# General Sir John Kotelawala Defense University

## Faculty of Computing

## Department of Computer Science

Group Project Undertaken in partial fulfillment of the requirement for the

BSc Information Technology/ Computer Science/ Computer Engineering/ Software Engineering Degree

**Intake 35**

# UNIVERSITY ELIGIBILITY CHECKING SYSTEM

## Design Report

| Group Details | | |
|---|---|---|
| **Group Number** | **Student Number** | **Student Name** |
| 12 | D/CE/18/0005 | ABTMAS Bandaranaike |
| | D/CS/18/0009 | DA Iddamalgoda |
| | D/CS/18/0015 | BLA Kalhari |
| | D/CE/18/0018 | IWMHD Bandara |

| Project Details | |
|---|---|
| **Project Title** | University Eligibility Checking System |
| **Supervisor** | Mrs. WPJ Premarathna |
| **Co-Supervisor** | |

**CONTENT**

## 1.0 Introduction

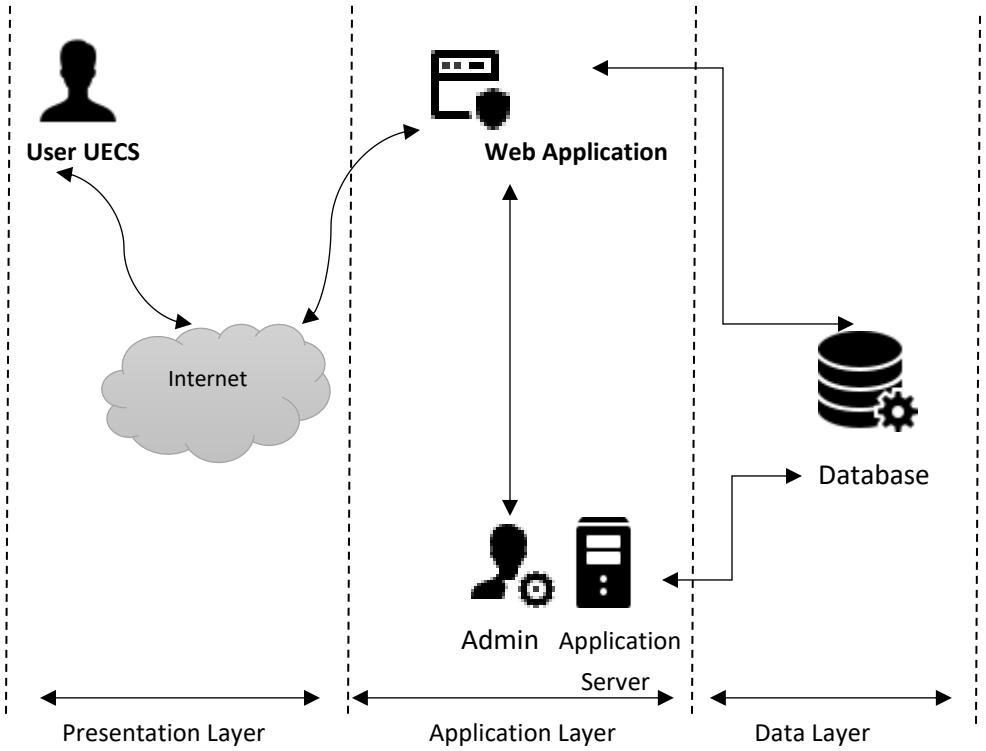
This document is the design report for the development of University Eligibility checking System with more enhancement features than the existing system. It contains details under 4 main chapters namely 'Overall System Architecture, Software Architecture, Data Design & Interface Design.

In the Overall system architecture, it is described about the presentation layer, application layer and the data link layer of the developing system with the aid of diagram. In the next section it is explained about the software architecture of the system. It contains the information about the modules which are presented in the system, roles and privileges of accessing these modules. First there is an overview of the overall software architecture which is followed by the list of modules and the detailed explanations about the key modules.

In this document 3rd section is dedicated for describing the data design of the system. It contains the conceptual data design, which is the EER model and then the conversion of EEP into the relations. This section also contains the database relationship diagrams followed by a description of tables in the data base with their attributes and their data types with lengths.

As the final section it contains the interface designs for various forms available in the system with brief description about each of them followed by a summary of this document to end the report.

## 2.0 Overall System Architecture




Architectural design defines the overall structure of the system and form a solution before moving on to the detail design or the low-level design which includes the design of specific components details. The architectural design is given according to the three-tier-architecture where overall design is spilt in to three layers of Client Tier, Application Tier and Data Tier.

### 2.1 Application Layer

The process of the UECS will be executed at this application layer. Application layer will interact with both presentation layer where the interfaces of UECS running & the data layer are where the information is stored. The information gathered from user inputs will be stored and subjected into filtering scenario and gives the output as requested by the users.

### 2.2 Data Layer

Data layer manage the data storage operations of the overall system where the database management applications are running. A database containing the details about degree programs with respect to each university and Z-scores will be stored in several tables in the UECS in order to improve the efficiency. Information gathered from the interfaces of the application layer will be stored in respective tables.

### 2.3 Presentation Layer

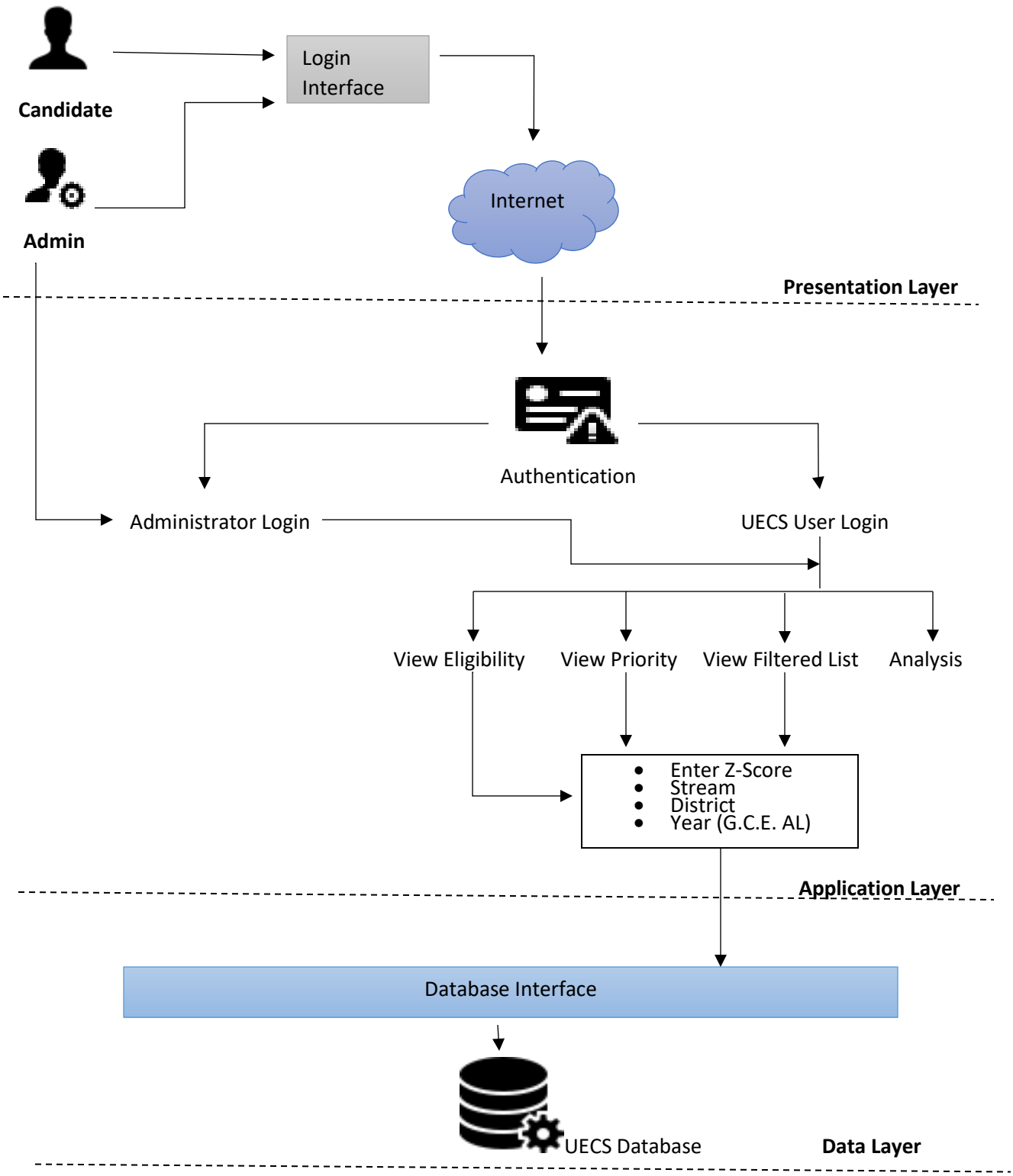
The presentation layer is responsible for the registering process of users in the UECS. In order to access the details of the system user must be first register as a member of the system. Information gathered at this layer will be provided into the application layer to manipulate according to the given instructions. Other interfaces are designed to get the inputs as required to provide the most accurate results for the user.

## 3.0 Software Architecture

Software architecture was based on modularized approach where the software is divided into parts. Each module is assigned to execute one or more tasks of the overall system in order to achieve the ultimate objectives expected. Since the program is developed using open resource platform such as PHP, HTML, CSS, will ease the development and maintenance of the system.

### 3.1 Overall Software Architecture

Following figure represents the overall software architecture of the developing system.



Main approaches of the system are then break into core modules to perform required tasks of the system. Further details of these modules and their tasks will be discussed in detailed in below sections.

### 3.2 Module Architecture

The developing UECS Web Application contains several modules to make the complete system. This section will describe about the organization of the modules that it consists.

#### 3.2.1 An Overview of Module Architecture

In the below shown is an enumerated list of requirements for the new system development.

Module 1: “Authentication/Login Module”

- 1.1: Administrator Login
- 1.2: User Login

Module 2: “View Priority”

- 2.1: Enter the Z-Score
- 2.2: Enter the stream
- 2.3: Enter the district
- 2.4: Enter the year

Module 3: “View Filtered List”

- 3.1: Enter the Z-Score
- 3.2: Enter the stream
- 3.3: Enter the district
- 3.4: Enter the year

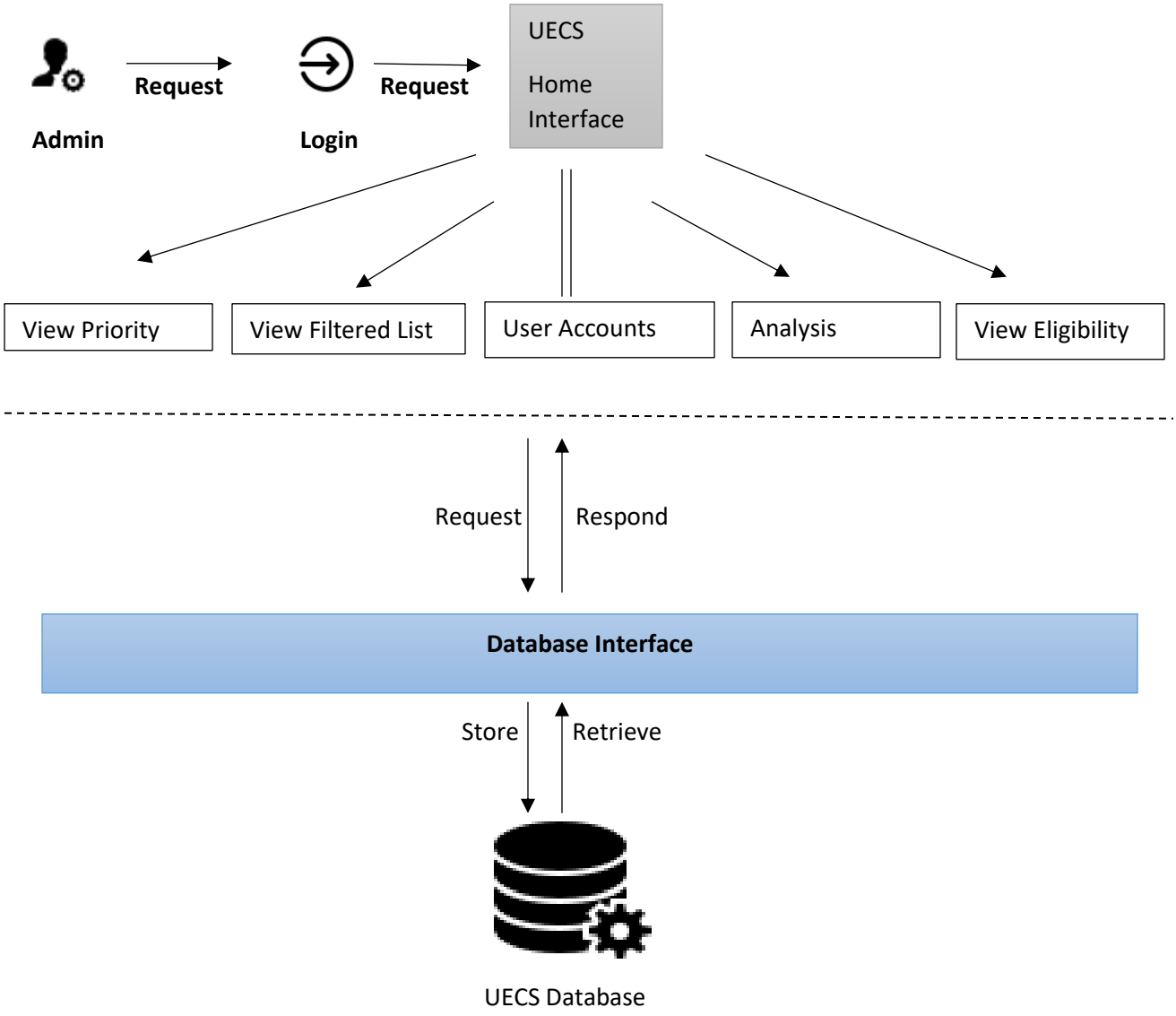
Module 4: “Analysis”

#### 3.2.2 Module Architecture in Detail

There are several modules for various functions in the developing UECS. In the below it is described in detail about those modules.

**Module1: "Authentication/Login Module"**

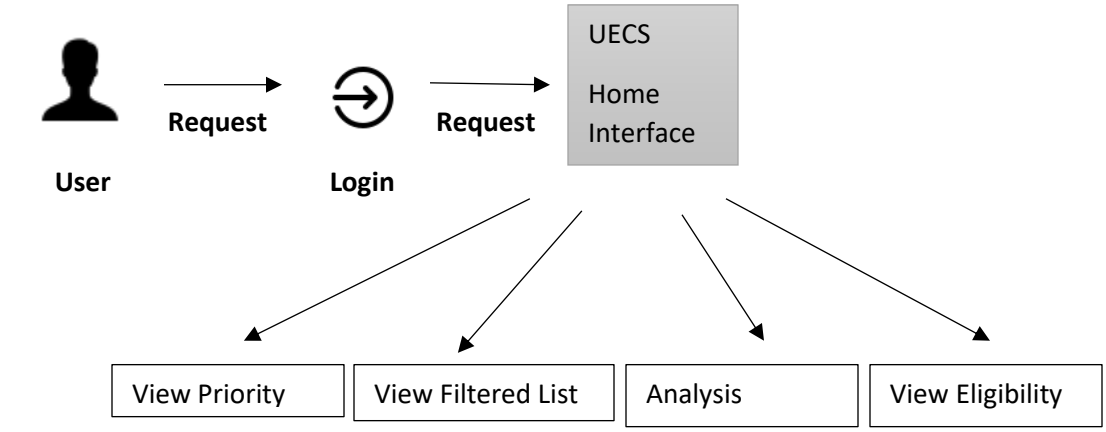
**1.1: Admin Login**

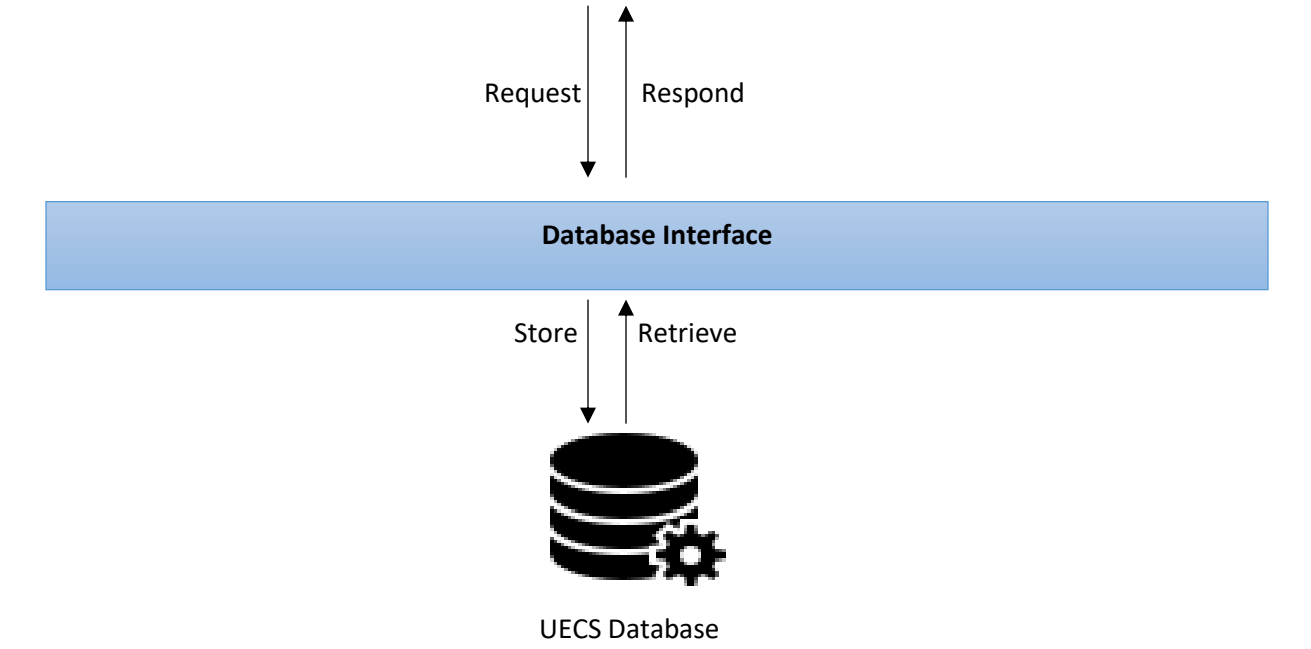



Administrator can log into the system and access every module that a user can access, and an additional module called Administration module.

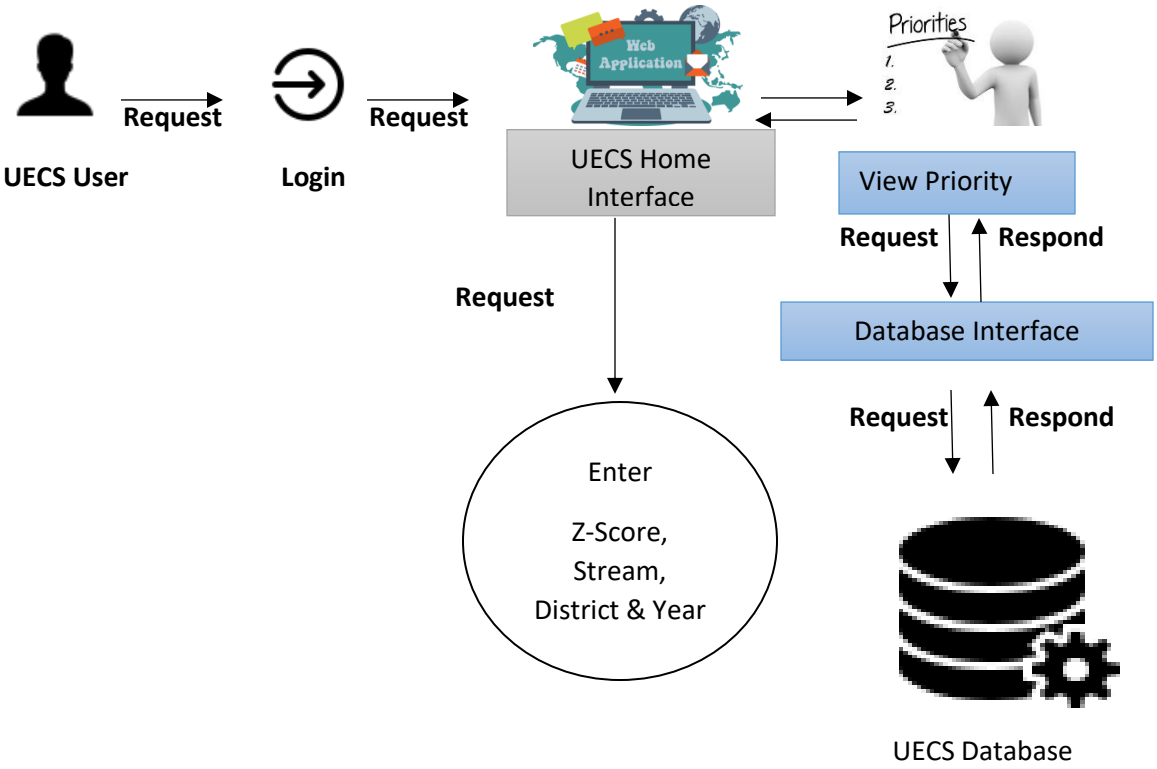
### 1.2: General User Login

Each general user of the system needs to be logged on to the system before using the system. Each user will be providing with a unique username and password to access the system by the administrator. When they are entered in the login interface those values will be verified using the data in the database and redirect the user to the system in order to achieve the results as user required.



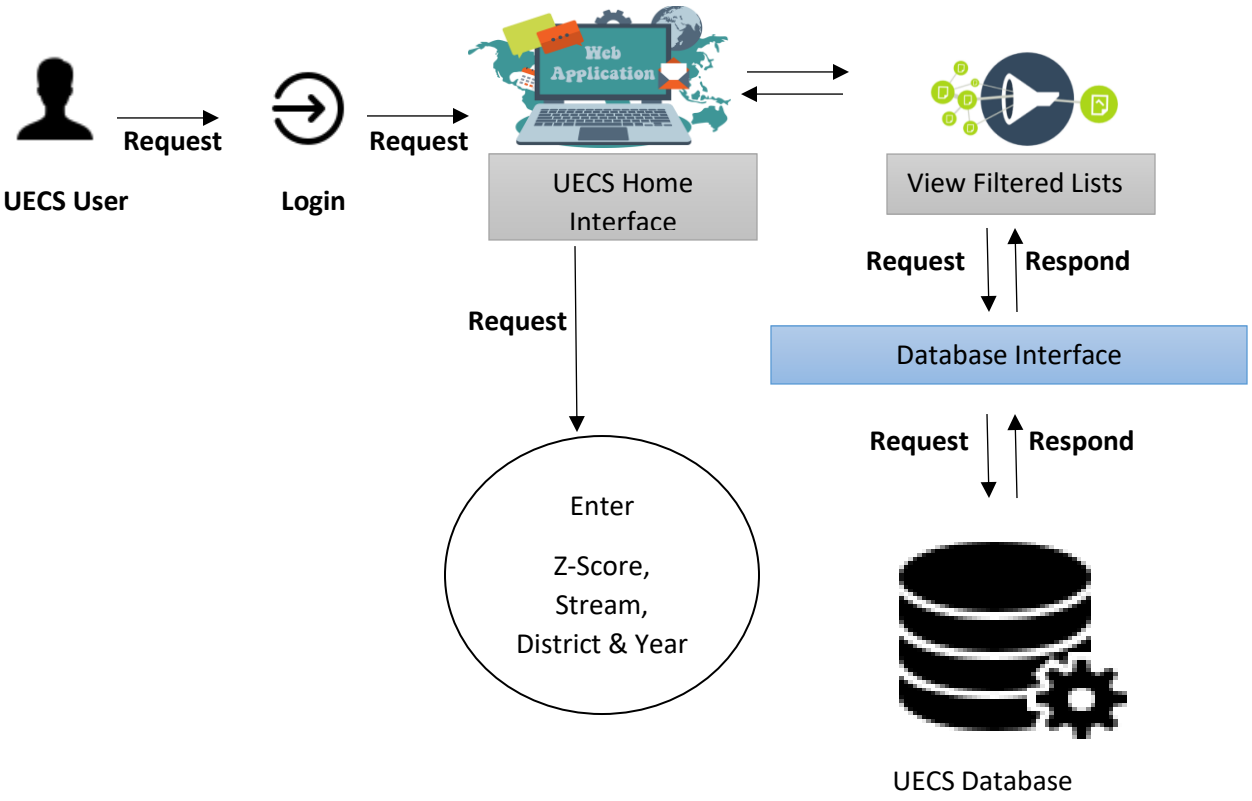



**Module 2: "View Priority"**



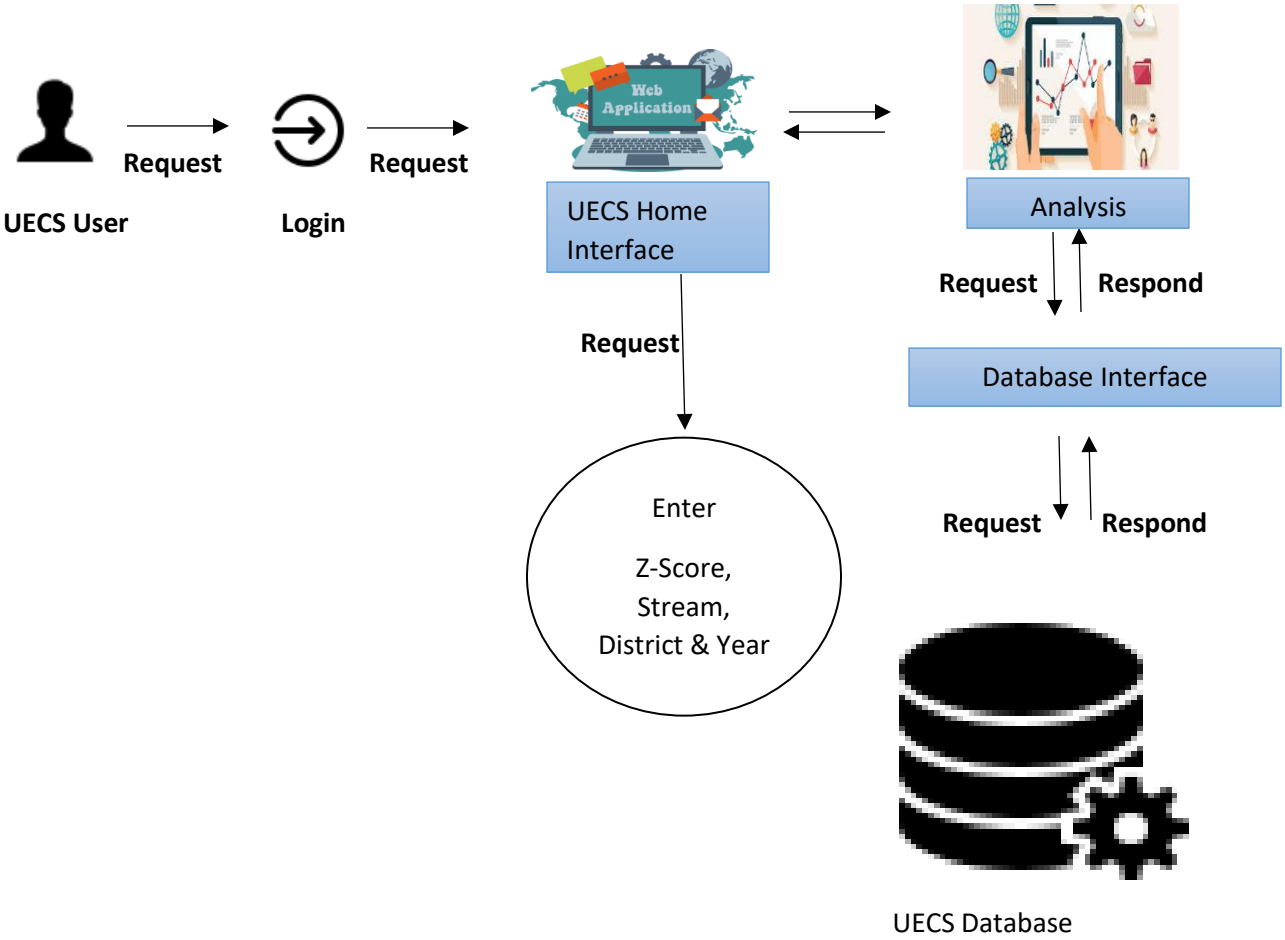

When the user is log into the system, he/she can request for their individual need by providing necessary information by filling the form. After that the updates are stored in the database and the output will be received as required. The degree programs the user is eligible are displayed according to the most priority.

**Module 3: "View Filtered List"**




When the user is log into the system, he/she can request for their individual need by providing necessary information by filling the form. After that the updates are stored in the database and the output will be the list of eligible degree programs which are filtered according to stream wise.

**Module 4: "Background Analysis"**



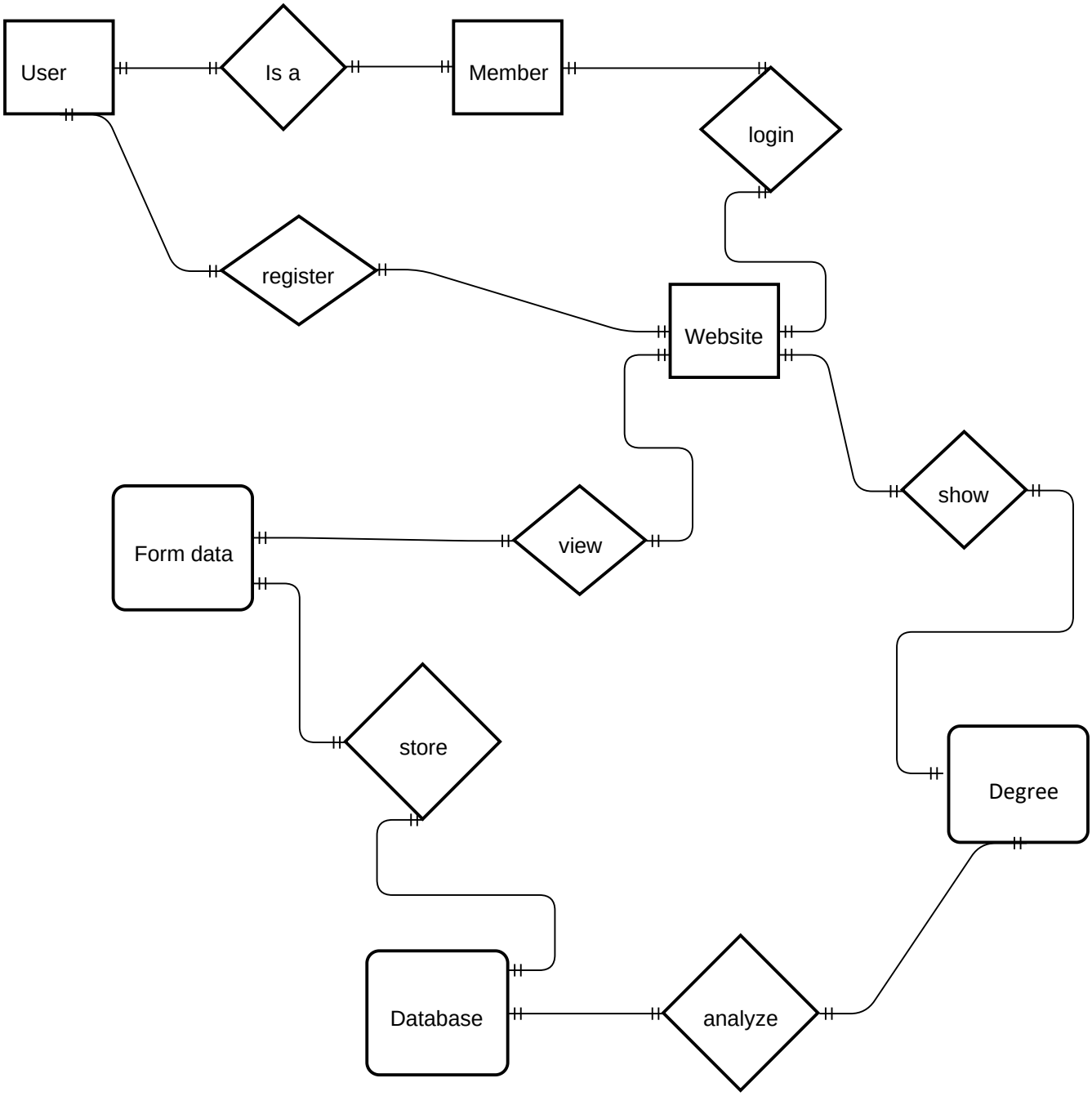

When the user log into the system, if he/she have no idea about the selection process for the universities based on previous year's results, through this module we provide a statistical analysis of previous records.

## 4.0 Data Design

The UECS will have a one centralized database in the server. MySQL database will be used for this purpose. This section of the document represents the conceptual data design of the system, the process of converting the EER diagram into tables and the database relationship diagram. Furthermore, the tables of the database are also stated with their attributes and data types.

### 4.1 Conceptual Database Design

Following figure shows the EER model for the UECS database.



### 4.2 Mapping of Logical Database to Relations

The above shown EER model is converted into relations using the 8-step process.

1st Step-Regular Entity Types

MEMBER (first_name, last_name, email, phone, index_number, district, stream, zscore, student_id)
USER (Username, email, Password, confirm_password)
DEGREE (Unicode, zscore, Stream, Course, University, District)
COMMENT (name, email)

2nd Step-Weak Entity Types

USER (Username, email, Password, confirm_password)
COMMENT (name, email)

3rd Step-Binary One-to-One

MEMBER (first_name, last_name, email, phone, index_number, district, stream, zscore, member_id, Username_FK)
USER (Username, email, Password, confirm_password)

4th Step-Binary One-to-Many

MEMBER (first_name, last_name, email, phone, index_number, district, stream, zscore, member_id)
COMMENT (name, email, member_id_FK)

5th Step-Binary Many-to-Many

MEMBER (first_name, last_name, email, phone, index_number, district, stream, zscore, member_id)
DEGREE (Unicode, zscore, Stream, Course, University, District)

6th Step-Mapping of Unary Relationships

No Unary Relationships here

7th Step-Mapping of Ternary Relationships

No Ternary Relationships here

8th Step-Mapping Super/Subtype Relationships

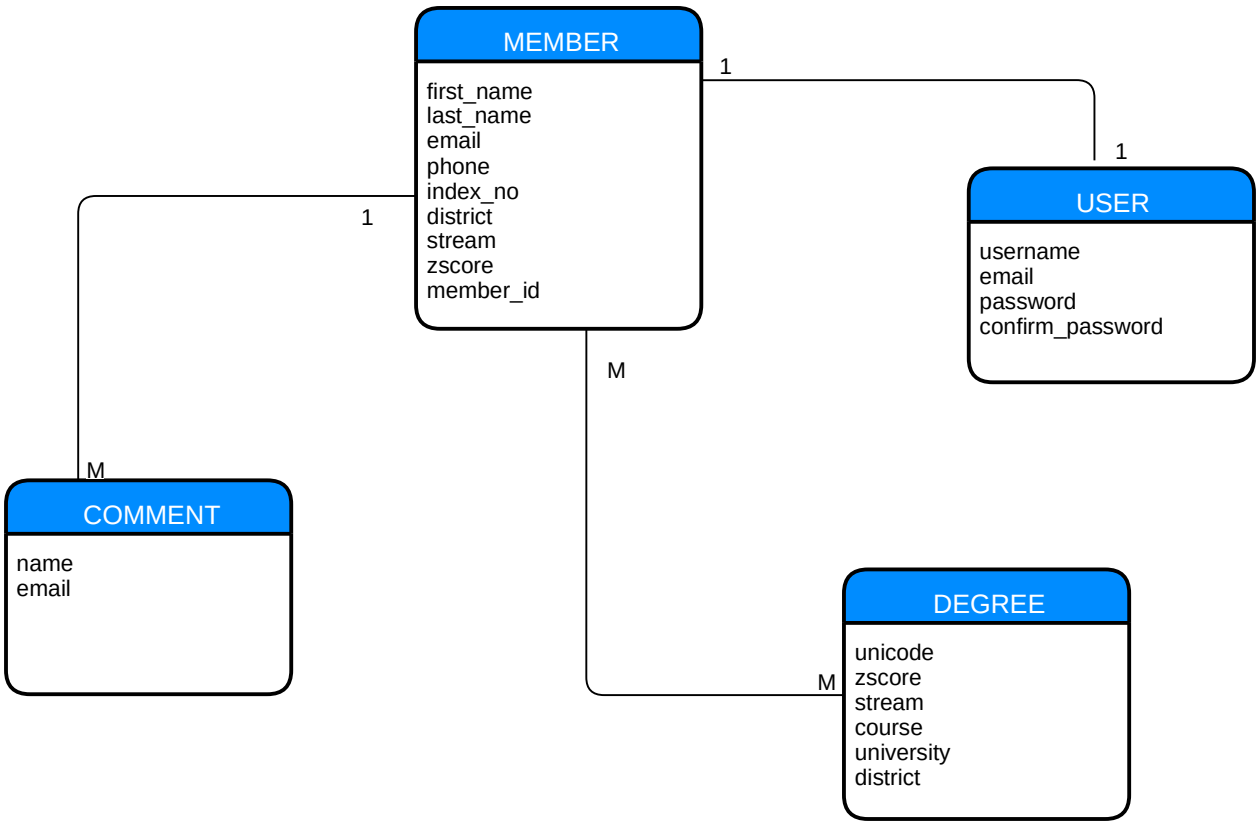
No Super/Subtype Relationships

After the normalization to 3rd normalization form there were no changes. Therefore, the final 3NF tables is given below.

MEMBER (first_name, last_name, email, phone, index_number, district, stream, zscore, member_id, Username_FK)
USER (user_id, Username, email, Password, confirm_password)
COMMENT (name, email, member_id_FK)
DEGREE_(Unicode, zscore, Stream, Course, University, District)

### 4.3 Database Relationship Diagram of the Database

The diagram shown below indicates the relationship among tables in the database.



### 4.4 Data Type Design for the Database

| TABLE:USER | | |
|---|---|---|
| **Attribute** | **Data Type** | **Length** |
| Username | Varchar | **255** |
| Email | Varchar | **255** |
| password | Varchar | **255** |
| Confirm_password | Varchar | **255** |

| TABLE: MEMBER | | |
|---|---|---|
| **Attribute** | **Data Type** | **Length** |
| First_name | Varchar | **255** |
| Last_name | Varchar | **255** |
| Email | Varchar | **255** |
| Phone | Text | |
| Index_no | Varchar | **12** |
| District | Varchar | **255** |
| Stream | Varchar | **255** |
| Zscore | Float | **5,4** |
| Member_id | Int | **11** |

| TABLE: DEGREE | | |
|---|---|---|
| **Attribute** | **Data Type** | **Length** |
| Unicode | Int | **11** |
| Zscore | Double | |
| Stream | Varchar | **100** |
| Course | Varchar | **255** |
| University | Varchar | **150** |
| District | Varchar | **255** |

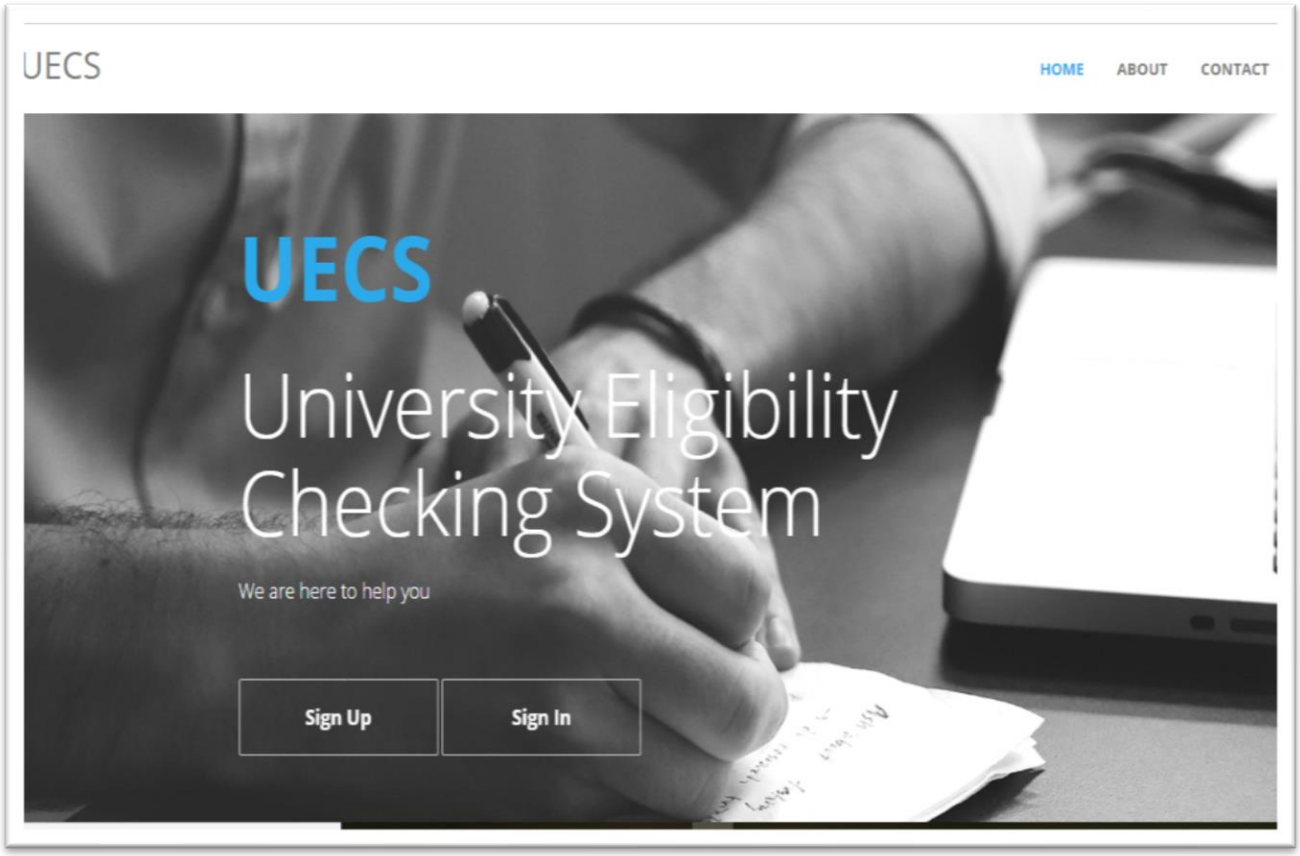
| TABLE: COMMENT | | |
|---|---|---|
| **Attribute** | **Data Type** | **Length** |
| Username | Varchar | |
| Email | Varchar | |

## 5.0 Interface Design

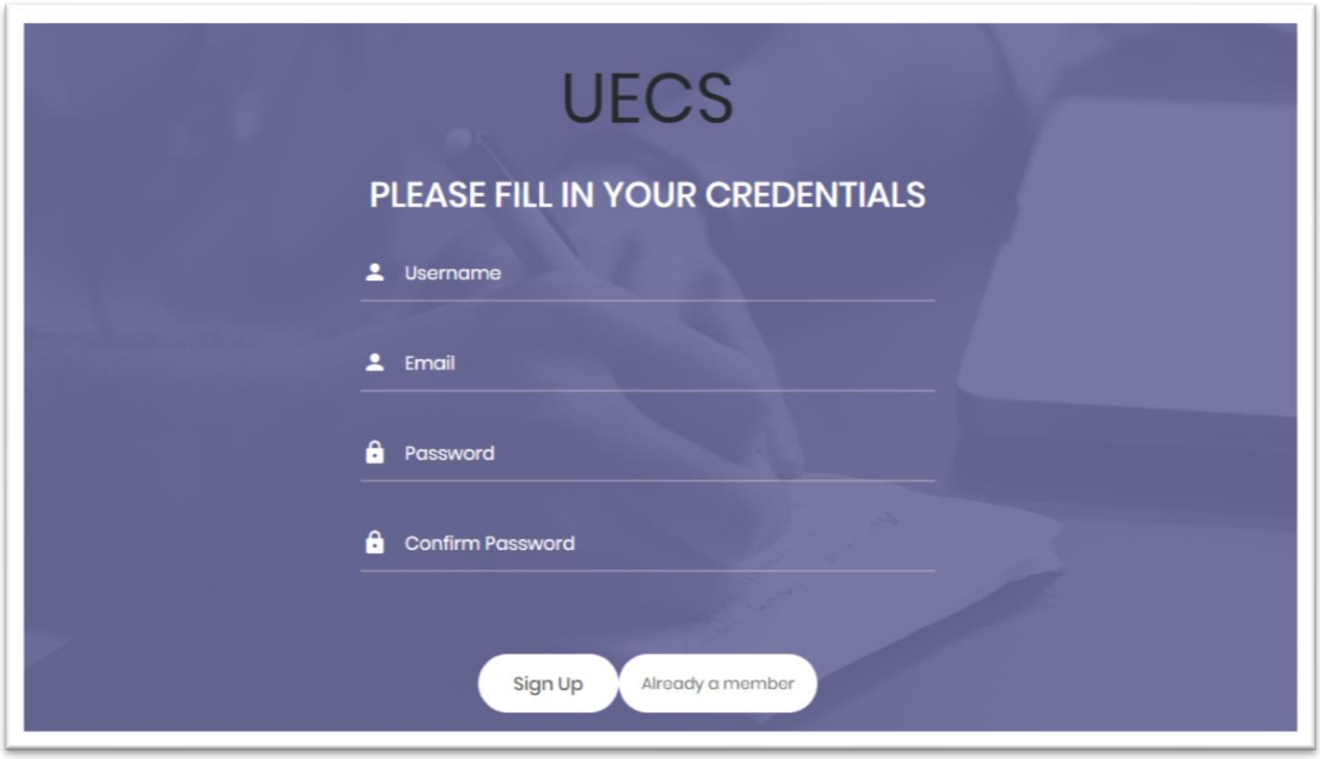
The user-friendly interface designs for the developing system are shown below.

### 5.1 User Login Interface

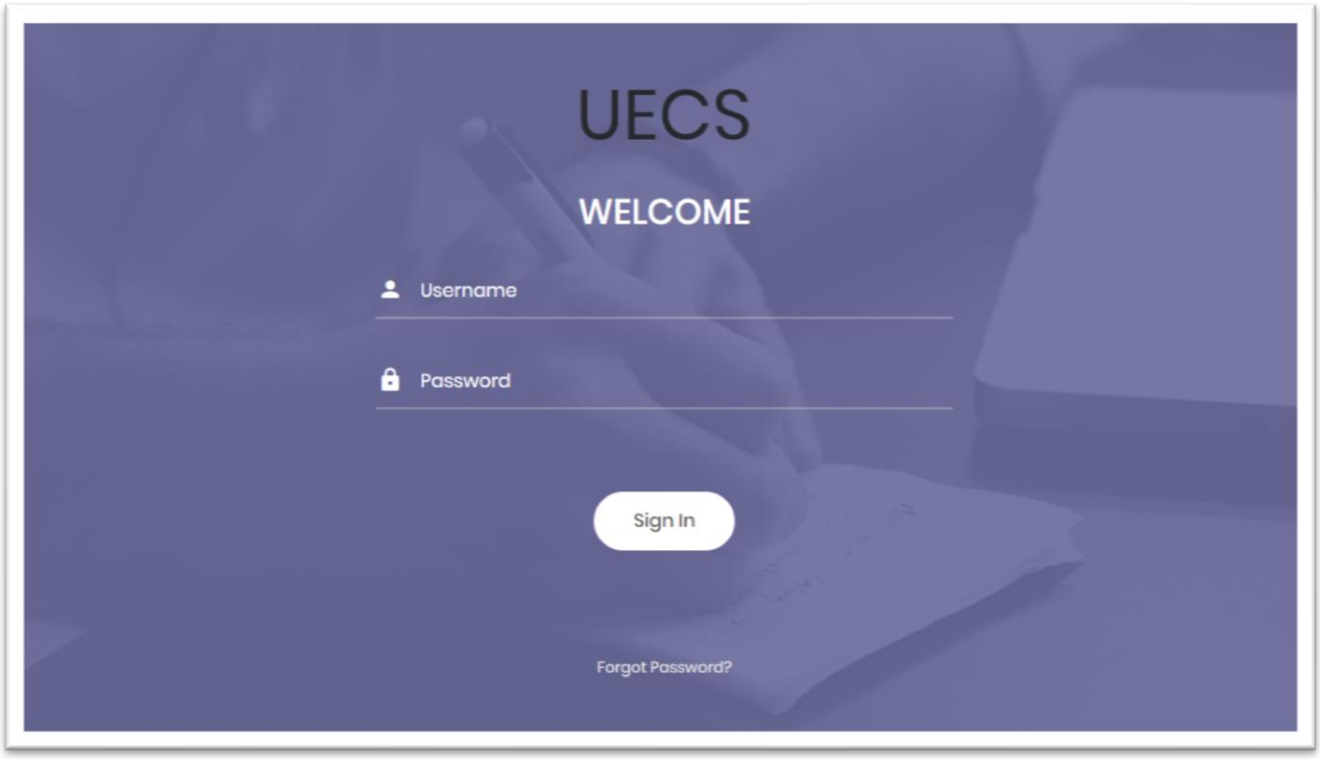
To enter the given options of UECS, users need to first register in the system. When submitting the username and password database checks the validity of entered username & password. Then it checks the user levels & provides the sign in window. Following figure shows the interface of the Sign Up & SignIn windows.



**SignUp**



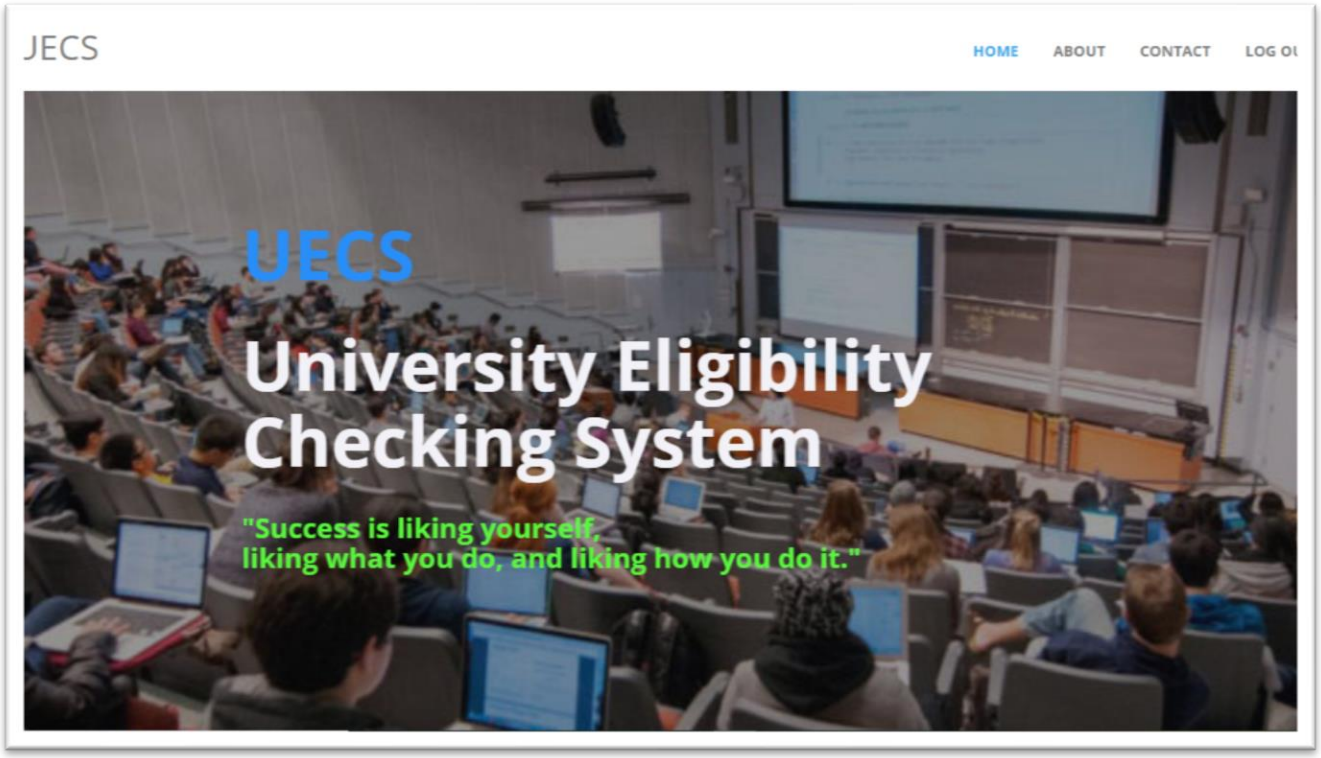

**SignIn**



### 5.2 Interface of Home Window

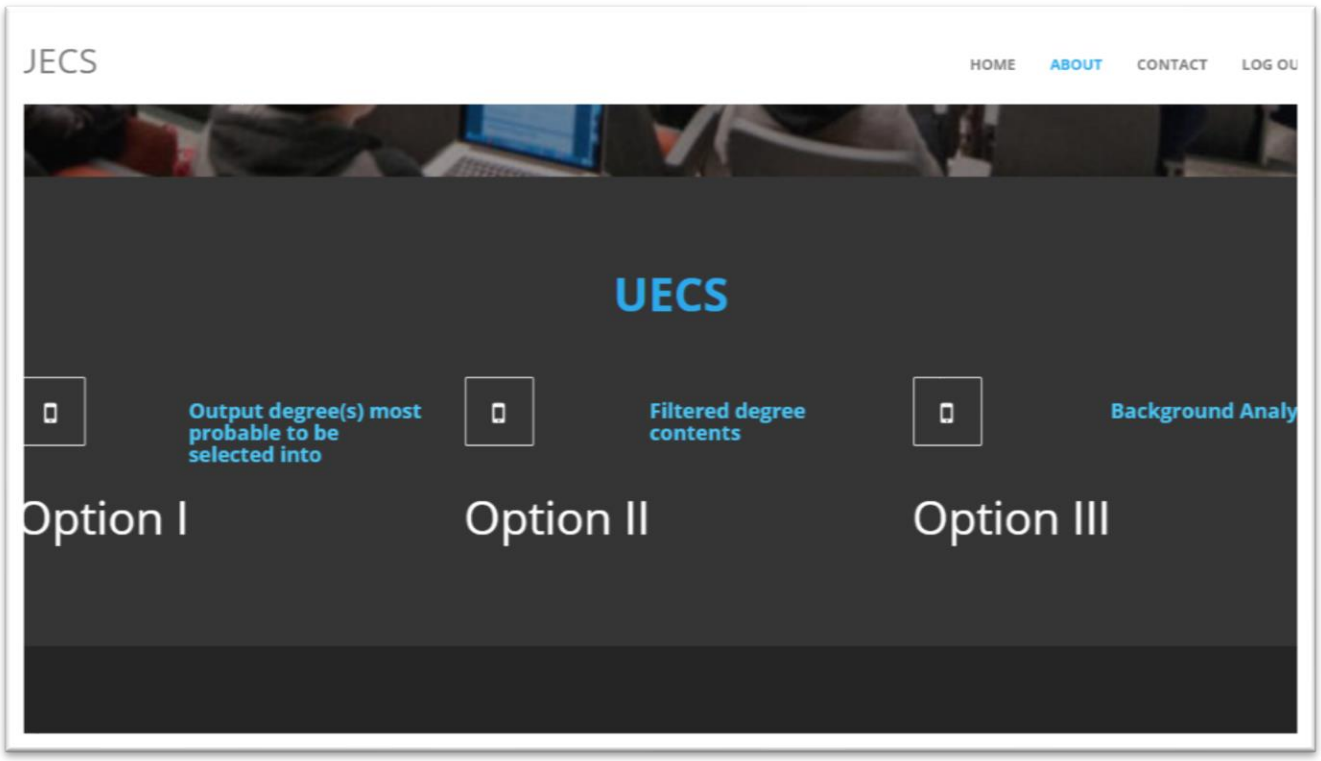
After successfully login, user gets the home window. It consists of a scroll up window while the menu tab displays above in return link back to the sub windows which contains home, about & contact inside the home window. Once the user is login to the system, he is free to use any of the options provided by the website.

Interface after a user is sign into the system.



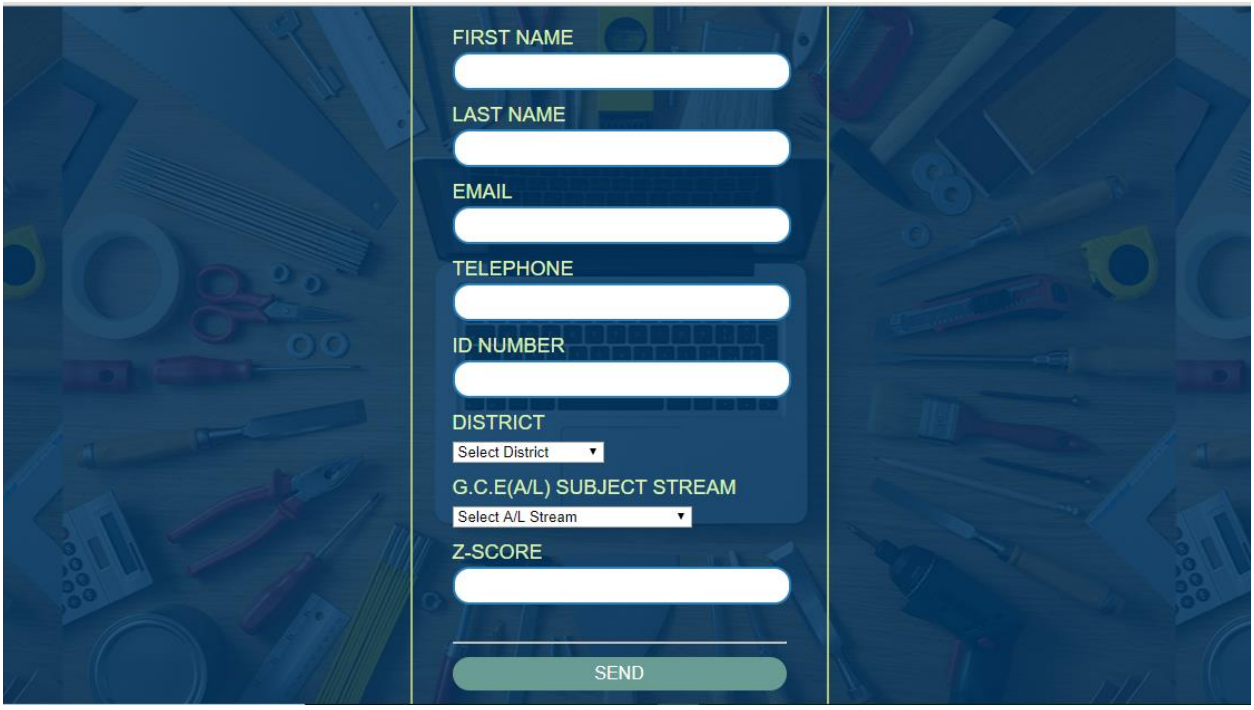
### 5.2.1 Interface of about tab

This tab contains all the main functions provided by the system. The following figure shows the interface of it.



### 5.2.1.1 Interface of option 1

Here when the user request for this section the site provide him the following form to enter his details.




"Output degrees most probable to be selected into"

Once he/she submit the form after completing, it will be stored in the database and the prioritized degrees will be sort out from most probable to least probable order.

### 5.2.1.2 Interface of option 2

"Filtered degree content stream wise"

The same above-mentioned form is displayed as soon as the user click on this link. So, after storing data provided by the user, the database will store the data and will provide a list of degree programs that he/she is eligible with.

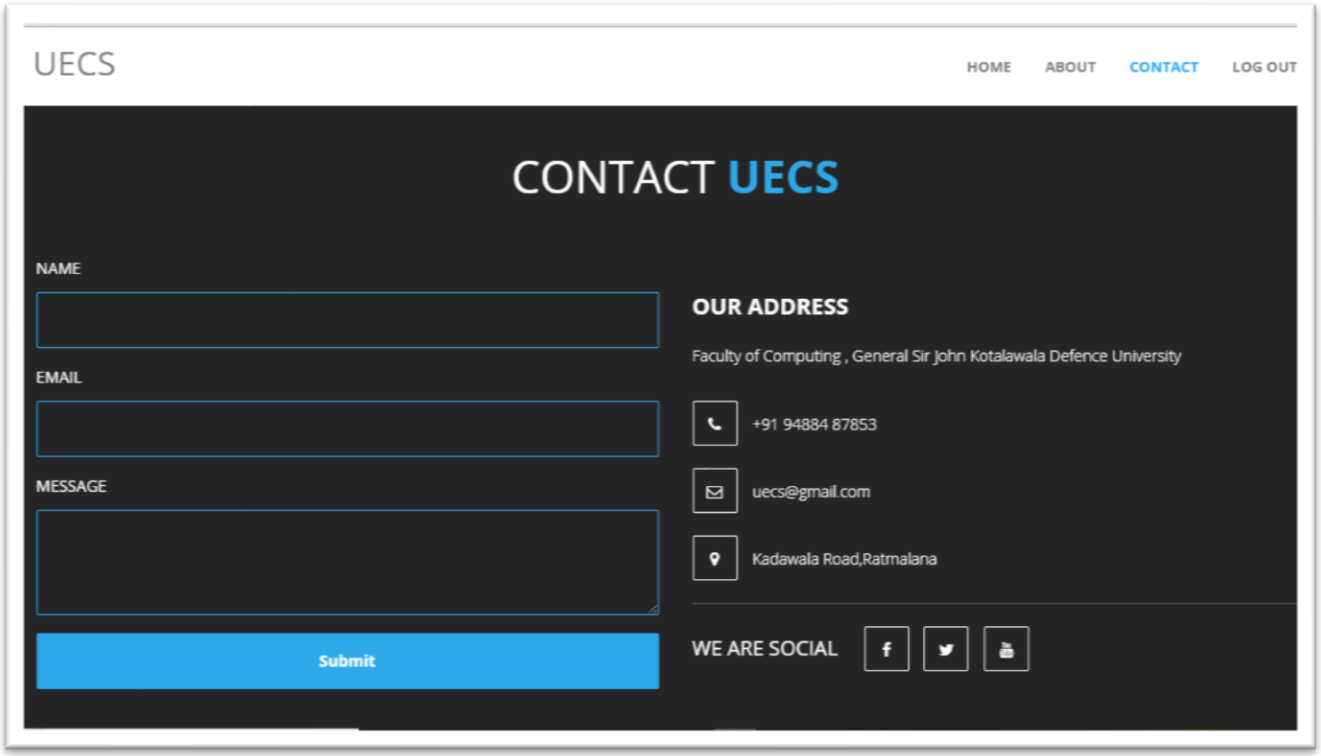
### 5.2.1.3 Interface of option 3

"Background research"

The user will able to get a clear idea about how the selections are made during the previous years and how the priorities are given in each university.

### 5.2.2 Interface of contact tab

Users can provide their opinions and more improvements that they are expecting from the system. Following figure shows the interface of the contact tab.



## 6.0 Summary

This document provides the Overall System Architecture, Software Architecture, Data Design & Interface Design regarding to the development of the computerized web based UECS system.

Overall architecture of this system is divided into 3 main layers namely presentation layer, application layer & data layer. Presentation layer focuses on the how the system interfaces are presented to the user while the application layer builds interaction between the presentation layer where the inputs of the interfaces and the data link layer where the required data are contained. Data layer includes the database management applications which will be helped to store data of each module. There is only one database to handle the records and this database will consist of several tables. Html will be used for the interface designing while php will be used as the scripting language for the system. phpMyAdmin & MySQL will be used as the database management application.

Software architecture of the developing system will be based on a modularized approach where system is divided into different modules. UECS will contain main 3 sections namely “Output degrees most probable to be selected into”, “filtered degree contents” & “Background analysis”. Each section will have specific functions under them.

When it comes to the data design there will be one central database using MySQL database management application hosted in the web server. Access to the database should be done through the network. An EER is modeled as for the requirements of the developing system and it is then converted into relations. Database contains several tables according to the logical design to store information about users, degrees etc. Relationships between these tables are also established according to the requirements.

All the interfaces & forms needed which are described above are designed with using html in order to produce user friendly interfaces.