INFORMATICS INSTITUTE OF TECHNOLOGY

Individual Coursework

Module Code	: DOC 333
Module Title	: Introduction to Programming Principles
Module Leader	: Ms. Tharushi Amarasinghe
Assessment Type	: Individual Course work
Issued Date	: 4th November 2024
Hand-in Date	: 25th November 2024
Weight	: 30%

Student ID : 20241154

Student Name : S.V.D. Jayasiri

Centre : Kurunegala

Executive Summary

This report describes the process of constructing a student attendance management system for the Foundation Certificate in Higher Education program at the Informatics Institute of Technology. It aims to improve the management of student data and attendance control for students across various campuses for efficient administration.

The project involves installing a Software system based on Python which has the following capabilities :-

1. Student enrollment: Enroll new students by taking their personal information, tutorial group and any other information.

2. Student data management: Perform view, edit and display student information.

3. Attendance Management: View and record attendance for groups or individual students for specified periods.

The phases of the development process respect the fundamental principles of programming and concentration on one of the following aspects :

- Human-Computer interaction :- It was anticipated how the interface would look like as all figures for this part are supplied in the assignment brief.

- Data structures :- Well-organized storage and provision of student and attendance details through the use of features provided by python.

- Error reporting :- Implementation of mechanisms that can fully evaluate and verify the input data and the operating system as a whole.

Moreover, the utility helps to discover :

- An algorithmic decomposition of the operationalized solution.

- The limitations and assumptions made with regard to the designed systems and their functionalities.

- The tests confirm the strength of the system with illustrations on positive

Acknowledgements

I would like to express my heartfelt gratitude to Mr. Kasun Jayananda, Mrs. Samindi Rathnayake & Mrs. Nirasha Egodagama for their invaluable guidance, mentorship, and support throughout the completion of this coursework. Their expertise and constructive feedback have been instrumental in enhancing my understanding and refining my approach to the subject matter.

I also extend my thanks to the Informatics Institute of Technology for providing an enriching learning environment and access to the necessary resources and tools that facilitated the successful completion of this work.

Finally, I would like to acknowledge the encouragement and moral support of my family and friends, whose unwavering belief in me has been a source of motivation throughout this journey.

Table of Contents

Table of Contents

List of Figures

List of Tables

1 Introduction

Towards this end, the aim of this project is to design an integrated students' attendance management system using python for the foundation certificate in higher education program at the Informatics Institute of Technology (IIT). This project has been developed to tackle the difficulties faced in the management of student information and attendance record systems across the campuses located in Colombo, Galle and Kurunegala. The proposed solution enhances the management process by making the storage and retrieval of data to the administrators efficient, accurate and dependable.

The scope of the project covers three main areas:

Student Enrollment: Aims at the inclusion of a new student's record where relevant student's information such as personal particulars, tutorial assemble, and campus place is provided.

Data Management: This involves the aspect of viewing, editing and displaying information in respect to particular and them several students.

Attendance Tracking: This function tracks and retrieves the statistics of attendance for given tutorial groups or for individual students for the requested periods of time.

In the development phase, basic principles of programming were strictly followed and include the design of robust and enhanced data structures, efficient error detection and user interaction devices as can be seen from the given assignment diagrams. Also the report describes algorithms that were used in implementing particular feature, such as controlling the feature parameters, defining and explaining assumptions and limitations and presenting the designed solution as a result of testing. Integrating these components makes the project a good opportunity to apply the theoretical materials and acquire desired skills for dealing with practical administrative problems.

2 Algorithm

2.1 Initialization :-

- Create a list as students to store student details.
- Create a list attendance to store attendance records.
- Set a variable running to True to control the main loop.

2.2 Main Menu :-

- While running is True:
    - Display the main menu options:
        1. Enrolling in a new student
        2. View details of a student
        3. View details of all students
        4. Update student details
        5. Mark attendance
        6. View attendance
        7. Exit
    - Prompt the user for their choice.

2.3 Enrolling a New Student :-

- If the user selects choice 1:
    - Prompt for student details (ID, NIC, first name, last name, birth date, address, phone number, tutorial group, and centre).
    - Validate inputs (e.g., ID length, NIC length, etc.).
    - Confirm if the user wants to save the details.
    - If confirmed, apply the student details to the students list.

2.4 View Details of a Student :-

- If the user selects choice 2:
    - Prompt for the student ID.

- Search for the student in the students list.
- If found, display the students' details.
- Ask if the user wants to view another student's details.

2.5 View Details of All Students :-

- If the user selects choice 3:
  - Prompt for a branch name.
  - Display all students' details that match the branch name.
  - Ask if the user wants to update student details.

2.6 Update Student Details :-

- If the user selects choice 4:
  - Prompt for the student ID.
  - Search for the students in the students list.
  - If found, prompt for new details (allow empty input to retain old values).
  - Confirm if the user wants to save the updated details.

2.7 Mark Attendance :-

- If the user selects choice 5:
  - Prompt for the centre, tutorial group, and date.
  - For each student in the students list, check if they belong to the specified group.
  - Prompt for attendance status (Present/Absent) for each student.
  - Confirm if the user wants to save the attendance records.

2.8 View Attendance :-

- If the user selects choice 6:
  - Prompt for the student ID and date range (from date to date).
  - Display attendance records for the specified student within the date range.

2.9 Exit the Program :-

- If the user selects choice 7:

- Set running to False to exit the main loop.

2.10 Invalid Choice Handling :-

- If the user inputs an invalid choice, display an error message and prompt again.

3 Test Cases

3.1 Figure 1

```
# Lists for Storage Student Data and Attendance

students = [[569873624,"9864215963","Kasun ","Weerasinghe","2007/02/14","Kuliyapitiya","0765968423","A","Kurunegala"],
            [697412385,"6542198753","Nadun ","Gunarathne ","2005/08/20","Anuradhapura","0743875213","C","Colombo"],
            [124698536,"5416872316","Pramud","Kulaweera  ","2004/11/05","Nikawaratiya","0713697874","A","Kurunegala"]]
attendance = []

# Figure 1:- Main Menu

running = True
while running :
    print("_" * 153)
    print()
    print("IIT Campus".center(75))
    print()
    print("Main Menu".center(77))
    print()
    print("1) Enroll a new student")
    print("2) View details of a student")
    print("3) View details of all students")
    print("4) Update student details")
    print("5) Mark attendance")
    print("6) View attendance")
    print("7) Exit")
    print()

    choice = input("                                                                    Your Choice : ")
```

Figure 1 : Main Menu Code

Table 1 : Main Menu Test Cases

Input	Expected Output	Received Output
print("_" * 153)	_____	_____
print("IIT Campus".center)	IIT Campus	IIT Campus
Your Choice : 1234	Invalid Choice. Please Try Again.	Invalid Choice. Please Try Again.

```
                                IIT Campus

                                Main Menu

1) Enroll a new student
2) View details of a student
3) View details of all students
4) Update student details
5) Mark attendance
6) View attendance
7) Exit

                                                        Your Choice : 1234
                    Invalid Choice. Please Try Again.
```

Figure 2 : Main Menu Test Case

3.2 Figure 2

```
# Figure 2:- Enroll a New Student

if (choice == "1") :
    student = []
    print("_" * 153)
    print()
    print("IIT Campus".center(75))
    print()
    print("Enroll a New Student".center(75))
    print()
    student_id = (input("Student ID        : "))
    while len(student_id)!= 9 :
        student_id = (input ("Invalid ID,Enter 9 Digits : "))
    nic = (input("NIC               : "))
    while len(nic)!= 10 :
        nic = (input ("Invalid NIC,Enter 10 Characters : "))
    first_name = input("First Name        : ")
    while len(first_name)> 10 :
        first_name = input ("Invalid First Name,Enter 10 Characters : ")
    last_name = input("Last Name         : ")
    while len(last_name)> 15 :
        last_name = input ("Invalid Last Name,Enter 15 Characters : ")
    birth_date = input("Birth Date        : ")
    address = input("Permanent Address : ")
    while len(address)> 15 :
        address = input ("Invalid Address,Enter 15 Characters : ")
    phone = input("Phone Number      : ")
    while len(phone)!=10 :
        phone = input ("Invalid Phone Number,Enter 10 Characters : ")
    tutorial_group = input("Tutorial Group    : ")
    centre = input("Centre            : ")
```

Figure 3 : Enroll a New Student Code 1

```
student.append(int(student_id))
student.append(nic)
student.append(first_name)
student.append(last_name)
student.append(birth_date)
student.append(address)
student.append(phone)
student.append(tutorial_group)
student.append(centre)

print()
confirm = input("Do you want to save the details (Yes/No)? : ").lower()
print()
if (confirm == "yes") :
    students.append(student)
    print()
    print("Student Details Saved Successfully.".center(75))
else :
    print()
    print("Student Details not Saved.".center(75))
```

Figure 4 : Enroll a New Student Code 2

Table 2 : Enroll a New Student Test Cases

Input	Expected Output	Received Output
Do you want to save the details (Yes/No)? **:** yes	Save details in students list	Student Details Saved Successfully.
NIC : 123	Invalid ID,Enter 9 Digits :	Invalid ID,Enter 9 Digits :
student.append(int(student_id))	Save student ID in students list from student list	Save student ID in students list from student list

```
                                IIT Campus

                            Enroll a New Student

Student ID          : 123
Invalid ID,Enter 9 Digits : 123456789
NIC                 : 8456923175
First Name          : Amith
Last Name           : Warnakula
Birth Date          : 2004/10/15
Permanent Address   : Narammala
Phone Number        : 0789652314
Tutorial Group      : C
Centre              : Colombo

Do you want to save the details (Yes/No)? : yes


                    Student Details Saved Successfully.
```

Figure 5 : Enroll a New Student Test Case

3.3 Figure 3

```
# Figure 3:- View Details of a Student

elif (choice == "2") :
    while True :
        print("_" * 153)
        print()
        print("IIT Campus".center(75))
        print()
        print("View Details of a Student".center(75))
        print()
        student_id = int(input("Enter Student ID : "))

        for student in students :
            if (student[0] == int(student_id)) :
                print("NIC : ", student[1])
                print("Phone Number : ", student[6])
                print("First Name : ", student[2])
                print("Last Name : ", student[3])
                print("Tutorial Group : ", student[7])
                print("Centre : ", student[8])
                break
        else :
                print("Student not Found".center(75))

        print()


        save = input("Do you Want to View Another Student's Details (Yes/No)? : ").lower()
        if (save == "yes") :
            continue
        if (save == "no") :
            print()
            print("Canceled View Another Student's Details".center(75))
            break
        else :
            print()
            print("Invalid Choice,Try Again".center(75))
            break
```

Figure 6 : View Details of a Student Code

Table 3 : View Details of a Student Test Cases

Input	Expected Output	Received Output
Enter Student ID : 569873624	View details of one student	NIC : 9864215963 Phone Number : 0765968423 First Name : Kasun Last Name : Weerasinghe

		Tutorial Group : A Centre : Kurunegala
Enter Student ID : colombo	Error Message Because, student id data type must be int	Error Message
Do you Want to View Another Student's Details (Yes/No)? : yes	Continue the loop Again	Continue the loop Again

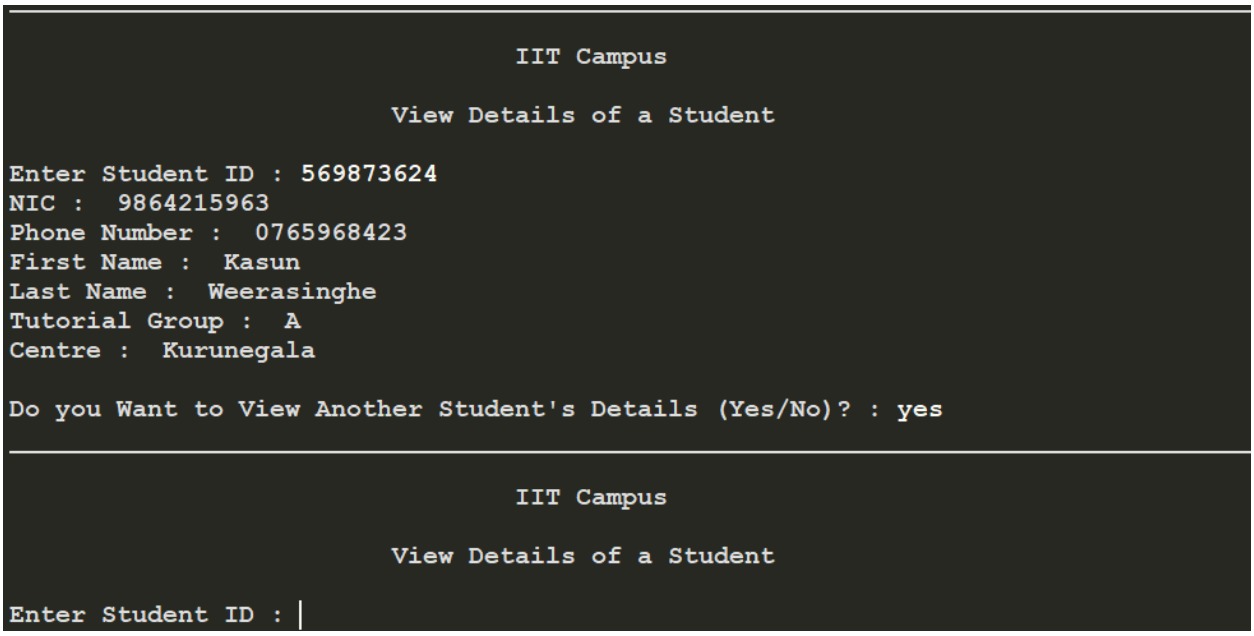

Figure 7 : View Details of a Student Test Case 1

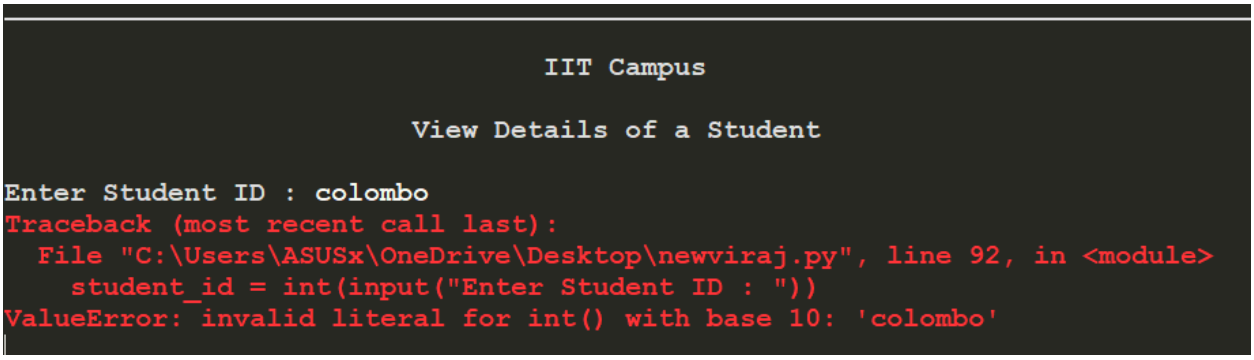

Figure 8 : View Details of a Student Test Case 2

3.4 Figure 4

```
# Figure 4:- View Details of All the Students

elif (choice == "3") :
    print("_" * 153)
    print()
    print("IIT Campus".center(75))
    print()
    print("View Details of All Students".center(75))
    print()

    branch = input("Branch Name : ")

    if (students) :
        print("|NIC|             |Student ID|         |First Name|         |Last Name|        |Tutorial Group|")
        for student in students :
            if (branch.lower() in student[8].lower() or not branch) :
                print(f"{student[1]}         {student[0]}             {student[2]}              {student[3]}              {student[7]}")
    print()
    confirm = input("Do you want to update student details (Yes/No)? : ").lower()
    if (confirm == "yes") :
        print()
        print("Please Select Choice = 4 to Update Student Details".center(75))
        continue
    if (confirm == "no") :
        continue
    else :
        print()
        print("Invalid Choice,Try Again".center(75))
        continue
```

Figure 9 : View Details of All the Students Code

Table 4 : View Details of All the Students Test Cases

Input	Expected Output	Received Output
Branch Name : kurunegala	View the details of Kurunegala students	Display Kurunegala students Details
Branch Name : galle	View the details of Galle students	No save information of Galle Students (Empty Details)
Do you want to update student details (Yes/No)? : yes	Update Student Details	Please Select Choice = 4 to Update Student Details

```
                                IIT Campus

                          View Details of All Students

Branch Name : kurunegala
|NIC|              |Student ID|          |First Name|          |Last Name|         |Tutorial Group|
2365987452          569873624             Nuran               Amunugama               A
5416872316          124698536             Pramud               Kulaweera                 A

Do you want to update student details (Yes/No)? : yes

               Please Select Choice = 4 to Update Student Details
```

Figure 10 : View Details of All the Students Test Case 1

```
                                IIT Campus

                        View Details of All Students

Branch Name : galle
|NIC|             |Student ID|          |First Name|          |Last Name|         |Tutorial Group|

Do you want to update student details (Yes/No)? : no
```

Figure 11 : View Details of All the Students Test Case 2

3.5 Figure 5

```
# Figure 5:- Update Student Details

elif (choice == "4") :
    print("_" * 153)
    print()
    print("IIT Campus".center(75))
    print()
    print("Update Student Details".center(75))
    print()
    student_id = input("Enter Student ID   : ")

    for student in students :
        if (student[0] == int(student_id)) :

            new_nic = input("New NIC          : ")
            while len(new_nic)!= 10 :
                if (new_nic == "") :
                    new_nic = student[1]
                    break
                new_nic = input ("Invalid NIC,Enter 10 Characters : ")

            new_first_name = input("New First Name     : ")
            while len(new_first_name)>10 :
                if (new_first_name == "") :
                    new_first_name = student[2]
                    break
                new_first_name = input ("Invalid First name,Enter less than 10 Characters : ")

            new_last_name = input("New Last Name      : ")
            while len(new_last_name)>15 :
                if (new_last_name == "") :
                    new_last_name = student[3]
                    break
                new_last_name = input ("Invalid Last name,Enter less than 15 Characters : ")
```

Figure 12 : Update Student Details Code 1

```
            new_birth_date = input("New Birth date    : ")
            if (new_birth_date == "") :
                new_birth_date = student[4]

            new_address = input("New Address       : ")
            while len(new_address)> 15 :
                if (new_address == "") :
                    new_address = student[5]
                    break
                new_address = input ("Invalid Address,Enter 15 Characters : ")


            new_phone_number = input("New phone number  : ")
            while len(new_phone_number)!=10 :
                if new_phone_number == "":
                    new_phone_number = student[6]
                    break
                new_phone_number = input ("Invalid phone number,Enter 10 digits : ")

            print()
            confirm = input("Do you want to save the details (Yes/No)? : ").lower()
            if (confirm == "yes") :
                student[1] = new_nic
                student[2] = new_first_name
                student[3] = new_last_name
                student[4] = new_birth_date
                student[5] = new_address
                student[6] = new_phone_number
                print()
                print("Student Details Updated Successfully.".center(75))
            else :
                print()
                print("Student Details not Saved.".center(75))
            break
    else :
        print()
        print("Student not Found".center(75))
```

Figure 13 : Update Student Details Code 2

Table 5 : Update Student Details Test Cases

Input	Expected Output	Received Output
Enter Student ID : 12345	Error Message	Student not Found
New NIC : 25684	Error Message	Invalid NIC,Enter 10 Characters :
Do you want to save the details (Yes/No)? : yes	New details replace with old details in lists	Student Details Updated Successfully.

```
                                IIT Campus

                           Update Student Details

Enter Student ID   : 12345

                             Student not Found
```

```
                                IIT Campus

                                 Main Menu

1) Enroll a new student
2) View details of a student
3) View details of all students
4) Update student details
5) Mark attendance
6) View attendance
7) Exit

                                                              Your Choice : 4
```

```
                                IIT Campus

                           Update Student Details

Enter Student ID   : 569873624
New NIC          : 25684
Invalid NIC,Enter 10 Characters : 2365987452
New First Name     : Nuran
New Last Name      : Amunugama
New Birth date     : 2003/08/08
New Address        : Nikaweratiya
New phone number   : 0723651489

Do you want to save the details (Yes/No)? : yes

                    Student Details Updated Successfully.
```

Figure 14 : Update Student Details Test Case

3.6 Figure 6

```
# Figure 6:- Mark Attendance

elif (choice == "5") :
    print("_" * 153)
    print()
    print("IIT Campus".center(75))
    print()
    print("Mark Attendance".center(75))
    print()
    centre = input("Enter Centre        : ")
    tutorial_group = input("Enter Tutorial Group : ")
    date = input("Enter Date          : ")
    group_attendance = []
    for student in students :
        if (student[7] == tutorial_group and student[8].lower() == centre.lower()) :
            print()
            status = input(f"Student ID = {student[0]} - Present/Absent? : ").lower()
            group_attendance.append([student[0],date,status])
    if len(group_attendance)==0 :
        print("Invalid centre or tutorial group.... No student found")
    print()
    confirm = input("Do you Want to Save the Attendance Details (Yes/No)? : ").lower()
    if (confirm == "yes") :
        print()
        attendance.extend(group_attendance)
        print("Action Completed.".center(75))
    else :
        print()
        print("Action Not Completed.".center(75))
```

Figure 15 : Mark Attendance Code

Table 6 : Mark Attendance Test Cases

Input	Expected Output	Received Output
Enter Centre : kurunegala Enter Tutorial Group : C Enter Date : 2024/11/15	Invalid centre or tutorial group.... No student found	Invalid centre or tutorial group.... No student found
Do you Want to Save the Attendance Details (Yes/No)? : no	Action Not Completed.	Action Not Completed.
Enter Centre : colombo Enter Tutorial Group : C Enter Date : 2024/11/20	Save the attendance details of entering centre, tutorial group, date & present/absent	Action Completed.

Student ID = 697412385 - Present/Absent? : present Do you Want to Save the Attendance Details (Yes/No)? : yes		

```
                                IIT Campus

                             Mark Attendance

Enter Centre         : kurunegala
Enter Tutorial Group : C
Enter Date           : 2024/11/15
Invalid centre or tutorial group.... No student found

Do you Want to Save the Attendance Details (Yes/No)? : no

                           Action Not Completed.
```

Figure 16 : Mark Attendance Test Case 1

```
                                IIT Campus

                             Mark Attendance

Enter Centre         : colombo
Enter Tutorial Group : C
Enter Date           : 2024/11/20

Student ID = 697412385 - Present/Absent? : present

Do you Want to Save the Attendance Details (Yes/No)? : yes

                           Action Completed.
```

Figure 17 : Mark Attendance Test Case 2

3.7 Figure 7

```
# Figure 7:- View Attendance

elif (choice == "6") :
    print("_" * 153)
    print()
    print("IIT Campus".center(75))
    print()
    print("View Attendance".center(75))
    print()
    student_id = int(input("Student ID : "))
    from_date = input("From Date   : ")
    to_date = input("To Date    : ")
    print()
    print("Date        | Present/Absent")
    for record in attendance:
        if (record[0] == student_id and (from_date) <= (record[1]) and (record[1]) <= (to_date)) :
            print(record[1], "        |" , record[2])
    print()
    return_main = input("Do you want to Direct to the Main Menu (Yes/No)? : ").lower()
    if (return_main == "no") :
        print()
        print("...Exited...".center(75))
        break
```

Figure 18 : View Attendance Code

Table 7 : View Attendance Test Cases

Input	Expected Output	Received Output
Student ID : 697412385 From Date : 2024/11/10 To Date : 2024/11/30	Date | Present/Absent 2024/11/20 | present	Date | Present/Absent 2024/11/20 | present
Do you want to Direct to the Main Menu (Yes/No)? : yes	Direct to the Main Menu	Direct to the Main Menu
Do you want to Direct to the Main Menu (Yes/No)? : no	Exit the programme	...Exited...

```
                                IIT Campus

                              View Attendance

Student ID : 697412385
From Date  : 2024/11/10
To Date    : 2024/11/30

Date       | Present/Absent
2024/11/20        | present

Do you want to Direct to the Main Menu (Yes/No)? : yes
```

```
                                IIT Campus

                                 Main Menu

1) Enroll a new student
2) View details of a student
3) View details of all students
4) Update student details
5) Mark attendance
6) View attendance
7) Exit

                                                        Your Choice : |
```

Figure 19 : View Attendance Test Case 1

```
Date       | Present/Absent

Do you want to Direct to the Main Menu (Yes/No)? : no

                                  ...Exited...
```

Figure 20 : View Attendance Test Case 2

4 Conclusion

The most interesting part of this project is the development of an attendance management system suited for the requirements of the Informatics Institute of Technology, which was built on python. The system manages to accomplish the tasks posed by the assignment regarding registration of students, modification of the database, and attendance capturing, as well as user friendliness through a simple interface and good error responses.

It is assembled this way with the use of the right data structures and algorithms to ensure good efficiency and scalability for future modifications. The rigorous testing stage validated the effectiveness of the application in practice for positive as well as negative scenarios, hence it can be used without any worries in different environments.

Lastly, this project was able to meet the goals that were set, however, they did emphasize on some of the best ‘programming for orientation’ practices in the industry. It has been an eye opener in development processes on how to come up with functional software systems that are suitable for its intended users, which is a crucial combination of theory and practice. This assignment is a great achievement in the process of fully understanding and implementing the concepts of programming and prototyping.