

**Computer Science and Engineering**

**Integrated University Department Information System**

**System Requirements Specification**

**Version 1.0**

Document Number: SRS-001

Project Team Number: **A6**

Project Team Members: **Munieshwar Ramdass (mr3420)**

**Ajay Shenoy (as7195)**

**August Tan (aot221)**

**REVIEW AND APPROVALS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Printed Name and Title** | **Function** | **Date** | **Signature** |
| Professor Strauss | Advisor | Fall 2014 | On File |
| Munieshwar Ramdass | Author, Reviewer | Fall 2014 | On File |
| Ajay Shenoy | Author, Reviewer | Fall 2014 | On File |
| August Tan | Author | Fall 2014 | On File |

**REVISION LEVEL**

|  |  |  |
| --- | --- | --- |
| **Date** | **Revision Number** | **Purpose** |
| October 13, 2014 | Version 1.0 | Initial Release |
| October 22, 2014 | Version 1.1 | Sections 7.2 and 7.3 added.  Document reformatted.  Minor details edited. |

**TABLE OF CONTENTS**

**1. INTRODUCTION .................................................................................................................. 5**

1.1 PURPOSE .................................................................................................................... 5

**2. SCOPE ...................................................................................................................................... 5**

2.1 IDENTIFICATION ...................................................................................................... 5

2.2 BOUNDS ..................................................................................................................... 5

2.3 OBJECTIVES .............................................................................................................. 6

**3. OVERALL SYSTEM OVERVIEW........................................................................................ 6**

3.1 CONTEXT DIAGRAM................................................................................................ 6

3.2 ADDITIONAL DESCRIPTIVE ITEMS ..................................................................... 7

**4. DOCUMENT OVERVIEW .................................................................................................. 10**

**5. REFERENCE DOCUMENTS .............................................................................................. 10**

**6. BUSINESS REQUIREMENTS ............................................................................................ 10**

6.1 TECHNOLOGY ........................................................................................................ 10

6.2 ECONOMICS ............................................................................................................ 10

6.3 REGULATORY AND LEGAL ................................................................................. 10

6.4 MARKET CONSIDERATIONS ............................................................................... 10

6.5 RISKS AND ALTERNATIVES ................................................................................ 11

6.6 HUMAN RESOURCES AND TRAINING .............................................................. 11

**7. SPECIFIC REQUIREMENTS (DESCRIPTIVE FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS) ....................................................................................... 11**

7.1 FUNCTIONAL DESCRIPTIVE DETAILED REQUIREMENTS ........................... 11

7.2 REQUIREMENT USE CASES ................................................................................. 12

*7.2.1 Use Case Diagrams .................................................................................... 12*

*7.2.2 Use Case Descriptions ................................................................................ 13*

7.3 NON-FUNCTIONAL DESCRIPTIVE DETAILED REQUIREMENTS ................. 16

*7.3.1 System Capabilities, conditions, and constraints ....................................... 17*

*7.3.2 Physical Resource Requirements ................................................................ 17*

*7.3.3 Environmental Conditions .......................................................................... 17*

*7.3.4 System Performance characteristics ........................................................... 18*

*7.3.5 Safety Requirements .................................................................................... 18*

*7.3.6 Security and Privacy Requirements ............................................................ 18*

*7.3.7 System Human Interfaces ............................................................................ 18*

*7.3.8 System Maintainability ................................................................................ 18*

*7.3.9 System Quality Factors ............................................................................... 18*

*7.3.10 Design and Construction Constraints ....................................................... 18*

*7.3.11 Personnel-Related Requirements .............................................................. 19*

*7.3.12 Training-Related Requirements ................................................................ 19*

*7.3.13 Logistics-Related Requirements ................................................................ 19*

*7.3.14 Packaging Requirements .......................................................................... 19*

*7.3.15 Precedence and Criticality Requirements ................................................ 19*

*7.3.16 Other non-functional Requirements .......................................................... 19*

**8. SYSTEM TEST PLAN REQUIREMENTS ........................................................................ 19**

**9. QUALIFICATION PROVISIONS ..................................................................................... 20**

**10. REQUIREMENTS TRACEABILITY ............................................................................. 20**

**11. EVOLUTION OF THE SRS ............................................................................................... 20**

**12. RATIONALE ....................................................................................................................... 20**

**13. NOTES .................................................................................................................................. 20**

**14. APPENDICES ...................................................................................................................... 20**

14.1 SCHEDULE TRACKING ....................................................................................... 20

14.2 DEFECT TRACKING ............................................................................................. 21

**1. INTRODUCTION**

This section documents the system requirements specifications (SRS) of the Integrated University Departmental Information System (IUDIS).

**1.1 Purpose**

This documentation will serve the client, development team, and software quality assurance group (SQA). The client must view and confirm that the details provided in this SRS fits their demands. The development team will use this document to produce the proposed product. The SQA group will test the product over its lifecycle using this document.

**2. SCOPE**

The Integrated University Departmental Information System is an organized approach intended to help universities regulate student records, laboratory administration, the ordering of goods and services, and the management of account receivable and payable. It should be linked to a wider University system responsible for staff salaries, student transcripts, etc. The IUDIS structure provides a new information system that is faster with a slicker user interface then the outdated and archaic systems, which makes it easier for students to access their records. Simultaneously, it will be effortless for staff and faculty to use the system for submitting grades and financial information. The new user interface allows for users to interact with the system seamlessly while being aesthetically pleasing, accommodating a broader spectrum of users. The structure of the information system will be rigorously secure to protect the confidentiality, integrity and availability of the individual user.

**2.1 Identification**

Integrated University Department Information System, Team A6, SRS Version 1.0

**2.2 Bounds**

The Integrated University Department Information System is expected to be used by university administrators, who will have complete access, and professors and students who will have limited access to the system. The university’s system may interact with the IUDIS as necessary. Unique to administrators, they can add and delete users. Professors may submit/change students’ grade and request lab materials and other items or personnel which may alter the budget of the university, as a result of extensive research and testing done by PHD student. Professor’s salary will be taken to account. Students may view grades and financial information. Students may make payments to their tuition and lab fees. Third party companies may access the IUDIS if they are a mundane part of the system in which they continuously provide goods/services, therefore building a tab for the university to monitor. The IUDIS will feature a departmental registrar. Ultimately, the IUDIS will only facilitate users of a specific department in a specific university and third party goods/services.

**2.3 Objectives**

Project Priority: Priority is placed on this project since the integration of the departmental budget of the computer science and engineering department is necessary for the account of the university’s budget itself.

Type of Delivery: Incremental

Deliverables: Due Date:

Project Proposal 10/7/2014

Software Requirements Specification (SRS) 10/13/2014

Software Project Management Plan (SPMP) 11/15/2014

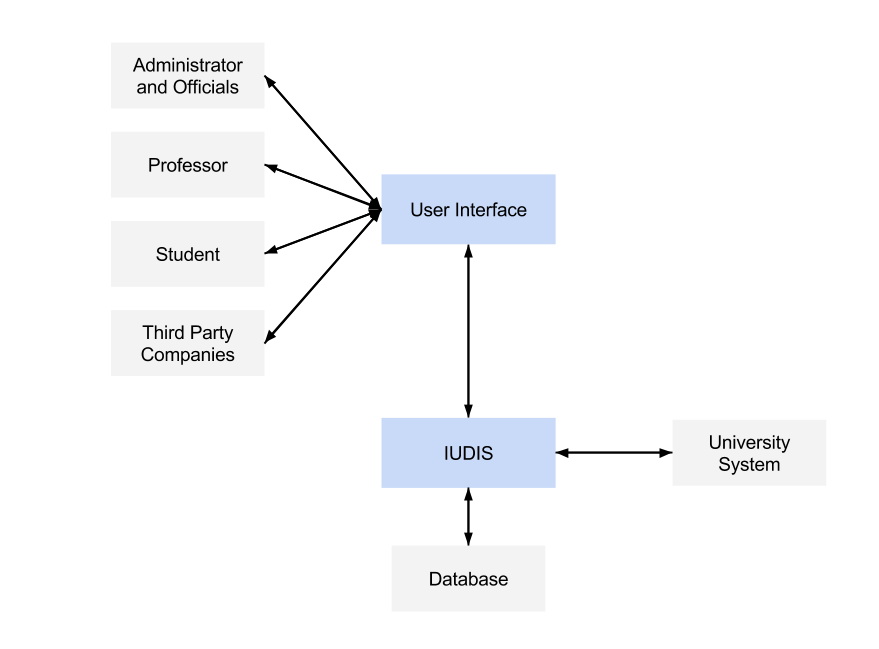
Software Analysis Specification (SAS) 11/19/2014

Software Design Document (SDD) 12/1/2014

**3. OVERALL SYSTEM OVERVIEW**

The Integrated University Departmental Information System is a smaller part of the larger university system that will interact with the budget system and the database system for the computer science department. The IUDIS is concerned with the users of the IUDIS, such as students, professors, advisors, other officials, and third party goods/service providers. The goal of the IUDIS is to please the user by having fast loading, easy to read, clearly arranged, and seamless navigation to all pages. Other factors that may affect the system is the load, the hard disk space, and the amount of down time the server faces. This product may be appear to be a combination of NYU’s Albert and Bursar systems.

**3.1 Context Diagram**



**3.2 Additional Descriptive Items**

Our product will have different classes of permissions and associated user that performs these functions.

* Administrator
  + Assign authorization level to users
  + Perform maintenance on accounts
  + Management of Registrar (permissions to all accounts)
  + Management of accounts receivable and payable
  + Ordering of goods and services
* Professor
  + Review and submit final grades for students
  + Send and review messages to other professors and students
  + Add/Remove from students into his or her class (limited permissions)
  + Consider personnel assistant(s)
  + Request for laboratory equipment
  + Acquire research fund
  + Edit/view own information
  + View student information (limited permissions)
* Student
  + Able to view final grades
  + Able to send and receive messages to other students and professors
  + Management of tuition
  + Able to register for department classes
  + Edit/view own information
  + View professor information (limited permissions)
* Third party companies
  + Billing university for services

Administrator/Officials

This user should have the most technical expertise. He or she will need to have experience with server maintenance and be able to manage a large user base of professors and students. It will be this user’s responsibility to make sure the IUDIS is running and that other users are being given the correct credentials.

Professor

This user needs to understand how to operate the website interface that will be provided by IUDIS. He or she will need minimal technical experience since they will be provided a user friendly interface. However, it will be necessary for them to perform more functions than the user as they will have to manage their classes. This includes publishing final grades and requesting materials for labs/classes. As a part of the department and university budget, a counter will be kept of the professor’s salary and research funds.

Student

This user is similar to the professor. They will have minimal technical experience as they will be provided an easy to use interface. Their level of technical ability will be lower than that of professors since students only need to be able to review their final grades, register for courses, and manage their tuition.

**Constraints**

Regulatory

The IUDIS will have to maintain a user base for large universities. In the case of New York University, this is 53,771 students plus other faculty members involved with the school. It is important that the IUDIS be able to maintain constant uptime over the semester for all users so that assignment submission and creation is not delayed.

Hardware limitations

Ideally, there would be no hardware limitations. We would be using the largest and fastest hardware for our servers. However, the IUDIS system strives to be efficient. Meaning that our hardware will be relatively low cost while able to perform at an optimal speed.

Interface to other applications

The IUDIS will associate with a University system and third party companies. Users will be able to access our system by logging in through it. We don’t want the system to be cluttered with excess services so IUDIS will be minimize its interface with other applications.

Operations

IUDIS will be keeping operations to the minimum. We want to be able to perform the necessities for a university and nothing more in order to keep it simple and efficient.

Control functions

The various user classes will have different control functions. The administrators will have no constraints and will be given full access to the system. Professors will have some constraints to their control functions. Their control functions will be limited to their classes. Students will have the most control function constraints. Their control is limited to their own personal accounts with the associated personal information.

Development language

Under the running idea of keeping IUDIS simple, the number of different development languages will be kept to a minimum. Ideally, there will be only one development language. However, we will be using additional languages if it means a faster service.

Communication protocols

All communication will be occurring internally within the system. There will be no interaction with users from IUDIS of one university and those from another university. For example, students will be able view their grades that professors post. Members may be able to message another member of the system directly.

Safety and security

To prevent any unauthorized users professors and students will be unable to create new accounts. That responsibility is delegated to the more experienced administrator who has access to the list users that need accounts.

Assumptions and dependencies

Server hardware that can manage a large University database is available. Enough technical experience from developers to create site is present. Project is funded entirely by the University. Users consist of students, professors, other faculty, and administrators.

Requirements subsets

The most important subset of requirements

* A database that can manage IUDIS
* Different type of user accounts with their associated permissions
* A web interface that allows users to interact IUDIS

Requirement subsets that can be delayed until future release

* A simple web interface that is user friendly
* A minimalist system that runs as fast as possible while being able to serve a University’s needs.

**4. DOCUMENT OVERVIEW**

This SRS document starts with a general overview of our system. It entails the purpose of the IUDIS, objectives, and descriptive items such as function and user characteristics. After the overview on the purpose and the functions of IUDIS, the document moves on to technical specifics and business aspects. It details how we plan on the system will be funded and describes the languages, framework, and server that it’ll be using.

After this document overview, the SRS document will go over the University system’s technology, how the system is going to be funded and generate revenue, and possible risks as well as their solutions. Detailed descriptions of each requirement can be traced from source to instances of artifact.

Finally, this documents details how the IUDIS will be tested for quality and maintained after development. It describes how the progress of this product will be tracked.

5. **REFERENCE DOCUMENTS**

Integrated University Department Information System, Team A6, Project Proposal Version 1.1

Integrated University Department Information System, Team A6, Software Project Management Plan Version 1.0

6. **BUSINESS REQUIREMENTS**

**6.1 Technology**

As universities grow larger, a smaller, independent system will be recommended to split the wider university system tasks into smaller jobs. Such a system would be a departmental system as described in this SRS. This way departmental systems will handle department task and report to main university system hereby cutting the work for the wider university system.

**6.2 Economics**

IUDIS must be cost efficient. It must able to assist all user in a timely fashion. Therefore, the server’s maintenance is critical.

**6.3 Regulatory and Legal**

Regulatory and Legal requirements will not be discussed.

**6.4 Market Considerations**

This system is made for very populated universities and these universities may choose to purchase the rights via subscription if it works with their larger university system.

**6.5 Risks and Alternatives**

Unexpected functionality may occur. To prevent this, administrators must rigorously test the system for flaws and fix it throughout the system’s lifecycle. Since this system can be accessed through the internet, hackers might be able to hack the database. To prevent this, an intranet can be use.

**6.6 Human Resources and Training**

Priority is placed heavily on training individuals in object oriented design. This is especially needed for information hiding. Training in the use of databases is also required. Knowledge in front and back-end web development is necessary.

**7. SPECIFIC REQUIREMENTS (DESCRIPTIVE FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS)**

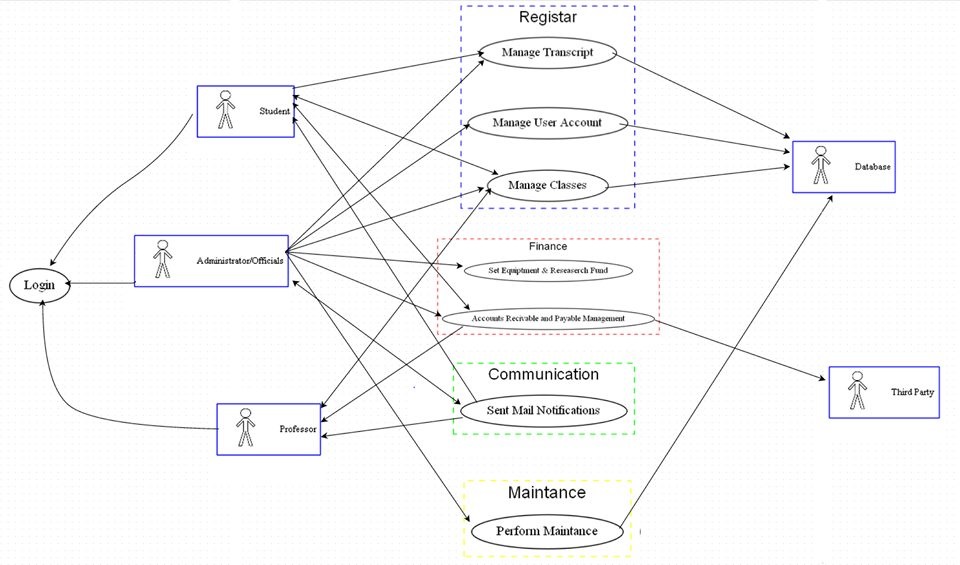
In this section, software requirements are specified so that developers may be able to satisfy these requirements. Definitions and other supporting details are listed within each requirement. *Specifically for section 7.1, four categories of functional requirements are described. In each category are the specific requirement along with its definitions.*

**7.1 Functional Descriptive Detailed Requirements**

1. There must be a method to handle the Management of Registrar.
   1. Administrators/Officials must be able to create/modify/delete/view class.
      1. Professors may vote for a prerequisite for a class.
      2. Professors may vote on credit level for a class.
      3. Officials must be able to set time for class.
      4. Professors may to vote and associate a lab and/or recitation to class.
   2. Administrators/Officials must be able to create/modify/delete/view any user account.
      1. Administrators must be able to assign user permissions.
      2. Administrators must have absolute control.
      3. University’s system’s administrators must have absolute control.
   3. Officials must be able to associate students and professors to classes.
      1. Professor must be able request to teach a class.
      2. Students must be able request to take a class.
      3. Professors must be able post grade for students in given class(es).
   4. Officials must be able to create account holds.
      1. Hold can depend on user financial status.
      2. Hold can depend on student prerequisite.
      3. Hold can depend on other predefined reasons by officials.
   5. There must be a way to update student transcript and public them.
      1. Student must be able to request Unofficial Transcript
      2. Student must be able to request Official Transcript
      3. Officials must be able to sign Official Transcript
      4. Grades and courses must be reflected on the student’s transcript
2. There must be a method to handle the Management of Finance.
   1. Administrators/Officials must be able to set tuition for students based on university system.
      1. Officials must be able allow third party company to allow student scholarship/grant/payments.
      2. Students must be able make payments via their means of paying.
   2. Officials from university system must be able to set salary for all department employees based on university’s system’s officials.
      1. A direct deposit system may be used from university system.
      2. A check mailing system may be used if direct deposit not used.
   3. Officials must be able to set Research funds.
      1. Professors must be able to request research funds.
      2. Officials must be able to approve/disapprove professor’s request for fund.
   4. Officials must be able to set department’s equipment fund.
      1. Professors must be able to request goods and services.
      2. Officials must be able to approve/disapprove professor request.
   5. Goods/services from third party sites must be paid.
      1. Officials may make payments.
      2. University’s system’s officials may handle payments.
3. There must be a method of Communication.
   1. The use of the integrated university’s emailing and mailing system will be used.
      1. Emails will primarily be notifications from user account changes
      2. Officials may choose to quickly send an email from user interface to the person of whom s/he is currently viewing
4. There must be a method to handle the Maintenance of IUDIS.
   1. Administrators must have access to entire IUDIS and its database.
      1. Administrators must be able to debug the entire system.
      2. Integrated permissions depend on larger university system

**7.2 Requirement Use Cases**

**7.2.1 Use Case Diagrams**

****

**7.2.2 Use Case Descriptions**

|  |  |  |
| --- | --- | --- |
| Login | | |
| Description | Admins, Officials, Professors and Students can use their credentials to log into the system with permissions suited based on their status. Incorrect credentials does not allow access. | |
| Pre-Conditions | User must be a part of university’s department that uses this software. User must have their account created by admin/official. | |
| Flows | Basic or Normal Flows | 1 User enters username and password.  2 System checks database for correct login credentials.  3 User now have access to their account. |
|  | Alternative Flows | 1 From step 1 above, if user enters wrong credentials, prompt error. Use case ends.  2 Use case ends if user quits/log out. |
| Post Conditions | Depending on who the user is, they will be able to make different transactions within the IUDIS. | |
| Special Requirements | Only one log in per device permitted. | |
| Extension Points |  | |

|  |  |  |
| --- | --- | --- |
| Manage Student Transcript | | |
| Description | Based on person that is logged into the system, user will either be able to edit and/or view transcript. | |
| Pre-Conditions | User must be logged in. | |
| Flows | Basic or Normal Flows | 1 Officials make and view students’ transcript.  2 Professor will then be able to edit grades.  3 Students will lastly be able to view grades on a transcript. |
|  | Alternative Flows | 1 Student may request a copy.  2 Transcript not available: present nothing to show or edit until Official makes it for student. |
| Post Conditions | There must always be an existing student transcript stored in the database, even if the student is no longer in the University. | |
| Special Requirements |  | |
| Extension Points |  | |

|  |  |  |
| --- | --- | --- |
| Management of Classes | | |
| Description | This is how the registrar’s classes are created and filled with students and associated to a professor that will teach the class. Based on user permissions, user will be able to do different things. | |
| Pre-Conditions | User must be logged in. | |
| Flows | Basic or Normal Flows | 1 Official creates (opens) a class  2 Professor may request teaching it  3 Professor may vote on class’s credit level and prerequisites.  4 Student may choose to register for class or remove themselves from class. |
|  | Alternative Flows | 1 Block student from registering for class if s/he does not have prerequisites.  2 Block professor from teaching if s/he is not fit to teach it.  3 Block student if his/her financial side is not taken care of; create an account hold in the Management of User Accounts used case. |
| Post Conditions | Either student is registered for classes or professor given a class. | |
| Special Requirements |  | |
| Extension Points |  | |

|  |  |  |
| --- | --- | --- |
| Management of User Accounts | | |
| Description | This is how administrators/officials will create and edit user accounts. Students and Professors will be able to view their account. | |
| Pre-Conditions | User must be logged in. | |
| Flows | Basic or Normal Flows | 1 Admins/officials can create user account, which user will be able to use in the department.  2 Professor will be able to view their account information  3 Student will be able to view their account information |
|  | Alternative Flows | 1 Account hold created if user’s financial side has not been taken care of. Access will be blocked to registrar until they handle their financial side in the Account Receivable and Payment Management used case.  2 Students may be restricted to prerequisites. |
| Post Conditions | There must be an existing account if creation or editing of user account is done. There must be no existing account if admin/official deletes account. | |
| Special Requirements |  | |
| Extension Points | Usability of account may be blocked and/or refer to Receivable and Payable use case. | |

|  |  |  |
| --- | --- | --- |
| Set Departmental Equipment and Research Fund | | |
| Description | This is where budget that is set aside for other things get taken into consideration. | |
| Pre-Conditions | User must be an Administrator or Official to access this used case. | |
| Flows | Basic or Normal Flows | 1 Tally fund from university necessities such as equipment.  2 Tally any requested research fund.  3 Admin/Official may view this resulting sum. |
|  | Alternative Flows |  |
| Post Conditions | The sum of fund must reflect on wider university system. | |
| Special Requirements |  | |
| Extension Points |  | |

|  |  |  |
| --- | --- | --- |
| Account Receivable and Payable Management | | |
| Description | This is where students pay their tuition, employees receive their payments, and Officials pay third party for their services. | |
| Pre-Conditions |  | |
| Flows | Basic or Normal Flows | 1 If student, s/he will view his/her balance. If official, view third party’s bill or view all employees pending paychecks. If professor, view paycheck.  2 Student may enter account s/he wishes to pay. If official, enter payment amount for third party transaction or send out employee paychecks.  3 If student, send student notification. If official, log transaction. |
|  | Alternative Flows | 1 Invalid amount: re-prompt student or official to enter amount from step 2 above.  2 Invalid date: Officials may only send paychecks out on certain dates (step 2 above). |
| Post Conditions | System logs and/or notifies via email of any transaction. | |
| Special Requirements |  | |
| Extension Points |  | |

|  |  |  |
| --- | --- | --- |
| Communicate | | |
| Description | Mailing and notifications will be handled here. | |
| Pre-Conditions | Depending on reason for mail or notification, this action might be invoked by another used case. | |
| Flows | Basic or Normal Flows | 1 If any used case perform an action, mail recipient (or the user that that action was performed on).  2 Officials may access emailing directly and send out emails to whoever they wish to send to. |
|  | Alternative Flows | 1 Account does not exist: send an error email back to the sender. |
| Post Conditions | A user gets a mail. | |
| Special Requirements |  | |
| Extension Points | Used cases from Registrar or Financial categories may access this use case at step 1. | |

|  |  |  |
| --- | --- | --- |
| Perform Maintenance | | |
| Description | Administrator will be allowed to shut system down and debug system or perform updates as necessary. | |
| Pre-Conditions | User must be Administrator. | |
| Flows | Basic or Normal Flows | 1 User can debug entire system |
|  | Alternative Flows | 1 Restart system. |
| Post Conditions | System must be up and running. | |
| Special Requirements | System must be shut down or unavailable to other users. | |
| Extension Points |  | |

**7.3 Non-Functional Descriptive Detailed Requirements**

IUDIS will provide a University with a system to manage their registrar, finance, communication, and maintenance. These services include student tuition, student grades, faculty salaries, and other expenses. The IUDIS will provide different utilities depending on the user’s class. For instance, an Administrator will be able to create new accounts, remove accounts, modify accounts, manage academic records, create/modify classes, and manage salaries, tuition payments, and any other expenses.

Meanwhile, a Professor will have some limitations compared to an Administrator. He or she will be assigned a class by an Official. He can also post student grades for his class. Lastly, a professor will be able to request funds for research.

Another user class will be Officials. They can create, modify, delete, or view any account. They can associate students and professors to classes.

The fourth user class that this system must support are students. They will be able to pay their tuitions and takes classes.

**7.3.1 System Capabilities, conditions, and constraints**

This system will be capable of managing a university’s finance and registrar. It will be designed with large universities in mind. Most importantly, this system will be able to perform efficiently with minimum downtime and maximum speed. One of the constraints of this system is that it will not be able to take online submissions of homework assignments as that would add too much weight to the system and slow it down.

**7.3.2 Physical Resource Requirements**

The physical requirement for this system will be a room large enough to house the server as well as computers that can provide a direct interface into these servers.

**7.3.2.1 Computer Hardware Requirements**

The server hardware must be able to support a user base of around 50,000, the average size of a large university.

The minimum server specs are: 4CPUs 2.6 GHZ and 4 gigs of Ram and 10TB Disk Space

**7.3.2.2 Computer Hardware Resource Requirements**

There will be at least 3 different servers: application server, database server, and a tunnel server. Also there will be a computer that will allow an administrator to directly interface with the system.

**7.3.2.3 Computer Software Requirements**

Server - SQL

Framework - Maven

Interface - HTML5, CSS3, JavaScript

Languages - Java 1.6

Any Browser

**7.3.2.4 Computer Communications Requirements**

Any computer must be able to sign on to the system provided that they have the correct credentials.

**7.3.3 Environmental Conditions**

Ideally, the servers will be kept in a cool room so that they do not overheat.

**7.3.4 System Performance characteristics**

The system should be able to manage a large amount of traffic. It should be able to handle at least 100 simultaneous user commands.

**7.3.5 Safety Requirements**

Anyone who is working near the server hardware should be careful not to cause any physical damage to it.

**7.3.6 Security and Privacy Requirements**

Users should have basic password protection knowledge. Passwords must be at least 12 characters long with uppercase and numerical characters. The server will have a firewall to protect it from being hacked into.

**7.3.7 System Human Interfaces**

Users will interact with the system via browser. They can log into the system and access all of its functionalities from a website.

**7.3.8 System Maintainability**

This server will be responsible for holding school records throughout the semester. The system should have a 99% uptime and as little maintenance as possible will be required.

**7.3.9 System Quality Factors**

The system should have a clean and simple interface while looking modern. Users should experience very little downtime and be able to process their transactions quickly.

**7.3.10 Design and Construction Constraints**

The system will be limiting students and professors from creating and submitting assignments online. The system must be able to be created by a small team (3 people).

**7.3.10.1 Life Cycle Model**

The IUDIS will be developed under Extreme Programming. The principle of extreme programming which is to minimize the number of features is also the focus of our system. Extreme programming will also allow us to accommodate to any changes in requirements to suit the University market.

**7.3.10.2 Policies and standards - Methods, tools, and techniques**

**7.3.11 Personnel-Related Requirements**

**7.3.12 Training-Related Requirements**

The system will be usable by a University student with an average computing background. He or she must be able to navigate the system’s website interface and perform transactions on it. If the user has little to no computing knowledge then a short 15 minute training session will be able to bring him or her up to speed.

**7.3.13 Logistics-Related Requirements**

The system must be able to accommodate a large user-base of around 50,000. Downtime, especially when grades are due, is a priority to prevent.

**7.3.14 Packaging Requirements**

None

**7.3.15 Precedence and Criticality Requirements**

None

**7.3.16 Other non-functional Requirements**

None

**8. SYSTEM TEST PLAN REQUIREMENTS**

It is up to the SQA group to test this project using its database and a university system (or replica university system) for which this product will be integrated to. Following that, the user interface will be tested by people unaware of programming. Usability, efficiency, and competency will be taken note of.

**9. QUALIFICATION PROVISIONS**

The outline of this document was provided by our supervisor. It will be modified with details while keeping the outline in place. Details are traceable from the Project Proposal.

**10. REQUIREMENTS TRACEABILITY**

Requirements discussed in this document are traceable to the Project Proposal. As a result, artifacts are traceable to the Project Proposal. Future artifacts will be associated with the appropriate requirement number.

**11. EVOLUTION OF THE SRS**

In the event of a change, the revision level section of this document will inform us of the change. The document version will also increment appropriately. Reasons for changes will also be noted within the new revision. A reference to the older version of this document will be kept in the Reference Documents section.

**12. RATIONALE**

The rationale behind the Integrated University Department Information System is to allow a seamless integration between huge university systems and the department system. The computer science department may have many small and independent functions from the main university system and so it should prove to be easy if the department handles its budget by itself while the larger university system tallies the department’s budget into its own budget. Furthermore, students’ grades and registration should be handled likewise.

**13. NOTES**

The project described in this Software Requirements Specification was chosen and modified from a list of projects provided by our advisor.

**14. APPENDICES**

**14.1 Schedule Tracking**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or  Deliverable | Who | Estimated | Actual | Difference |
| Initial SRS | Munieshwar Ramdass | 5 | 6 | 1 |
| Initial SRS | Ajay Shenoy | 6 | 5.5 | 0.5 |
| Initial SRS | August Tan | 4 | 5 | 1 |
| Initial SRS | Summary | 15 | 16.5 | 2.5 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or  Deliverable | Who | Estimated | Actual | Difference |
| Final SRS | Munieshwar Ramdass | 3 | 3 | 0 |
| Final SRS | Ajay Shenoy | 4 | 3 | 1 |
| Final SRS | August Tan | 3 | 3 | 0 |
| Final SRS | Summary | 10 | 9 | 1 |

**Cumulative**

|  |  |  |  |
| --- | --- | --- | --- |
| Who | Estimated | Actual | Difference |
| Munieshwar Ramdass | 8 | 9 | 1 |
| Ajay Shenoy | 10 | 8.5 | 1.5 |
| August Tan | 7 | 8 | 1 |

**14.2 Defect Tracking**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or  Deliverable | Who | Estimated | Actual | Difference |
| Initial SRS | Munieshwar Ramdass | 16 | 10 | 6 |
| Initial SRS | Ajay Shenoy | 10 | 10 | 0 |
| Initial SRS | August Tan | 15 | 10 | 5 |
| Initial SRS | Summary | 41 | 30 | 11 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or  Deliverable | Who | Estimated | Actual | Difference |
| Final SRS | Munieshwar Ramdass | 5 | N/A | N/A |
| Final SRS | Ajay Shenoy | 2 | N/A | N/A |
| Final SRS | August Tan | 4 | N/A | N/A |
| Final SRS | Summary | 11 | N/A | N/A |

**Cumulative**

|  |  |  |  |
| --- | --- | --- | --- |
| Who | Estimated | Actual | Difference |
| Munieshwar Ramdass | 21 | N/A | N/A |
| Ajay Shenoy | 12 | N/A | N/A |
| August Tan | 19 | N/A | N/A |