**University Department Information System**

Software Design Document

Name : Kola Deepti (120CS0151)

Lab Section: S1

Workstation: Kalam Lab LA-1

Date: (01/31/2023)

**TABLE OF CONTENTS**

1. [INTRODUCTION 2](#_bookmark0)
   1. [Purpose 2](#_bookmark1)
   2. [Scope 2](#_bookmark2)
   3. [Overview 2](#_bookmark3)
   4. [Reference Material 2](#_bookmark4)
   5. [Definitions and Acronyms 2](#_bookmark5)
2. [SYSTEM OVERVIEW 2](#_bookmark6)
3. [SYSTEM ARCHITECTURE 2](#_bookmark7)
   1. [Architectural Design 2](#_bookmark8)
   2. [Decomposition Description 3](#_bookmark9)
   3. [Design Rationale 3](#_bookmark10)
4. [DATA DESIGN 3](#_bookmark11)
   1. [Data Description 3](#_bookmark12)
   2. [Data Dictionary 3](#_bookmark13)
5. [COMPONENT DESIGN 3](#_bookmark14)
6. [HUMAN INTERFACE DESIGN 4](#_bookmark15)
   1. [Overview of User Interface 4](#_bookmark16)
   2. [Screen Images 4](#_bookmark17)
   3. [Screen Objects and Actions 4](#_bookmark18)
7. [REQUIREMENTS MATRIX 4](#_bookmark19)
8. [APPENDICES 4](#_bookmark20)

### INTRODUCTION

## Purpose

The University Department Information System software’s purpose is to eliminate the lot of book keeping activities to record details of various students, the new registrations , their grades(semester-wise) and also the software should be able to keep track of the inventories of the department. It should also allow updation of data.

This software should also take care of the Department information system keeps track of the Department accounts and research projects.

## Scope

This software system will be a University Information system for a the members of the department of an organization.

This system will be designed to maximize the administrative, academic and overall productivity of each and every department by providing tools to assist in automating the technical procedures and processes, which would otherwise have to be performed manually.

By maximizing the users work eﬃciency and production the system will meet the users needs while remaining easy to understand and use. It is a user-friendly portal to interact, manage, access and edit the information.

## Overview

The product will be a standalone application and may be run on multiple systems within an

Internet network. The product will require a keyboard, mouse and monitor to interface with

the users. The minimum hardware requirements for the product are specified in this document.

## Reference Material

*https://www.studocu.com/row/document/nahda-university/computer-science/srs-for-university-management-system/27752841*

### SYSTEM OVERVIEW

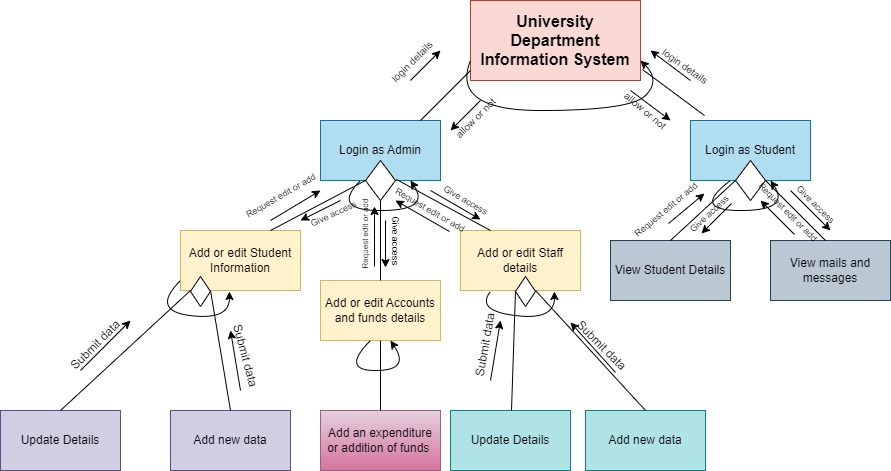
Give a general description of the functionality, context and design of your project. Provide any background information if necessary.

### SYSTEM ARCHITECTURE

## Architectural Design

This is a modular program structure which explains the relationships between the modules to achieve the complete functionality of the system. This is a high level overview of how

Responsibilities of the system were partitioned and then assigned to subsystems. We identified each high level subsystem and the roles or responsibilities assigned to it and how these subsystems collaborate with each other in order to achieve the desired functionality. The main purpose is to gain a general understanding of how and why the system was decomposed, and how the individual parts work together.



Structure Chart of UDIS

## Decomposition Description

The software is designed such as every module is independent of each other. Login portal will pop up at first and after logging in as admin or student then there will be different accessibility according to that. There are functions such as edit or update student/ staff info and accounts management for admin. The student has limited access such as viewing his/ her details and viewing mails and messages sent to the student.

## Design Rationale

There are many functions or processes which are iterative or will go on loop for several times, such as, if login failed for some reason such as incorrect username or password then the software will ask to reenter till it’s logged in successfully. There is also decision making for various options after going to a particular page.

### DATA DESIGN

## Data Description

We’ll use a relational database to store different set of data, such as Student database, Staff database, Accounts database, Examination database and many more. There will also be connections or joins between those databases to omit redundancy.

## Module Specifications

This UDIS is designed such as every module is independent of each other. Login portal will pop up at first and after logging in as admin or student then there will be different accessibility according to that.

* **Login as admin**: There are functions such as edit or update student/ staff info and accounts management for admin.
  + Update/ Add student info: The admin will ask permission to add/ update data and then submit it to student database.
  + Update/ Add accounts details: The admin will ask permission to add/ update data and then submit it to Accounts database.
  + Update/ Add Staff details: The admin will ask permission to add/ update data and then submit it to staff database.
* **Login as student:** The student has limited access such as viewing his/ her details and viewing mails and messages sent to the student.
  + View Student details: Student can only view their marks/details/sitting chart etc.
  + View mails/ Messages: Viewing the mails and messages which are sent by professors or admin.

There are many functions or processes which are iterative or will go on loop for several times, such as, if login failed for some reason such as incorrect username or password then the software will ask to reenter till it’s logged in successfully. There is also decision making for various options after going to a particular page.

## Requirement Tracing

The department should have proper trained and skilled coders and software designers to make and maintain the required software. The resources required for designing the software is the complete understanding of the problem to be solved and what functions are needed. This can be best known by end users, what the system should be designed to do.

### COMPONENT DESIGN

In this section, we take a closer look at what each component does in a more systematic way. If

you gave a functional description in section 3.2, provide a summary of your algorithm for each function listed in 3.2 in procedural description language (PDL) or pseudocode. If you gave an OO description, summarize each object member function for all the objects listed in 3.2 in PDL or pseudocode. Describe any local data when necessary.

### HUMAN INTERFACE DESIGN

## Overview of User Interface

Describe the functionality of the system from the user’s perspective. Explain how the user will be able to use your system to complete all the expected features and the feedback information that will be displayed for the user.

## Screen Images

Display screenshots showing the interface from the user’s perspective. These can be hand­ drawn or you can use an automated drawing tool. Just make them as accurate as possible. (Graph paper works well.)

## Screen Objects and Actions

A discussion of screen objects and actions associated with those objects.

### REQUIREMENTS MATRIX

Provide a cross­reference that traces components and data structures to the requirements in your SRS document.

Use a tabular format to show which system components satisfy each of the functional requirements from the SRS. Refer to the functional requirements by the numbers/codes that you gave them in the SRS.

### APPENDICES

*This section is optional.*

Appendices may be included, either directly or by reference, to provide supporting details that could aid in the understanding of the Software Design Document.