Software Requirements Specification

for

University Department Information System

(UDIS)

Version 1.0 approved

Prepared by

Nikhil Tudaha (19CS10045)

Seemant G. Achari (19CS10055)

Rajas Bhatt (19CS30037)

Submitted On

20 March 2021

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Nikhil Tudaha	20 March 2021	Draft Formalization	1.0.0

1. Introduction

1.1 Purpose

It is proposed to develop a software to automate various bookkeeping activities associated with the day-to-day operations of the department. The manual system has lot of disadvantages and time consuming. Hence, there is a need to develop a systemized platform that caters to all the needs of the department. These include adding new students, registering courses, entering grades, maintaining track of department activities like department inventory, academic information, and keeping a track of department expenditures.
In addition to these, it is proposed that proper methods for addition of new courses, department publications and projects be included.

1.2 Intended Audience

This document is intended for various administrative officials of the department and future developers and testers. The Department Secretary is advised to have a good look at this document so that they have utmost comfort while using the system.

1.3 Product Scope

This software has any advantages over any manual system
1. Acts as an interface for administration to easily access and update department records.
2. Time efficient and removes data redundancy.
3. No maintenance problem.
4. Easy backup at a local database, in which data is securely stored.
5. Highly secure. The database is not changeable without adequate authorization.
6. Easy updating of academic information of the students.
7. Easy monitoring of the department account, inventory and department academics.

1.4 References

The following books and sites can be approached during the development phase

- *https://docs.python.org/3/library/tk.html (Tkinter library official document)*
- *https://sqlite.org/docs.html*
- *https://www.youtube.com/watch?v=Wb0DM9I8RDo (SQL Lecture in CS50)*
- *https://docs.python.org/3/library/tkinter.ttk.html (Themed Widgets for Tkinter)*
- *https://www.geeksforgeeks.org/python-gui-tkinter/ (Tkinter guide in GeeksForGeeks)*
- *https://stackoverflow.com*

2. Overall Description

2.1 Product Perspective

This is a new, self-contained product, which will be used by the department secretary to increase efficiency in accessing and managing department and student records. This product is not the extension or a subsystem of any other product.

2.2 Product Functions

This product provides the following functions:

- *Student*
  - *View Student*
  - *Register Courses for Student*
  - *Add a new Student*
  - *Enter Grades of the student*
  - *Save Details on local machine*

- *Department*
  - *Academics*
    - *View and Add new Courses*
    - *View and Add new Publications*
    - *View and Add new Projects*
  - *Inventory*
    - *Search for items by name and type*
    - *Purchase items*
  - *Account*
    - *View Passbook of Bank Account*
    - *Add money to Bank Account*

2.3 User Classes and Characteristics

There is only one end user and administrator at present, i.e. the department secretary. In later additions, new users like students (who can view their information) and professors (who can add grades directly) may be added.
A non-human user may be a printer, which can be used to print the locally saved document.

2.4 Operating Environment

This product will operate on Microsoft Windows 10 and Linux. It has the following dependencies:

- *Python (version 3.6 and greater)*
- *Tkinter*
- *sqlite3 (inbuilt in Python 3.6 or greater)*
- *fpdf (for saving the details as a PDF)*

2.5 Design and Implementation Constraints

Assuming there would be around 3000-3500, students in a department, the SQL querying should be fast and efficient. The department secretary is the only one who can use the system, and hence appropriate credentials for initial login are necessary.

2.6 Assumptions and Dependencies

- It is assured that the details entered are apt and correct in the first go itself. No editing option has been provided.
- The credentials of the admin, once entered into the database (while building the application) cannot be changed.

3. External Interface Requirements

3.1 User Interfaces

- *Login Screen*
  - *Enter username and password*
  - *Sign in*
- *Home Screen*
  - *Choose student or department menu*
  - *Sign out*
- *Student Main Screen*
  - *View Student*
    - *Search by roll no. and name*
    - *Save details as PDF*
  - *Register Courses for Student*
    - *Search by roll no. and name*
    - *Select the courses and register*
  - *Add a new Student*
    - *Enter roll no. (unique)*
    - *Enter name*
    - *Enter address*
    - *Enter year of joining*
    - *Select degree/program from the given options*
  - *Enter Grades of the student*
    - *Search by roll no. and name*
    - *Select appropriate grade for each ongoing course*
- *Department Main Screen*
  - *Academics*
    - *View and Add new Courses*
      - *Search for course by name*
      - *Add new course*
        i. *Enter course code (unique)*
        ii. *Enter course name*
        iii. *Enter teacher name*

- *View and Add new Publications*
  - *Add publications*
    i. *Enter professor name*
    ii. *Enter date*
    iii. *Enter name of the publication*
- *View and Add new Projects*
  - *Add new project*
    i. Enter professor name
    ii. Enter duration
    iii. Enter allotted grant from department account
  - *Modify project status*

- *Inventory*
  - *Search for items by name and type*
    - *Enter name*
    - *Choose type*
  - *Purchase items*
    - *Enter name*
    - *Choose item type*
    - *Enter rate*
    - *Enter quantity*
    - *Enter location*

- *Account*
  - *View Passbook of Bank Account*
  - *Add money to Bank Account*
    - *Enter date*
    - *Enter organization*
    - *Enter amount*
    - *Choose purpose*

3.2 Hardware Interfaces

There is only one possible software-hardware interaction and that is printing of the generated student PDF.

3.3 Software Interfaces

- *A modern operating system like Windows 10 or any recent Linux distributions.*
- *A database file named UDIS.db which stores the backend and all the valuable information. It contains the various tables needed by the application, such as:*
  - *all_students (stores information of all students)*
  - *all_courses (stores information of all available courses)*
  - *courses_taken (stores information of all students taking respective courses)*
  - *publication_details (stores information of all publications)*
  - *project_details (stores information of all projects)*
  - *inventory_details (stores information of all inventory items)*
  - *account_details (stores information of all account transactions)*

4. Functional Requirements

4.1 Administrative Login and Use

- *Can enter various details of the student such as his name, address, course registered, etc., when he takes admission.*
- *Can view various details of each student including performance, courses taken, courses completed and backlogged courses.*
- *At the beginning of each semester, can register students for courses.*
- *Should be able to enter grades for each student at the end of the semester, and update the performance details, including the SGPA and the CGPA of the student.*
- *Should be able to keep track of inventories of the department, such as equipment, furniture, etc. and their location.*
- *Should be able to enter funds into the department account received as grants from various sources and key in the details of the sources.*
- *Should be able to buy various inventory items by withdrawing adequate money from the bank account.*
- *Should be able to keep track of various research projects running in the department, add new ones and update their status.*
- *Should be able to query details of the bank account and view account passbook.*

5. Non Functional Requirements

5.1 Performance Requirements

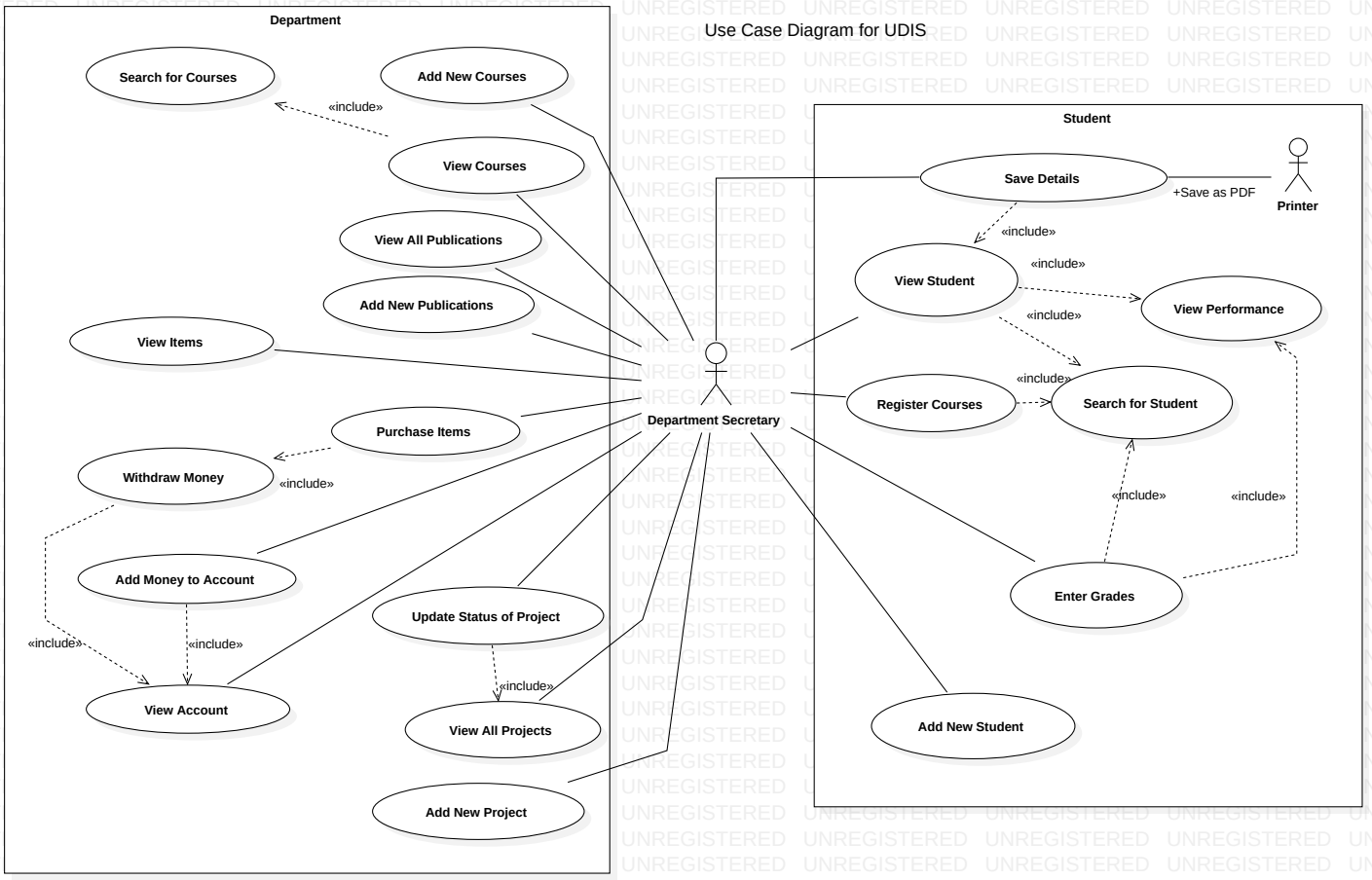
- *Database management system should be efficient and cost effective.*
- *It should support quick searching for students, courses across big databases in a short span of time.*

5.2 Software Quality Attributes

- *Should be user friendly*
- *Should be fast, secure and operating system independent*

Model::Use Case Diagram1

Use Case Diagram for UDIS



Model.main

Class Diagram for UDIS