

# **CANDREAM**

*Project Report Submitted By*

**AMALA MARIYAT**

**Reg. No.:AJC20MCA-2010**

*In Partial fulfillment for the Award of the Degree Of*

**MASTER OF COMPUTER APPLICATIONS (2 YEAR)  
APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY**



**AMAL JYOTHI COLLEGE OF ENGINEERING  
KANJIRAPPALLY**

[Affiliated to APJ Abdul Kalam Technological University, Kerala.  
Approved by AICTE, Accredited by NAAC with 'A' grade. Koovappally, Kanjirappally,  
Kottayam, Kerala – 686518]

**2020-2022**

**DEPARTMENT OF COMPUTER APPLICATIONS**  
**AMAL JYOTHI COLLEGE OF ENGINEERING**  
**KANJIRAPPALLY**



**CERTIFICATE**

This is to certify that the Project report, “**CANDREAM**” is the bonafide work of **AMALA MARIYAT (Reg.No:AJC20MCA-2010)** in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2021-2022.

**Ms. Anit James**  
**Internal Guide**

**Rev. Fr. Dr. Rubin Thottupurathu Jose**  
**Coordinator**

**External Examiner**

**Rev. Fr. Dr. Rubin Thottupurathu Jose**  
**Head of the Department**

## **DECLARATION**

I hereby declare that the project report “**CANDREAM**” is a bonafide work done at Amal Jyothi College of Engineering, towards the partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2022.

**Date:**

**AMALA MARIYAT**

**KANJIRAPPALLY**

**Reg. No: AJC20MCA-2010**

## ACKNOWLEDGEMENT

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our manager **Rev. Fr. Dr. Mathew Paikatt** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev. Fr. Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinator **Rev. Fr. Dr. Rubin Thottupurathu Jose** for his valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also like to express sincere gratitude to my guide, **Ms. Anit James** for her inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

AMALA MARIYAT

## **ABSTRACT**

Candream is a website that offers assistance to students who are delaying their master's degree relocation to Canada. More students are seeking higher education in Canadian nations as a result of the rapid expansion of global education. According to statistics, around 14% of international students in Canada are from India. Therefore, we are here to integrate all immigration procedures into a single system with modules for administration, students, higher education, and immigration consulting. Students can check and select advisors, colleges, and courses here based on the information provided about them. Due to the fact that the administration verifies the consultancies, it will assist students in selecting the one that is most appropriate for them. Students may submit applications to numerous universities for a variety of courses. Students submit applications and supporting documentation to the consultancy, which can then forward the applications to the institution after confirming them. The university can then examine and accept student application materials. When a student's application is submitted, reviewed by consultants, and accepted by universities, an email message is sent. Universities produce reports for accepted students. Consultancies pay the fees added by universities for the students, while students pay the costs added by consultancies.

# CONTENT

| <b>Sl. No</b> | <b>Topic</b>                         | <b>Page No</b> |
|---------------|--------------------------------------|----------------|
| <b>1</b>      | <b>INTRODUCTION</b>                  | <b>1</b>       |
| <b>1.1</b>    | <b>PROJECT OVERVIEW</b>              | <b>2</b>       |
| <b>1.2</b>    | <b>PROJECT SPECIFICATION</b>         | <b>2</b>       |
| <b>2</b>      | <b>SYSTEM STUDY</b>                  | <b>4</b>       |
| <b>2.1</b>    | <b>INTRODUCTION</b>                  | <b>5</b>       |
| <b>2.2</b>    | <b>EXISTING SYSTEM</b>               | <b>6</b>       |
| <b>2.3</b>    | <b>DRAWBACKS OF EXISTING SYSTEM</b>  | <b>6</b>       |
| <b>2.4</b>    | <b>PROPOSED SYSTEM</b>               | <b>6</b>       |
| <b>2.5</b>    | <b>ADVANTAGES OF PROPOSED SYSTEM</b> | <b>6</b>       |
| <b>3</b>      | <b>REQUIREMENT ANALYSIS</b>          | <b>8</b>       |
| <b>3.1</b>    | <b>FEASIBILITY STUDY</b>             | <b>9</b>       |
| <b>3.1.1</b>  | <b>ECONOMICAL FEASIBILITY</b>        | <b>9</b>       |
| <b>3.1.2</b>  | <b>TECHNICAL FEASIBILITY</b>         | <b>9</b>       |
| <b>3.1.3</b>  | <b>BEHAVIORAL FEASIBILITY</b>        | <b>10</b>      |
| <b>3.2</b>    | <b>SYSTEM SPECIFICATION</b>          | <b>12</b>      |
| <b>3.2.1</b>  | <b>HARDWARE SPECIFICATION</b>        | <b>12</b>      |
| <b>3.2.2</b>  | <b>SOFTWARE SPECIFICATION</b>        | <b>12</b>      |
| <b>3.3</b>    | <b>SOFTWARE DESCRIPTION</b>          | <b>12</b>      |
| <b>3.3.1</b>  | <b>PHP</b>                           | <b>12</b>      |
| <b>3.3.2</b>  | <b>MYSQL</b>                         | <b>13</b>      |
| <b>3.3.3</b>  | <b>CODEIGNITER</b>                   | <b>14</b>      |
| <b>4</b>      | <b>SYSTEM DESIGN</b>                 | <b>15</b>      |
| <b>4.1</b>    | <b>INTRODUCTION</b>                  | <b>16</b>      |

|              |                                         |           |
|--------------|-----------------------------------------|-----------|
| <b>4.2</b>   | <b>UML DIAGRAM</b>                      | <b>16</b> |
| <b>4.2.1</b> | <b>USE CASE DIAGRAM</b>                 | <b>17</b> |
| <b>4.2.2</b> | <b>SEQUENCE DIAGRAM</b>                 | <b>20</b> |
| <b>4.2.3</b> | <b>STATE CHART DIAGRAM</b>              | <b>22</b> |
| <b>4.2.4</b> | <b>ACTIVITY DIAGRAM</b>                 | <b>24</b> |
| <b>4.2.5</b> | <b>CLASS DIAGRAM</b>                    | <b>26</b> |
| <b>4.2.6</b> | <b>OBJECT DIAGRAM</b>                   | <b>28</b> |
| <b>4.2.7</b> | <b>COMPONENT DIAGRAM</b>                | <b>29</b> |
| <b>4.2.8</b> | <b>DEPLOYMENT DIAGRAM</b>               | <b>30</b> |
| <b>4.3</b>   | <b>USER INTERFACE DESIGN</b>            | <b>31</b> |
| <b>4.4</b>   | <b>DATA BASE DESIGN</b>                 | <b>34</b> |
| <b>5</b>     | <b>SYSTEM TESTING</b>                   | <b>45</b> |
| <b>5.1</b>   | <b>INTRODUCTION</b>                     | <b>46</b> |
| <b>5.2</b>   | <b>TEST PLAN</b>                        | <b>47</b> |
| <b>5.2.1</b> | <b>UNIT TESTING</b>                     | <b>47</b> |
| <b>5.2.2</b> | <b>INTEGRATION TESTING</b>              | <b>48</b> |
| <b>5.2.3</b> | <b>VALIDATION TESTING</b>               | <b>48</b> |
| <b>5.2.4</b> | <b>USER ACCEPTANCE TASTING</b>          | <b>48</b> |
| <b>5.2.5</b> | <b>SELENIUM TESTING</b>                 | <b>49</b> |
| <b>6</b>     | <b>IMPLEMENTATION</b>                   | <b>56</b> |
| <b>6.1</b>   | <b>INTRODUCTION</b>                     | <b>57</b> |
| <b>6.2</b>   | <b>IMPLEMENTATION PROCEDURE</b>         | <b>58</b> |
| <b>6.2.1</b> | <b>USER TRAINING</b>                    | <b>58</b> |
| <b>6.2.2</b> | <b>TRAINING ON APPLICATION SOFTWARE</b> | <b>58</b> |
| <b>6.2.3</b> | <b>SYSTEM MAINTENANCE</b>               | <b>59</b> |
| <b>6.2.4</b> | <b>HOSTING</b>                          | <b>59</b> |
| <b>7</b>     | <b>CONCLUSION &amp; FUTURE SCOPE</b>    | <b>61</b> |

|            |                         |           |
|------------|-------------------------|-----------|
| <b>7.1</b> | <b>CONCLUSION</b>       | <b>62</b> |
| <b>7.2</b> | <b>FUTURE SCOPE</b>     | <b>62</b> |
| <b>8</b>   | <b>BIBLIOGRAPHY</b>     | <b>63</b> |
| <b>9</b>   | <b>APPENDIX</b>         | <b>65</b> |
| <b>9.1</b> | <b>SAMPLE CODE</b>      | <b>66</b> |
| <b>9.2</b> | <b>SCREEN SHOTS</b>     | <b>73</b> |
| <b>9.3</b> | <b>PLAGARISM REPORT</b> | <b>77</b> |



## List of Abbreviation

|      |                                      |
|------|--------------------------------------|
| IDE  | - Integrated Development Environment |
| HTML | - Hyper Text Markup Language.        |
| CSS  | - Cascading Style Sheet              |
| SQL  | - Structured Query Language          |
| UML  | - Unified Modeling Language          |

# **CHAPTER 1**

## **INTRODUCTION**

## 1.1 PROJECT OVERVIEW

A website called CANDREAM supports students who are waiting to immigrate to Canada in order to pursue higher education. One system, CANDREAM, can handle the entire processes. More students are seeking higher education in Canadian nations as a result of the rapid expansion of global education. According to statistics, around 14% of international students in Canada are from India. Therefore, we are here to integrate all immigration procedures into a single system with modules Admin, Students, University, and Immigration Consultancy. Students can view consultancies and universities here based on the information provided. By assessing the registration of consultancies and universities into the system by Admin, this would assist students in selecting the option that best suits them. It will also help students avoid fraudulent consultancies.

Students submit applications for various programmes at various institutions through various consultants. The technology allows students to upload their documents. Universities and consulting firms verify student applications using the documentation submitted by the applicants.

Here, the student can check universities, courses, and consultancies through the student portal before applying for admission. Administrator will examine the applications and confirm their approval. Admin can also oversee colleges and consulting firms.

## 1.2 PROJECT SPECIFICATION

The system includes 4 modules. They are:

### 1. Admin

- Login
- Verify and Manage Universities
- Verify and Manage Consultancies
- Change Password

### 2. Student

- Registration
- Login
- Check Consultancies
- Check Universities
- Check Courses
- Apply for Admission
- Check Application Status
- Fee Payment

---

**3. Consultancy**

- Registration
- Login
- Verify or Reject Student Applications
- Forward Student Application to University
- Apply University Contract
- Check Student Application Status
- Add Fees

**4. University**

- Registration
- Login
- Verify or Reject Student Applications
- Admit Student
- Approve University Contract

## **CHAPTER 2**

### **SYSTEM STUDY**

## 2.1 INTRODUCTION

System analysis is the process of gathering data, analyzing it, identifying issues, and making recommendations for system changes. It is the problem-solving process that necessitates effective communication between system users and system developers.

The most crucial stage of any system development process is system analysis or study. In-depth research and analysis are done on the system. The system analyst assumes the position of an interrogator and probes the inner workings of the existing system. The system's input is recognized, and the system is viewed as a whole. The various processes can be connected to the organizational outputs. The goal of system analysis is to identify the problem, find the relevant and crucial factors, analyze and combine the many components, and choose the best or, at the very least, most acceptable course of action.

The procedure needs to be thoroughly investigated utilizing a range of approaches, such as surveys and interviews. The data acquired by various sources needs to be thoroughly evaluated in order to draw a conclusion. The conclusion is knowing how the system functions. This system is known as the current one. Now, problem areas have been identified after a careful examination of the current system. Now the designer takes on the role of a problem-solver and attempts to fix the problems the company is experiencing. The solutions are replaced with proposals. The best option is then selected after an analytical comparison of the proposal and the existing system. The proposal is presented to the user with the option to accept or reject it. On the basis of user requests, the plan is assessed for any necessary revisions.

Preliminary research is the process of gathering and analyzing data in order to use it in later system studies. Initial research requires strong collaboration between system users and developers since it involves problem-solving. It carries out several feasibility studies. These investigations give a rough idea of the system activities, and this information can be utilized to choose the methods to employ for effective system research and analysis.

## 2.2 EXISTING SYSTEM

In existing system there are different websites which we have to visit and register for each step of the entire migration process. It may have to spend more time. And also, we cannot know what is the current status of the application. Delay in getting the information that the application is canceled will result in a lot of time wastage. And the majority of the processes cannot be done through automated system in the existing system. There is no way to detect fraud consultancies there. Because, lots of consultancy pages are available in internet, it is difficult to identify which is genuine or not.

## 2.3 DRAWBACKS OF EXISTING SYSTEM

- The existing system cannot guarantee authenticity and credibility of consultancies.
- Most of them are not easily affordable.
- The user does not have any means for the follow up on their application status with the educational institutions.

## 2.4 PROPOSED SYSTEM

The suggested system is intended to address every drawback of the current system. For business growth, it is essential to have a system that is more user-friendly and appealing to users; as a result, the system is suggested. Admin in our proposed system have access to view and validate all consultancies and universities. Students, admin, consultants, and universities are among the users of the planned system. The planned system aims to create a network of upgraded utilities. The approach guarantees avoiding fraudulent universities and consultancies while minimizing manual labour.

## 2.5 ADVANTAGES OF PROPOSED SYSTEM

The system's implementation and design are both rather simple. The system requires extremely little system resources and operates in almost all environments. It features the following things:

### **Avoid Fraud Users: -**

The consultancies and universities who are registered in the system can only be logged in after verifying and accepting by the admin. The documents and other details which are provided by the consultancies and universities can be viewed by the admin for verification. This avoids many scams in immigration consulting process.

**Ensure data accuracy: -**

The suggested remedy eliminates errors produced by people when inputting user information during registration.

**Better service: -**

The system won't be overburdened with hard copy storage. We can also save time and resources by carrying out the same task in a different way. Without losing any information, the data can be stored for a longer period of time.



## **CHAPTER 3**

# **REQUIREMENT ANALYSIS**

### **3.1 FEASIBILITY STUDY**

To ascertain if the project will, after completion, achieve the goals of the organisation in relation to the work, effort, and time put in it, a feasibility study is carried out. The developer can forecast the project's usefulness and potential future thanks to a feasibility study. A feasibility study's foundation is the determination of a system proposal's viability, which is based on its impact on the organization, capacity to meet user needs, and effective use of resources. As a result, a feasibility evaluation is frequently performed before a new application is approved for development. The document discusses the project's viability and includes a number of criteria, including technical ones, that were carefully considered during the feasibility study for this project.

#### **3.1.1 Economical Feasibility**

Cost-benefit analyses are required to support the evolving system. Must make sure that the project receiving the most attention is the one that will yield the best results the quickest. The price of setting up a new system is one of the variables.

Some significant financial queries raised during the initial probe include the following:

- The costs look into the entire system?
- The cost of the hardware and software?

The benefits manifest themselves as lower costs or less costly errors.

There are no manual costs involved with the suggested system because it was developed as part of a project. Additionally, the availability of all necessary resources suggests that the system might be implemented at a reasonable cost.

The costs associated with the CANDREAM project were divided into three groups: system costs, development costs, and hosting costs. All estimates show that the project was created at a reasonable cost. As all of its development was done using open-source software.

#### **3.1.2 Technical Feasibility**

The system must first undergo a technical evaluation. The assessment of its viability must be built upon an overview design of the system's requirements in terms of input, output, programmes, and procedures. After identifying an outline system, the inquiry must next recommend the type of equipment, necessary steps for building the system, and methods of operating the system once it has been designed.

---

The following technical issues were encountered during the investigation:

- Does the proposed technology function with existing technology?
- If developed, can the system grow?

### **3.1.3 Behavioral Feasibility**

The project should be designed in such a way that the necessary functionality and performance are met within the constraints. The fact that a newer version of the same software still functions with an older version means that the system can still be used even though the technology may become outmoded with time. Consequently, there aren't many restrictions on this project. The system was developed with PHP for the front end and a MySQL server for the back end; the project can be finished technically. The computer used has a potent Intel i3 core processor, 4GB of RAM, and a 1TB hard drive. Behaviorally Susceptible.

The following inquiries are part of the suggested system:

- Is there enough assistance for the users?
- Will the suggested system harm people?

The project would be advantageous since it will accomplish the goals when it is developed and put into action. After carefully analysing all behavioural elements, the proposal is behaviorally practicable.

The GUI of CANDREAM is easy to use and doesn't require any training for users.

---

## 3.2 SYSTEM SPECIFICATION

### 3.2.1 Hardware Specification

Processor - Intel core i3

RAM - 4 GB

Hard disk - 1 TB

### 3.2.2 Software Specification

Front End -PHP

Backend -MYSQL

Framework -Codeigniter 3

Clienton PC -Windows 7 and above.

Technologies used - JS, HTML5, J Query, CSS

## 3.3 SOFTWARE DESCRIPTION

### 3.3.1 PHP

PHP is a server-side scripting language used for both web development and other types of programming. PHP is now installed on more than 244 million websites and 2.1 million web servers. The PHP group now produces the reference implementation of PHP, which was first developed by Rasmus Ledorf in 1995. PHP is a recursive acronym that originally stood for personal Home page but is now used to refer to PHP: Hypertext Preprocessor. The PHP processor module on a web server translates PHP code to create the finished web page. Instead of calling an external file, PHP commands can be directly put into an HTML source file to handle data. Additionally, it has grown and is now equipped with a command-line interface. Instead of calling an external file, PHP commands can be directly put into an HTML source file to handle data. The GNU General Public License is incompatible with PHP because it has evolved to include a command-line interface and can be used independently due to restrictions on the usage of the term PHP (GPL). The majority of web servers allow for the free deployment of PHP, which can also be used independently as a shell on almost all platforms and operating systems

### 3.3.2 MySQL

The most well-known Open-Source SQL database management system, MySQL, was developed, distributed, and supported by Oracle Corporation. On the MySQL website, you may find the latest information about the MySQL program. A database management system is MySQL.

A database is a planned collection of data. Anything might be it, from a simple grocery list to a photo gallery to the vast amount of information in a company network. Data included in a computer database must be added to, accessed, and processed using a database management system, such as MySQL Server. Because computers are so good at processing enormous amounts of data, database management systems—whether employed as standalone programmes or as a component of other applications—are crucial to computing. MySQL databases are relational.

A relational database stores the data in separate tables rather than consolidating it into one enormous warehouse. The database structures are stored in physical files that are optimized for speed. A flexible programming environment is offered by the logical model, which includes objects like databases, tables, views, rows, and columns. Just a few examples of the rules you may create to control the relationships between various data fields include one-to-one, one-to-many, unique, compulsory or optional, and "pointers" between different tables. Since a well-designed database upholds these constraints, your application won't ever run into inconsistent, duplicate, outdated, or missing data.

MySQL stands for "Structured Query Language" with the SQL prefix. The most popular standard language for accessing databases is SQL. Depending on your programming environment, you might explicitly enter SQL (for example, to generate reports), incorporate SQL statements into other languages' code, or use a language-specific API that obscures the SQL syntax. By way of the ANSI/ISO SQL Standard, SQL is defined. Since its inception in 1986, the SQL standard has undergone multiple revisions. The 1992 standard, "SQL92," is referred to in this document.

The 1999 version of the standard is referred to as "SQL: 1999," and the most recent version is referred to as "SQL: 2003." The SQL Standard as it exists at any one time is referred to as "the SQL standard."

- **MySQL software is Open Source.**

Considering that the software is open source, anyone can use and alter it. Anyone can use and download the MySQL software for free online. You have the right to look at the source code and make any necessary changes. The MySQL software adheres to the GPL (GNU General Public License), which outlines what you can and cannot do with the programme under specific conditions. If the GPL makes you uncomfortable or if you need to incorporate MySQL code into a for-profit application, you can buy a commercially licenced version from us. Check out the MySQL Licensing Overview for more information.

- **The MySQL Database Server is incredibly quick, trustworthy, scalable, and simple to use.**

MySQL Server can run smoothly on a desktop or laptop and needs little to no maintenance in addition to your other apps, web servers, and other software. If you dedicate an entire system to MySQL, you can change the settings to utilize all the RAM, CPU, and I/O power. Client/server or embedded systems can use MySQL Server.

A multithreaded SQL server, multiple client programmes and libraries, management tools, and a wide range of application programming interfaces make up the client/server system known as the MySQL Database Software (APIs). In addition, we offer MySQL Server as a built-in multi-threaded library that you can incorporate into your software to produce a standalone solution that is more compact, quick, and easy to use.

### **3.3.3 CODEIGNITER**

PHP powers the CodeIgniter framework. CodeIgniter includes libraries, a user-friendly interface, a logical access structure, plug-ins, helpers, and other resources that make it easier to solve the complicated PHP functions while still retaining excellent speed. It speeds up the creation of a fully interactive, dynamic website while streamlining the PHP code. It supports MySQL 4.1 or newer and PHP version 5.2.6 or newer. Your web becomes more reliable, and your code is simpler to read and update. It is a lightweight, cost-free toolkit that is simpler to install.

## **CHAPTER 4**

### **SYSTEM DESIGN**

## 4.1 INTRODUCTION

Design is the first step in the development of any engineered system or product. Design is a creative process. A good design is the key to a system that works effectively. "Design" is the process of using many approaches and concepts to thoroughly outline a process or a system so that it can be physically implemented. The process of using several approaches and concepts to specify a tool, a procedure, or a system in sufficient detail to enable its physical actuality is one way to put it. Software design serves as the technical foundation of the software engineering process, regardless of the development paradigm used. This programme has through the best design phase imaginable, fine-tuning all efficiency, performance, and accuracy levels, just like any systematic technique. During the design stage, a user-oriented document is transformed into a document for programmers or database employees. The two phases of system development are the logical and physical designs.

## 4.2 UML DIAGRAM

The artefacts of the software system are specified, visualised, constructed, and documented using a standard language called UML. A draught of the UML 1.0 specification was provided to the Object Management Group (OMG), which was in charge of creating UML, in January 1997.

UML stands for Unified Modeling Language. UML is different from other well-known programming languages like C++, Java, COBOL, etc. Software blueprints are made using a visual language called UML. UML is referred to as an all-purpose visual modelling language for the visualisation, specification, construction, and documentation of software systems. Although representing software systems is the most popular use of UML, it is not the only one. Additionally, it is employed to model non-software-based systems. the process flow in the manufacturing facility, etc. Despite the fact that UML is not a programming language, tools that use UML diagrams can generate code in a variety of languages. Direct connections exist between UML and object-oriented analysis and design. Now an OMG standard, UML has undergone standardisation. Using all the components and connections, a complete UML diagram illustrating the system is produced. The UML diagram's appearance is the most important factor in the entire process.



The additional components are used to complete it. These eight diagrams are included in UML.

- Class diagram
- Object diagram
- Use case diagram
- Sequence diagram
- Activity diagram
- State chart diagram
- Deployment diagram
- Component diagram

#### **4.2.1 USE CASE DIAGRAM**

A use case diagram is a visual representation of the interactions between system components. A approach for identifying, outlining, and organizing system requirements is called a use case. The word "system" here refers to a thing that is being created or run, like a website for mail-order product sales and services. UML (Unified Modeling Language), a standard language for the modelling of real-world objects and systems, uses use case diagrams.

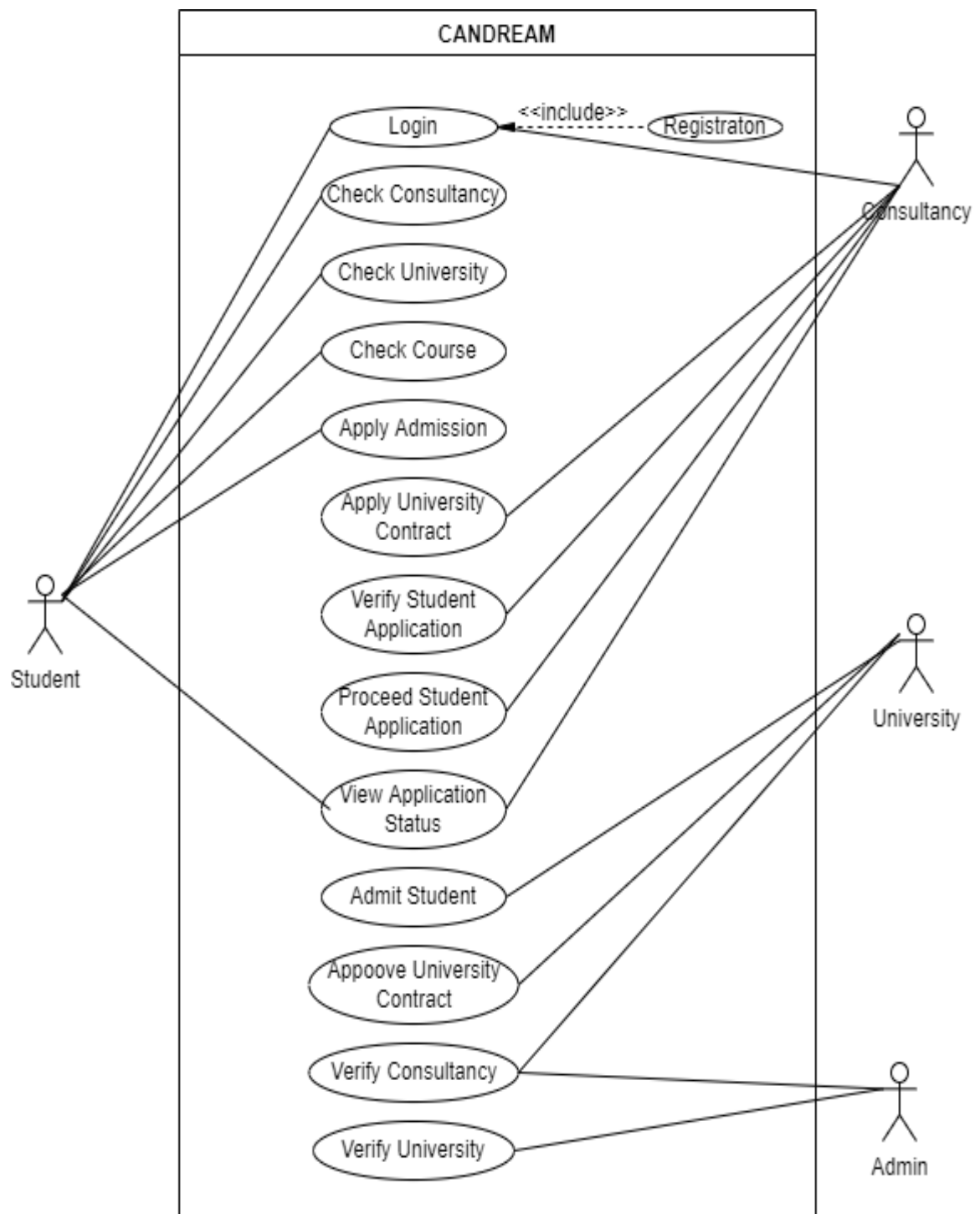
Just a few examples of system objectives include planning the overall requirements, validating a hardware design, testing and debugging a software product that is still in development, creating an online help reference, or performing a task centred on customer service. For instance, customer support, item ordering, catalogue updating, and payment processing are examples of use cases in a setting of product sales. A use case diagram consists of four components.

- The border that establishes the system of interest's position in reference to its surroundings.
- Performers are usually system users who may be recognised by the roles they play.
- The use cases, which are the specific roles that members of the system and those nearby play.

- The connections and interactions between the actors and use cases.

Use case diagrams are created to depict a system's functional requirements. To create an effective use case diagram after identifying the aforementioned things, we must adhere to the following rules.

- The name of a use case is highly important. It is important to choose a name that makes it obvious what tasks are being carried out.
- Give actors names that are appropriate.
- Make sure the diagram makes relationships and dependencies apparent.
- Relationships of all kinds should not be attempted to be included because the diagram's primary function is to define the needs.
- Whenever it is necessary to make a point clear, take notes.

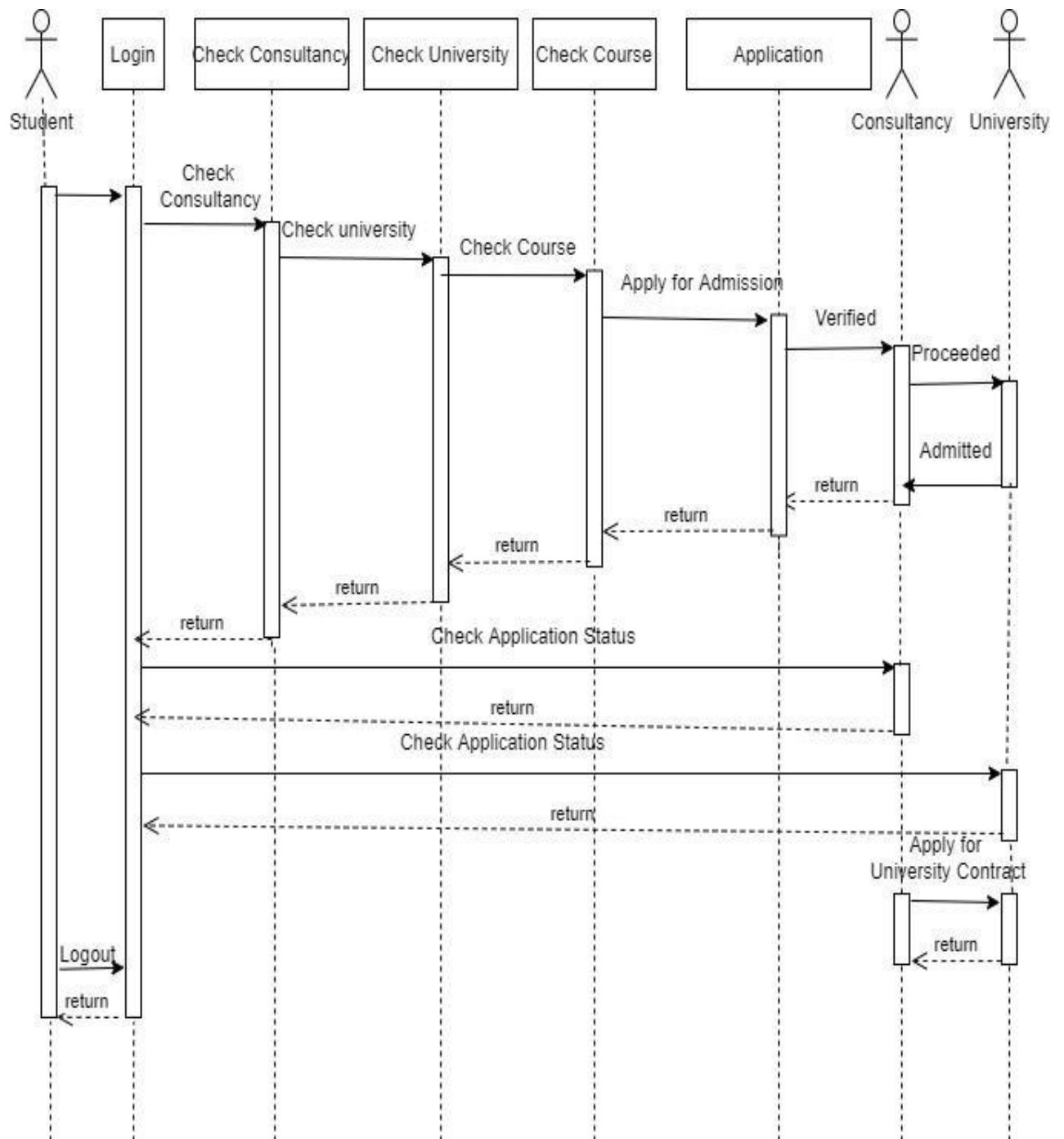


### **4.2.2 SEQUENCE DIAGRAM**

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

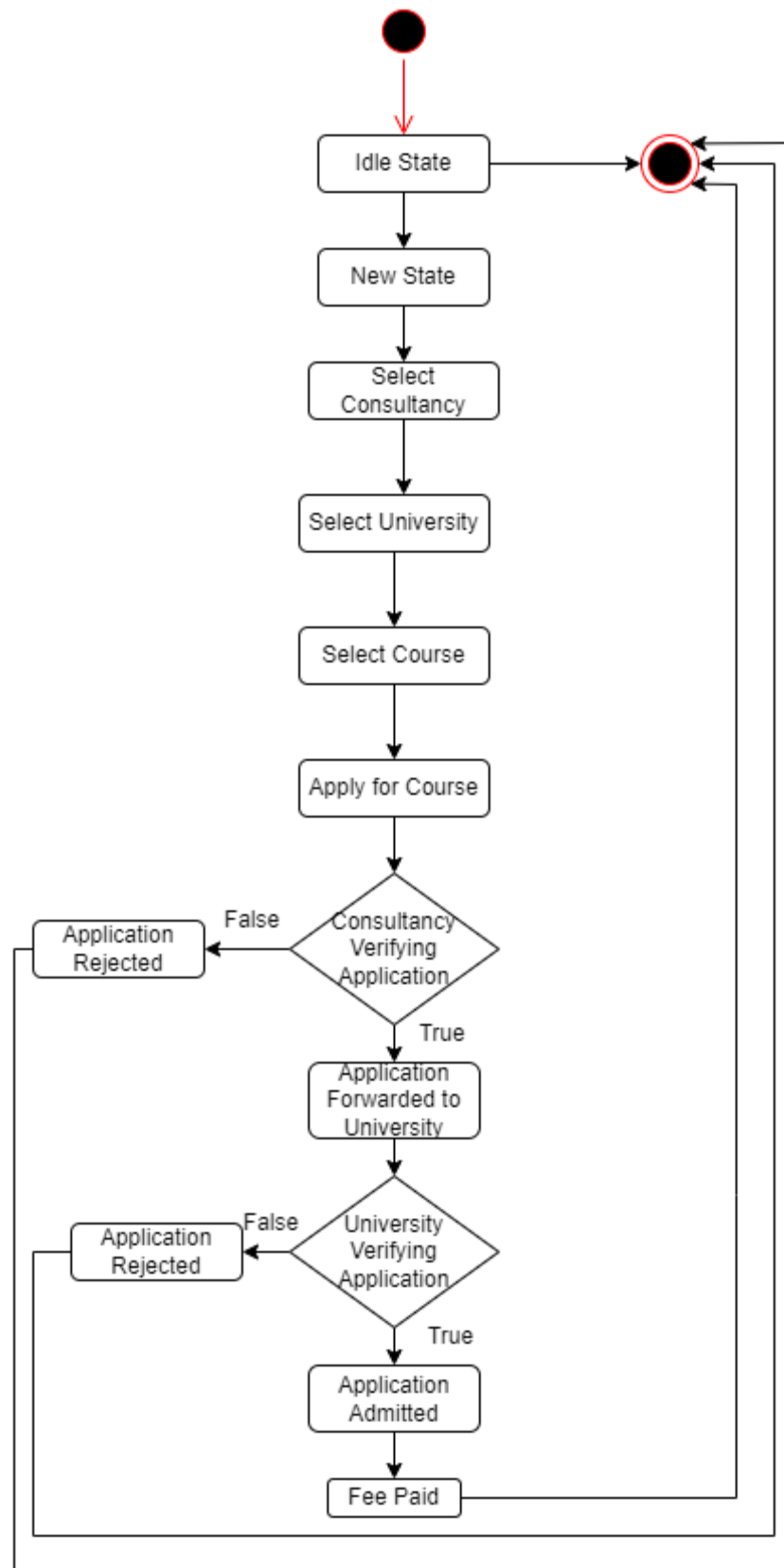
#### **Uses of sequence diagrams –**

- Used to model and visualize the logic behind a sophisticated function, operation or procedure.
- They are also used to show details of UML use case diagrams.
- Used to understand the detailed functionality of current or future systems.
- Visualize how messages and tasks move between objects or components in a system.



### **4.2.3 STATE CHART DIAGRAM**

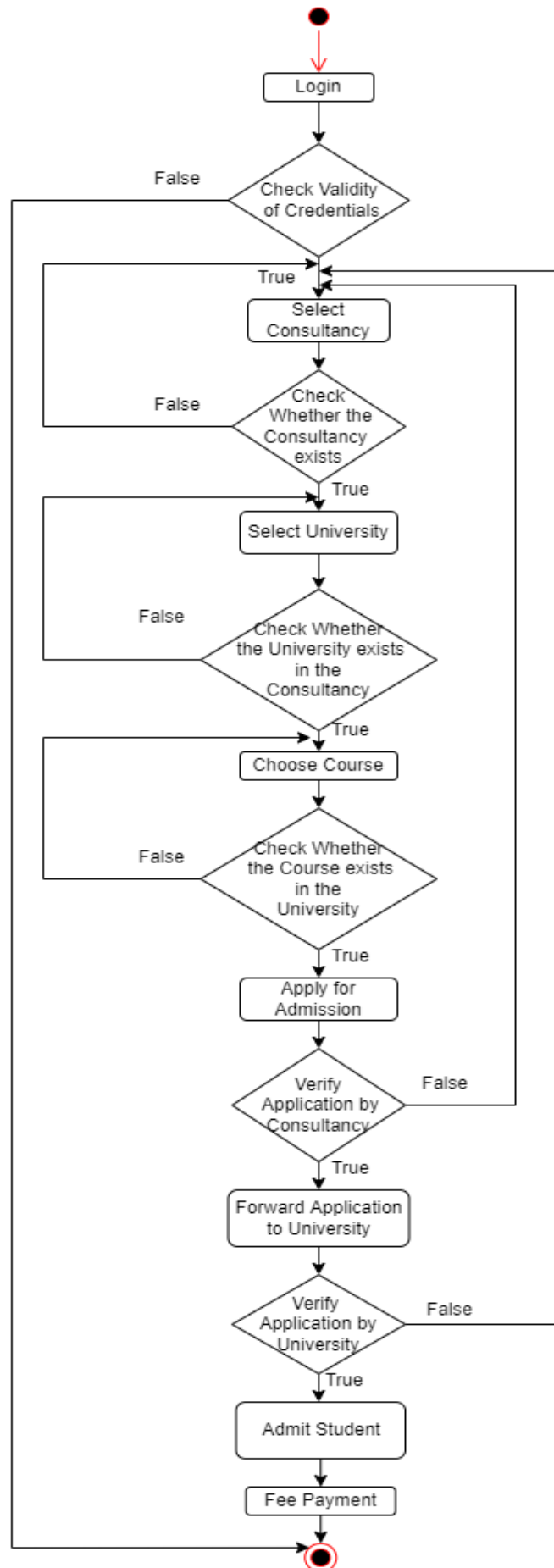
State diagrams are used to depict how a software system behaves. A class, a subsystem, a package, or even a complete system's behaviour can be represented by a state machine diagram in a UML model. State charts and state transition diagrams are other names for it. State chart diagrams give us a useful approach to represent the communications or interactions that take place between external entities and a system. The event-based system is modeled using these diagrams. With the aid of an event, a state of an object can be managed. To depict the multiple states of an entity within the application system, state chart diagrams are utilized.



#### **4.2.4 ACTIVITY DIAGRAM**

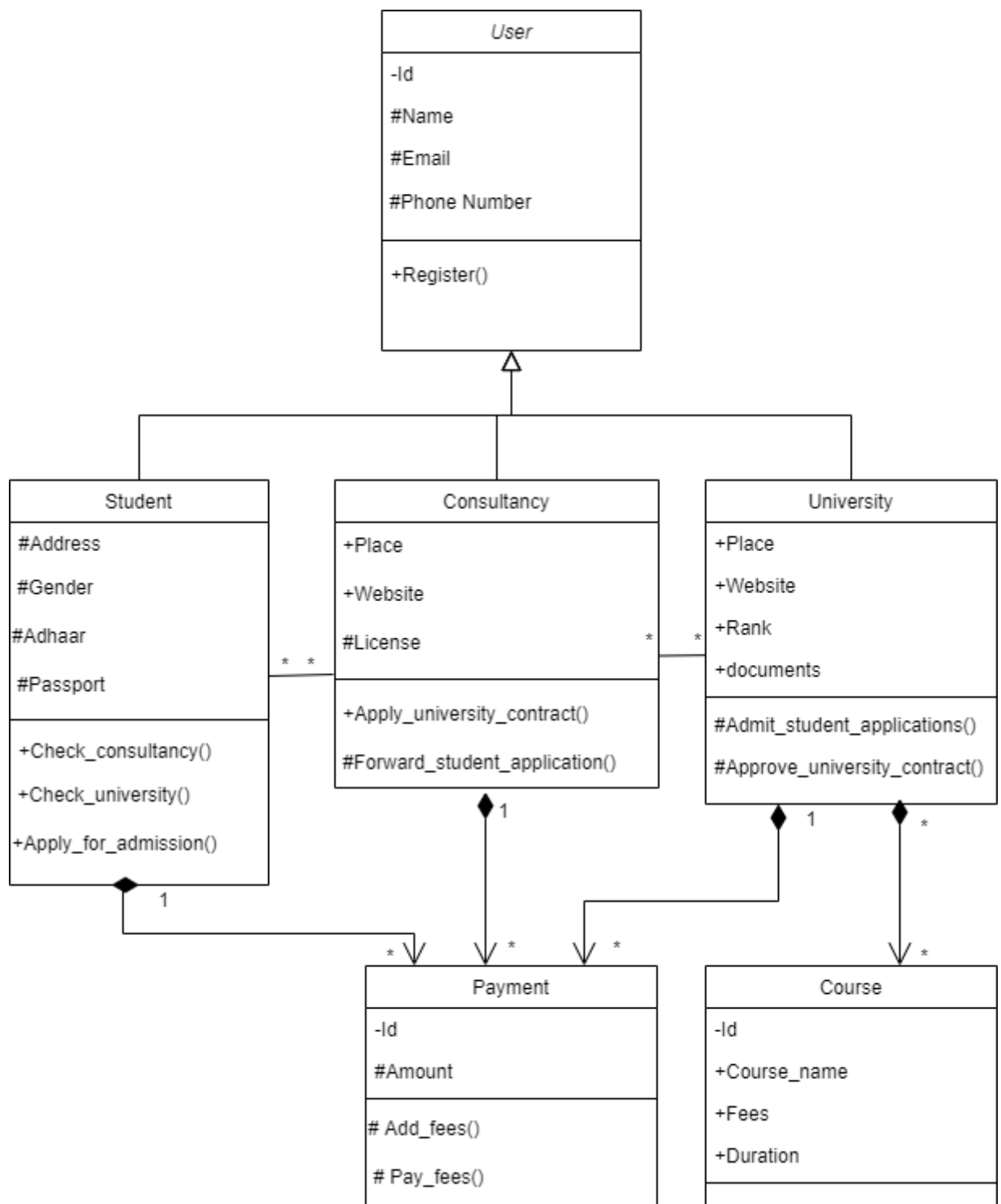
Activity diagrams show how multiple levels of abstraction of activities are coordinated to produce a service. Typically, an event must be accomplished by some operations, especially when the operation is meant to accomplish several different things that call for coordination. Another common requirement is how the events in a single use case relate to one another, especially in use cases where activities may overlap and require coordination. It is also appropriate for simulating how a set of related use cases operate together to reflect business workflows.





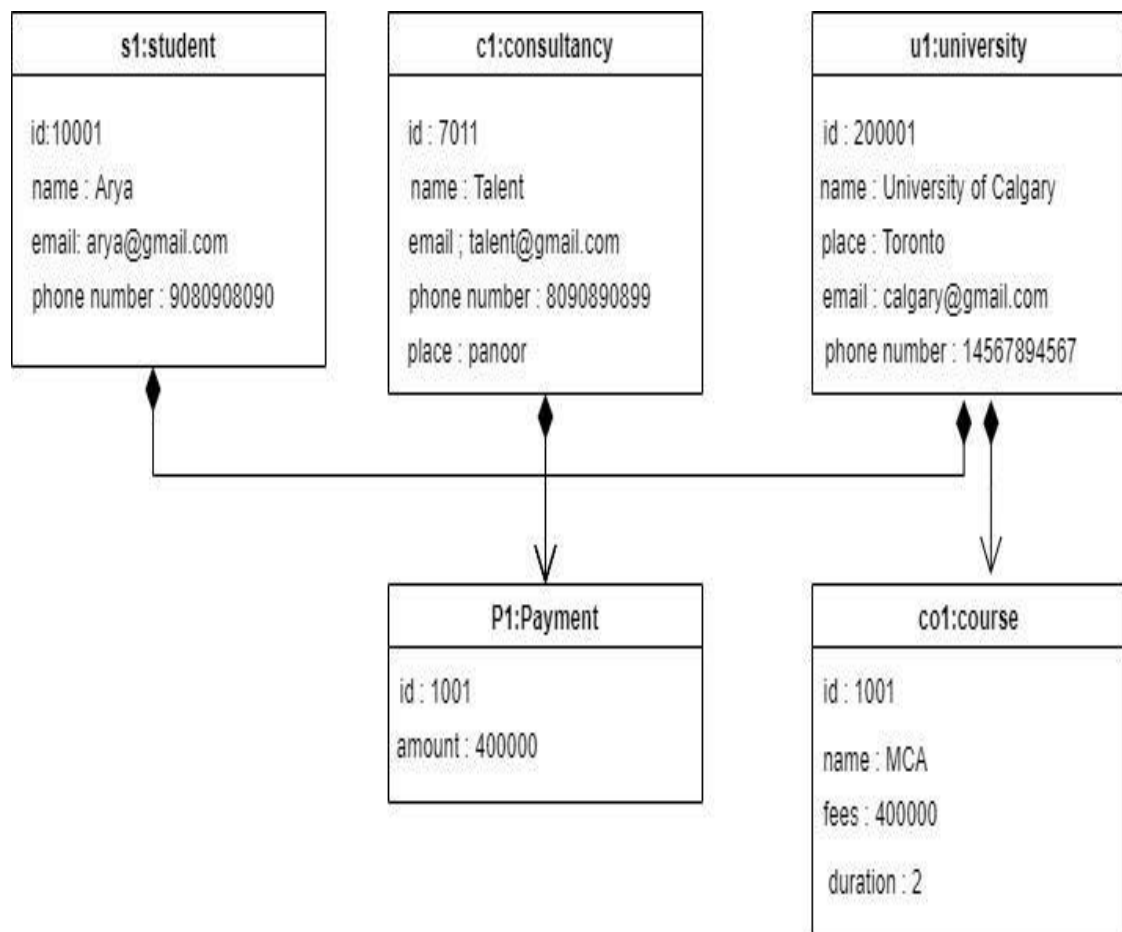
### **4.2.5 CLASS DIAGRAM**

Static diagrams include class diagrams. It represents the application's static view. Class diagrams are used to create executable code for software applications as well as for visualizing, explaining, and documenting various elements of systems. The characteristics and functions of a class are described in a class diagram, along with the restrictions placed on the system. Because they are the only UML diagrams that can be directly translated with object-oriented languages, class diagrams are extensively utilized in the modelling of object oriented systems. A collection of classes, interfaces, affiliations, collaborations, and constraints are displayed in a class diagram. It also goes by the name "structural diagram."



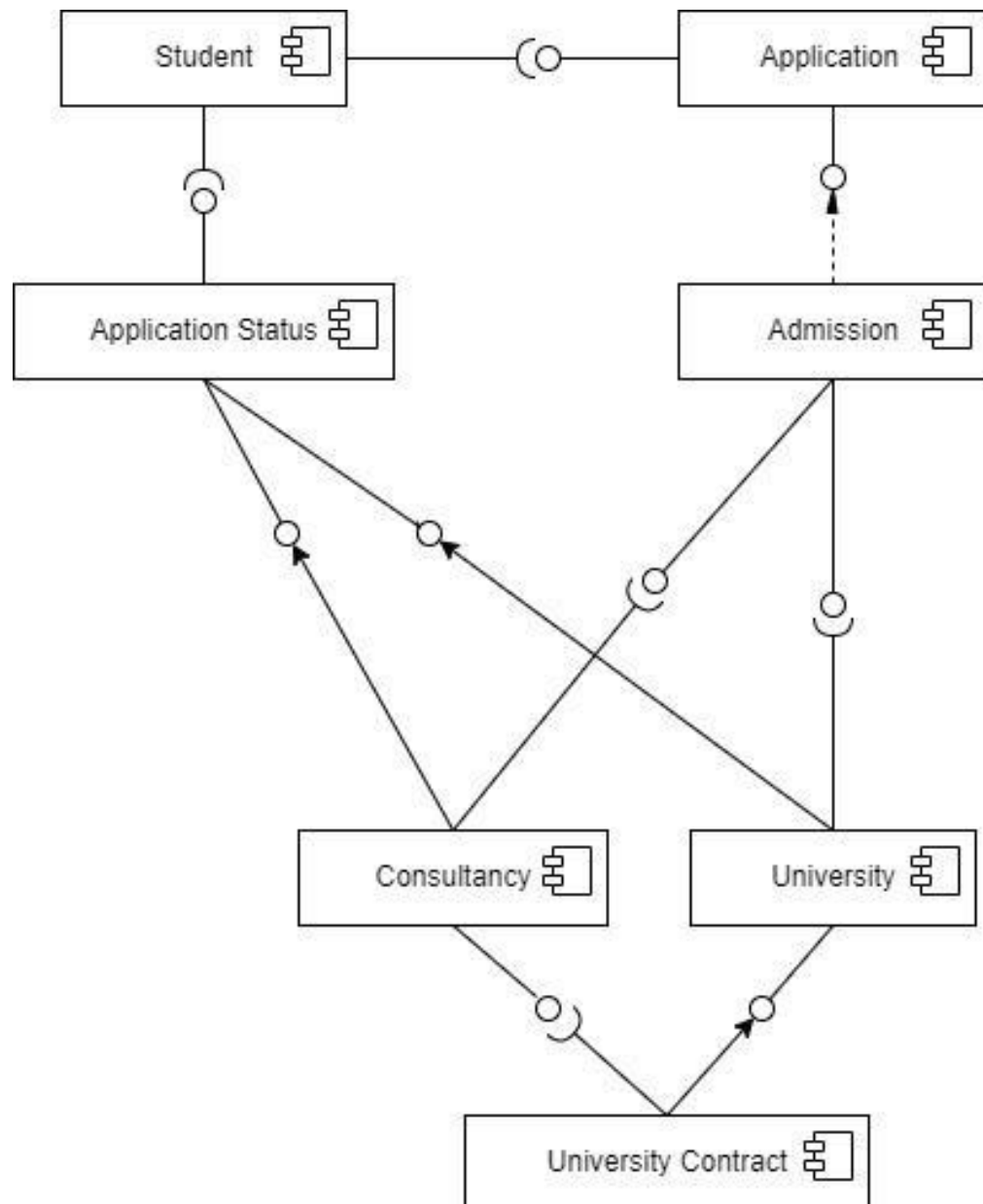
## 4.2.6 OBJECT DIAGRAM

Since class diagrams are the source of object diagrams, class diagrams are a prerequisite for object diagrams. An instance of a class diagram is represented by an object diagram. Class and object diagrams both use the same fundamental ideas. The static view of a system is also represented by object diagrams, but this static view represents a momentary snapshot of the system. Using a group of objects and their relationships as an example, object diagrams are employed.



### 4.2.7 COMPONENT DIAGRAM

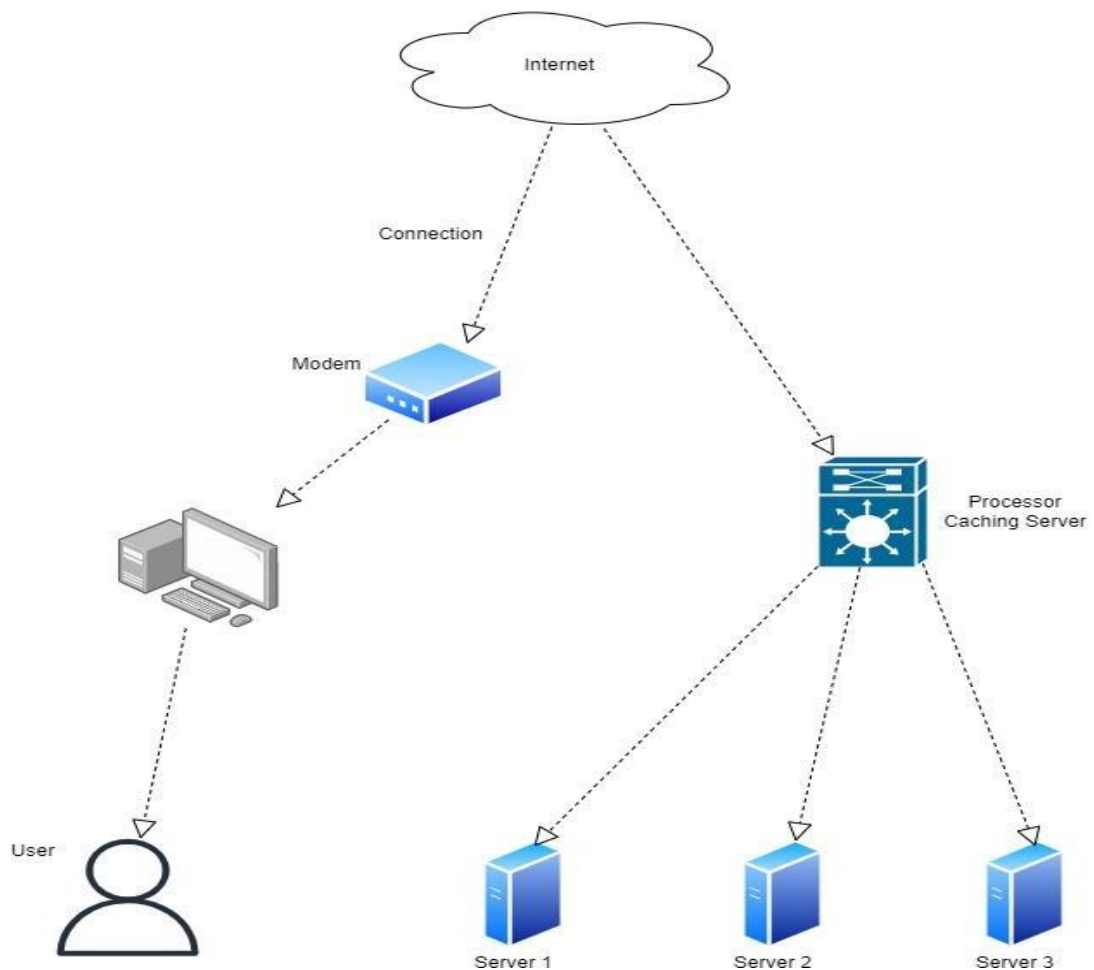
Component diagrams vary in their behaviour and character. Component diagrams are used to represent the system's physical components. The things that are physically present in a node include executable, libraries, files, documents, etc. The arrangement and connections between the components in a system are depicted using component diagrams. Systems that can be executed are also created using these diagrams.



### 4.2.8 DEPLOYMENT DIAGRAM

Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed. Deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.

Deployment Diagram



## 4.3 USER INTERFACE DESIGN

### 4.3.1-INPUT DESIGN

#### Student Registration Form



The registration form is displayed over a background image of a large, historic university building with a prominent clock tower. In the foreground, a large crowd of students is sitting on the grass, many with their arms raised. The form consists of the following fields and a submit button:

| Field Label   | Field Type    |
|---------------|---------------|
| FIRST NAME    | Text Input    |
| LAST NAME     | Text Input    |
| EMAIL         | Text Input    |
| PHONE NUMBER  | Text Input    |
| HOUSE NAME    | Text Input    |
| CITY          | Text Input    |
| PINCODE       | Text Input    |
| STATE         | Text Input    |
| QUALIFICATION | Text Input    |
| REGISTER      | Submit Button |

## Login Form



A woman with long dark hair, wearing a pink sweater and a blue headset, is sitting at a desk in a home office. She is looking at a computer monitor. Overlaid on the image is a login form with the following fields and buttons:

- SIGN UP (button)
- USERNAME (text input)
- PASSWORD (password input)
- USERTYPE (text input)
- LOGIN (button)

## Consultancy Registration Form



Three business professionals (two women and one man) are sitting around a table in a meeting room, looking at a laptop. Overlaid on the image is a consultancy registration form with the following fields and buttons:

- NAME (text input)
- LICENCE NO. (text input)
- LOCATION (text input)
- EMAIL (text input)
- PHONE NUMBER (text input)
- REGISTER (button)



## 4.3.2 OUTPUT DESIGN

### Student Module

#### Application Form

The screenshot shows a web browser window with the URL `localhost/candream_ci/index.php/Welcome/applyForAdmission1`. The page title is "Candream". The navigation bar includes links: Home, Consultancy, Apply for Admission, Application Status, Applications, and Logout. The user is logged in as `aparna@gmail.com`. The main content area is titled "APPLY FOR ADMISSION" and features a form with the following fields:

- Course:
- University:
- Consultancy:

A green "Save" button is located below the form. The background of the page shows a blurred image of a school building and children playing in a courtyard.

#### University Contract Request

The screenshot shows a web browser window with the URL `localhost/candream_ci/index.php/Consultancycontroller/applyForUniversityContract`. The page title is "Candream". The navigation bar includes links: Home, Student Applications, University Contract, and Logout. The user is logged in as `talent@gmail.com`. The main content area is titled "University Request" and features a form with the following fields:

- University:
- Request:

Below the form is a table with the following data:

| University               | Status   |
|--------------------------|----------|
| Royal Roads University   | Pending  |
| University of Victoria   | Accepted |
| University of Lethbridge | Accepted |
| University of Alberta    | Accepted |
| University of Calgary    | Rejected |

## 4.6. DATABASE DESIGN

A database is a structured system with the capacity to store information and allows users to retrieve stored information quickly and effectively. Any database's primary goal is its data, which demands protection.

There are two stages to the database design process. The user needs are obtained in the first step, and a database is created to as clearly as possible meet these criteria. This process, known as information level design, is carried out independently of all DBMSs.

The design for the specific DBMS that will be used to construct the system in issue is converted from an information level design to a design in the second stage. Physical Level Design is the stage where the characteristics of the particular DBMS that will be used are discussed. Parallel to the system design is a database design. The database's data arrangement aims to accomplish the two main goals listed below.

- Data Integrity
- Data independence

### 4.6.1 Relational Database Management System (RDBMS)

In a relational model, the database is shown as a set of relations. Each relation resembles a file or table of records with values. A row is referred to as a tuple, a column heading is referred to as an attribute, and the table is referred to as a relation in formal relational model language. A relational database is made up of a number of tables, each with its own name. Each row represents a group of associated values.

#### Relations, Domains & Attributes

A relation is a table. Tuples are the units of a table's rows. An ordered group of  $n$  elements is a tuple. Attributes are referred to as columns. Every table in the database has relationships already established between them. This guarantees the integrity of both referential and entity relationships. A group of atomic values make up a domain  $D$ . Choosing a data type from which the domain's data values are derived is a typical way to define a domain. To make it easier to understand the values of the domain, it is also helpful to give it a name. Each value in a relation is atomic and cannot be broken down.

## Relationships

- Key is used to create table relationships. Primary Key and Foreign Key are the two principal keys that are most crucial. With the use of these keys, relationships for entity integrity and referential integrity can be created.
- Entity Integrity forbids the use of null values for any Primary Key.
- It is required by referential integrity that no Primary Key have null values.
- Referential Integrity There must be a Primary Key value in the same domain for each unique Foreign Key value. Super Key and Candidate Keys are additional keys.

### 4.6.2 Normalization

The simplest possible grouping of data is used to put them together so that future changes can be made with little influence on the data structures. The formal process of normalizing data structures in a way that reduces duplication and fosters integrity. Using the normalization technique, superfluous fields are removed and a huge table is divided into several smaller ones. Anomalies in insertion, deletion, and updating are also prevented by using it. Keys and relationships are two notions used in the standard form of data modelling. A row in a table is uniquely identified by a key. Primary keys and foreign keys are the two different kinds of keys.. A primary key is an element, or set of components, in a table that serves as a means of distinguishing between records from the same table. A column in a table known as a foreign key is used to uniquely identify records from other tables. Up to the third normal form, all tables have been normalized.

It means placing things in their natural form, as the name suggests. By using normalization, the application developer aims to establish a coherent arrangement of the data into appropriate tables and columns, where names may be quickly related to the data by the user. By removing recurring groups from the data, normalization prevents data redundancy, which puts a heavy strain on the computer's resources. These consist of:

- Normalize the data.
- Select appropriate names for the columns and tables.
- Select a suitable name for the data.

**First Normal Form**

According to the First Normal Form, each attribute's domain must only include atomic values, and each attribute's value in a tuple must be a single value from that domain. In other words, 1NF forbids using relationships as attribute values within tuples or relations within relations. Single atomic or indivisible values are the only attribute values that are permitted under 1NF. The data must first be entered into First Normal Form. This can be achieved by separating data into tables of a similar type in each table. Depending on the needs of the project, a Primary Key or Foreign Key is assigned to each table. For each nested relation or non-atomic attribute, new relations are formed in this process.

**Second Normal Form**

No non-key attribute should be functionally dependent on a portion of the main key for relations where the primary key has several attributes, according to Second Normal Form. This involves breaking down each partial key into its dependent characteristics and setting up a new relation for each one. Keep the original primary key and any properties that are entirely dependent on it in your database. This procedure aids in removing data that depends only on a small portion of the key. If a connection meets all the requirements for first normal form for the main key and every non-primary key feature of the relation is completely dependent on its primary key alone, then and only then is the relation said to be in second normal form.

**Third Normal Form**

Relation should not have a non-key attribute that is functionally determined by another non-key attribute or by a collection of non-key attributes, according to the Third Normal Form. The primary key should not be transitively dependent, in other words. The non-key attributes that functionally determine other non-key attributes are decomposed in this way put up in relation. This procedure is used to eliminate anything not wholly dependent on the Primary Key. Only when a relation is in second normal form and, more importantly, when its non-key characteristics do not depend on those of other non-key attributes, is it considered to be in third normal form.

**TABLE DESIGN****1.tbl\_login**

| Field     | Data type   | Constraints | Description                  |
|-----------|-------------|-------------|------------------------------|
| login_id  | Integer     | Primary key | Login id of user             |
| user_type | Varchar(20) | Not null    | Type of user                 |
| Username  | Varchar(20) | Not null    | Username of user             |
| Password  | Password    | Not null    | Password of user             |
| reffer_id | Integer     | Foreign key | Id of user from user's table |
| Status    | Boolean     | Not null    | Status of user               |

**2.tbl\_university**

| Field           | Data type    | Constraints | Description                                    |
|-----------------|--------------|-------------|------------------------------------------------|
| university_id   | Integer      | Primary Key | Id of university                               |
| university_name | Varchar(30)  | Not Null    | Name of university                             |
| Place           | Varchar(30)  | Not Null    | Place of university                            |
| Areacode        | Varchar(11)  | Not Null    | Email of university                            |
| Phone           | Varchar(11)  | Not Null    | Phone number of university                     |
| Email           | Varchar(30)  | Not Null    | Email of university                            |
| Website         | Varchar(30)  | Null        | Website of university                          |
| Remarks         | Varchar(100) | Null        | Remarks about university                       |
| Rank            | Integer      | Null        | World Ranking of universities                  |
| Documents       | Varchar(300) | Not Null    | Essential documents for verifying universities |
| Status          | Boolean      | Not Null    | Status of university                           |

**3.tbl\_consultancy**

| Field            | Data type    | Constraints | Description                                   |
|------------------|--------------|-------------|-----------------------------------------------|
| consultancy_id   | Integer      | Primary key | Id of consultancy                             |
| consultancy_name | Varchar(30)  | Not null    | Name of Consultancy                           |
| Place            | Varchar(30)  | Not null    | Location of consultancy                       |
| Pin              | Varchar(7)   | Not null    | Pin code of Consultancy                       |
| email_id         | Varchar(30)  | Not null    | Email id of Consultancy                       |
| phone_no         | Varchar(11)  | Not null    | Phone number of consultancy                   |
| Website          | Varchar(30)  | Not null    | Website of Consultancy                        |
| Remarks          | Varchar(100) | Null        | Remarks about Consultancy                     |
| Documents        | Varchar(300) | Not null    | Essential documents for verifying Consultancy |
| Status           | Boolean      | Not null    | Status of Consultancy                         |

**4.tbl\_course**

| Field       | Data type   | Constraints | Description    |
|-------------|-------------|-------------|----------------|
| Course_id   | Integer     | Primary key | Id of course   |
| Course_name | Varchar(30) | Not null    | Name of course |

**5.tbl\_university\_courses**

| Field                | Data type   | Constraints | Description                      |
|----------------------|-------------|-------------|----------------------------------|
| university_course_id | Integer     | Primary key | Id of University Course          |
| course_id            | Integer     | Foreign key | Id of Course(tbl_course)         |
| Fees                 | Varchar(20) | Not null    | Fees for Course                  |
| Duration             | integer     | Not null    | Duration of the Course           |
| university_id        | Varchar(20) | Foreign key | Id of university(tbl_university) |
| Status               | Boolean     | Not null    | Status of University Courses     |

**6.tbl\_university\_contract**

| Field          | Data type | Constrain   | Description                   |
|----------------|-----------|-------------|-------------------------------|
| contract_id    | Integer   | Primary key | Education id of Student       |
| cunsultancy_id | Integer   | Foreign key | Id of Student(tb_student)     |
| university_id  | Integer   | Not null    | Id of Course(tb_university)   |
| Status         | Integer   | Not null    | Status of University Contract |

**7.tbl\_admission**

| Field          | Data type   | Constrains  | Description                          |
|----------------|-------------|-------------|--------------------------------------|
| application_id | Integer     | Primary key | Id of Application                    |
| student_id     | Integer     | Foreign key | Id of Student(tbl_login)             |
| consultancy_id | Integer     | Foreign key | Id of Consultancy(tbl_consultancy)   |
| university_id  | Integer     | Foreign key | Id of University(tbl_university)     |
| course_id      | Integer     | Foreign key | Id of Course(tbl_course)             |
| first_name     | Varchar(20) | Not null    | First name of Applicant              |
| middle_name    | Varchar(20) | Not null    | Middle name of Applicant             |
| last_name      | Varchar(20) | Not null    | Last name of Applicant               |
| birth_date     | Date        | Not null    | Birth date of Applicant              |
| Gender         | Varchar(20) | Null        | Gender of Applicant                  |
| Nationality    | Varchar(20) | Not null    | Nationality of Applicant             |
| State          | Varchar(20) | Not null    | State of Applicant                   |
| City           | Varchar(20) | Not null    | City of Student                      |
| Pin            | Varchar(11) | Not null    | Pin of Student                       |
| Phone          | Varchar(11) | Not null    | Phone number of Student              |
| Email          | Varchar(30) | Not null    | Email of Student                     |
| Guardian       | Varchar(30) | Not null    | Name of Gaurdian of Student          |
| document_id    | Integer     | Foreign key | Id of Student Document(tbl_document) |
| Status         | Boolean     | Not null    | Status of student Applications       |



**8.tbl\_document**

| Field                | Data type    | Constrain   | Description                              |
|----------------------|--------------|-------------|------------------------------------------|
| Document_id          | Integer      | Primary key | Id of student document                   |
| Photo                | Varchar(111) | Not null    | Photo of Student                         |
| Signature            | Varchar(111) | Not null    | Signature of Student                     |
| Transcripts          | Varchar(111) | Not null    | Transcripts of Student                   |
| Ielts                | Varchar(111) | Not null    | Ielts Score Card of Student              |
| Resume               | Varchar(111) | Not null    | Resume of Student                        |
| Educred              | Varchar(111) | Not null    | Education Credentials of Student         |
| Graducertificate     | Varchar(111) | Not null    | Graduation Certificate of Student        |
| Passport             | Varchar(111) | Not null    | Passport of Student                      |
| visa/eta             | Varchar(111) | Not null    | Visa or ETA(Estimated Time of Arrival)   |
| Adhar                | Varchar(111) | Not null    | Adhaar Card of Student                   |
| Pan                  | Varchar(111) | Not null    | Pan Card of Student                      |
| bank_passbook        | Varchar(111) | Not null    | Bank Passbook of Student                 |
| police_clearance     | Varchar(111) | Not null    | Police Clearance Certificate             |
| self_declaration     | Varchar(111) | Not null    | Self Declaration                         |
| Medical_document     | Varchar(111) | Not null    | Medical Clearance Certificate of Student |
| Prooffund            | Varchar(111) | Not null    | Proof of Fund for Students               |
| Vaccine              | Varchar(111) | Not null    | Vaccine Certificate of Student           |
| reason_fro_rejection | Varchar(50)  | Null        | Reason for Rejection                     |
| applied_date         | timestamp    | Not null    | Application date                         |
| Status               | Boolean      | Not null    | Status of student                        |

---

**9.tbl\_payment**

| Field            | Data type | Constrain   | Description                       |
|------------------|-----------|-------------|-----------------------------------|
| payment_id       | Integer   | Primary key | Payment id of Applicant           |
| admission_id     | Integer   | Foreign key | Id of Application( tb_admission)  |
| amount           | Integer   | Not null    | Fees added to Student             |
| transaction_id   | Integer   | Not null    | Id of Transaction                 |
| student_paid     | Integer   | Not null    | Whether Student paid the fees     |
| Consultancy_paid | Integer   | Not null    | Whether Consultancy paid the fees |
| Created_time     | Timestamp | Not null    | Time and date of payment          |
| status           | Integer   | Not null    | Status of the payment             |

## **CHAPTER 5**

### **SYSTEM TESTING**

## 5.1 INTRODUCTION

Software testing is the process of carefully controlling the execution of software in order to determine whether it behaves as intended. The words verification and validation are frequently used in conjunction with software testing. Validation is the process of examining or evaluating a product, including software, to determine whether it complies with all relevant specifications. Software testing is merely one type of verification; others include reviews, analyses, inspections, and walk through. Verifying that what has been specified matches what the user truly desired is the process of validation.

Other activities which are often associated with software testing are static analysis and dynamic analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without actually executing the code. Dynamic analysis looks at the behaviour of software while it is executing, to provide information such as execution traces, timing profiles, and test coverage information.

Testing is a collection of activities that can be planned ahead of time and carried out in a methodical manner. Testing starts with individual modules and progresses to the integration of the full computer-based system. There are many rules that can be used as testing objectives, and testing is necessary for the system testing objectives to be successful. As follows: A programme is tested by being run with the goal of identifying any errors.

- A excellent test case is one that has a good chance of revealing a previously unrecognized error.
- A test that finds an error that has not yet been found is successful.

If a test is successfully carried out in accordance with the aforementioned aims, it will reveal software bugs. Additionally, testing shows that the software functions seem to operate in accordance with the specification and that the performance requirements seem to have been satisfied. There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

Testing for correctness is meant to ensure that a programme performs exactly as it was intended to. This is much harder than it might initially seem, especially for big programmes.

---

## 5.2 TEST PLAN

A test plan suggests a number of required steps that need be taken in order to complete various testing methodologies. The activity that is to be taken is outlined in the test plan. A computer programme, its documentation, and associated data structures are all created by software developers. It is always the responsibility of the software developers to test each of the program's separate components to make sure it fulfills the purpose for which it was intended. In order to solve the inherent issues with allowing the builder evaluate what they have developed, there is an independent test group (ITG). Testing's precise goals should be laid forth in quantifiable language. The test strategy should include information on the mean time to failure, cost to detect and correct problems, remaining defect density or frequency of occurrence, and test work hours per regression test. The testing levels include:

- Unit testing
- Integration Testing
- Data validation Testing
- Output Testing

### 5.2.1 Unit Testing

Unit testing concentrates verification efforts on the software component or module, which is the smallest unit of software design. The component level design description is used as a guide when testing crucial control paths to find faults inside the module's perimeter. the level of test complexity and the untested area determined for unit testing. Unit testing is white-box focused, and numerous components may be tested simultaneously. To guarantee that data enters and exits the software unit under test properly, the modular interface is tested. To make sure that data temporarily stored retains its integrity during each step of an algorithm's execution, the local data structure is inspected. To confirm that each statement in a module has been executed at least once, boundary conditions are evaluated. Finally, each path for managing errors is examined.

Before starting any other test, tests of data flow over a module interface are necessary. All other tests are irrelevant if data cannot enter and depart the system properly. An important duty during the unit test is the selective examination of execution pathways. Error circumstances must be foreseen in good design, and error handling paths must be put up to cleanly reroute or halt work when an error does arise. The final step of unit testing is boundary testing.

---

Software frequently fails at its limits. In the Sell-Soft System, unit testing was carried out by treating each module as a distinct entity and subjecting them to a variety of test inputs. The internal logic of the modules had some issues, which were fixed. Each module is tested and run separately after coding. All unused code was eliminated, and it was confirmed that every module was functional and produced the desired outcome.

### **5.2.2 Integration Testing**

Integration testing is a methodical approach for creating the program's structure while also carrying out tests to find interface issues. The goal is to construct a programme structure that has been determined by design using unit tested components. The programme as a whole is tested. Correction is challenging since the size of the overall programme makes it challenging to isolate the causes. As soon as these mistakes are fixed, new ones arise, and the process repeats itself in an apparently unending cycle. All of the modules were integrated after unit testing was completed in the system to check for any interface inconsistencies. A distinctive programme structure also developed when discrepancies in programme structures were eliminated.

### **5.2.3 Validation Testing or System Testing**

The testing process comes to an end here. This involved testing the entire system in its entirety, including all forms, code, modules, and class modules. Popular names for this type of testing include system tests and black box testing.

The functional requirements of the software are the main emphasis of the black box testing approach. That example, using Black Box testing, a software engineer can create sets of input conditions that will fully test every programme requirement.

The following sorts of problems are targeted by black box testing: erroneous or missing functions, interface errors, data structure or external data access errors, performance errors, initialization errors, and termination errors.

### **5.2.4 Output Testing or User Acceptance Testing**

User approval of the system under consideration is tested; in this case, it must meet the needs of the company. When developing, the programme should stay in touch with the user and perspective system to make modifications as needed. With regard to the following points, this is done:

- Input Screen Designs,
- Output Screen Designs,

The aforementioned testing is carried out using a variety of test data. The preparation of test data is essential to the system testing process. The system under investigation is then put to the test using the prepared test data. While testing the system, faults in the test data are once again found and fixed using the above testing procedures. The fixes are also noted for later use.

### **5.2.5 Selenium Testing**

An open-source programme called Selenium automates web browsers. A single interface can be used to create test scripts in a number of programming languages, including Ruby, Java, Node JS, PHP, Perl, Python, and C#. Web application cross-browser compatibility testing is automated by the Selenium testing tool. It is used to make that web apps, whether they are responsive, progressive, or traditional, are of a high calibre. Selenium is open-source software.

| Project Name: Candream - Student Immigration Support System |                                      |                                       |                                  |                                                        |                    |
|-------------------------------------------------------------|--------------------------------------|---------------------------------------|----------------------------------|--------------------------------------------------------|--------------------|
| Login Test Case                                             |                                      |                                       |                                  |                                                        |                    |
| Test Case ID: Fun_1                                         |                                      |                                       | Test Designed By: Amala Mariyat  |                                                        |                    |
| Test Priority<br>(Low/Medium/High): High                    |                                      |                                       | Test Designed Date:19-07-2022    |                                                        |                    |
| Module Name: Login Screen                                   |                                      |                                       | Test Executed By: Ms. Anit James |                                                        |                    |
| Test Title: Verify login with valid username and password   |                                      |                                       | Test Execution Date: 20-07-2022  |                                                        |                    |
| Description: Test the Login Page                            |                                      |                                       |                                  |                                                        |                    |
| Pre-Condition: User has valid username and password         |                                      |                                       |                                  |                                                        |                    |
| Step                                                        | Test Step                            | Test Data                             | Expected Result                  | Actual Result                                          | Status (Pass/Fail) |
| 1                                                           | Navigation to Login Page             |                                       | Login Page should be displayed   | Login page displayed                                   | Pass               |
| 2                                                           | Provide Valid username               | Username :<br>aparna@gmail.com        | User should be able to Login     | User Logged in and navigated to User Dashboard         | Pass               |
| 3                                                           | Provide Valid Password               | Password:1234@Asdf                    |                                  |                                                        |                    |
| 4                                                           | Click on Sign In button              |                                       |                                  |                                                        |                    |
| 5                                                           | Provide Invalid username or password | Username : aparna<br>Password:1234sdf | User should not be able to Login | Message for enter valid email id or Password displayed | Pass               |
| 6                                                           | Provide Null username or Password    | Username : null<br>Password: null     |                                  |                                                        |                    |
| 7                                                           | Click on Sign In button              |                                       |                                  |                                                        |                    |



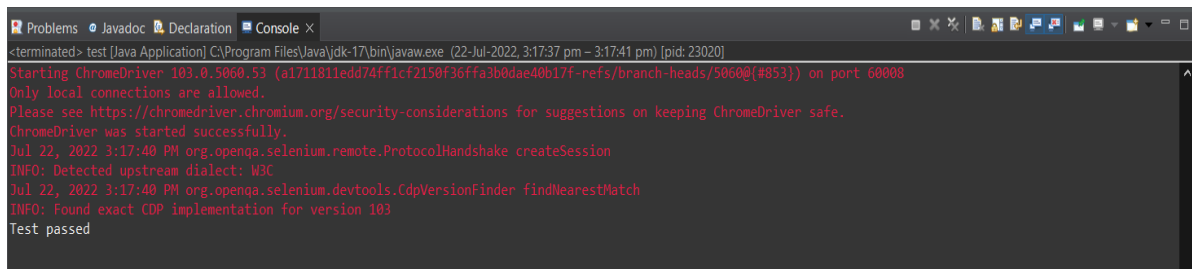
**Post-Condition:** User is validated with database and successfully login into account. The account session details are logged in database.

### Code package

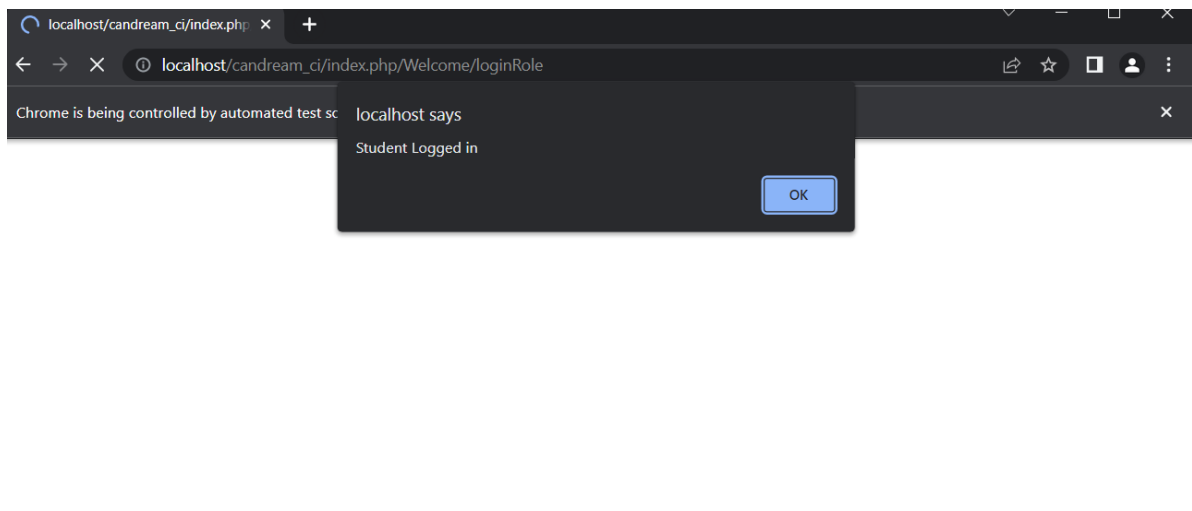
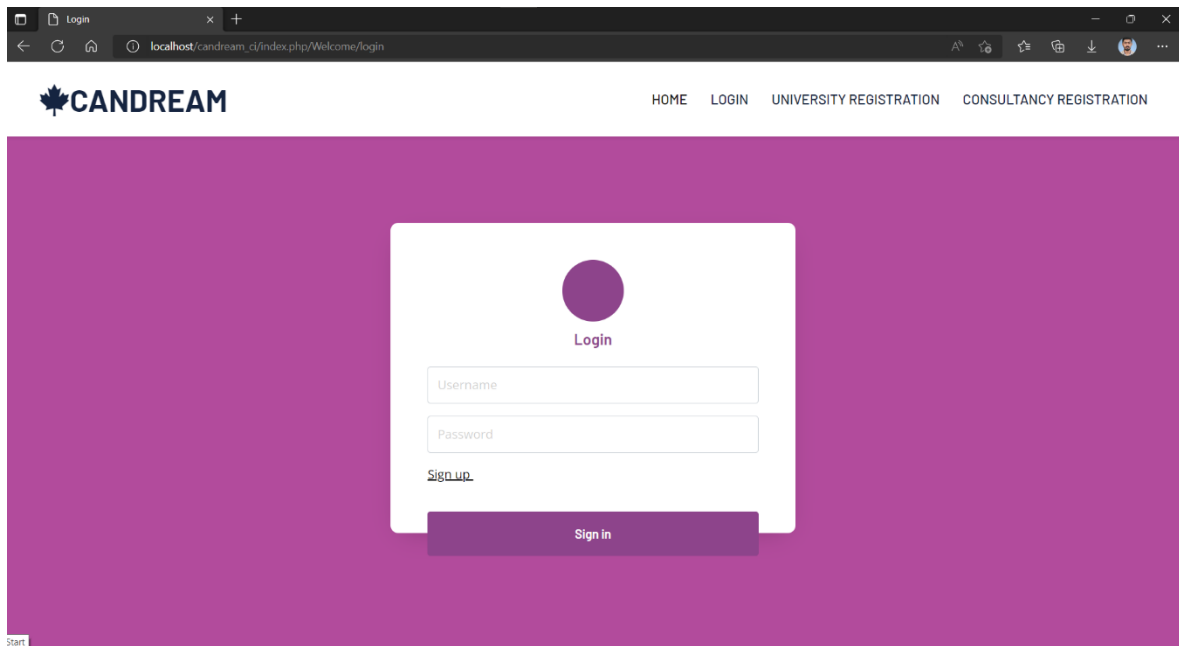
```
package seleniumtest;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class test {
    public static void main(String[] args) {

        System.setProperty("webdriver.chrome.driver", "C:\\Users\\jenso\\Downloads\\chromedriver_w
in32\\chromedriver.exe" );
        WebDriver driver=new ChromeDriver();
        driver.get("http://localhost/candream_ci/index.php/Welcome/login");
        driver.findElement(By.id("username")).sendKeys("aparna@gmail.com");
        driver.findElement(By.id("pass")).sendKeys("1234@Asdf");
        driver.findElement(By.id("submit")).click();
        if(actualUrl.equalsIgnoreCase(expectedUrl)) {
            System.out.println("Test passed");
        } else {
            System.out.println("Test failed");
        }
    }
}
```

## Screenshots



```
<terminated> test [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (22-Jul-2022 3:17:37 pm - 3:17:41 pm) [pid: 23020]
Starting ChromeDriver 103.0.5060.53 (a1711811edd74ff1cf2150f36ffa3b0dae40b17f-refs/branch-heads/50600@#853)) on port 60008
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Jul 22, 2022 3:17:40 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected upstream dialect: W3C
Jul 22, 2022 3:17:40 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found exact CDP implementation for version 103
Test passed
```



| Project Name: Candream - Student Immigration Support System |                                   |                                                                 |                                       |                                                      |                    |
|-------------------------------------------------------------|-----------------------------------|-----------------------------------------------------------------|---------------------------------------|------------------------------------------------------|--------------------|
| University Registration Test Case                           |                                   |                                                                 |                                       |                                                      |                    |
| Test Case ID: Fun_11                                        |                                   |                                                                 | Test Designed By: Amala Mariyat       |                                                      |                    |
| Test Priority (Low/Medium/High): High                       |                                   |                                                                 | Test Designed Date:20-07-2022         |                                                      |                    |
| Module Name: Login Screen                                   |                                   |                                                                 | Test Executed By: Ms. Anit James      |                                                      |                    |
| Test Title: Verify University Details                       |                                   |                                                                 | Test Execution Date: 21-07-2022       |                                                      |                    |
| Description: Test the University Registration Page          |                                   |                                                                 |                                       |                                                      |                    |
| Pre-Condition: University has valid details                 |                                   |                                                                 |                                       |                                                      |                    |
| Step                                                        | Test Step                         | Test Data                                                       | Expected Result                       | Actual Result                                        | Status (Pass/Fail) |
| 1                                                           | Navigation to Login Page          |                                                                 | Login Page should be displayed        | Login page displayed                                 | Pass               |
| 2                                                           | Provide Valid University Name     | Name: University of Calgary                                     | University should be able to Register | University Registered in and navigated to User Login | Pass               |
| 3                                                           | Provide Valid Place of University | Place: Toronto                                                  |                                       |                                                      |                    |
| 4                                                           | Provide Valid Area Code           | Area code: 234                                                  |                                       |                                                      |                    |
| 5                                                           | Provide Valid Phone Number        | Phone: +16727373465                                             |                                       |                                                      |                    |
| 6                                                           | Provide Valid Email               | Email: <a href="mailto:calgary@gmail.com">calgary@gmail.com</a> |                                       |                                                      |                    |
| 7                                                           | Provide Valid Website             | Website: <a href="http://www.calgary.com">www.calgary.com</a>   |                                       |                                                      |                    |
| 8                                                           | Provide Valid Remarks             | Remarks: Good                                                   |                                       |                                                      |                    |
| 9                                                           | Provide Valid Rank                | Rank: 43                                                        |                                       |                                                      |                    |
| 10                                                          | Provide Valid Password            | Password:1234@Asdf                                              |                                       |                                                      |                    |
| 11                                                          | Provide Password Again            | Confirm Password:1234@Asdf                                      |                                       |                                                      |                    |

|   |                                   |                                                  |                                     |                                                        |      |
|---|-----------------------------------|--------------------------------------------------|-------------------------------------|--------------------------------------------------------|------|
| 4 | Click on Register button          |                                                  |                                     |                                                        |      |
| 5 | Provide Invalid details           | University Name: ABC University<br>Place: Aparna | User should not be able to Register | Message for enter valid email id or Password displayed | Pass |
| 6 | Provide Null username or Password | University Name: null<br>Place: null             |                                     |                                                        |      |
| 7 | Click on Sign In button           |                                                  |                                     |                                                        |      |

**Post-Condition:** User is validated with database and successfully login into account.

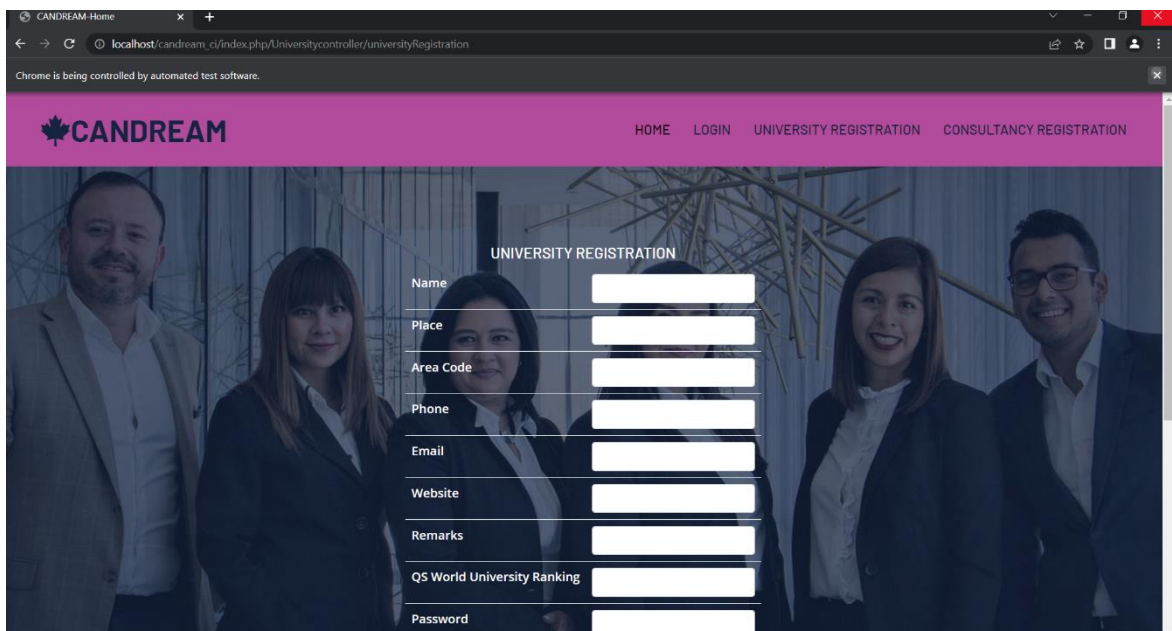
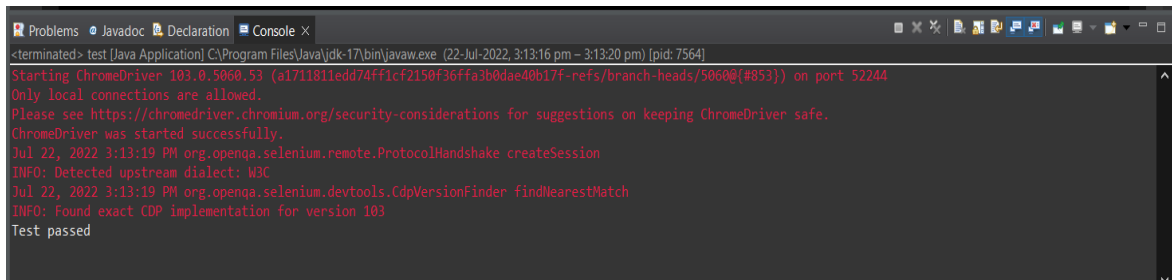
The account session details are logged in database.

### Code package

```
package seleniumtest;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class test {
public static void main(String[] args) {
    System.setProperty("webdriver.chrome.driver","C:\\\\Users\\jenso\\Downloads\\chromedriver_win32\\chromedriver.exe" );
    WebDriver driver=new ChromeDriver();
    driver.get("http://localhost/candream_ci/index.php/Universitycontroller/universityRegistration");
    driver.findElement(By.id("name")).sendKeys("Amala");
    driver.findElement(By.id("place")).sendKeys("Kannur");
    driver.findElement(By.id("area")).sendKeys("686");
    driver.findElement(By.id("phone")).sendKeys("9876543213");
    driver.findElement(By.id("email")).sendKeys("Amala@gmail.com");
    driver.findElement(By.id("website")).sendKeys("www.amalamariyat.com");
    driver.findElement(By.id("remarks")).sendKeys("good");
    driver.findElement(By.id("rank")).sendKeys("12");
    driver.findElement(By.id("pass")).sendKeys("Amala@1234");
```

```
driver.findElement(By.id("repass")).sendKeys("Amala@1234");
driver.findElement(By.id("submit")).click();
String actualUrl="http://localhost/candream_ci/index.php/Welcome/login";
String expectedUrl= driver.getCurrentUrl();
if (actualUrl.equalsIgnoreCase(expectedUrl)) {
    System.out.println("Test passed");
} else {
    System.out.println("Test failed");
}
}
```

## Screenshots



## **CHAPTER 6**

### **IMPLEMENTATION**

## 6.1 INTRODUCTION

The project's implementation phase is where the conceptual design is transformed into a functional system. Gaining the users' trust that the new system will function, be accurate and effective, can be seen as the most important stage in creating a successful new system. User documentation and training are its main concerns. Usually, conversion happens either during or after the user's training. Implementation is the process of turning a newly revised system design into an operational one, and it simply refers to placing a new system design into operation.

The user department now bears the most of the workload, faces the most disruption, and has the biggest influence on the current system. If the implementation is not well thought out or managed, confusion and mayhem may result.

Implementation encompasses all of the steps used to switch from the old system to the new one. The new system could be entirely different, take the place of an existing manual or automated system, or it could be modified to work better. A reliable system that satisfies organizational needs must be implemented properly. System implementation refers to the process of actually using the built system. This comprises all the processes involved in switching from the old to the new system. Only after extensive testing and if it is determined that the system is operating in accordance with the standards can it be put into use. The system personnel assess the system's viability. The effort necessary for system analysis and design to implement the three key components of education and training, system testing, and changeover will increase in proportion to how complicated the system being implemented is. The following tasks are included in the implementation state:

- Careful planning.
- Investigation of system and constraints.
- Design of methods to achieve the changeover.

---

## 6.2 IMPLEMENTATION PROCEDURES

Software implementation refers to the complete installation of the package in its intended environment, as well as to the system's functionality and satisfaction of its intended applications. The software development project is frequently commissioned by someone who will not be using it. People have early reservations about the software, but we must watch out that they do not become more resistant by making sure that:

- The new system's advantages must be known to the active user.
- Their belief in the software grows.
- The user receives the appropriate instruction so that he feels confident using the application.

Before examining the system, the user must be aware that the server software needs to be running on the server in order to access the results. The actual process won't happen if the server object is not active and functioning on the server.

### 6.2.1 User Training

The purpose of user training is to get the user ready to test and modify the system. The people who will be involved must have faith in their ability to contribute to the goal and benefits anticipated from the computer-based system. Training is more necessary as systems get more complicated. The user learns how to enter data, handle error warnings, query the database, call up routines that will generate reports, and execute other important tasks through user training.

### 6.2.2 Training on the Application Software

The user will need to be instructed on the new application software after receiving the appropriate basic training on computer awareness. This will explain the fundamental principles of how to use the new system, such as how the screens flow, how they are designed, what kinds of errors can occur while entering data, how to fix those errors, and how each entry is validated. The information needed by the particular user or group to utilize the system or a specific component of the system should then be covered as the program's training on the application is being given. This training may differ depending on the user group and the level of hierarchy.



### **6.2.3 System Maintenance**

The upkeep of systems is the riddle. A software product is actively functioning when it is in the maintenance stage of its lifespan. After a system has been implemented successfully, it should be appropriately maintained. System maintenance is a crucial phase in the software development life cycle. Maintenance is necessary for a system to be adaptable to changes in the system environment. Of all, maintaining software entails much more than merely "Finding Mistakes."

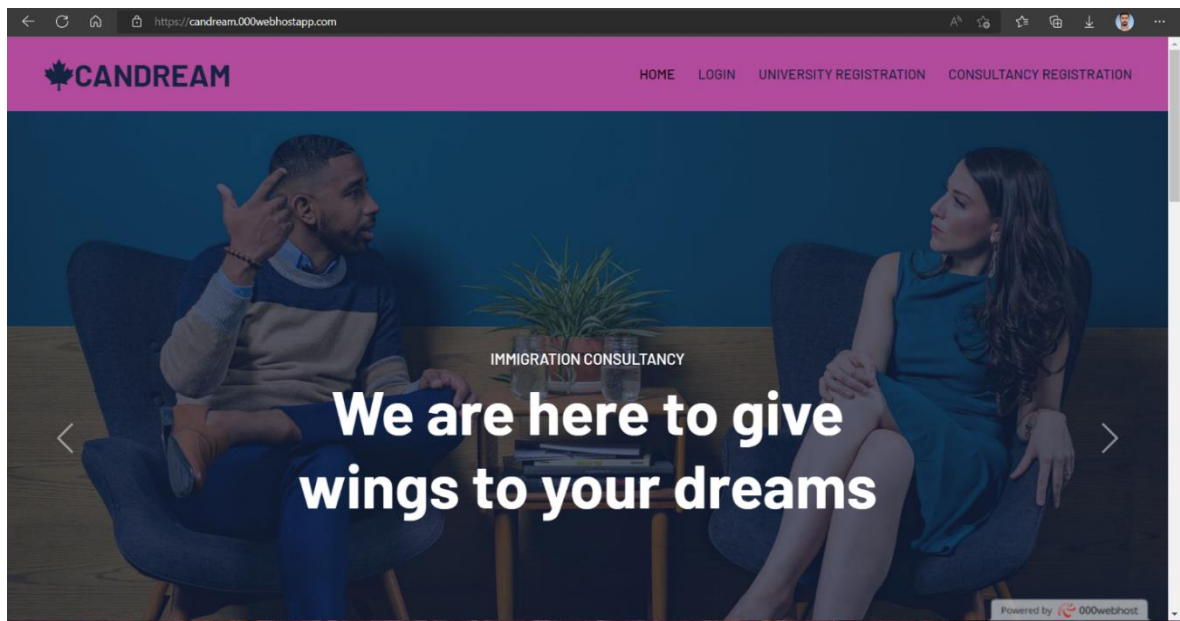
### **6.2.4 Hosting**

Free web hosting frequently has such little features and capabilities that customers must pay to obtain what they require. However, 000webhost is a free website hosting service that offer a number of significant advantages, including a website builder, WordPress support and no advertisements. Customers can upgrade to a paid plan to obtain significantly more features and benefits, but according to our surveys, 000webhost is the best free web hosting option for people with limited budget.

#### **Setting up 000webhost :**

- i. Head to 000webhost.com, and click that free sign up button.
- ii. Now go to your email account and confirm the link that we've sent. Once verified, you can start creating a website.
- iii. When you click the create button, there's a prompt asking why you sign up with us. Choose your reason.
- iv. After that, you'll get a form to fill the website's name and password. This step will add the website to your 000webhost sites list.
- v. There are three options available to build your website: using site builder, WordPress, or uploading a website that you've developed.
- vi. Click install now, and you'll get another form to create the webmaster login. When you finished, click the install button.

Link : <https://candream.000webhostapp.com/index.php/Welcome/logout>



## **CHAPTER 7**

### **CONCLUSION AND FUTURE SCOPE**

## **7.1 CONCLUSION**

This project is a web application which developed to support the students who seeks education in Canada. Now student can apply for canadian Universities. Consultancies will verify their applications and forward to the Universities. Universities receives the applications of students from the consultancies which are accepted by them. Admin can also manage Universities, Consultancies. It is build as a secure system with user friendly interface. Students have to register into before login. Admin can be logged in with their login credentials.

## **7.2 FUTURE SCOPE**

- The system can be extended by providing services related to various visa categories such as permanent, job, visiting, etc. For its registered members on the system on the basis of their profession or the purpose of their visit to the country.
- Direct communications can be done with Students, Consultancies and Universities through chat or other mediums.
- Application Processes for more universities in more different countries can be added to the system.

## **CHAPTER 8**

## **BIBLIOGRAPHY**

**REFERENCES:**

- <https://www.codeigniter.com/userguide3/tutorial/index.html>
- <https://www.tutorialspoint.com/codeigniter/index.htm>
- <https://www.javatpoint.com/css-tutorial>
- <https://www.tutorialspoint.com/php/index.html>
- <https://www.w3schools.com/php/default.asp>
- [www.w3schools.com/sql/](https://www.w3schools.com/sql/)

## **CHAPTER 9**

## **APPENDIX**

## 9.1 Sample Code

### welcome.php

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Welcome extends CI_Controller {
    public function __construct()
    {
        parent::__construct();
        $this->load->helper('form');
        $this->load->helper(array('form', 'url'));
        $this->load->library('form_validation');
        $this->load->library('session');
        $this->load->helper("file");
        $this->load->model("OnlineModel")
    }
    public function index()
    {
        $this->load->view('index');
    }
    public function login(){
        $this->load->view('student/login');
    }
    public function about(){
        $this->load->view('about.html');
    }
    public function signup(){
        $this->load->view('student/signup');
    }
    public function service(){
        $this->load->view('service.html');
    }
    public function contact(){
        $this->load->view('contact.html');
    }
    public function studentViewUniversity(){
        if( $this->session->userdata('username')){
```



---

```

    $data['c_name'] = $this->input->get_post('c_name');
    $id = $this->input->get_post('c_name');
    $data['university'] = $this->OnlineModel-
    >get_accepted_universities_by_consultancy_id($id);
        $this->load->view('student/student_header');
        $this->load->view('student/student_view_university',$data);
        $this->load->view('student/student_footer');
    }else{
        $this->index();
    }
}

public function studentViewCourses(){
    if( $this->session->userdata('username')){
        $data['u_name'] = $this->input->get_post('u_name');
        $data['course'] = $this->OnlineModel-
        >get_courses_by_university_id($data['u_name']);
        $this->load->view('student/student_header');
        $this->load->view('student/student_view_courses',$data);
        $this->load->view('student/student_footer');
    }else{
        $this->index();
    }
}

public function addSignup(){
    $data['username'] = $this->input->get_post('username');
    $data['password'] = $this->input->get_post('password');
    $data['repassword'] = $this->input->get_post('repassword');
    $data['status'] = '1';
    if($data['password'] == $data['repassword']){
        $existance = $this->OnlineModel->checkUsernameExistance($data);
        if($existance == true){
            $res = $this->OnlineModel->addSignup($data);
            ?> <script type="text/javascript">
            alert('Signup succesfull');
            location.href = "<?php echo base_url(); ?>";
            </script> <?php

            $this->login();
        }else{

```

---

```

        ?> <script type="text/javascript">
            alert('Email,username or phone already used by another user');
        </script> <?php
            $this->login();
        }
    }else{
        ?> <script type="text/javascript">
            alert('password mismatch');
        </script> <?php
    }
}

public function studentHome(){
    if( $this->session->userdata('username')){
        $this->load->view('student/student_header');
        $this->load->view('student/student_home');
        $this->load->view('student/student_footer');
    }else{
        $this->index();
    }
}

public function studentLogin()
{
    $login_type=$this->input->get_post('login_type');
    $data['username']=$this->input->get_post('username');
    $data['password']=$this->input->get_post('pass');
    $data['status']='1';

    if($login_type==1){
        $res=$this->OnlineModel->checkLogin($data);
        if($res==true){
            $username=$data['username'];
            $this->load->library('session');
            $this->session->set_userdata('username', $username);
            echo $this->session->userdata('username').'</h5>';
            ?> <script type="text/javascript">
                alert('Login successful');
            </script> <?php
                $this->email($data);
            }
        }
    }
}

```

```
        $this->studentHome();
    }else{
        ?> <script type="text/javascript">
        alert('Incorrect username or password');
        </script> <?php
        $this->login();
    }
}
else if ($login_type==2) {
    $res=$this->OnlineModel->checkLogin_consultancy($data);
    if($res==true){
        $username=$data['username'];
        $this->load->library('session');
        $this->session->set_userdata('username', $username);
        ?> <script type="text/javascript">
            alert('Login successful');
        </script> <?php
        $this->load->view('consultancy/consultancy_header');
        $this->load->view('consultancy/consultancy_home');
        $this->load->view('consultancy/consultancy_footer');
    }else{
        ?> <script type="text/javascript">
            alert('Incorrect username or password');
        </script><?php
    }
}
else if ($login_type==3) {
    $res=$this->OnlineModel->checkLogin_university($data);
    if($res==true){
        $this->session->set_userdata('email', $email);
        ?> <script type="text/javascript">
            alert('Login successful');
        </script> <?php
        $this->load->view('university/university_header');
        $this->load->view('university/university_home');
        $this->load->view('university/university_footer');
    }else{
```

```
        ?> <script type="text/javascript">
        alert('Incorrect username or password');
        </script><?php
    }
}
}
public function adminLogin(){
    $data['username']=$this->input->get_post('username');
    $data['password']=$this->input->get_post('password');
    $data['status']='1';
    $res1=$this->OnlineModel->checkAdminLogin($data);
    ?> <script type="text/javascript">
    alert('Admin login succesfull');
    </script> <?php
}
public function checkConsultancy(){
    if( $this->session->userdata('username')){
        $data['consultancy']=$this->OnlineModel->getConsultancy();
        $this->load->view('student/student_header');
        $this->load->view('student/student_check_consultancy',$data);
        $this->load->view('student/student_footer');
    }else{
        ?> <script type="text/javascript">
        alert('Please login again');
        </script> <?php
        $this->login();
    }
}
public function view_consultancy_by_id($id=null){
    $id=$this->input->get_post('id');
    $data['consultancy']= $this->OnlineModel->get_consutancy_by_id($id);
    $this->load->view('student/student_header');
    $this->load->view('student/student_single_consultancy',$data);
    $this->load->view('student/student_footer');
}
public function view_university_by_id($id=null){
    $id=$this->input->get_post('id');
    $data['university']= $this->OnlineModel->get_university_by_id($id);
```

```
$this->load->view('student/student_header');
$this->load->view('student/student_single_university',$data);
$this->load->view('student/student_footer');
}
public function admissionStatus(){
    if( $this->session->userdata('username')){
        $username=$this->session->userdata('username');
        $res=$this->OnlineModel->get_student_id_by_uname($username);
        $data['student_id'] = $res['logno'];
        $data['admission_status']=$this->OnlineModel-
        >admissionStatus( $data['student_id']);
        $data1['admissionStatus']=$data['admission_status']->row_array();
        $data['course_name']=$this->OnlineModel-
        >getCourse_name_by_id($data1['admissionStatus']['course_id']);
        $this->load->view('student/student_header');
        $this->load->view('student/student_view_admission_status',$data);
        $this->load->view('student/student_footer');
    }else{
        ?> <script type="text/javascript">
        alert('Please login again');
        </script> <?php
        $this->login();
    }
}
public function checkEligibility(){
    if( $this->session->userdata('username')){
        $data['eligibility'] = $this->input->get_post('eligibility');
        $data['course']=$this->OnlineModel->getCourse();
        $this->load->view('student/student_header');
        $this->load->view('student/student_check_eligibility',$data);
        $this->load->view('student/student_footer');
    }
    else{
        ?> <script type="text/javascript">
        alert('Please login again');
        </script> <?php
        $this->login();
    }
}
```

---

 }

Student\_check\_consultancy.php

```

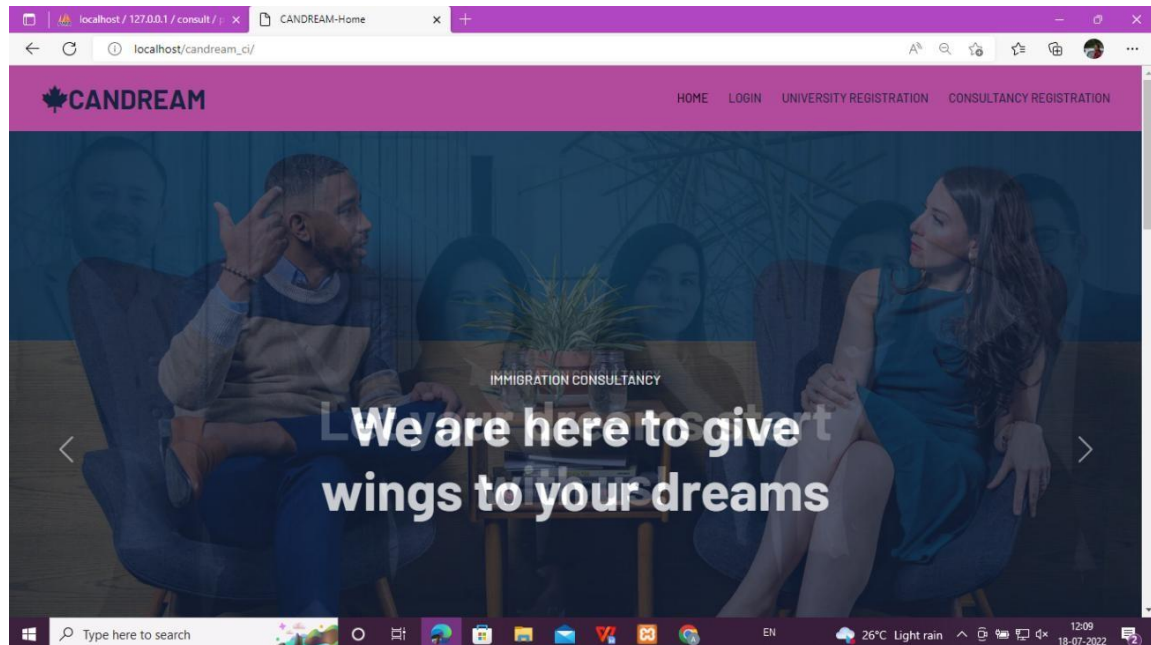
<section id="home" class="w3l-banner py-5">
  <div class="banner-content">
    <div class="container py-4">
      <div class="row align-items-center pt-sm-5 pt-4">
        <center>
          <h1><b>CONSULTANCY<b></h1><br>
        </center>
      </div>
    </div>
  </div>
  <?php
  $n=1;
  foreach($consultancy->result() as $row)
  {?>
    <figure class="snip1559">
      <div class="profile-image"></div>
      <figcaption>
        <h3><?php echo $row->cunsultancy_name; ?></h3>
        <h5><?php echo $row->phone; ?></h5>
        <p><?php echo $row->place; ?></p>
        <div class="icons"><a href=""></a>
          <a href="view_consultancy_by_id?id=<?php echo $row->cunsultancy_id ?>">View Consultancy
          </a><br>
          <a href="studentViewUniversity?c_name=<?php echo $row->cunsultancy_id ?>">View universities</a>
        </div>
      </figcaption>
    </figure --!>
  <?php $n++;}
  ?>
</div>
</div>
</div>
</section>

```

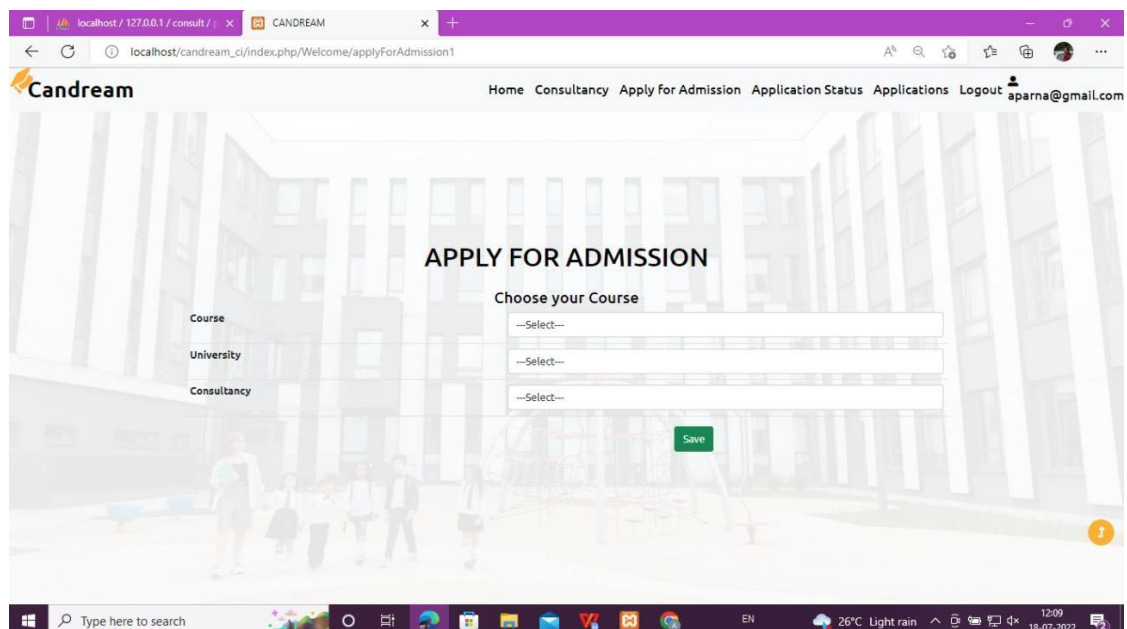
## 9.2 Screen Shots

### Student Module

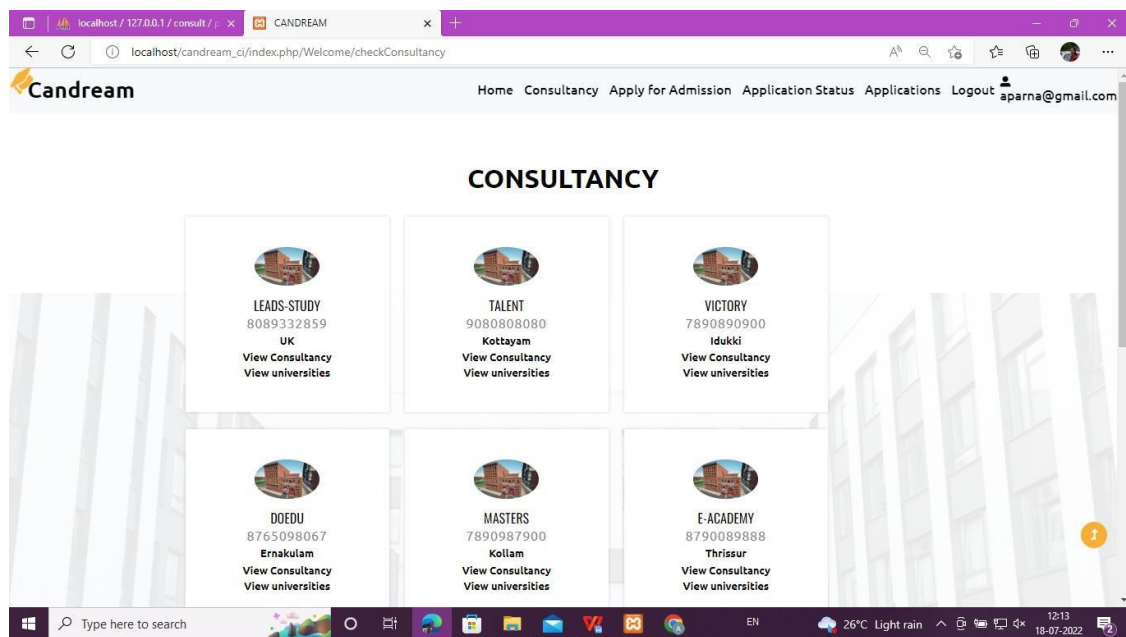
#### Index Page



#### Application First Page

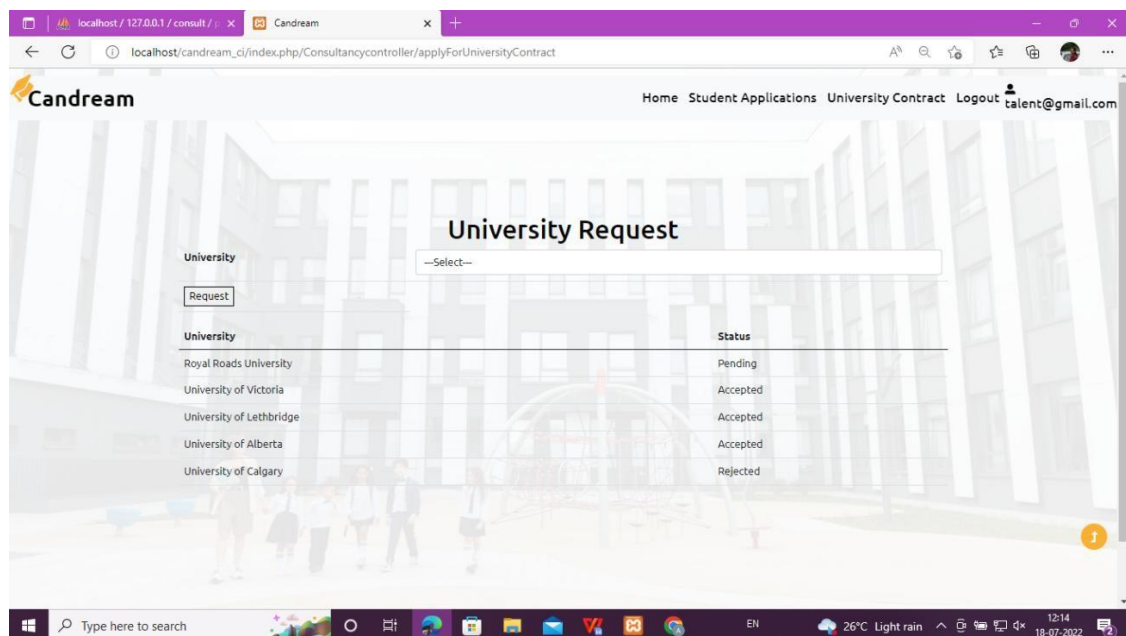


## Check Consultancy Page



## Consultancy Module

### University Request Page





## Student Application Management

**Student Applications**

| Application ID | Student   | Phone      | Email           | Course | University            | View                 | Status               | Comments |
|----------------|-----------|------------|-----------------|--------|-----------------------|----------------------|----------------------|----------|
| 1021           | Ameya N   | 9008008989 | ameya@gmail.com | MBA    | University of Calgary | <a href="#">View</a> | university Approved  |          |
| 998            | alen n nm | 8089332859 | alen@gmail.com  | MBA    | University of Calgary | <a href="#">View</a> | Consultancy Verified |          |

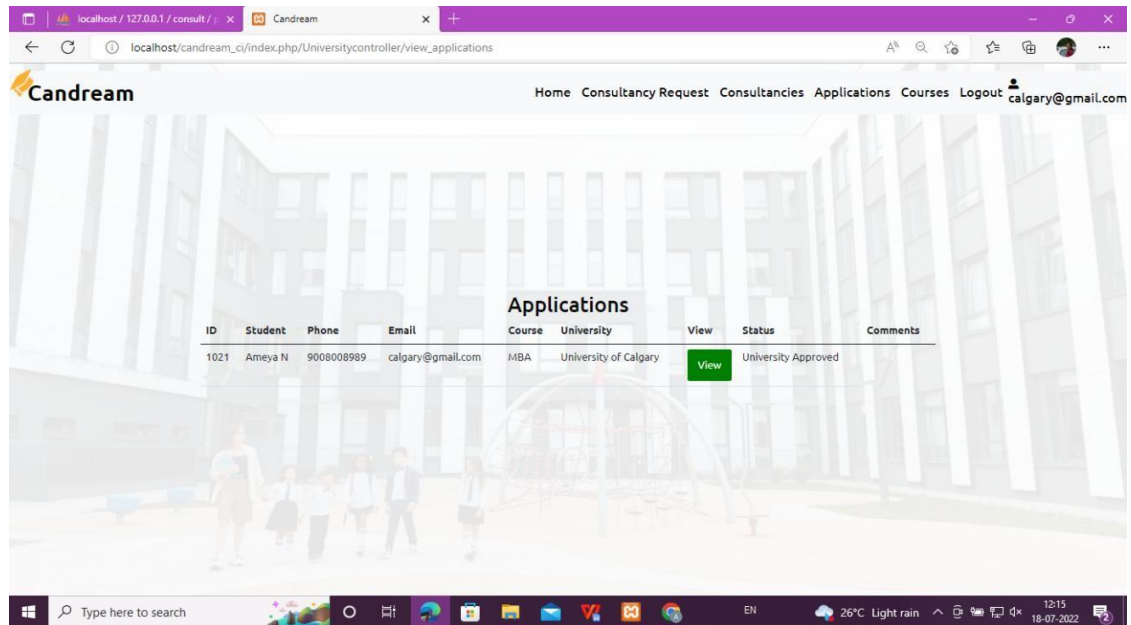
## University Module

### University Contract Management

**Contract Requests**

| Name        | Email                | Phone      | Website     | Place    | View             | Document | Action               |
|-------------|----------------------|------------|-------------|----------|------------------|----------|----------------------|
| leads-study | consultancy@gmail.co | 8089332859 | www.uni.com | UK       | View Consultancy | Accepted | <a href="#">Save</a> |
| Talent      | talent@gmail.com     | 9080808080 | talent      | Kottayam | View Consultancy | Rejected | <a href="#">Save</a> |

## Student Application Management



The screenshot shows a web browser window with the URL `localhost/candream_ci/index.php/Universitycontroller/view_applications`. The page title is "Candream" and the user is logged in as `calgary@gmail.com`. The navigation menu includes: Home, Consultancy Request, Consultancies, Applications, Courses, and Logout.

### Applications

| ID   | Student | Phone      | Email             | Course | University            | View                 | Status              | Comments |
|------|---------|------------|-------------------|--------|-----------------------|----------------------|---------------------|----------|
| 1021 | Ameya N | 9008008989 | calgary@gmail.com | MBA    | University of Calgary | <a href="#">View</a> | University Approved |          |

The Windows taskbar at the bottom shows the search bar, task view button, and several application icons. The system tray displays the date and time as 12:15 on 18-07-2022, along with weather information: 26°C, Light rain.

PAPER NAME

**CanDream.pdf**

WORD COUNT

**7040 Words**

CHARACTER COUNT

**37255 Characters**

PAGE COUNT

**45 Pages**

FILE SIZE

**504.9KB**

SUBMISSION DATE

**Jul 22, 2022 10:30 AM GMT+5:30**

REPORT DATE

**Jul 22, 2022 10:32 AM GMT+5:30****● 13% Overall Similarity**

The combined total of all matches, including overlapping sources, for each database.

- 13% Internet database
- 1% Publications database
- Crossref database

**● Excluded from Similarity Report**

- Small Matches (Less than 15 words)

## 13% Overall Similarity

Top sources found in the following databases:

- 13% Internet database
- 1% Publications database
- Crossref database

### TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

|   |                                    |          |     |
|---|------------------------------------|----------|-----|
| 1 | <b>docshare.tips</b>               | Internet | 3%  |
| 2 | <b>dev.mysql.com</b>               | Internet | 2%  |
| 3 | <b>geeksforgeeks.org</b>           | Internet | 2%  |
| 4 | <b>issuu.com</b>                   | Internet | 1%  |
| 5 | <b>slideshare.net</b>              | Internet | <1% |
| 6 | <b>scribd.com</b>                  | Internet | <1% |
| 7 | <b>docplayer.net</b>               | Internet | <1% |
| 8 | <b>engineeringminiprojects.com</b> | Internet | <1% |
| 9 | <b>medium.com</b>                  | Internet | <1% |

|    |                                                                     |     |
|----|---------------------------------------------------------------------|-----|
| 10 | <b>coursehero.com</b><br>Internet                                   | <1% |
| 11 | <b>best-computer-courses-in-chandigarh.blogspot.com</b><br>Internet | <1% |
| 12 | <b>software-application.blogdrive.com</b><br>Internet               | <1% |
| 13 | <b>vsbec.com</b><br>Internet                                        | <1% |
| 14 | <b>elibrary.buse.ac.zw:8080</b><br>Internet                         | <1% |
| 15 | <b>whatis.techtarget.com</b><br>Internet                            | <1% |
| 16 | <b>deivasathish.blogspot.com</b><br>Internet                        | <1% |
| 17 | <b>tutorialspoint.com</b><br>Internet                               | <1% |