Software Requirements Specification

for

Online University Admission System

Version 1.0 approved

Prepared by

B. VIGNESH

VIT University

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

Student admissions are a vital part of any university's running because students are what keep a University alive. The student admission is one of the most important activities within a university as one cannot survive without students. A poor admissions system can mean fewer students being admitted into a university because of mistakes or an overly slow response time.

The process begins with a potential student completing an application form through the Universities and Colleges Admissions Service, the first step for students is to apply directly to the university through a custom online form.

The next step is for the Admissions service center has to review the application and ensure that all of the required information has been provided, from the form itself to the supplementary documentation, such as language and degree certificates. If any of the required information is missing, it is the secretary for the department to which the application concerns that contacts the potential student and arranges for the delivery of the outstanding data.

The application in its entirety is then forwarded, complete with a recommendation, to the respective department's Admissions Tutor, who has the final say as to whether each potential student is accepted or rejected. Before making a decision, the Admissions Tutor reviews the application and the additional documentation, comparing the academic credentials to a list of university rankings and previous, similar applications.

1.1 Purpose

The purpose of this SRS document is to specify software requirements of the Online Admission for the university. It is intended to be a complete specification of what functionality the admission provides. The main purpose of the system is to automate the task carried out by different peoples in the organization to perform the student admission. Specific design and implementation details will be specified in a future document.

1.2 Document Conventions

- Items that are intended to stay in as part of your document are in **bold**
- Explanatory comments are in *italic* text.
- Plain text is used where you might insert wording about your project

1.3 Project Scope

This project's aim is to automate the system, pre-checking the inclusion of all required material and automatically ranking each student's application based on a number of criteria. These criteria include the ranking of their university, their grade at said university and their language grade Certificate. The data used by the system is stored in a database that will be the centre of all information held about students and the base for the remainder of the process after the initial application has been made. This enables things to be simplified and considerably quickened, making the jobs of the people involved easier. It supports the current process but centralizes it and makes it possible for decisions to be made earlier and easier way.

1.3.1 Goals

The main goal of the system is to automate the process carried out in the organization with improved performance and realize the vision of paperless admission. Some of the goals of the system are listed below:

- Manage large number of student details.
- Manage all details of student who registered for the course and send appropriate details about the course to the students account.
- Create student accounts and maintain the data's effectively.
- View all the details of the students.
- Create the statistical reports to facilitate the finance department work.
- Manage the details of hostellers and facilitate the allotment of hostels rooms for the students.
- Reduce the work load in interview the students for selection and Counseling should be very effective rather then direct methods.
- Activities like updating, modification, deletion of records should be easier.
- The System must support Undo the Previous activities if any Problem Occurs.

1.3.2 Objectives of the Proposed System:

The aim of the proposed system is to address the limitations of the current system. The requirements for the system have been gathered from the defects recorded in the past and also based on the feedback from users of previous metrics tools. Following are the objectives of the proposed system:

- **Reach to geographically scattered students.** One of the important objectives of the admission system is communicate with all the students scattered geographically.

- **Reducing time in activities.** Reduce the time taken process the applications of students, admitting a student, conducting the online examination, verify student marks, and send call letters to selected students.

- **Centralized data handling**. Transfer the data smoothly to all the departments involved and handle the data centralized way.

- **Paperless admission with reduced manpower**. Reduce the manpower needed to perform all the admission and administration task by reducing the paper works needed.

- **Cost cutting**. Reduce the cost involved in the admission process.

- **Operational efficiency**. Improve the operational efficiency by improving the quality of the process.

1.4 Abbreviations

- **Verification:** Student verifies the marks they scored in the online entrance exam conducted by the university.

- **Counseling:** University conduct the online Counseling to admit the students in the respective Courses.

- **Course Catalog:** Course Catalog contains all the details about the course and schedule of the course. It is generated by the Superior Persons like Register in the university.

- **Maintenance:** Student information's are maintained in a separate Log for maintenance.

- **Registration:** To participate in the entrance exam conducted by the University, the student must provide all the details about him. This process is called Registration.

- **Deletion:** If the course not like by most of the students and less number of applications are getter from the students means the Course details is temporarily stopped by the administrator.

- **Student Log:** Student information's are maintained in a separate log for future reference and retrieved for any contacting Purpose.

- **Eclipse:** Open Source developed by IBM to support development of complex Java projects in a simple way and it provides easiest way to develop more dynamic web applications that is run on anywhere.

- **HTML:** Hyper Text Markup Language is a markup language used to design static web pages**.**

- **EJB:** Enterprise Java Beans.

- **DB2**: DB2 Database is the database management system that delivers a flexible and cost effective database platform to build robust on demand business applications.

- **WAS:** Web sphere application server is an application server that runs business applications and supports the J2EE and web services standards.

- **HTTP**: Hypertext Transfer Protocol is a transaction oriented client/server protocol between web browser & a Web Server.

- **TCP/IP:** Transmission Control Protocol/Internet Protocol, the suite of communication protocols used to connect hosts on the Internet. TCP/IP uses several protocols, the two main ones being TCP and IP.

1.5 Benefits of the system

As with most real world activities, there are numerous benefits to using a software system for university admissions. The most apparent to this project is the unification of the entire process.

Another benefit of a software system is the use of a central database. This database is the basis for all actions in the system and can be trivially updated and used to aid in all of the system's processes, meaning all of the required information is stored in one central location and thus is easily accessible. This is a far more reasonable storage method than a paper-based file system, where the time of traveling to and physically searching the records for the required information could be a burden. Human error could also be a factor in that mistakes could be made in the filing process which would not occur in a well written database system and mistakes or changes on physical records can be messy to correct.

Software systems are also much faster at performing certain tasks than humans, meaning that time can be saved performing processes such as sending communication e-mails, creating recommendations and the comparison of applications. This also means that these tasks can be done solely by the system, freeing up those involved to perform more important tasks.

In the long term, if methods or minor details concerning the admissions process at universities changes, this can be reflected in potentially minor changes to the code of the system, to retrain employees rather than having regarding the new practices.

1.6 References

- *The document in this file is adopted from the IEEE* Guide to Software Requirements Specifications *(Std 830-1993).*

- Basic Record Structure for designing and developing an OO System given by OMG.

- *Appendix A contains use cases for most of the functionality of the system.*

1.7 Technologies

- J2EE: Application Architecture.
- DB2: Database.
- Eclipse: Development Tool.
- WAS: Web Server.
- Rational: Design Tool.

1.8 Overview

SRS will include two sections.

Overall Description will describe major components of the system, interconnection and external interfaces.

Specific Requirements will describe the functions of actors, their role in the system and constraints.

1.8.1 Overall Description:

The rest of this document will give further details on the overall product description, including the hardware, software, and communications interfaces, product functions, user characteristics, and any assumptions that will be made.

1.8.2 Specific Requirements:

The document will also include the specific requirements needed. These will include the functions, performance, design, and software attributes.

This document is organized in a logical manner and is easy to follow. Readers should refer to the table of contents, appendices, or index if looking for something in specific. Otherwise, reading this document from start to finish will start with a vague description and get more specific and detailed as changing sections and reading further.

2. Overall Description

2.1 Product Perspective

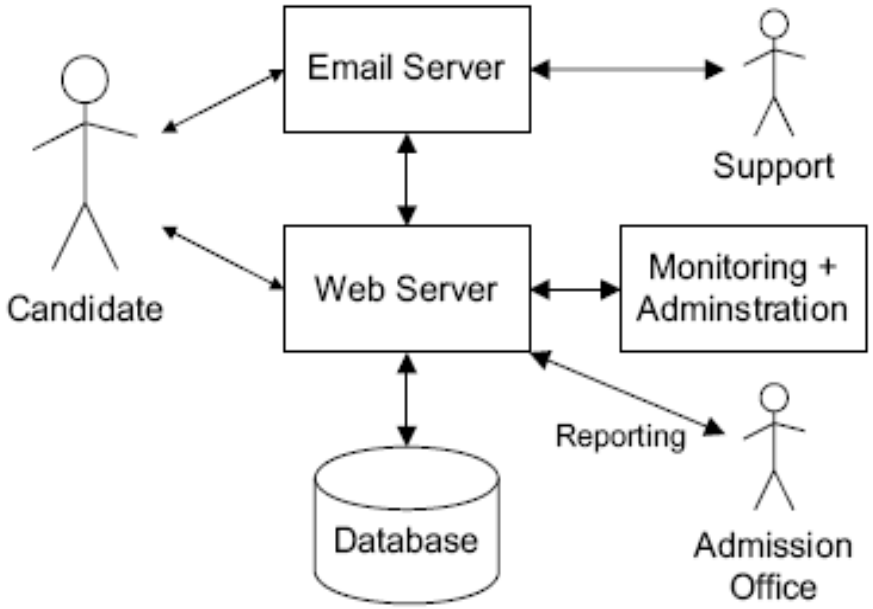

Figure 1: Model of the System

- The web pages (XHTML/JSP) are present to provide the user interface on customer client side. Communication between customer and server is provided through HTTP/HTTPS protocols.

- The Client Software is to provide the user interface on system user client side and for this TCP/IP protocols are used.

- On the server side web server is EJB and database server is for storing the information.

2.1.1 System Interfaces

- **Client on Internet:** Web Browser, Operating System (any)
- **Client on Intranet:** Client Software, Web Browser, Operating System (any)
- **Web Server:** WAS, Operating System (any)
- **Data Base Server:** DB2, Operating System (any)
- **Development End:** Eclipse (J2EE, Java, Servlets, JSP), DB2, OS (Windows), Web server.

2.1.2 Hardware Interfaces

Client Side			
	Processor	**RAM**	**Disk Space**
Internet Explorer 6.0	Pentium II at 500 MHz	64 MB	1 GB
Server Side			
Web sphere application server V5.0	Pentium III at 1 GHz	512 MB	2 GB
DB2 V8.1	Pentium III at 1 GHz	512 MB	1GB (Excluding data size)

2.1.3 Communication Interface

- Client on Internet will be using HTTP/HTTPS Protocol.
- Client on intranet will be using TCP/IP protocol.

2.1.4 Memory Constraints

- **Hardware memory:** The growth of university is unpredictable; to resolve the future problems occurs while enhancing the system is controlled by larger memory as possible. So the memory constraint in the server side is extended up to 1TB.

2.1.5 Site Adaptation requirements

- No site adaptation is necessary in this project. Because the University admission system is portable. The entire system is transported to wherever it is needed. No external dependendancies are in place and operation of the system will never change due to location.

2.2 Product Features

Some of the features are identified for the software. They are listed below:

- **View Course Information's:** The student must able log as student and see all details about course without any constraints.

- **Apply for Course:** The student can able download the application form or register for the course online.

- **Verification of Marks:** The system must allow the student verify marks through online.

- **Advanced Enquiry support:** Enable the students to ask and clear doubts.

- **Online Counseling:** The administrator can able to send the call letters for the short listed candidates, if the student not able contact directly respective authorized persons, than the system must facilitate the online Counseling.

- **Report Generation:** The system supports generation of reports based on different criteria.

- **Record maintenance:** The system also must keep track the statistical reports of daily activities of the Student Registration Process.

- **Online Examination:** Enable the student to write the exams through online in effective way compare with paper based process.

2.3 User Classes and Characteristics

2.3.1 User Characteristics

The Student should have the basic idea to operate (use) the system and he already has the experience to work in the internet (browser). Default Language is English.

2.3.2 User Classes

Some of the users identified for this system through use case analysis are listed below:

- Students
- Data entry operators
- Tutors
- Administrators
- Admission Authorities

2.4 Design and Implementation Constraints

Some of the design and implementation constraints identified are listed below:

- Student is not allowed to register for more than three courses.
- Student not has any rights to edit any data in the system.

- Student pays the application fees in VPP or DD or MO to register for Course.
- Online Payment facility may be restricted if the university not want this facility for some reasons.
- This system is not support distributed database Facility.
- System is limited to HTTP/HTTPS Protocols.

2.5 User Documentation

- Online documentation facility is available for the students to assess them for the easy use.
- A specific document should be prepared for the maintenance of the system and should say the system in easiest way.

2.6 Assumptions and Dependencies

- Courses are already created and information's available for use.
- Roles and responsibilities are already established.
- Administrator is already created.

2.7 Apportioning of Requirements

It is possible in the future that a few additional features be implemented into this system.

- **Management System:** This will allow the system to manage effectively the other resources in the easiest way.
- **Training Facility:** This will allow effectively train the staffs and improve the quality of education in the institution.

3. System Requirements and Analysis:

The following sections will introduce the numerous requirements of the system from the point of view of different users and will introduce a number of decisions that have been made regarding implementation. These sections also attempt to somewhat describe the role of each user group in the system, discussing their individual roles through the functions they can perform.

3.1 User Interface

- The user interface for this system will have to be simple and clear. Most importantly, the ages must be easy to read, easy to understand and accessible. The color scheme should be appropriate to provide familiarity with the university and there should be no contrast issues.

3.2 Student View Functionality:

- **Registration and Login System:** Applicants will carry out their own registration, providing the system with a way to associate a user to their application(s). This will enable the system to display personalized information when the user logs in and certain information, such as name and address, to be added to each application automatically. Giving each student a specific ID will also allow a user to apply to a number of courses, while giving the system a way to prevent unnecessary duplication of applications.

- **Application System:** The application process will be as straightforward as possible, using an intuitive form layout, with the necessary information being completed in stages. When regarding supplementary documentation, such as degree transcripts, these could be uploaded through the form in digital format, upon which it will be saved to the database and associated with the application, being accessible by both the admissions office staff and tutors.

- **Update Details:** Applicants, admissions staff and tutors will all have the ability to update their personal details at any time. Applicants, however, will also be able to update their

application details. After the user has confirmed the update, an e-mail is dispatched with the original and new details as confirmation. The only time an application will be locked for editing will be when it has been submitted to a tutor for review, after which point the application will no longer be accessible by the user.

3.3 Admissions View Functionality:

- **Create New Application:** Registering is not something admissions office staff or tutors will be required to complete. These accounts will be created by the admissions office to prevent unauthorized users obtaining global access, with the login information being given to the appropriate user.

- **Create Application:** For the sake of keeping the system centralized and accessible, should an application be received by post, the admissions office staff would enter the details into a specialized application form. This form is very much like the student view application form, however none of the information is automatically filled in and an account is automatically created for the user.

- **View Submitted Applications:** Viewing all of the recently submitted applications is something the admissions office will do on a regular basis. A list of all the submitted applications, oldest to newest to prevent some applications remaining unread, will be viewable, each of which expandable to view the entire details. This list will be a set size, for example the last two days, but this value will be variable to enable more or fewer applications to be displayed.

- **Generate Emails:** For most users, who apply through the website, communication can be handled most effectively by e-mails. These will be less formal than the documents created by the system, but nonetheless will convey the same information. The admissions office staff will select the type of communication required, based on templates, and include any additional required information and the system will automatically send the e-mail to the correct user.

- **Generate Documents:** For those users who apply by post, communication cannot be carried out through emails and instead formal documents must be created including all of the required information to be posted back to the applicant. This function of the system will generate a number of such documents ranging from acceptance letters to letters regarding missing information.

- **View Logs**: Whenever an action has been completed, a time stamped log should be created by the system, detailing the action completed and the user who performed it for reference purposes. These logs should be viewable by the admissions office staff and by default should display the logs for the past two hours.

- **Edit/Add Universities**: From time to time, a university's rank may change in the tables used by the admissions office. Since this table will be held by the system for automatic ranking of applications, it would be wise to include the ability to edit this information. A member of the admissions office staff will be able to view the list of universities included in the university ranking and edit its details, including name, rank and location.

3.4 Tutor:

- **View Approved Application**: Much like the view submitted applications page for admissions office staff, view approved applications will list the applications, oldest to newest, that were deemed of a suitable quality to forward to an admissions tutor. The main difference with the approved applications is that each is only sent to one tutor, thus there is no need for a locking mechanism.

- **Compare Application**: In some cases, decisions about an application will be simple, given that the application might be exceptionally good or exceptionally bad. If, however, an application is similar to other, previous applications, the tutor may have a more difficult decision to make and inconsistencies may be introduced. Using the automatic ranking of applications a tutor will be able to see a list of applications with a similar ranking. This list will have a default length of 5, for example, but this will be extendible if more comparisons are needed, and the list will include applications of the same rank as well as slightly higher and lower ranks.

3.5 System

- **Validation:** On the completion of each form in the system, the system will use a set of validation functions to ensure that information is of the right type in each field.

- **Make Recommendations:** The system should be able to make recommendations to the tutor which will be decided once an application has been submitted by the admissions office. The system could make its recommendation based on the ranking of the application, where any rank above a certain threshold would be accepted and any below would be rejected.

- **Statistics:** If the admissions office so wishes, they should be able to view statistics gathered by the system regarding applications. These statistics should be displayed on a page with individually expandable sections, such as extending the number of applications received from the past year to the past two years.

- **Report Generation:** Generate reports based on the selected criteria.

4. Supplementary Requirements

- **Immediate Feedback:** The System must try to answer all the queries of the students and it should provide immediate feedback after getting any request from the students. The system must provide the illusion to the students that, they are contact the real peoples for process the Admission task.

- **Reduce the Cost of Admission Process:** The main aim of the System is to reduce the cost needed for Admission Process, so it automatically reduces the manual power needed to perform the entire task and improve the quality of the work.

- **Increase the Quality of the Process:** The System facilitates the work of the universities and the same time it must reduce the work load of the organization with expected quality. Quality in the sense, the system try to avoid the mistakes that are usually happen during the Admission Process because names of the students sometimes missed in the selected list and call letters for the students also not send properly to the qualified students.

- **Make the Interface Simple as Possible:** The System must provide the simple and easy interface for beginners and also provide facilities for technical peoples who are using the system. The interface must be simple as possible.

- **Reduced Time:** To perform any task time is one of the important factors to consider. If the system not utilize properly time, than the entire aim of system is fails and the system is fails to reach its goal. So time take to process all these activities should be less but the output should be effective.

- **Make the System as Global Unit:** The System must provide facilities to tie up with any other existing system and transformation of messages between that other existing system should be not depend upon any other server architecture and any other platform.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Some Performance requirements identified is listed below:

- The database shall be able to accommodate a minimum of 10,000 records of students.
- The software shall support use of multiple users at a time.

There are no other specific performance requirements that will affect development.

5.2 Security Requirements

Some of the factors that are identified to protect the software from accidental or malicious access, use, modification, destruction, or disclosure are described below. Specific requirements in this area could include the need to:

- Utilize certain cryptographic techniques
- Keep specific log or history data sets
- Assign certain functions to different modules
- Restrict communications between some areas of the program
- Check data integrity for critical variables
- Later version of the software will incorporate encryption techniques in the user/license authentication process.
- The software will include an error tracking log that will help the user understand what error occurred when the application crashed along with suggestions on how to prevent the error from occurring again.
- Communication needs to be restricted when the application is validating the user or license. (i.e., using https).

5.3 Portability Requirements

Some of the attributes of software that relate to the ease of porting the software to other host machines and/or operating systems. This may include:

- Java is used to develop the product. So it is easiest to port the software in any environment.

5.4 Maintainability

The user will be able to reset all options and all stored user variables to default settings.

5.5 Reliability

Some of the attributes identified for the reliability is listed below:

- All data storage for user variables will be committed to the database at the time of entry.
- Data corruption is prevented by applying the possible backup procedures and techniques.

5.6 Usability requirements

Some of the usability requirements identified for this system are listed below:

- A logical interface is essential to an easy to use system, speeding up common tasks.
- Error prevention is integral to the system and is provided in a number of formats from sanity checks to limiting free-text input.

5.7 Availability

- All cached data will be rebuilt during every startup. There is no recovery of user data if it is lost. Default values of system data will be assigned when necessary.

5.8 Software System Attributes

There are a number of attributes of software that can serve as requirements. It is important that required attributes by specified so that their achievement can be objectively verified. The following items provide a partial list of examples.

- The input system will allow for inputting numbers, operands, special symbols and letters of the alphabet.

6. Change management Process

As a team, we will update and evaluate our SRS document every week as we make changes in our design and requirements. We will add new detailed information which will include: research, references, charts and graphs, and more specifications and requirements that we find along the way in the designing and implementation of the product.

7. Document Approvals

We have no document approvals as of this time.

8. Supporting information

Appendix A: Glossary

- **ACADEMIC PROGRAM –** An academic program is a broad category for the student's area of academic interest. An academic program "owns" students and is that organizational entity to which a student applies, is admitted, and ultimately graduates from.

- **ACADEMIC INSTITUTION –** Each campus of the University of Missouri will be an academic institution. Separate Institutions allow us to maintain separate code and rule tables for each campus while keeping all four campuses in the same database.

- **ACADEMIC YEAR –** Each term is associated with an academic year for purposes of reporting and financial aid accumulation. (However, a student may have any summer term work changed to point at either the preceding or subsequent academic year.) Accounting is done at a term level and then summarized into a fiscal year which usually parallels an academic year.

- **AP –** Accounts Payable module within the People Soft Finance System. Will be used for issuing student refund and third party sponsor refund checks.

- **COURSE –** A course offered by a school, usually described in the course catalog. A course has a standard syllabus and credit level, although these may be modified at the class level. Courses can contain multiple components such as lecture, discussion, and lab. Courses are not term or even academic year specific, but they do have an effective or starting date. Courses may be entered in pending status.

- **COURSE LIST –** A listing of courses that is used to satisfy an academic or enrollment requirement. Course lists must be established before academic requirements are developed.

- **TERM FEES --** The Term Fee rate table allows us to charge different rates based on the number of units a student is taking in a particular term. We can differentiate the course load by academic structure, campus, location of the course and how the course is taught as well as other student attributes.

- **TUITION GROUP –** A Tuition Group is a group of students who are charged the same set of fees under the same general rules.

- **WASH PERIOD –** That period of time (expressed in days) during which credits and charges resulting from registration add/drop activity “wash” against one another.

Appendix B: Analysis Models

Data Flow Diagram 1:

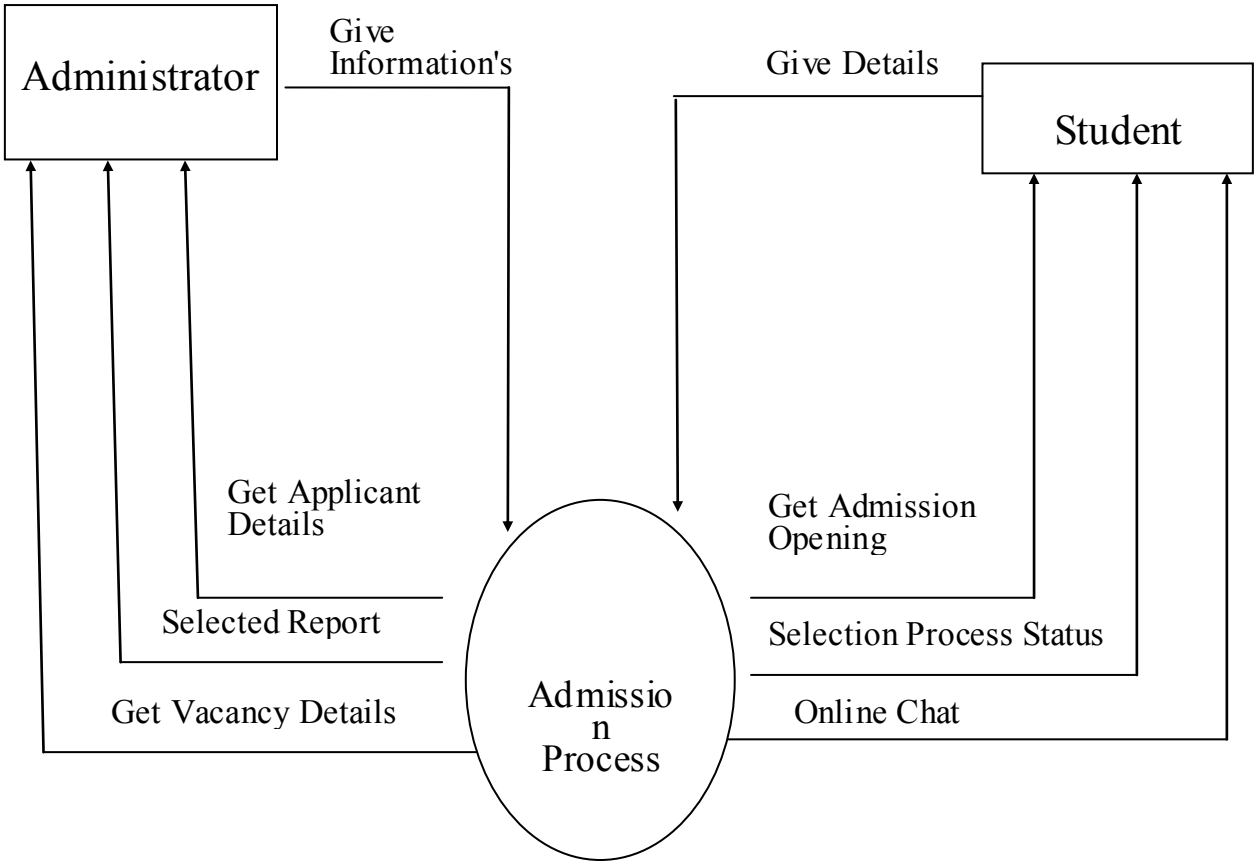

Figure 2: Data Flow Diagram level 0

Data Flow Diagram 2:

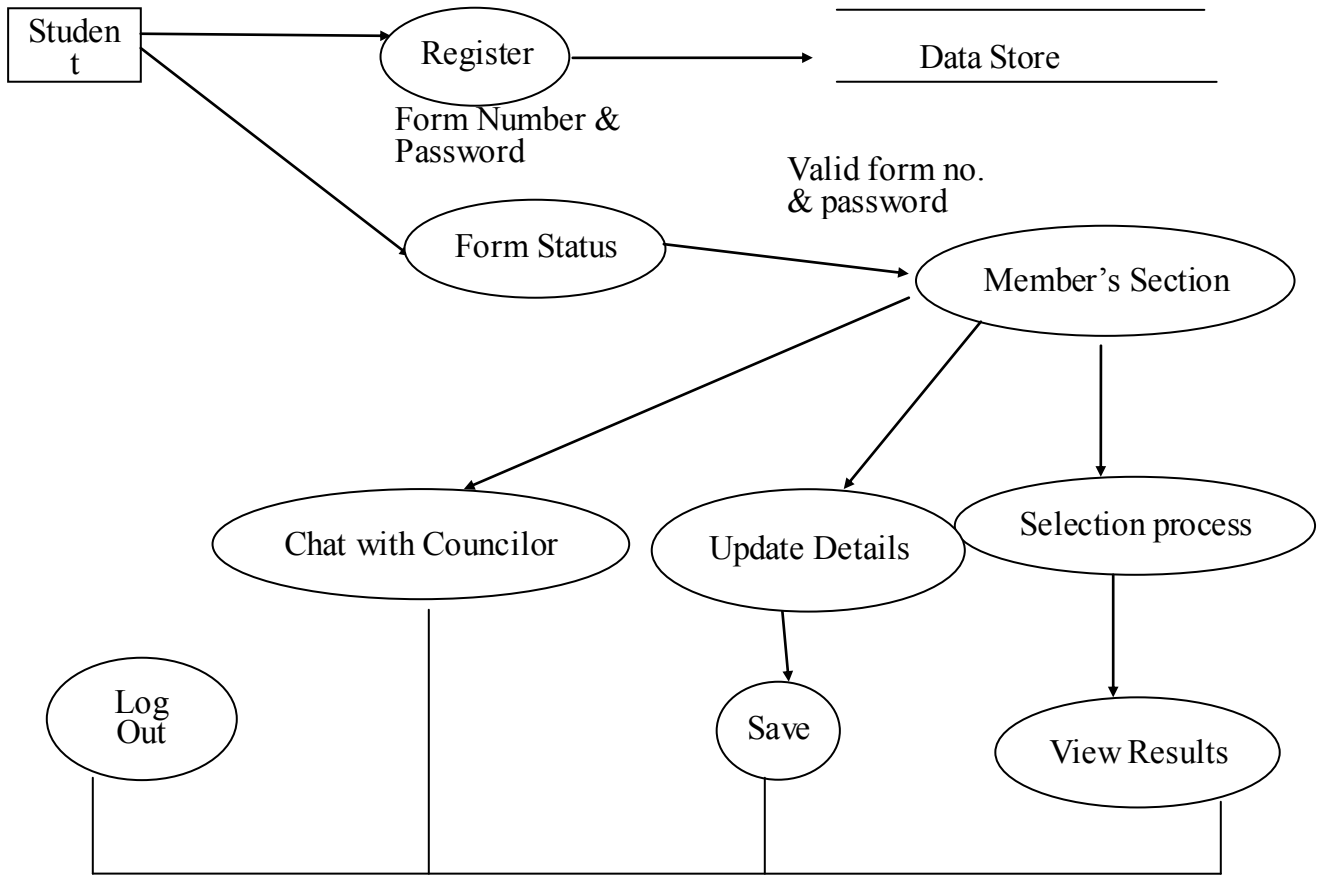

Figure 3: Data Flow Diagram

Overall Use Case Diagram:

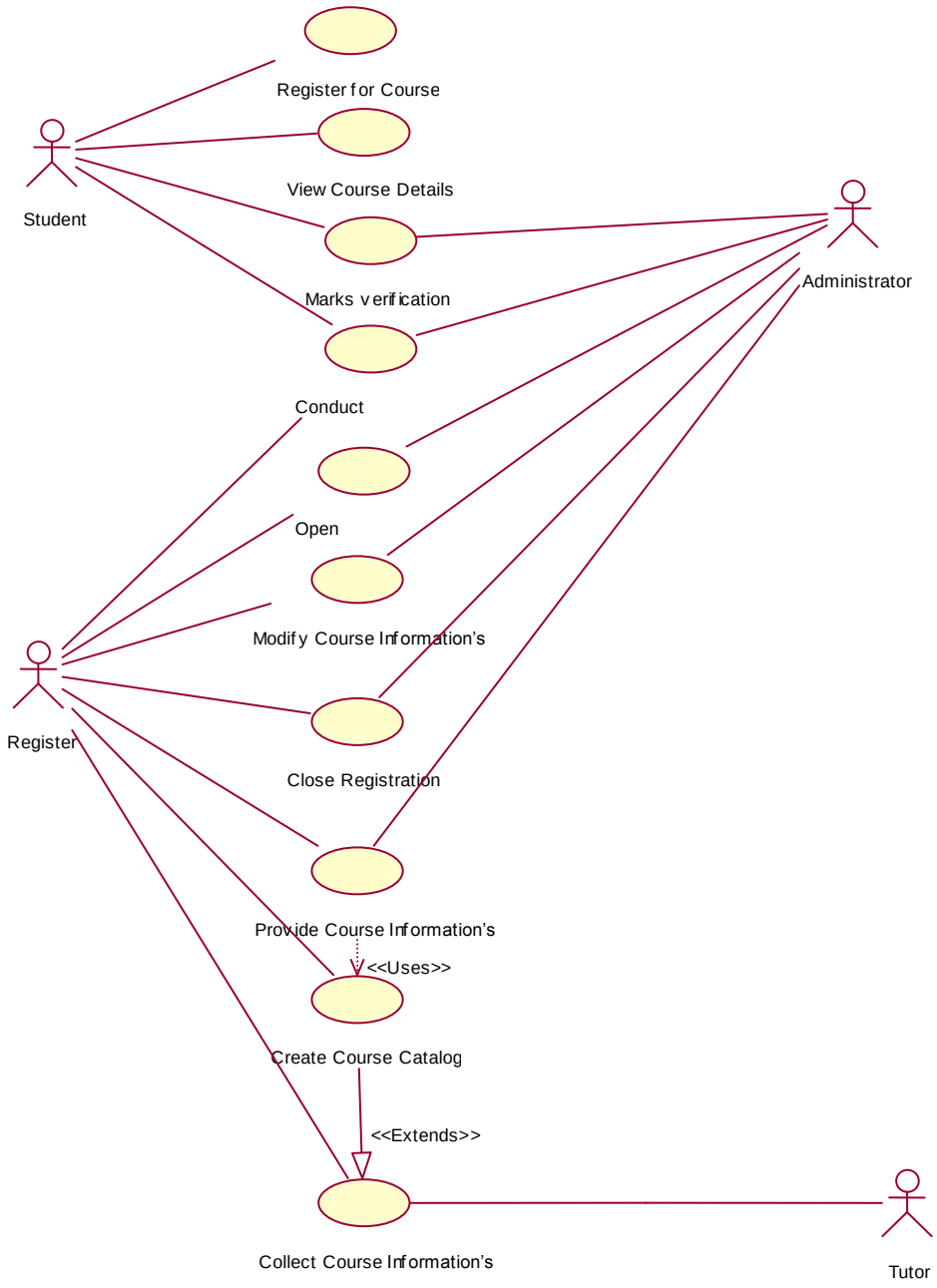

Figure 4: Use Case Diagram