Assignment 4

Software Requirements Specifications

FOR

University Department Information System

Group – 58
Kaustubh Hiware – 14CS30011
Surya Midatala – 14CS30017

Table of contents :-

1. Introduction

1.1 Purpose

This Software Requirements Specifications document is intended to describe the software requirements of the University Department Information System (UDIS). It explains the features and constraints of the system.

1.2 Document Convention

🧍	Actor
➡️	Connection
⬭	Activity
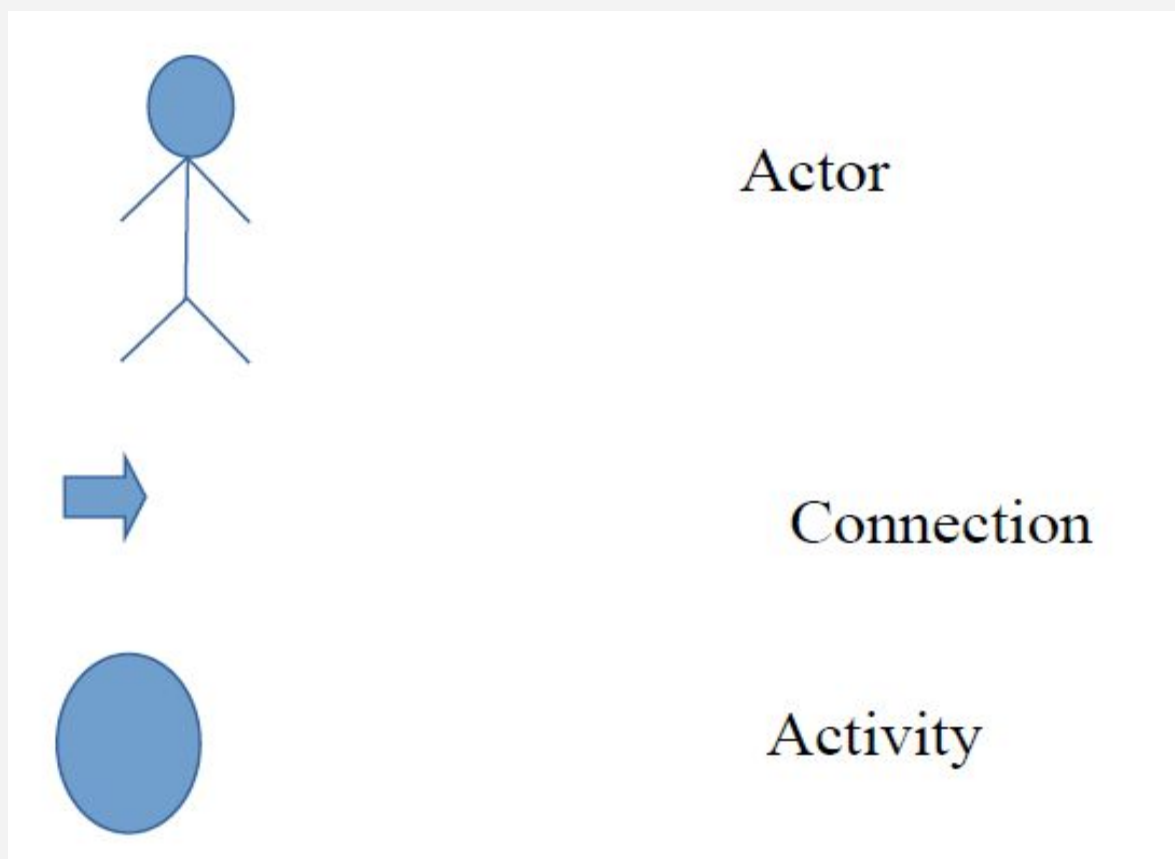
1.3 Scope

- The UDIS software is meant to manage the information database of a university department.
- It allows the user to store, modify and display the student and course lists, inventory, research information, publication details, and cashbook of a department.
- This software is intended to be used by the department secretary alone.

1.4 References

- IEEE standard for the SRS template

  830-1984 — IEEE Guide to Software Requirements Specifications. 1984.
- Class slides

1.5 Overview

- Part 2 of this SRS, the General Description section, gives a brief overview of the functionality of the product.
- Part 3, the Requirements Specification section, is written primarily for the developers and describes in technical terms the details of the functionality of the product.

1.6 Glossary

Term	Definition
UDIS	Abbreviation for University Department Information System
Course	A series of lectures in a particular subject leading to examination and/or qualification.
Student	A person who pursues one or more courses in the department.
Grade	A particular level of quality, level or value a student receives for a particular course.
Inventory	A list of property or goods.
Item	An individual article or unit , which is a member of the inventory.
Transaction	An instance of spending/receiving money , extending to details of research work ongoing in the department.
Database	A structured set of data which may be accessible in ways more than one.

2. General Description

2.1 System Environment

The system is accessed by one user i.e. the department secretary through a login page. Any changes or queries to the databases are done through the secretary. The UDIS consists of storing, viewing information and modifying them when requested.

2.2 Functional Requirements Specifications

This section briefly outlines the use cases of the department secretary.

Secretary Use Cases

The secretary has the following set of use cases:
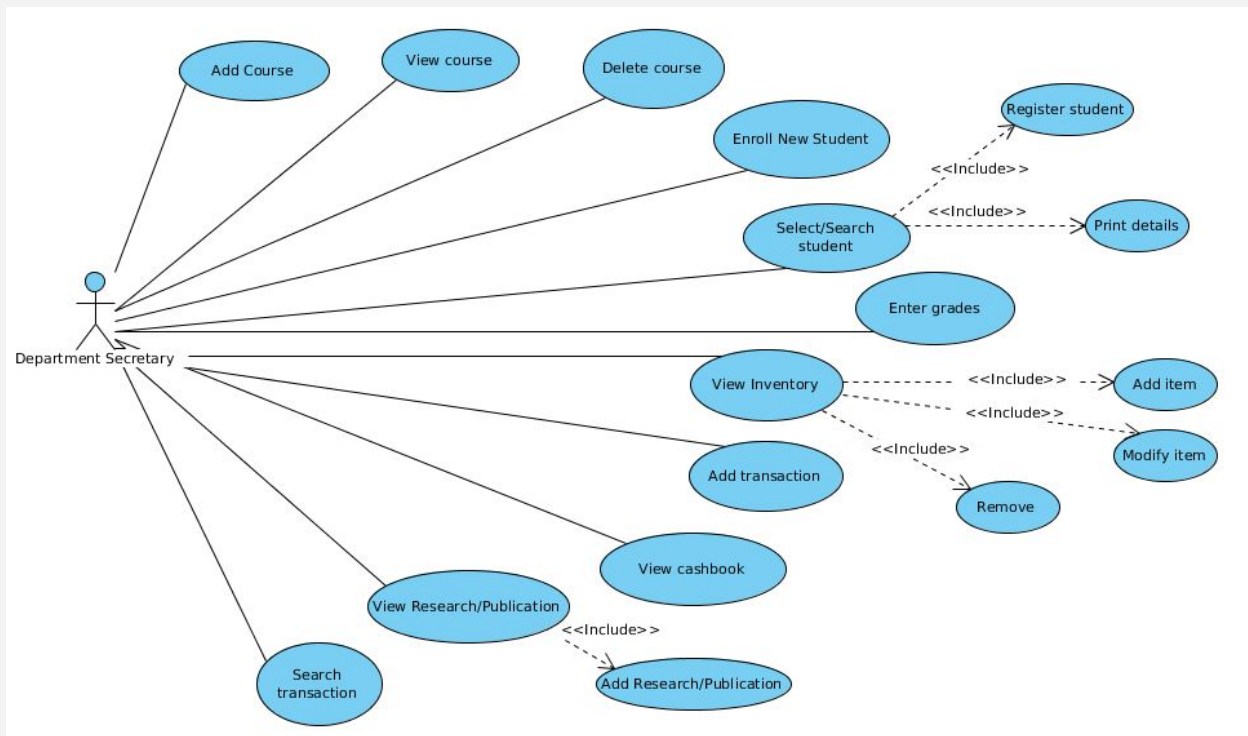
2.2.1 Use Case: Modify courses

Brief Description:

The secretary adds a new course, deletes or modifies an existing course in the department.

2.2.2 Use Case: Manage student

Brief Description:

The secretary enrolls a new student to the department and registers the student for courses. The secretary can also search the students by roll number.

2.2.3 Use Case: Enter grades

Brief Description:
The secretary enter the grades of all the students in a particular course.

2.2.4 Use Case: Manage Inventory

Brief Description:
The secretary manages an inventory by adding, removing and viewing inventory items.

2.2.5 Use Case: Manage Treasury

Brief Description:
The secretary adds transactions such as funds, grants and other expenditure and also views the cash book with separate logs for the funds/grants and expenditures.

2.2.6 Use Case: Manage Research/Publication

Brief Description:
The secretary manages the research and publications of the department by adding and viewing them when necessary.

2.2.7 Use Case: Search Transactions

Brief Description:
The secretary sends queries for transactions in the cash book. The results are displayed in a table in the same window.

2.3 User Characteristics

The user is expected to be informed about the system and have the knowledge to use a search engine, drop down lists, forms and similar tools. For the user interface specifications , please refer to the file "Interface Diagram " for the poor man's prototypes of the software.

2.4 Non-Functional Requirements

- The software has been developed keeping Windows as the primary OS to serve in mind , as it is the most popular choice. It may also be used in Linux . However , if that method fails , the user may have to run the project through a Java IDE like Eclipse or Netbeans.
- The language chosen for development is Java , mainly because of two reasons :
  - ❏ It is one of the most popular language for UI development.
  - ❏ Since javascript will be used if online storage is made available , there will not be major code reforms due to the similarities in java and javascript.

- This particular software will be developed using Netbeans 8.1

2.5 General Constraints

- The current software stores database in the form of a file on the local system on which it is run . A future implementation would be to facilitate the storage of online database using MySQL/SQLite for this process.

- The current system supports storage/logs for only one department . A platform where multiple department secretaries are able to save their individual logs would be much handy.

- The current software allows access to only one user . It would be better if students were able to access this software in order to enroll themselves , view their grades , etc similar to the ERP system of IIT Kharagpur.

- The login screen demands password and provides no way to reset the password . If online storage is used, a new password could be generated by sending a mail to the stored e-mail of the secretary . Also protection could be extended to not only password , but finger recognition or face recognition , but this is beyond the scope of the current usage.

2.6 Assumptions and Dependencies

- It is assumed that only the secretary can access this software and does not delete the record generated by the software.It is also assumed that the secretary remembers the password for login purpose.

- The software is designed to run on Windows platform directly by clicking on the jar file . For using in Linux , it will first have to be marked as executable using the command : chmod u+x UDIS.jar after changing the directory in the terminal.

- Java JRE(Java Runtime Environment) version 7 or later (preferably 8) must be set up in the system in which the software is wished to be run.

3. Requirements Specification

3.1 External Interface Requirements
None

3.2 Functional Requirements
This section explains in detailed steps, the use cases covered in part 2.2

Secretary Use Cases

3.2.1 Use Case: Modify Course

1. Secretary logs into the system.
2. Main window opens.
3. **Trigger :** Secretary selects the Academics tab
4. Secretary selects among
   a. Adding new course
   b. Deleting existing course
   c. Viewing existing courses
5. **Basic Path :** The respective window opens
6. **Postcondition :** Courselist database is updated.

3.2.2 Use Case: Manage Student

1. Secretary logs into the system.
2. Main window opens.
3. Secretary selects the Academics tab
4. **Trigger :**Secretary selects among
   a. Adding new student
      i. This option also brings up the register students window
   b. Registering student for courses
   c. Viewing existing students
   d. Search for student by roll number
5. The respective window opens **.**
6. **Postcondition :** Studentlist database is updated.

3.2.3 Use Case: Enter Grades

1. Secretary logs into the system
2. Main window opens
3. Secretary selects the Academics tab
4. **Trigger :**Secretary selects the enter grades button
5. Window opens showing list of all courses
6. Selecting a course shows a table with fields for entering grades
7. Secretary enters the grades of the students
8. **Exception Path :** Secretary does not grade all students
9. **Postcondition :** If all the students are graded, each student's data is updated

3.2.4 Use Case: Manage Inventory

1. Secretary logs into the system
2. Main window opens
3. Secretary selects the Inventory tab
4. The tab displays the current inventory items
5. **Trigger :**Secretary selects among
     a. Add item
     b. Delete item
     c. Modify item
6. The respective window opens.
7. **Postcondition :** Itemlist database is updated.

3.2.5 Use Case: Manage Treasury

1. Secretary logs into the system
2. Main window opens
3. Secretary selects the Treasury tab
4. **Trigger :**Secretary selects among
     a. Add transaction such as grant/fund/expenditure
     b. View cash book
5. The respective window opens.
6. **Postcondition :** CashBook database is updated.

3.2.6 Use Case: Manage Research/Publication

1. Secretary logs into the system
2. Main window opens
3. Secretary selects the Research/Publication tab
4. The window shows the current research and publications
5. **Trigger :**Secretary can select among
     a. Add research
     b. Add publication
     c. View research and publication
6. The options are selected by using the appropriate radio buttons
7. The corresponding window opens
8. **Postcondition :** CashBook database is updated.

3.2.7 Use Case: Search Transactions

1. Secretary logs into the system
2. Main window opens
3. **Trigger:** Secretary selects the Search tab
4. The window shows the current publications
5. Secretary enters query into the search bar
6. The corresponding results are displayed in the table

3.3 Non Functional Requirements

The data descriptions of each of these data entities is as follows:

Course data entity

Data item	Type	Description	Comment
Name	String	Name of the course	
Credit	String	Credits it accounts for	
Grade	Int	The evaluation result	Varies for each student
Status	String	Whether subject is cleared or not	If a student is enrolled in a course , it may be cleared/ current/ backlog
StudentsOnRoll	Int	Number of students enrolled for that subject	

| Professor | String | Name of professor in charge for that course | |

Student data entity

Data item	Type	Description	Comment
Name	String	Name of the student	
Address	String	Address for the student	
Phone	String	Contact number for the student	
Mail	String	Email id for the student	
Subjects	Arraylist of courses	A collection of courses present/past the student has enrolled in	The status of course determines ongoing courses
Roll	Int	Roll number	Generated by the system at time of enrollment

sgpa	ArrayList of float	Array of SGPA of the student across various semesters	
cgpa	ArrayList of float	Array of CGPA of the student across various semesters	
PrevCredits	Int	Total of previous subjects ' credits which the student had enrolled in	Used for computation of CGPA at the end of every semester

Item data entity

Data item	Type	Description	Comment
Name	String	Name of the item	
Location	String	Location or details for the item	
Price	double	Price of the item	Counted as a transaction

Transaction data entity

Data item	Type	Description	Comment

Title	String	Title of the transaction	
Authority	String	Authority responsible for the transaction	
Details	String	Any details needed to be saved	
Investment	double	The negative amount involved in a transaction	
Profit	double	The positive amount involved in a transaction	
Type	String	Type of string	Necessary to differentiate cost , income , research , etc

NOTE : A transaction must contain both investment and profit , as some research may include both of them