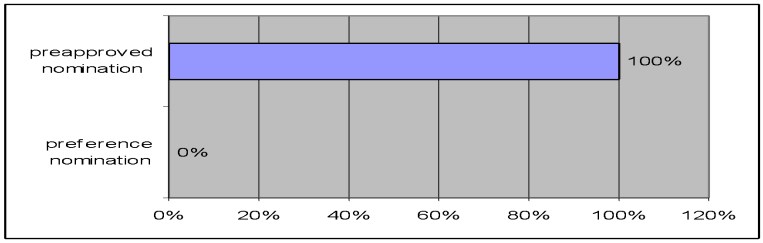
1. Give a grade of how the current thesis management system helps you with managing thesis topics!



2. Give a grade of how the current thesis management system helps you with allocating students to topic!



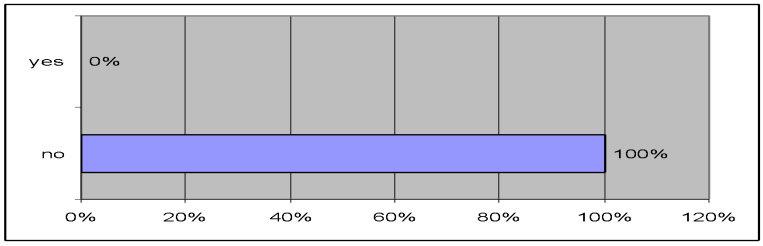
3. Which nomination process you prefer to handle?



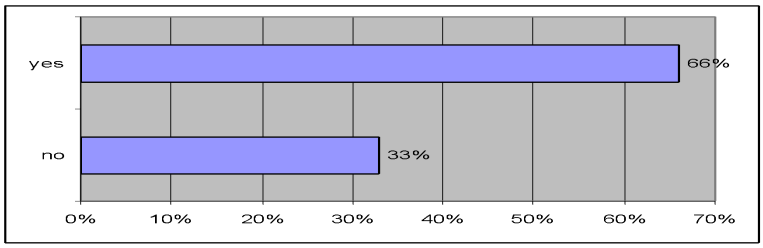
Why?

All admin respondents prefer handling preapproved nomination process because it is less work and simpler.

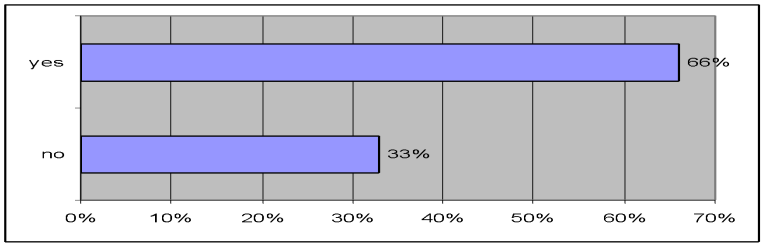
4. Do you feel it is exhausting or tedious to send requests to the supervisors when students submit nomination form?



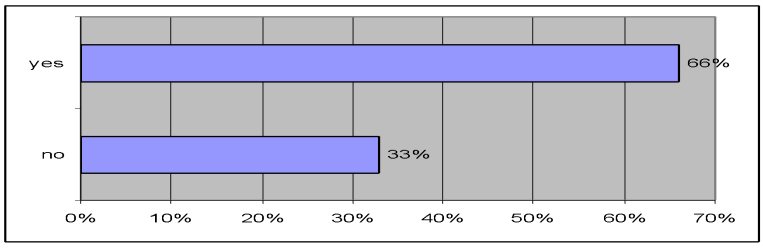
5. Do you think the current system is quick enough on handling nomination process?



6. The current system involves admin work for both preapproved nomination and preference nomination during topic allocation. Do you think it would be a great idea to have a system that bypasses admin work?



7. Do you think the current system should provide login account for students to prevent submitting nomination for others?


yes
no
66%
33%
0%
10%
20%
30%
40%
50%
60%
70%

APPENDIX B

Results of Usability Tests on WebTMS

The results presented here are based on the usability tests that are run on our new thesis management system, WebTMS during the period 6 – 20 July 2004. The usability tests are taken on different type of users: student, staff, and admin, each is surveyed with the same set of questionnaire to evaluate the quality and user interface design of the web site. There are 10 students, 3 staffs and 3 administrators that participated on the usability tests.

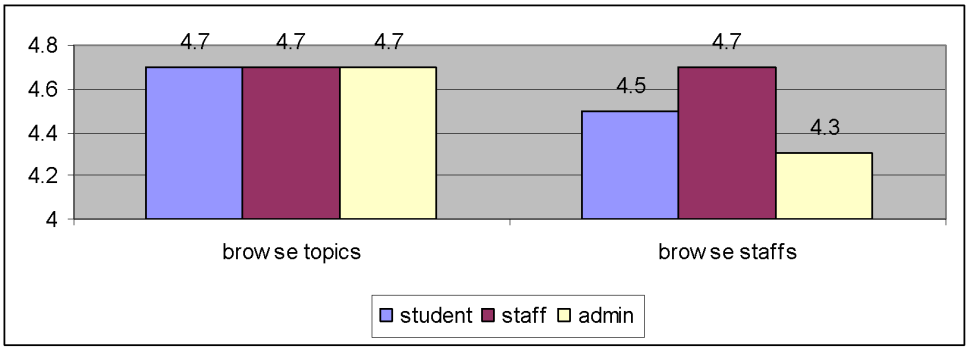
Summary of findings:

In general, participants gave high usability ratings to the functionalities and features provided by the system. One interesting finding on the usability tests is that the number of students preferring preference nomination is almost the same with the ones preferring preapproved nomination. This is considered as an improvement in the preference nomination system provided by WebTMS over the one provided by CSE thesis management system. The overall comment made by participants is that there are major improvements on the browsing/searching system and automation on the nomination process. Additional feedbacks about features are to incorporate undo operation and reordering for topics in the Topic Cart. Other suggestions are to display the topic requests that are sent to the same supervisor at once and to allow students getting approval while nominating topics. These two features have already been taken into consideration in the development phase and have been decided not to be implemented in WebTMS. Displaying topic request sent to the same supervisor will enable the supervisor to accept the topic request with the lowest preference rank; hence, it would violate the students' expectation to have the topic request sent in the order of preference. Allowing approval to be given to students while preference

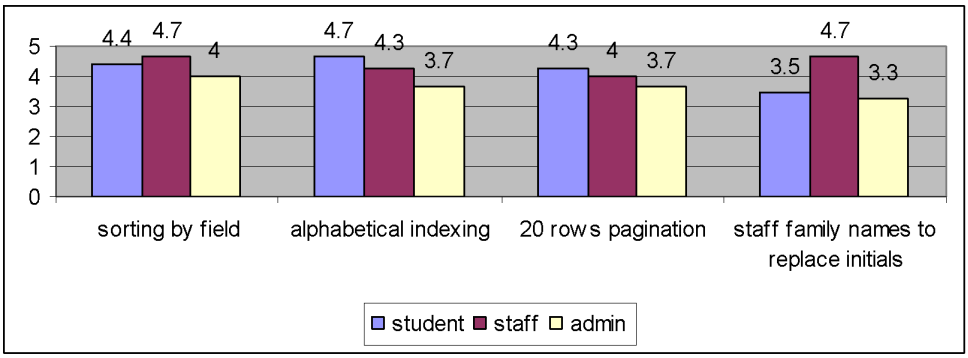
nomination is being processed is considered more flexible, i.e. when staffs grant approval to the students, their nomination is automatically canceled. However, it is not convenient and secure for the students if supervisors grant approval deliberately before the students approach them.

Participant Responses: (the rating that are presented on the charts are calculated from average of all grades that are given from each different type of users)

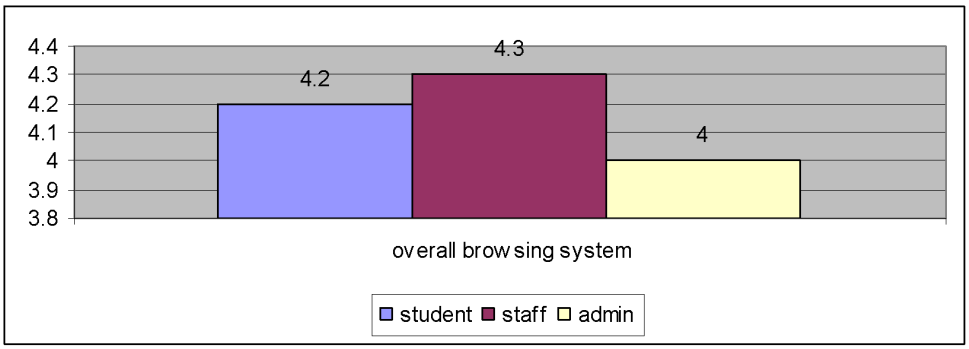
1. Give grades from 1 (very poor) to 5 (very good) to the following functionalities of browsing system in terms of ease of use!



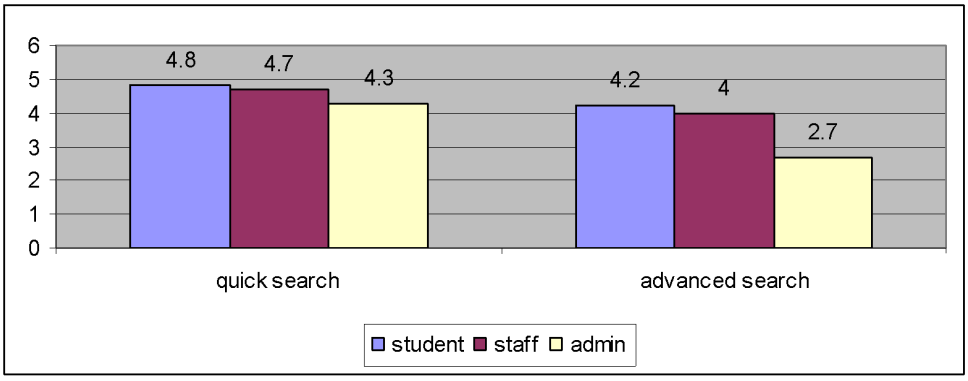
Give grades from 1 (very poor) to 5 (very good) to the following features in terms of usefulness to browse topics!



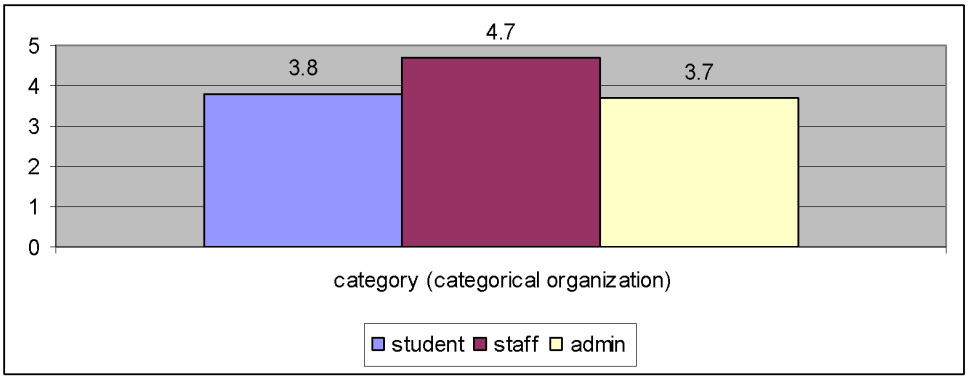
Give a grade from 1 (very poor) to 5 (very good) to the overall browsing system of WebTMS!



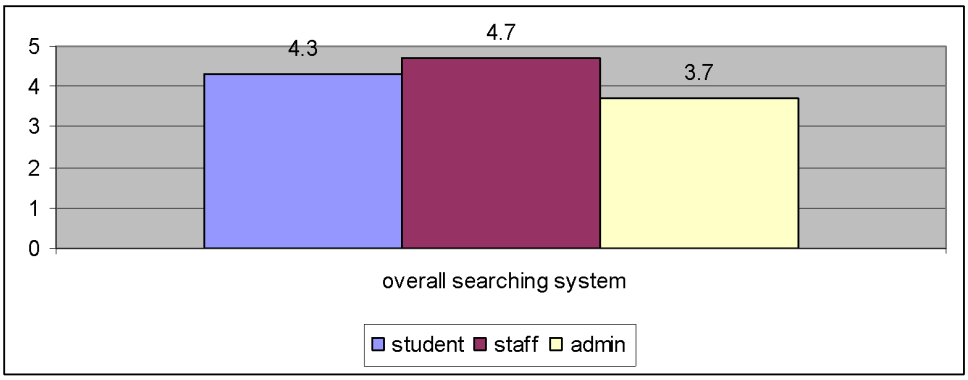
2. Give grades from 1 (very poor) to 5 (very good) to the following functionalities of searching system in terms of ease of use!



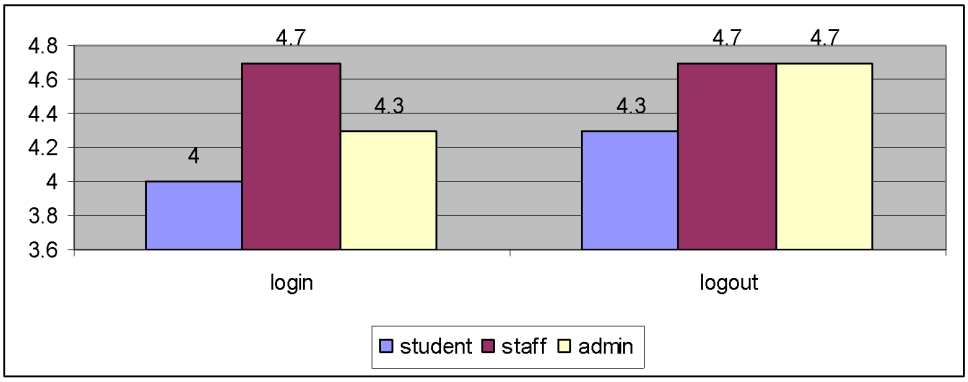
Give a grade from 1 (very poor) to 5 (very good) to the following feature in terms of usefulness to search topics!



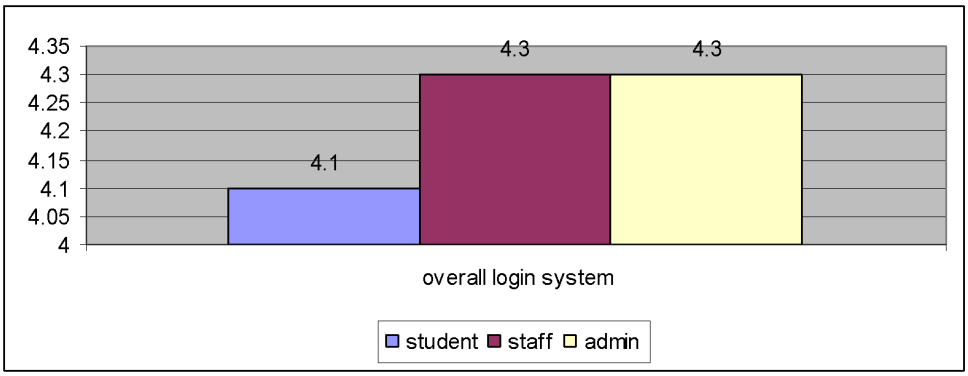
Give a grade from 1 (very poor) to 5 (very good) to the overall searching system of WebTMS!



3. Give grades from 1 (very poor) to 5 (very good) to the following functionalities of login system in terms of ease of use!

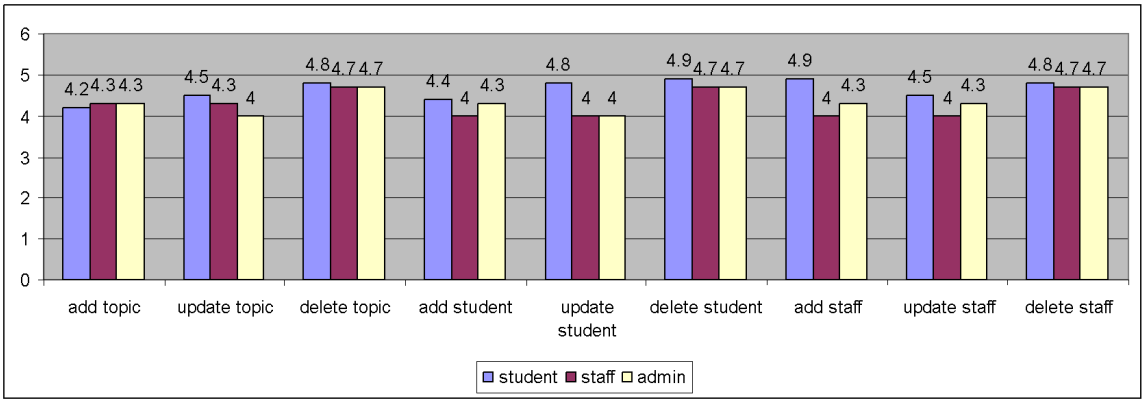


Give a grade from 1 (very poor) to 5 (very good) to the overall login system of WebTMS!

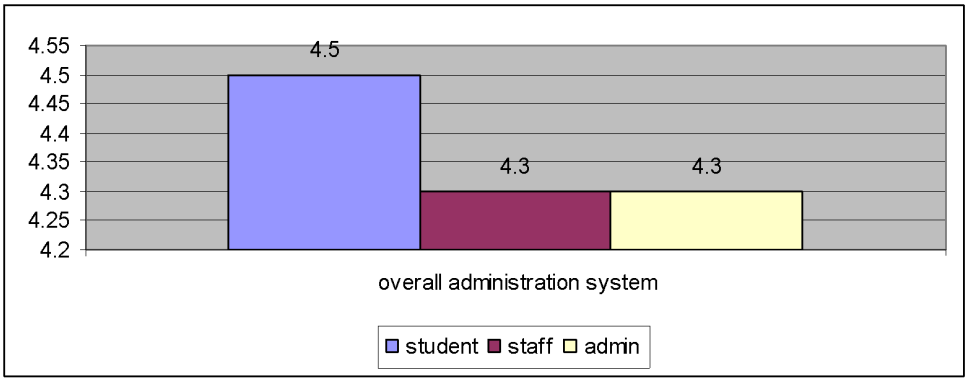


4. Give grades from 1 (very poor) to 5 (very good) to the following functionalities

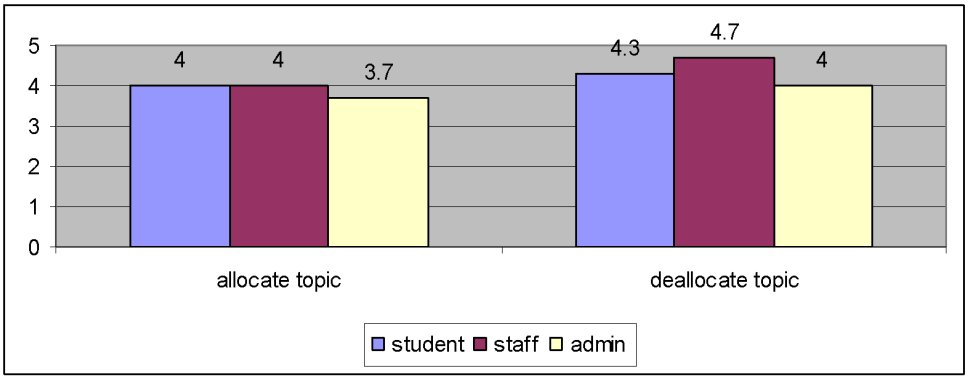
of administration system in terms of ease of use!



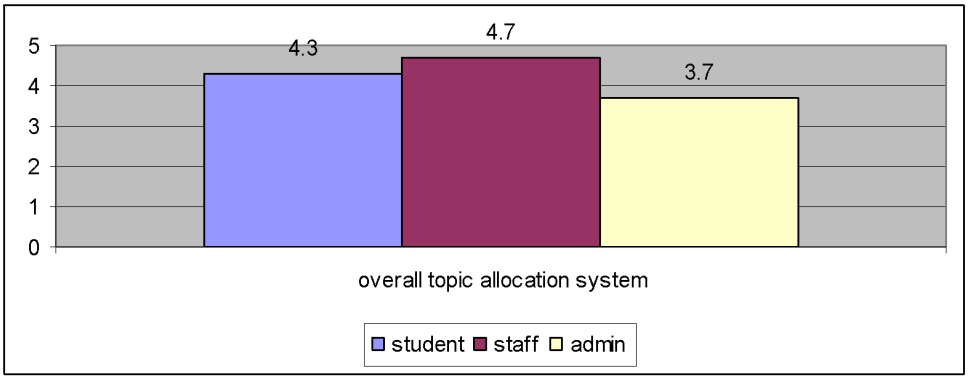
Give a grade from 1 (very poor) to 5 (very good) to the overall administration system of WebTMS!



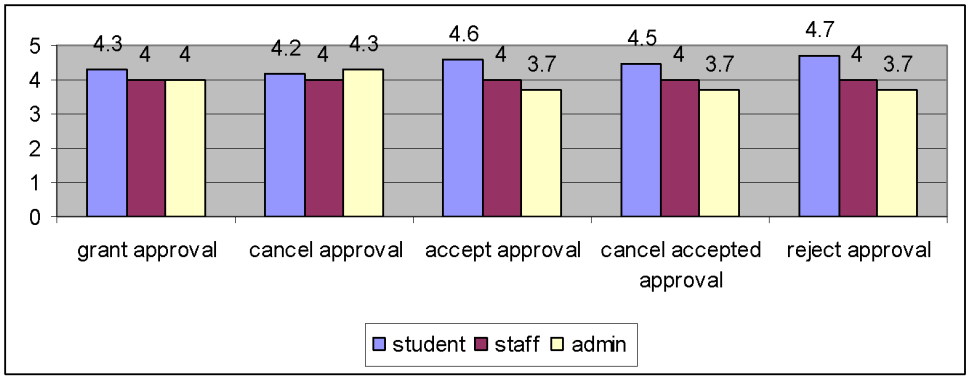
5. Give grades from 1 (very poor) to 5 (very good) to the following functionalities of topic allocation system in terms of ease of use!



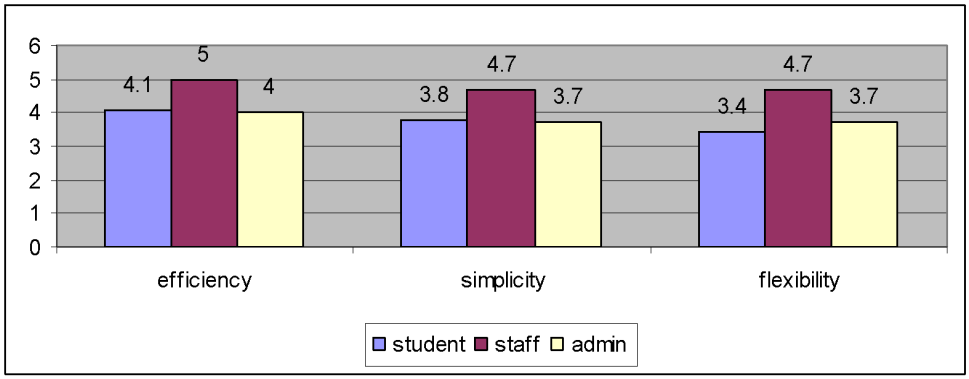
Give a grade from 1 (very poor) to 5 (very good) to the overall topic allocation system of WebTMS!



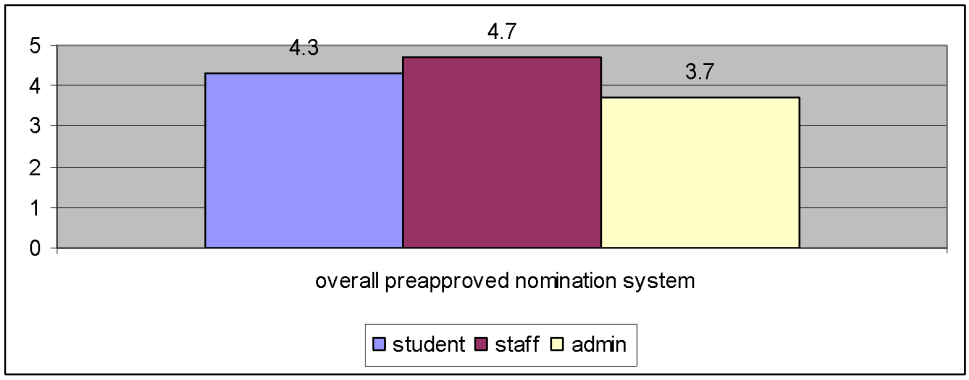
6. Give grades from 1 (very poor) to 5 (very good) to the following functionalities of preapproved nomination system in terms of ease of use!



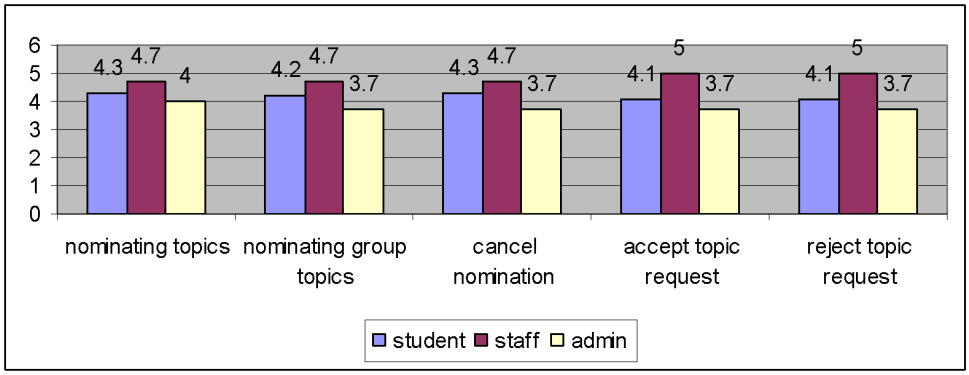
Give grades from 1 (very poor) to 5 (very good) to the following attributes of the preapproved nomination system!



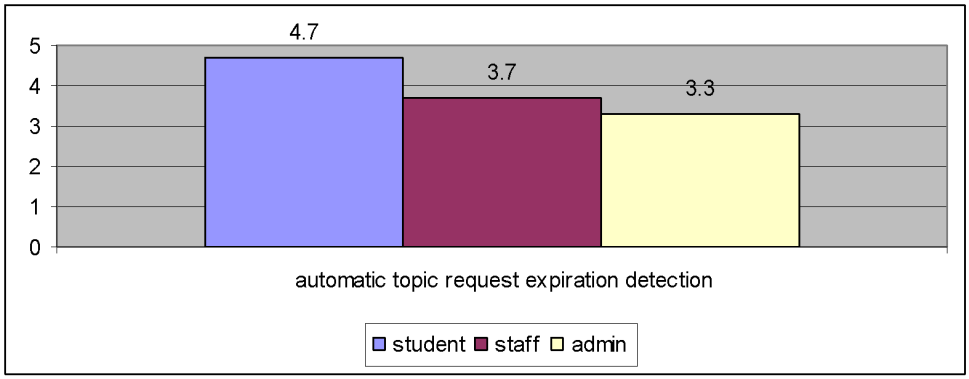
Give a grade from 1 (very poor) to 5 (very good) to the overall preapproved nomination system of the WebTMS!



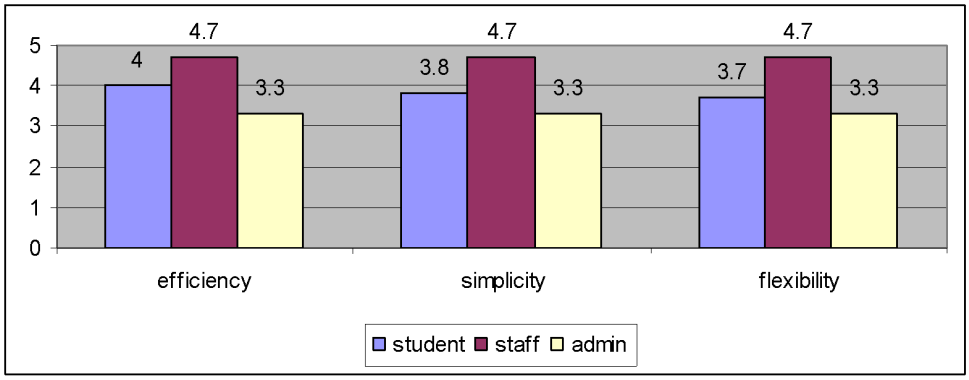
7. Give grades from 1 (very poor) to 5 (very good) to the following functionalities of preference nomination system in terms of ease of use!



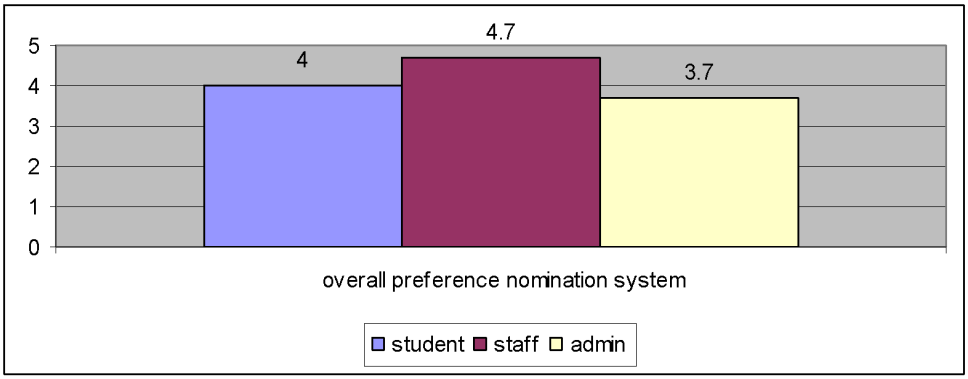
Give grades from 1 (very poor) to 5 (very good) to the following features in terms of usefulness to manage topic requests!



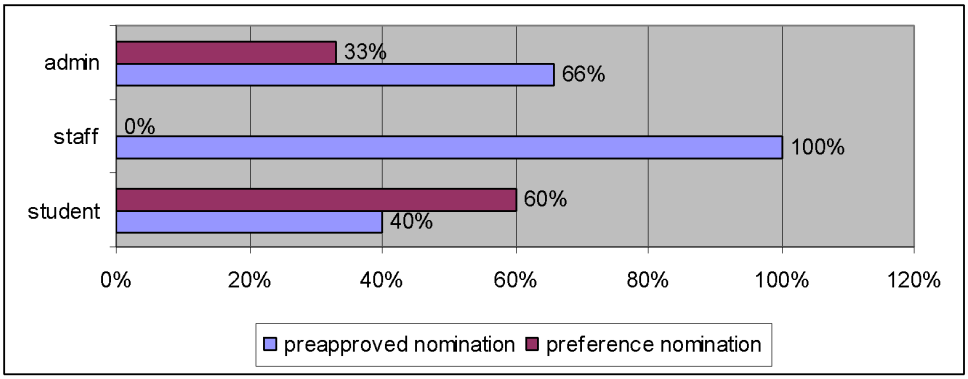
Give grades from 1 (very poor) to 5 (very good) to the following attributes of the preference nomination system!



Give a grade from 1 (very poor) to 5 (very good) to the overall preference nomination system of the WebTMS!



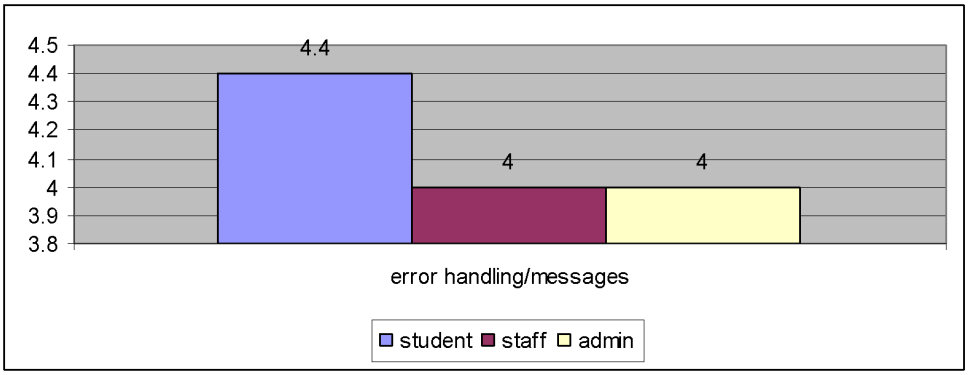
8. Which nomination process you prefer from WebTMS?



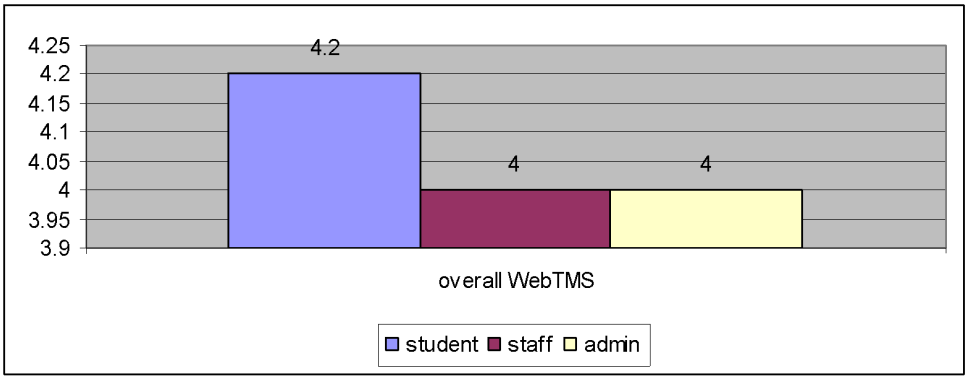
Why?

Most of the students chose preference nomination from WebTMS because it is more efficient way to getting allocated to a topic compared to the preapproved nomination that involve additional overheads such as contacting the supervisor at the first time. Students, staffs and administrators prefer preapproved nomination because it is felt necessary to have a meeting and understand the topic before students undertake the thesis project. In addition, some students also add that group formation in preapproved nomination is much simpler than preference nomination.

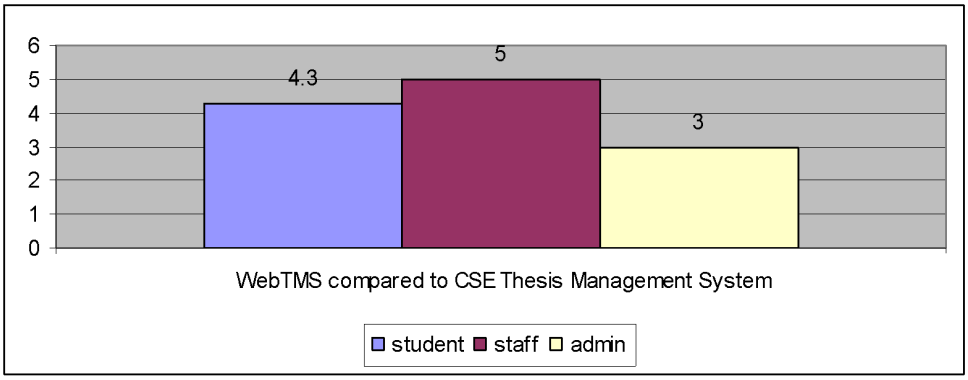
9. Give a grade from 1 (very poor) to 5 (very good) to the error handling/messages help you recover from errors!



10. Give a grade from 1 (very poor) to 5 (very good) to overall WebTMS!



(This question is only applicable for those who have ever used CSE Thesis Management System). Give a grade from 1 (much worse) to 5 (much better) to the WebTMS compared to the CSE Thesis Management System!



11. Any Comments?

A general comment made by participants is that there is a big improvement on the browsing and searching system, as well as automation on the nomination process. One of the staff participants suggests that the system should support undo operation because some of the process could be harmful, such as deallocation of students from a topic. Another staff also adds that the system should allow students to nominate by preference while also retrieve approval from another staff. There is also a suggestion from one of student participants to display the topic requests at once if the topic requests are sent to only one supervisor. Another comment is that the system should be able to reorder the preference topics in the Topic Cart without having to drop the selected topics first.

References

1. Fowler, Martin, 2004, UML Distilled: A Brief Guide to The Standard Object Modelling Language (3rd Edition), Booch Jacobson Rumbaugh, US.

2. Connalen, Jim, 2003, Building Web Applications with UML Second Edition, Booch Jacobson Rumbaugh, US.

3. Quatrani, Terry, 2002, Visual Modelling with Rational Rose 2002 and UML, Booch Jacobson Rumbaugh, US.

4. Elmasri and Navathe, 2003, Fundamentals of Database Systems (4th Edition), Addison-Wesley, US.

5. Kendall and Kendall, 2002, Systems Analysis and Design (5th Edition), The University of New South Wales, Australia.

6. Momjian, 2001, PostgreSQL: Introduction and Concepts (1st Edition), Addison-Wesley, US.

7. Carey, Patrick, 1999, New Perspectives on Creating Web Pages with HTML, Course Technology, US.

8. Zandstra, Matt, 2002, Teach Yourself PHP4 in 24 Hours, SAMS, US.

9. Williams, Hugh E, Lane, David, 2002, Web Database Applications with PHP & MySQL (1st Edition), O'Reilly, US.

10. Sklar, David, Trachtenberg, Adam, 2002, PHP Cookbook (1st Edition), O'Reilly, US.

11. Benatallah, B, Rabhi, F, Mehandjiev, N, 2002, E-Commerce Enabling Technologies, The University of New South Wales, Australia.

12. Holmes, Merlyn, 2002, Web Usability & Navigation, McGraw Hill/Osborne, US.

13. Rosenfeld, Louis, Morville, Peter, 1998, Information Architecture for the World Wide Web (1st Edition), O'Reilly, US.

14. Wodtke, Christina, 2002, Information Architecture (1st Edition), New Riders, US.

15. Nielsen, Jakob, 2002, Designing Web Usability, New Riders AIGA, US.

16. Garett, Jesse James, 2002, The Elements of User Experience (1st Edition), New Riders AIGA, US.

17. PHP.net, 2002, PHP: Hypertext Preprocessor, http://www.php.net

18. Nielsen, J, Ten Usability Heuristics, http://www.useit.com/papers/heurisitc/heuristic.html