Software Design Specification

<Smart Degree Choice>

Project Code:

SU18-SES22-CP01

Internal Advisor:

Mr. Hussam Ali

Project Manager:

Dr Muhammad Ilyas

Project Team:

Name	Roll No:	Role
Ehtisham Bashir	BSEF18E018	Team Leader
Shahtaj Amjad	BSEF18E014	Team Member
Laiba Shahid	BSEF18E049	Team Member

Submission Date:

14-Jan-2022

_____
Project Manager's Signature

Document Information

Category	Information
Customer	University of Sargodha
Project	Smart Degree Choice
Document	Software Design Specification
Document Version	1.0
Identifier	<SU18-SES22-CP01>
Status	Final Draft
Author(s)	Ehtisham Bashir BSEF18E018 Shahtaj Amjad BSEF18E014 Laiba Shahid BSEF18E049
Approver(s)	Dr Muhammad Ilyas (Project Manager)
Issue Date	04-Jan 2022
Document Location	D:\7th Semester\CP\DesignDocument
Distribution	1. Mr. Hussam Ali (Advisor) 2. Dr Muhammad Ilyas (PM) 3. Project Office

Definition of Terms, Acronyms and Abbreviations

Term	Description
DFD	Data Flow Diagram
OOAD	Object Oriented Analysis and Design
GUI	Graphical User Interface

Table of Contents

1. Introduction

1.1 Purpose of Document

This document provides a design overview of the degree choice software. The application deals with the scenario in which allow students to do choice degree wisely. It describes the detailed design of the application. Serve as a means to clearly outline the project features. The intended audience for the design document includes the developers and testers documentation writers. It also serves the purpose of making the functionality clear to end users.

1.2 Project Overview

Degree choice systems have been deployed for many years in our country. However, in many parts of our country students cannot make choice because of several reasons. To illustrate, sometimes people may not be in their own region and due to this fact, they cannot continue their education based on their interests. Especially needed in order to get youth motivated to continue their studies. It would also help in rural areas where students don't have anyone to provide them guidance in this regard. There are also many students who intend to study, but they don't get enough guidance to choose the field of their interest. This web-based application will help the students to wisely make decisions and opt for their universities which will best suit them.

1.3 Scope

The software produced will be an online smart degree choice. This software is being developed for use by students with a simple and self-explanatory GUI [1]. This software that can be used by the students to choose the field which suits them according to their interest will help them choose the best possible university.

- The software will be used by the students to pursue a degree which will be the best according to their mindset.
- The main objective of this software is to help the students who face difficult in opting for their higher-level degrees after getting an intermediate degree.
- It will maintain the database of all the universities providing all courses.

2. Design Considerations

The consideration related to design of the project including assumptions and dependencies are provided with the possible risks and volatile areas.

2.1 Assumptions and Dependencies

- It is assumed that all diagrams presented in this document convey the intended meaning.
- No miss functionality is shown in this regard.

- It is assumed that all work units are designed very carefully.

- No ambiguity is leaved behind, all parts are clearly understood.

- All team members have appropriate expertise in their assigned responsibilities.

2.2 Risks and Volatile Areas

- New requirements and advancement in technology can cause a lot of changes during development.
- The functionality present might be extended with passage of time.
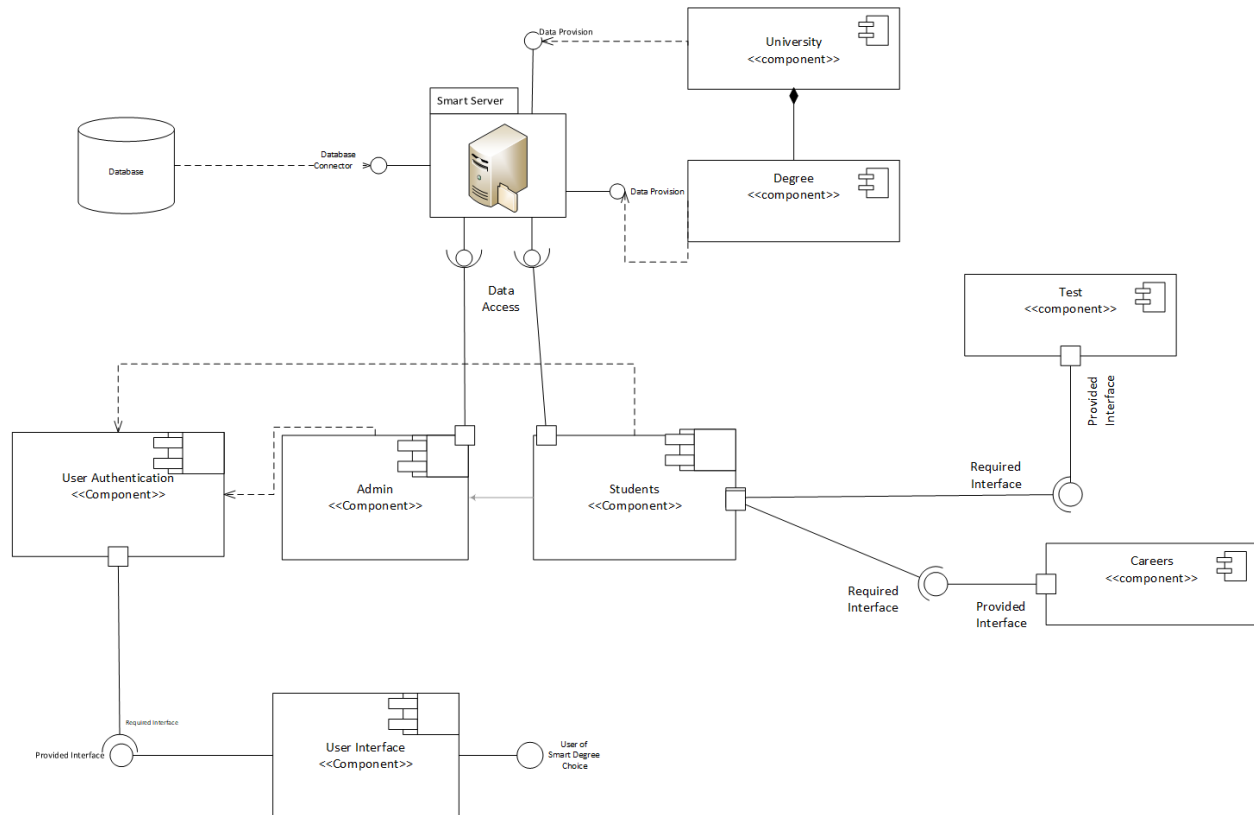- Tight deadlines can cause problems.

3. System Architecture

3.1 System Level Architecture.

3.1.1 Deployment Diagram
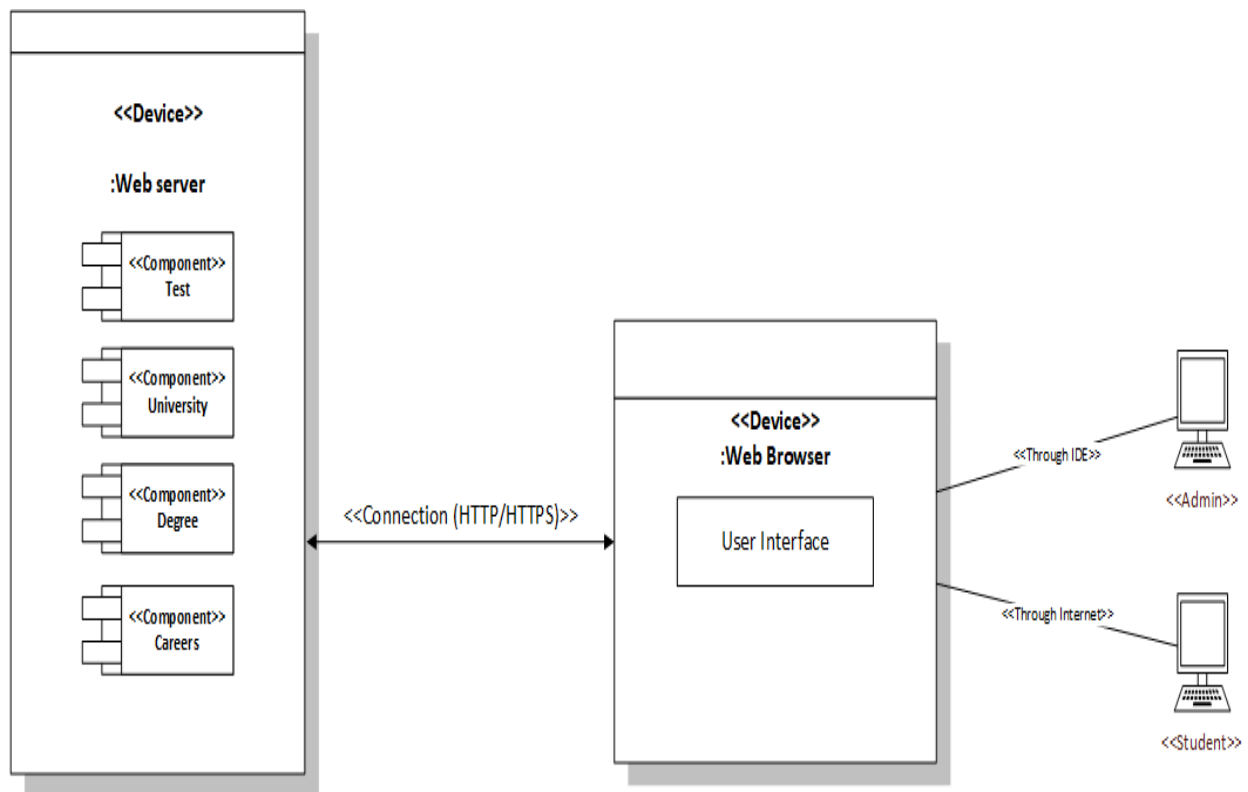
3.2 Sub-System / Component / Module Level Architecture

3.2.1   Component Diagram

4. Design Strategies

Our development process focusses on classes and their objects. That's why we will use OOAD [2] strategy with Top to Bottom approach during our design and construction.

4.1 OOAD Strategy

An object-oriented design approach is better suited to a work-based design as in the case of object-oriented, real-world businesses can easily be used in the computer world. Also, other basic behaviors **[3]** such as polymorphism, inheritance, abstraction, and encapsulation can be used in this way. Important features of OOAD are

- Class
- Object
- Inheritance
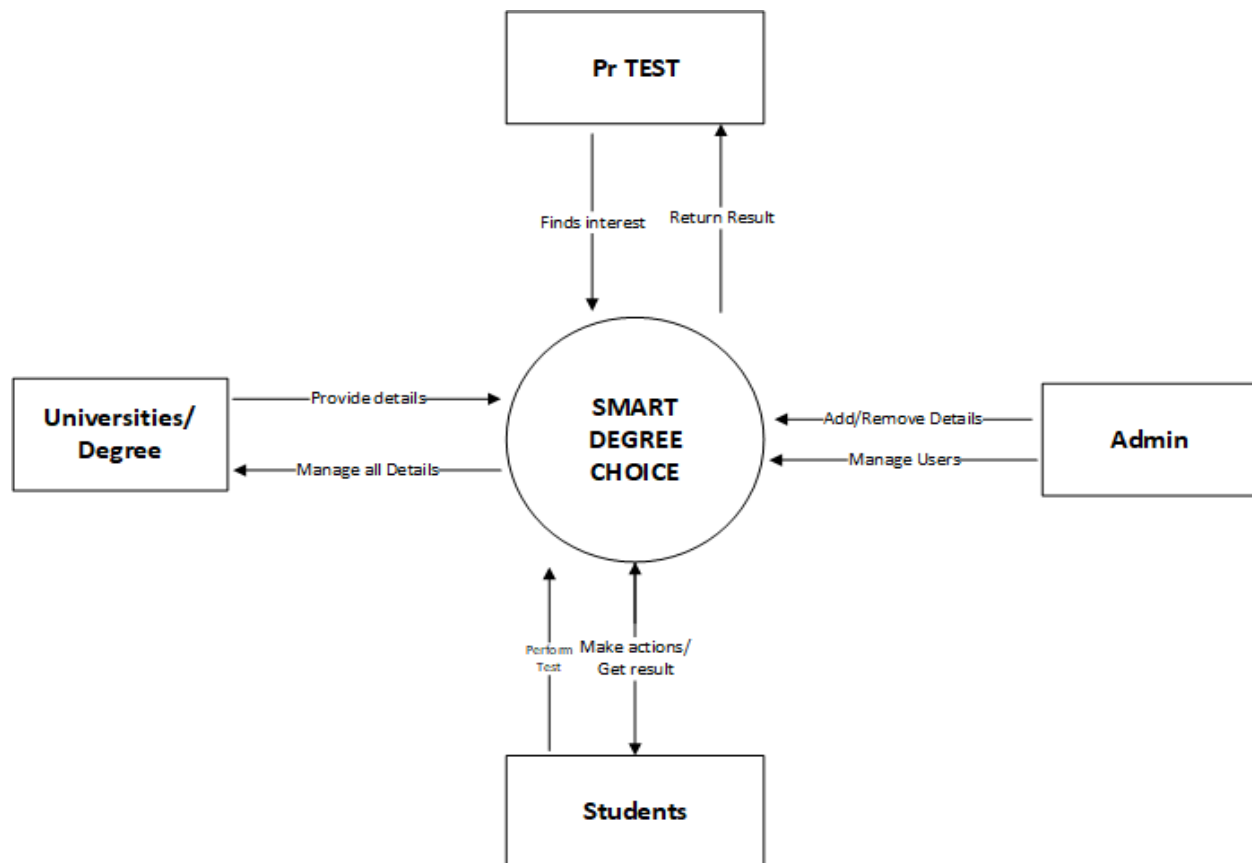- Encapsulation
- Polymorphism

4.2 Top-Down Approach

In top-down [4] approach, the class at the top of the hierarchy is design, written and integrated first. Due to following benefits we will use top-down approach.
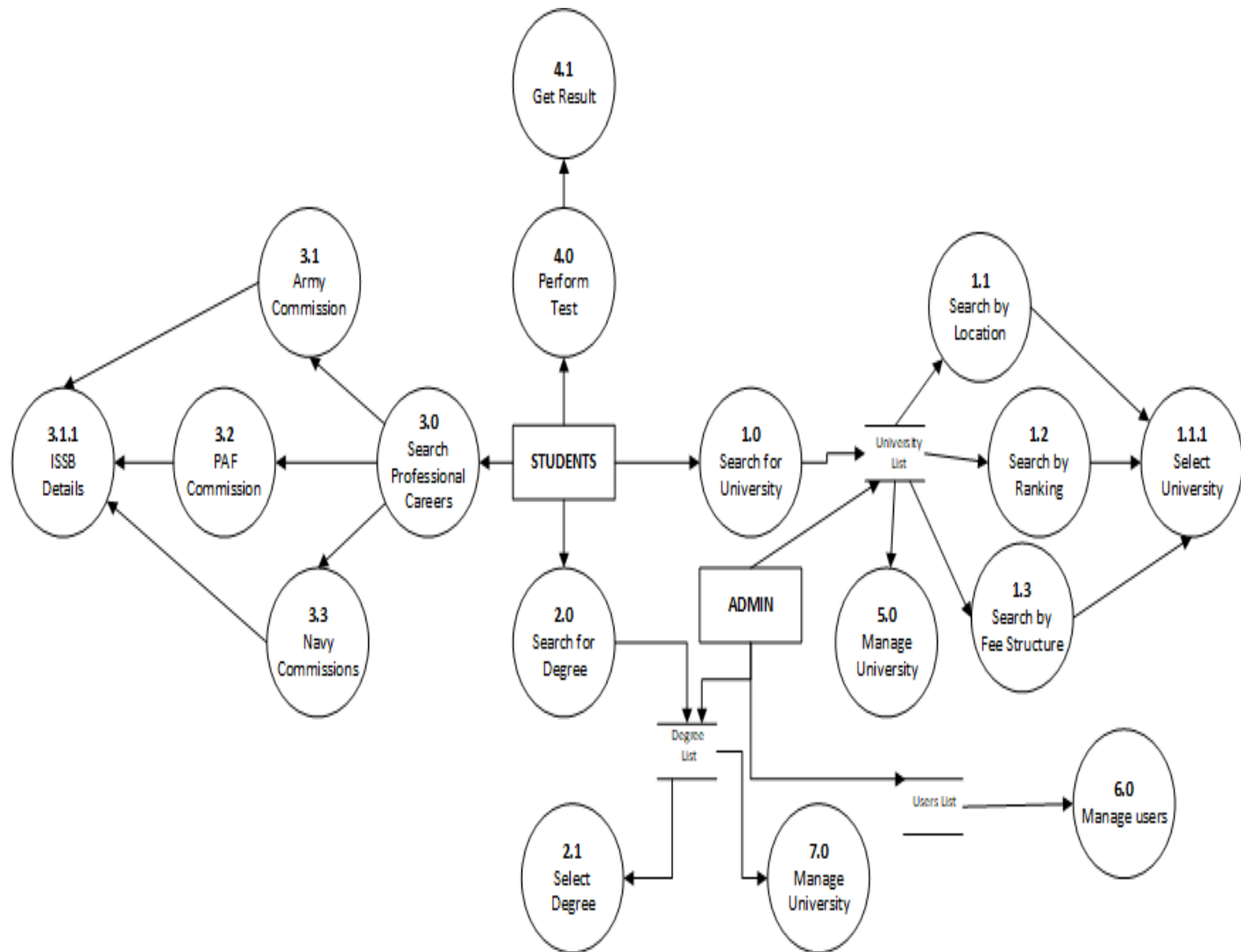
- The control logic of the system is tested relatively early.

- All the classes at the top of the hierarchy are exercised a lot so that big, conceptual, design problems are exposed quickly.

- If you plan it carefully, you can complete a partially working system early in the project.

- If the user-interface parts are at the top, you can get a basic interface working quickly and flesh out the details later.

- The morale of both users and programmers benefits from getting something visible working early.
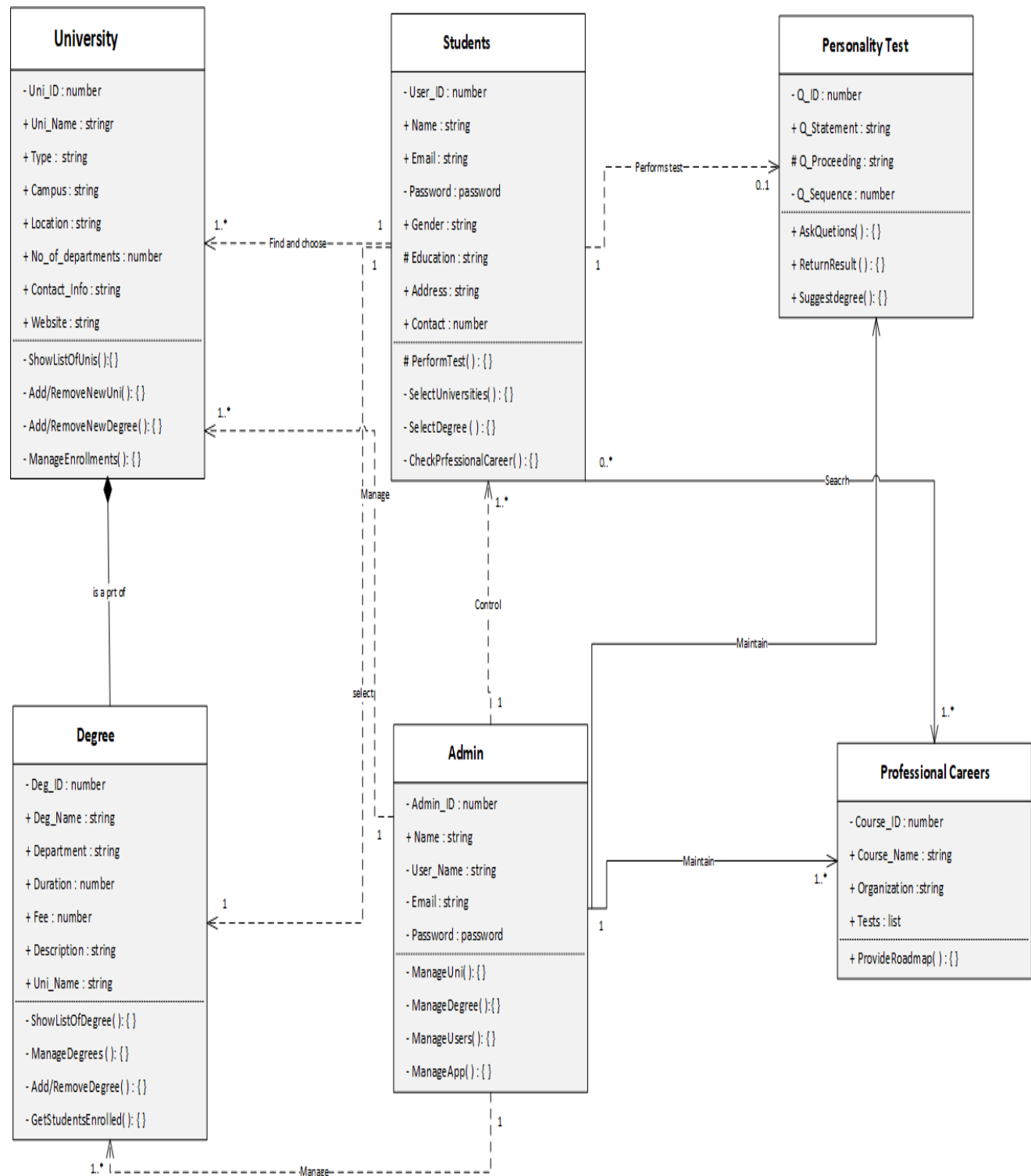
5. Detailed System Design
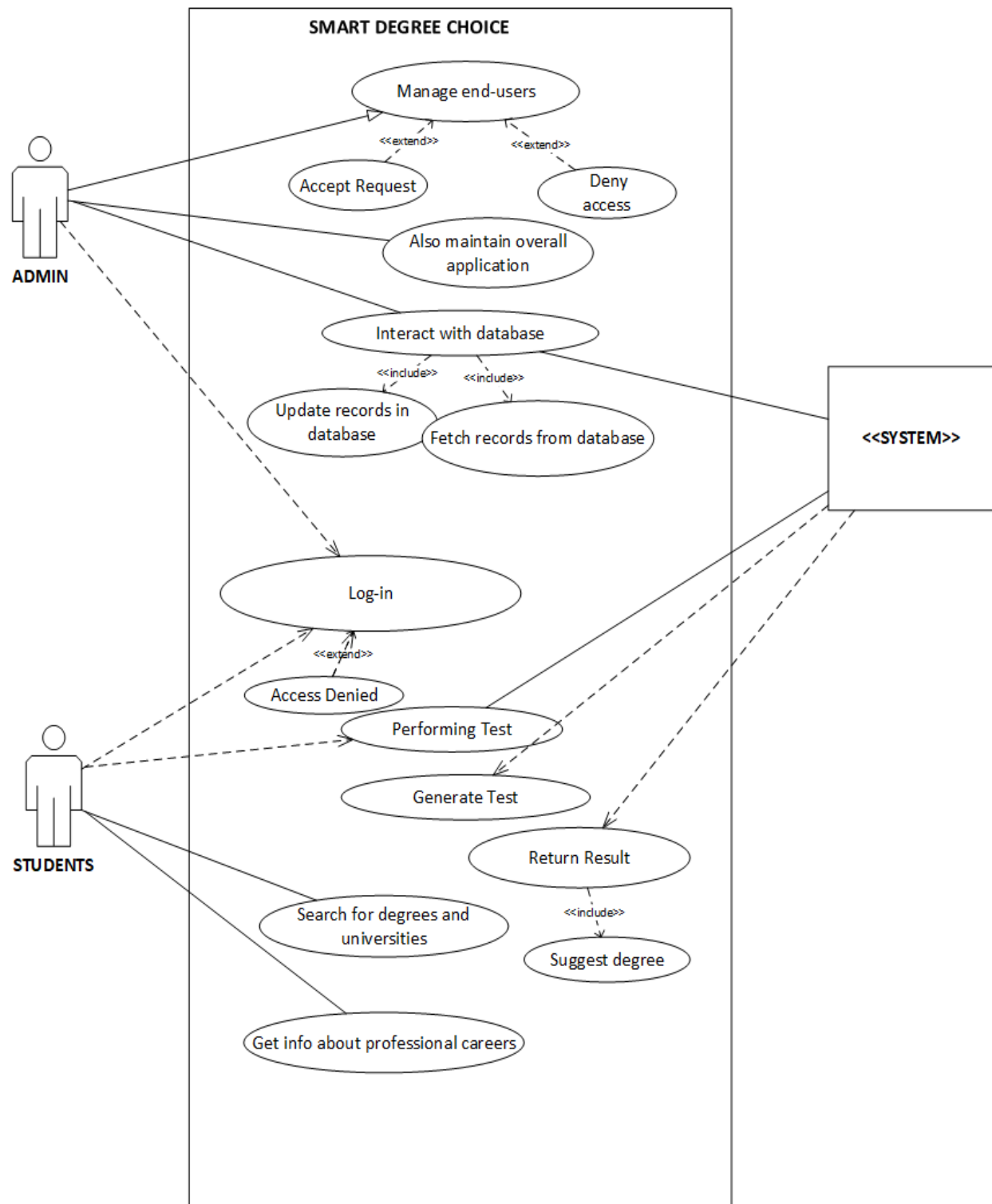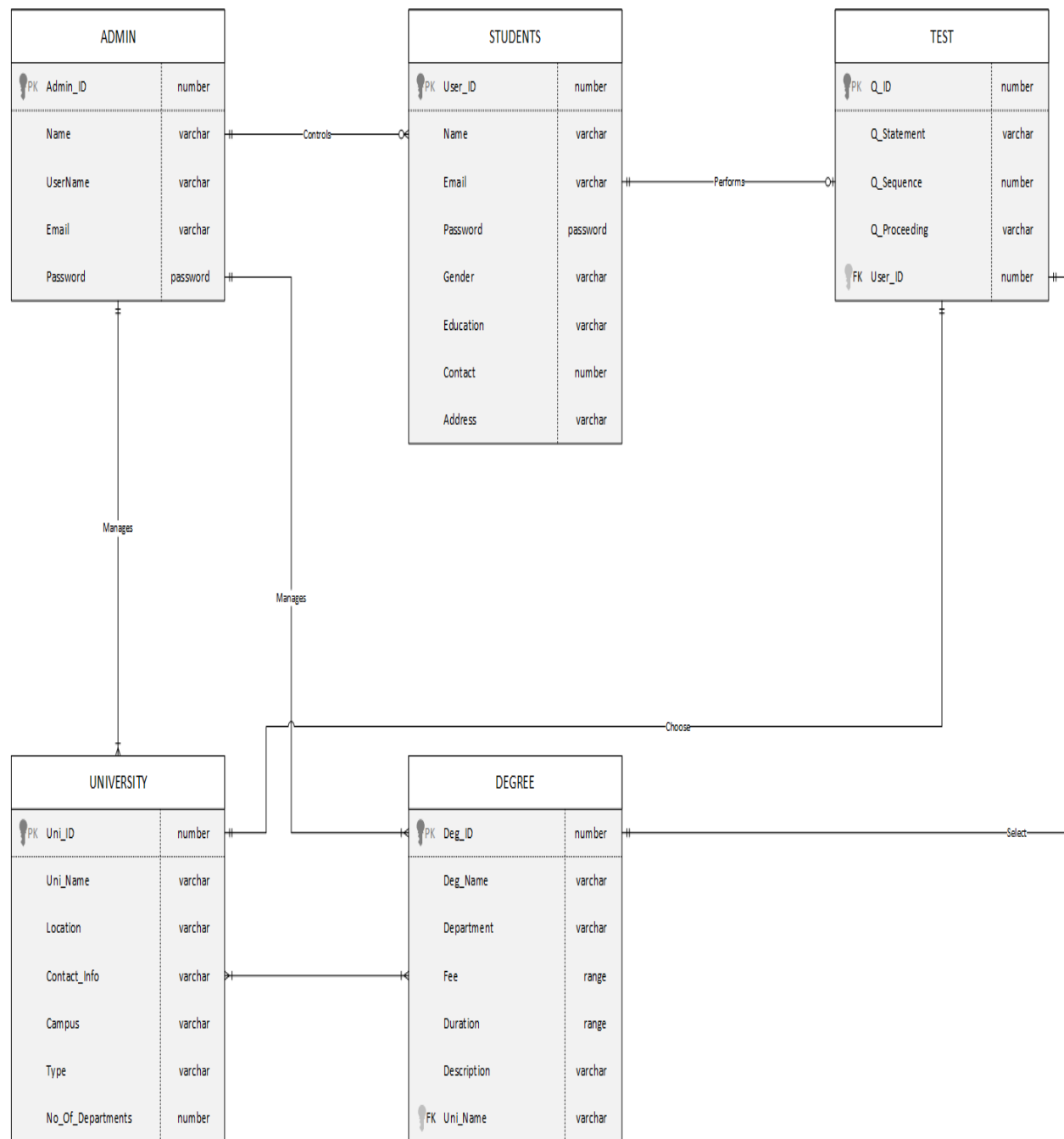
5.1 DFD (0 Level / Context Level)
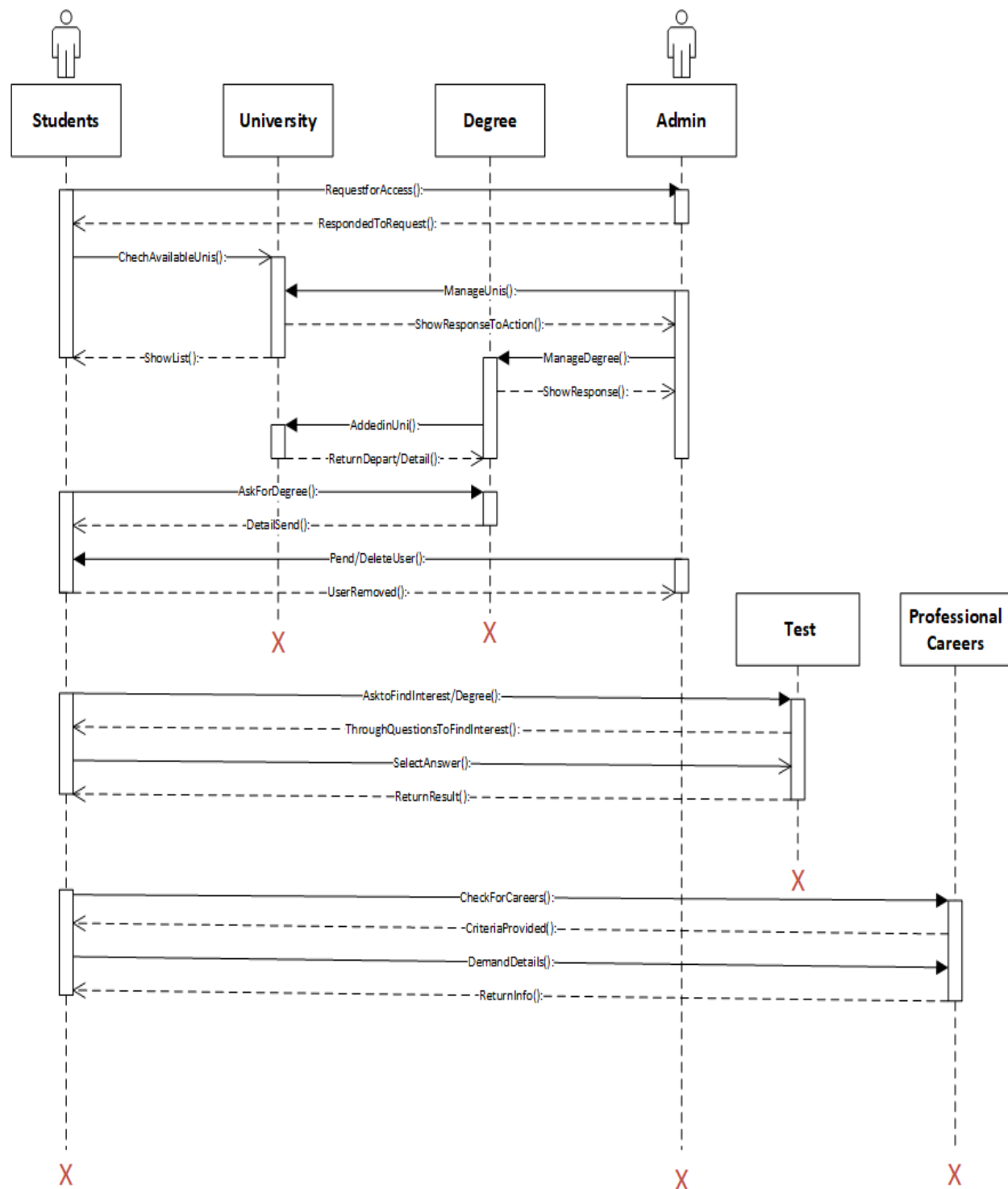
5.2 DFD (2 Level)

5.3 Class Diagram

University

- Uni_ID : number
+ Uni_Name : stringr
+ Type : string
+ Campus : string
+ Location : string
+ No_of_departments : number
+ Contact_Info : string
+ Website : string

- ShowListOfUnis( ):{ }
- Add/RemoveNewUni( ): { }
- Add/RemoveNewDegree( ): { }
- ManageEnrollments( ): { }

Students

- User_ID : number
+ Name : string
+ Email : string
- Password : password
+ Gender : string
# Education : string
+ Address : string
+ Contact : number

PerformTest( ) : { }
- SelectUniversities( ) : { }
- SelectDegree ( ) : { }
- CheckPrfessionalCareer( ) : { }

Personality Test

- Q_ID : number
+ Q_Statement : string
# Q_Proceeding : string
- Q_Sequence : number

+ AskQuetions( ) : { }
+ ReturnResult ( ) : { }
+ Suggestdegree( ) : { }

Degree

- Deg_ID : number
+ Deg_Name : string
+ Department : string
+ Duration : number
+ Fee : number
+ Description : string
+ Uni_Name : string

- ShowListOfDegree( ): { }
- ManageDegrees ( ): { }
- Add/RemoveDegree( ) : { }
- GetStudentsEnrolled( ): { }

Admin

- Admin_ID : number
+ Name : string
- User_Name : string
- Email : string
- Password : password

- ManageUni( ): { }
- ManageDegree( ):{ }
- ManageUsers( ): { }
- ManageApp( ) : { }

Professional Careers

- Course_ID : number
+ Course_Name : string
+ Organization :string
+ Tests : list

+ ProvideRoadmap( ) : { }

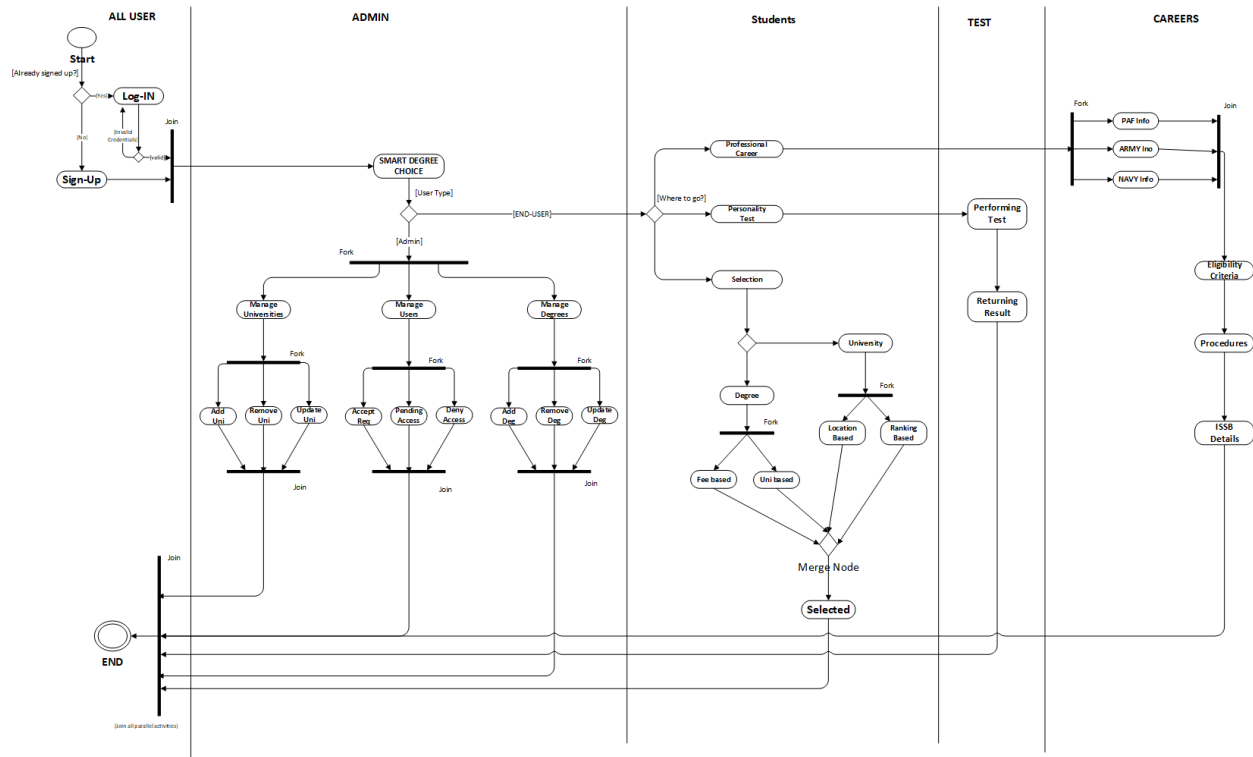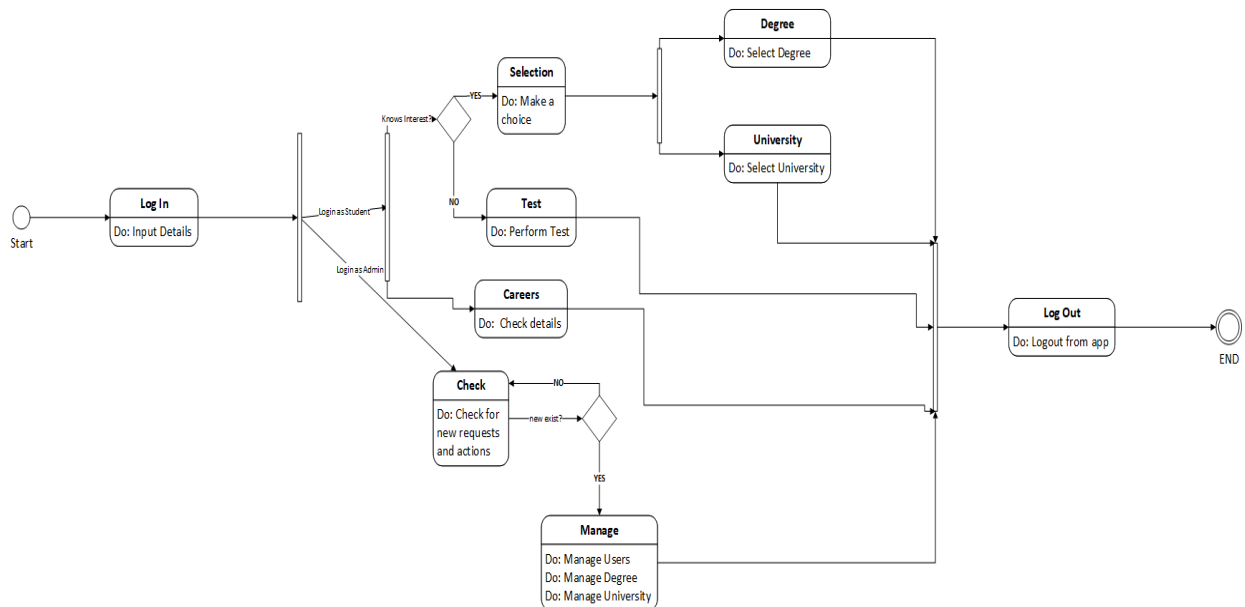## 5.4 Use Case Diagram

5.5 ER-Diagram

5.6 Sequence Diagram

5.7 Activity Diagram

5.8 State Chart/ Transition Diagram

6. References

[1] Britannica, "graphical user interface," Britannica, [Online]. Available: https://www.britannica.com/technology/graphical-user-interface. [Accessed 01 Jan 2022].

[2] Tutorialspoint, "Software Design Strategies," Tutorialspoint, [Online]. Available: https://www.tutorialspoint.com/software_engineering/software_design_strategies.htm. [Accessed 03 Jan 2022].

[3] Study, "Five Basic Concepts of Object-Oriented Design," Study, [Online]. Available: https://study.com/academy/lesson/five-basic-concepts-of-object-oriented-design.html. [Accessed 07 Jan 2022].

[4] Asana, "Top-down approach vs. bottom-up approach:," Asana, [Online]. Available: https://asana.com/resources/top-down-approach. [Accessed 10 Jan 2022].

Review:

I have read this design document several time and I'm pretty sure that there is nothing wrong or misspell. All the work units are designed very carefully. And all the unit members are highly expertise, loyal and responsible for their respective jobs. Further on I have submitted it to our project supervisor for more exploration.