

Abstract

What if one day the world we perceive is not enough for us? As humans, our desire to see beyond what we already have has brought so many innovations and technological advancements. And one day, we found a way to create everything we wanted to see in real life with computers and incorporate them into our environment. This happened thanks to augmented reality, but you may be wondering, what is augmented reality exactly?

AR, which is becoming a shining star with the increasing interest in the metaverse trend every day, is certain to be a significant factor in many industries, from entertainment to online shopping. Of course, in order to keep up with all these developments, it is necessary to understand what augmented reality really is, its true nature, and what it promises.

Augmented reality (AR) is the integration of virtual data produced in the computer environment into the real world in a way that we can perceive with our sense organs. The data produced in the virtual environment such as images, animations, 3D models, videos, graphics are transferred to the real world in real-time.

Supporting the data we perceive with enriching elements such as depth and viewing objects from different angles make it easier to understand how the objects actually look in our 3D world.

Acknowledgments

I would first like to thank my supervisor, Professor Marwa Abdelfatah, whose expertise was invaluable in formulating the project questions and methodology. Your insightful feedback pushed me to sharpen my thinking and brought my work to a higher level.

I respect and thank Dr Marwa, for providing me an opportunity to do the project work and giving us all support and guidance which made me complete the project duly. I am extremely thankful to her for providing such a nice support and guidance, although she had busy schedule managing the corporate affairs.

The success and final outcome of this project required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of my project. All that I have done is only due to such supervision and assistance and I would not forget to thank them.

List of Figures

3.1	Student Use case	7
3.2	Admin Use case	8
3.3	Guest Use case	9
3.4	Class Diagram	11
3.5	object Diagram	12
3.6	Package Diagram	13
3.7	Usecase Diagram	14
3.8	Student Activity Diagram	16
3.9	Admin Activity Diagram	17
3.10	Guest Activity Diagram	18
3.11	Student State machine Diagram	19
3.12	Admin State machine Diagram	20
3.13	Class Diagram	21
3.14	Admin Sequence Diagram	22
3.15	Guest Sequence Diagram	23
3.16	Student Sequence Diagram	24
3.17	Admin Communication Diagram	25
3.18	Guest Communication Diagram	26
3.19	Student Communication Diagram	27
4.1	Scale-invariant equation	36
4.2	Connect to database Function	37
4.3	Show registered courses Function	38
4.4	Check Registered courses Function	39
4.5	Logout Function	40
4.6	Login Function	41
4.7	Exam info Function	42
4.8	Show profile Function	43
4.9	Update profile Function	44
4.10	Show basic Information Function	45
4.11	Show Transcript Function	46
4.12	College Logo	47
4.13	Home	48

4.14 Home	49
4.15 Labs and Halls	50
4.16 Information about College Departments	51
4.17 Information about College Departments	52
4.18 Welcome To Student Panel	53
4.19 Login To Profile	54
4.20 Check Username and Password	55
4.21 Edit Profile	56
4.22 Show Transcript	57
4.23 Register your Courses	58
4.24 Successful Registration	59
4.25 Fail Registration	60
4.26 Show Registered Courses	61
4.27 ID detection	62
4.28 Detect Sign	62
4.29 Show Sign Information	63

Contents

Abstract	i
Acknowledgments	ii
List of Figures	iii
1 Introduction	3
2 Background	5
3 Specification - (SRS)	6
3.1 Functional Requirements	7
3.2 Non-Functional Requirements	9
3.3 System analysis and design	11
4 Implementation	28
4.1 Steps and Problems	28
4.2 Vuforia's Algorithm	32
4.3 Main Functionality and Results	37
5 Conclusions and Future work	64
5.1 Conclusion	64
5.2 Future Work	64
A Collected materials from	65

Chapter 1

Introduction

Augmented reality (AR) was first conceived as a science-fiction concept to support the creative narrative in films, books and plays. Our imagination took us as far as visualising digitally displayed information in thin-air, yet no one could put this into practice. The technology has come a long way since the conception of its possibilities. We're no longer confined to the limits of our imagination since augmented reality has become a real-life and widely applicable phenomenon.

Augmented reality essentially adds digital elements such as text, images, videos and audio to a screen. When done effectively, it creates the illusion that the holographic content is part of the physical world around the user. Whilst a similar principle to virtual reality (VR), an AR experience does not immerse a user into an artificial environment. Rather, AR alters a users' perception of reality by adding three-dimensional objects, sounds, videos and graphics to their view of reality.

Overlaying superimposed digital content onto a screen can be applied in different ways. Consider Snapchat and Instagram filters that change a user's face into a zebra, cowboy or kitten. Mindlessly opening Google Maps or Waze is a form of interacting with augmented reality. The digital navigation that guides us through streets and across continents is a highly utilised tool.

In order to retrieve and show relevant digital content to users, AR uses computer vision, simultaneous localization, mapping and depth tracking – using sensor data to calculate the distance to objects in the user's field of vision. For instance, when using a navigation app, computer vision processes the location and objects captured by the camera and recognises it. The program then puts labels onto the surface, and the same process happens each time

the user holds the camera in front of the location that has previously been mapped. This is markerless AR.

The applications will provide you a completely new experience of Augmented Reality. And, The app will become more interactive for you.

you will find new features to use in the app. The mobile app will imitate college website in Augmented Reality by making virtual 3D buttons to access the website functionality, make navigators in college and display the places of halls and classes. which will make the app easier to use, more fun to use, more logically, offers a creative world and saves time and effort of the students. It allows to preview the Functionalities of the system to the student virtually. Enabled student to view, interact and experience the app virtually on the digital platform.

Chapter 2

Background

All universities currently need a system to help students and facilitate them in their university life, the system helps to know the schedules of students, know the dates of professors and their whereabouts, help students register subjects, know exam dates, etc.

This system also uses by visitors to know information about the college and may be used by parents to know the dates and more details about the college.

We have created a system for the college, and we have added AR technology to the system to add fun in using the system and make it easier, and by adding this technology the student can go to places in the college such as halls and laboratory rooms as well as library and rooms of professors so not waste the student's time in search of places.

The Student will use back camera to interact with the system.

The Stakeholders of the system are Students , administrator and guests, and guests are doctors , teacher assistants , graduated student , parents of students , newcomers and anyone who wanted to know anything about the college such as college history ,College Departments , college achievements , master fields information, faculty Members by Using a new method by raising up mobile back camera to Capture the sign of the doctor to show brief summary of doctor.

The technologies we use to Develop the system are Dart programming language, Flutter framework, Vuforia engine , unity 3D, visual studio, API network, BloC state managment, Flutter unity package for integration, C for unity, open CV with vuforia framework, GoDaddy Mysql Panel.

We use GoDaddy server to store the data on it.

[utf8]inputenc graphicx

Chapter 3

Specification - (SRS)

A software requirements specification is a document that describes requirements for a software product, program or set of programs. Requirements in the software requirements specification are expressed in normal language and are not concerned with technical implementation. That's what the design documents are for. In software development, the software requirements specification represents the results of the requirements analysis and describes the requirements of the software under development. Though it is traditionally created as a document, it can also be created in different forms, for example – a very simple one – in spoken form.

3.1 Functional Requirements

This section outlines the use cases for each of the users separately. The Admin, the guest and the Student while the Student is main actor in this system. The student login to the system and student will interact with college

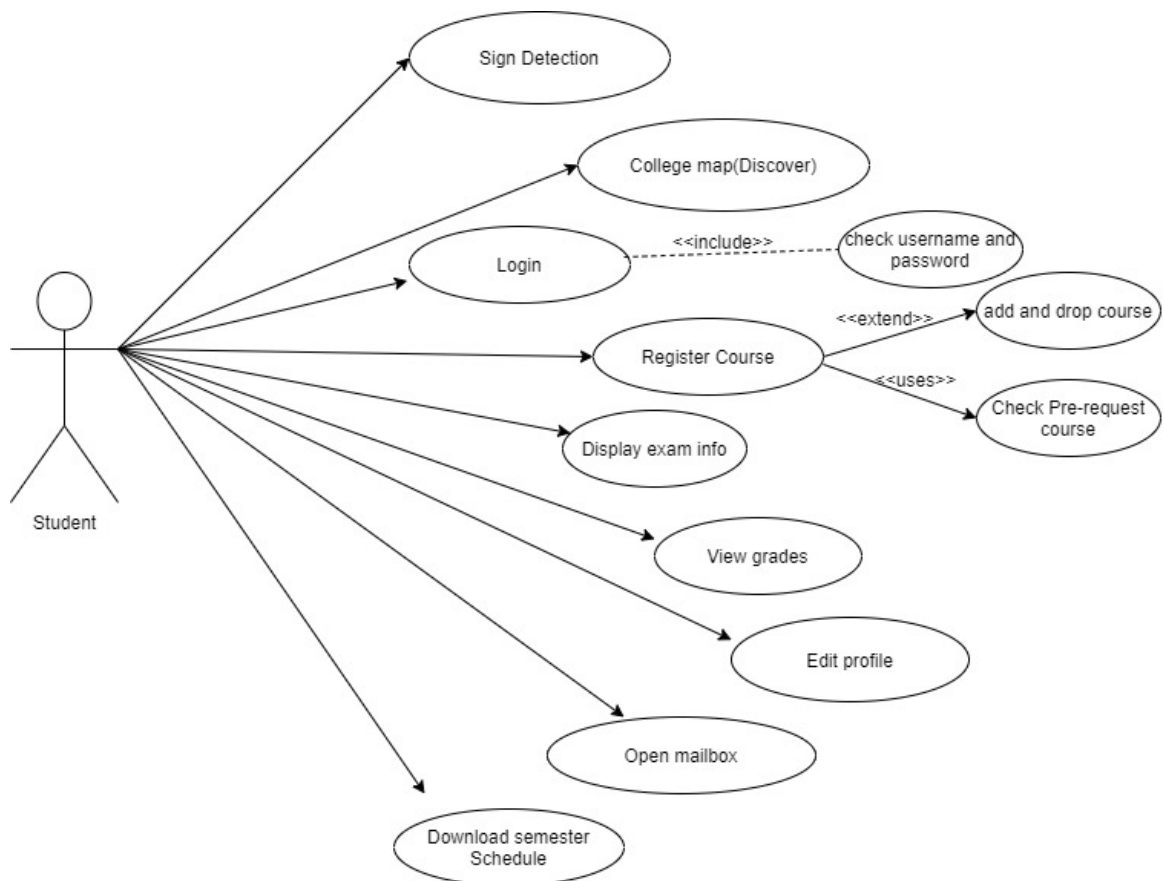


Figure 3.1: Student Use case

system by 3D virtual buttons to access System Functionalities of the system.

1. Register Courses
2. View grade
3. Display exam info (if exists)
4. Edit Profile
5. Open mailbox
6. Download Semester Schedule
7. use Sign Detection

8. use college map

Admin Use case

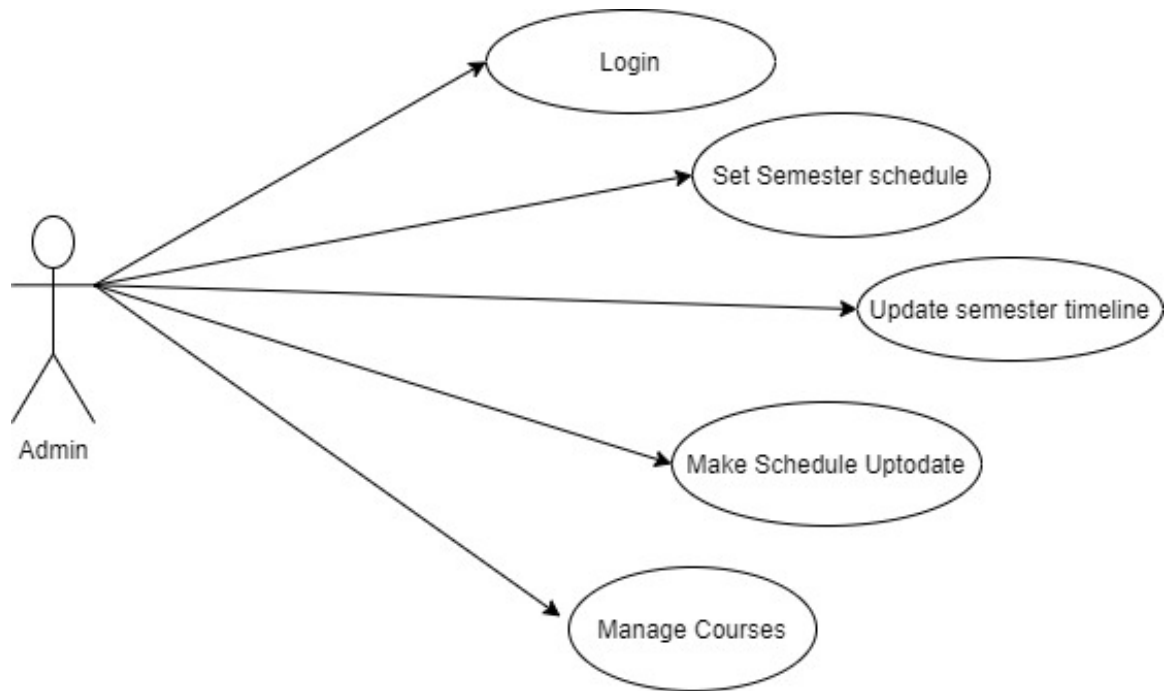


Figure 3.2: Admin Use case

The administrator will login to the system and do the below Functionalities.

1. Set Semester Schedule
2. Manage Courses(add course , delete course)
3. Make Schedule up to date
4. Update semester Timeline

Guest Use case

The administrator will open the System directly without login to access below functionality.

1. Sign Detection
2. College Map
3. Open College info

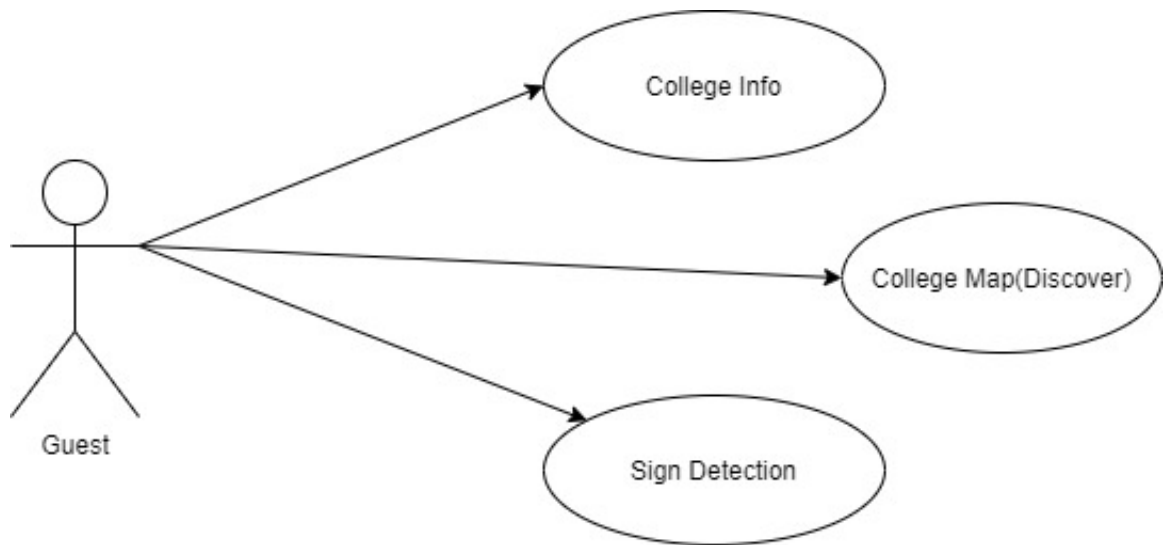


Figure 3.3: Guest Use case

3.2 Non-Functional Requirements

Non-Functional Requirement (NFR) specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software system.

Accessibility

The system must be available in 24 hours in every day with no more than 20 minutes down time per day

Security and Safety

The system cannot affect or damage user device while accessing the system over internet. The system must handle the logout and login through session, the database must be secured to prevent leak or loss data, and the system should be encrypting the data of user during login, Assign certain functions to different modules.

Usability

The system should have user friendly interface to easy use system by users.

Performance

The system must respond of system in time less than 15 seconds for users, the system should be compatible with all mobiles, The system must response the operation notification in time less

Reliability

The system must be designed in modular manner to easy use in software maintenance, the program should be reliable and provide catching of exceptions to prevent system crashes or data validation failure.

Quality**Scalability**

3.3 System analysis and design

Class Diagram

a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

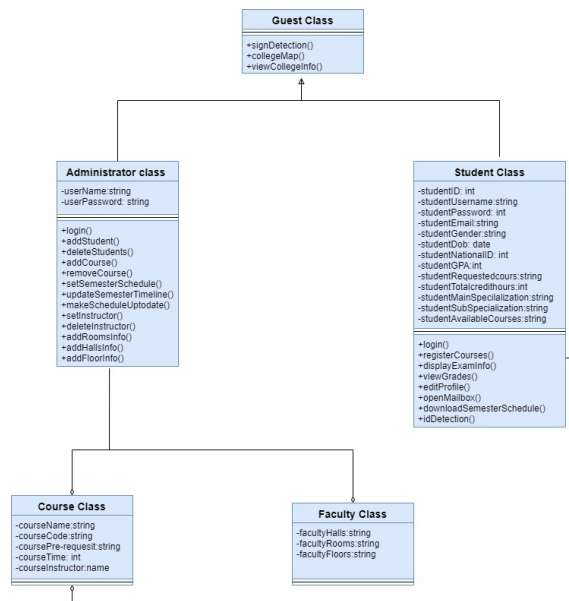


Figure 3.4: Class Diagram

College system class diagram describes the structure of the system classes, their attributes, methods and the relationship among objects. The main classes of the college system are Guest, Administrator, Student, course and faculty.

Object Diagram

Object diagrams describe the static structure of a system at a particular time. They can be used to test class diagrams for accuracy.

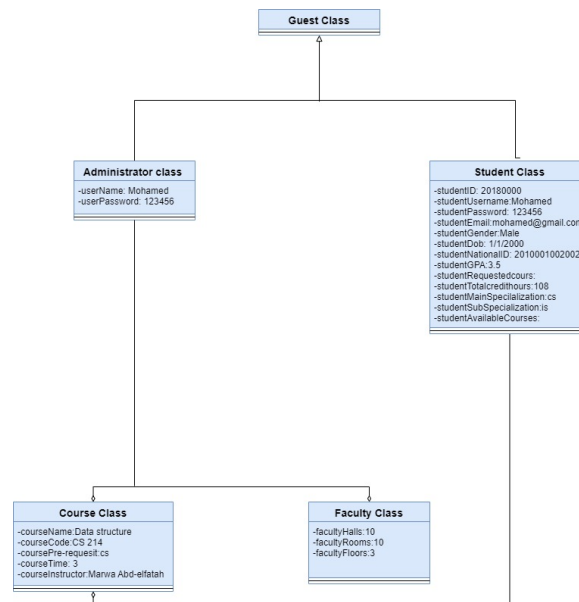


Figure 3.5: object Diagram

This object Diagram Representing the college System. It specially shows the instance process of a System at a particular time in a system. There are objects such as Administrator, Student, Course and Faculty. The Administrator and the Student inherit some functions from the guest. Course and Faculty aggregate with Administrator and course aggregate with student.

Package Diagram

Package diagrams are structural diagrams used to show the organization and arrangement of various model elements in the form of packages.

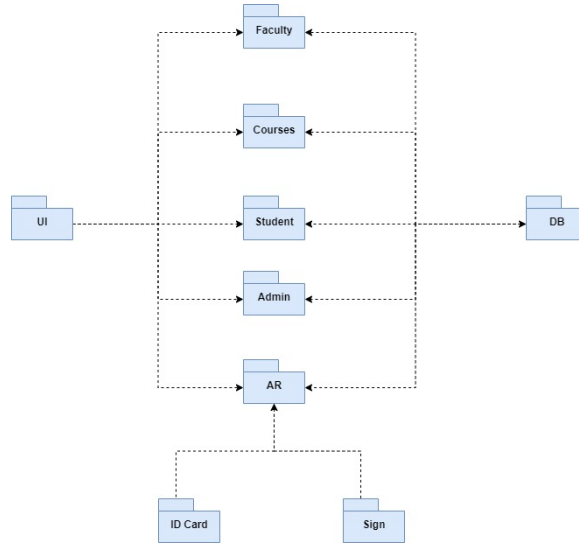


Figure 3.6: Package Diagram

Package diagram is used to show dependency between packages. we can't use the components in the system without user interface to interact with. so, everything in the system dependent on user interface. the packages like Faculty, courses, student and admin dependent on database. we can't get information about them and help them in the system without database. Augmented Reality(AR) feature cannot be used without ID card if the student want to know information about courses or timetable, guest also cannot use AR feature to know about doctors and halls without raise his/her back camera to a doctor sign or hall or room sign. we sure need database in AR feature to get the information from it.

Use Case Diagram

To model a system, the most important aspect is to capture the dynamic behavior. Dynamic behavior means the behavior of the system when it is running/operating. Only static behavior is not sufficient to model a system rather dynamic behavior is more important than static behavior. The internal and external agents are known as actors. Use case diagrams consists of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system. The purpose of use case diagram is to capture the dynamic aspect of a system.



Figure 3.7: Usecase Diagram

The relationship between and among the actors and the use cases

Guest :use cases of guest are view college information , open sign detection and open college map.

Student :use cases of student are Open sign detection ,open college map ,Register courses ,Display exam info ,view grades ,Edit profile , open mailbox and download semester schedule.

Admin :use cases of admin are Set semester schedule , make schedule up to date and manage courses

Activity Diagram

- Activity diagram is another important diagram in UML to describe the dynamic aspects of the system.
- Activity diagram is basically a flowchart to represent the flow from one activity to another activity.
- The activity can be described as an operation of the system.
- The control flow is drawn from one operation to another.
- It captures the dynamic behavior of the system.
- Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system.

Student Activity Diagram

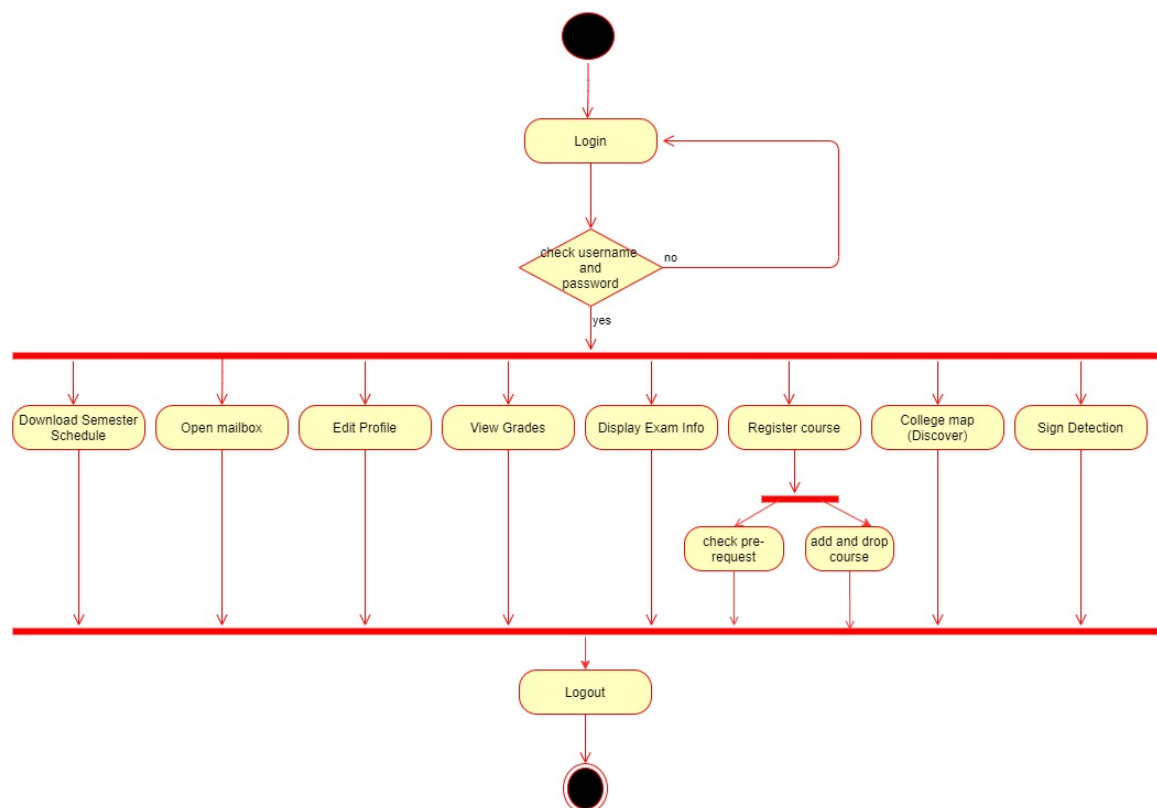


Figure 3.8: Student Activity Diagram

This is the student activity diagram of college app which shows the flow of student

activity, first admin login to the system by using username and password then check validity of input data before accessing the system. after validation the student will use app via back camera . The camera should detect student's ID card so the 3D buttons appear on the screen and the student could interact with system by this 3D buttons and accessing functionalities of the system like edit profile and view grades.

Admin's Activity Diagram

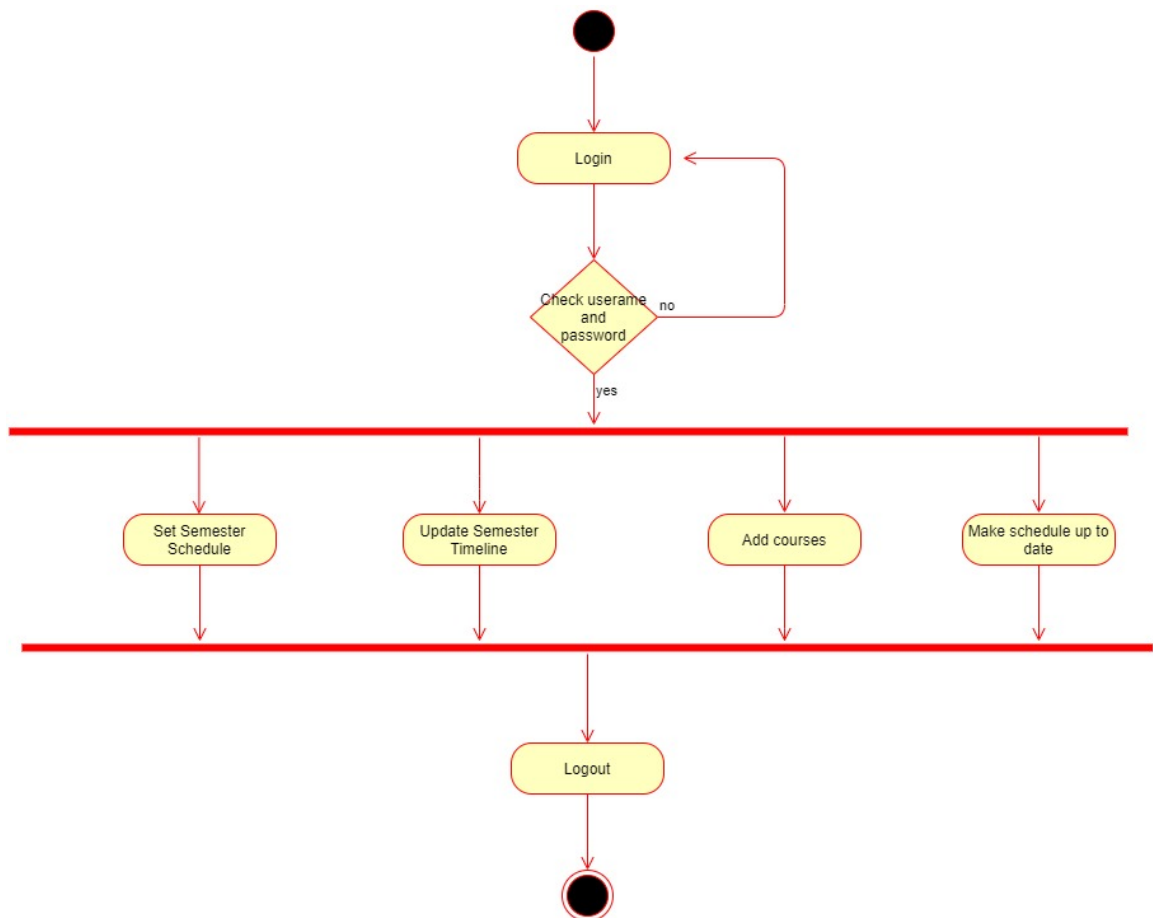


Figure 3.9: Admin Activity Diagram

This is the admin activity diagram of college app which shows the flow of admin activity, first admin login to the system by user name and password then check validity of input data. after validation the admin could access his functionalities like manage courses .

Guest's Activity Diagram

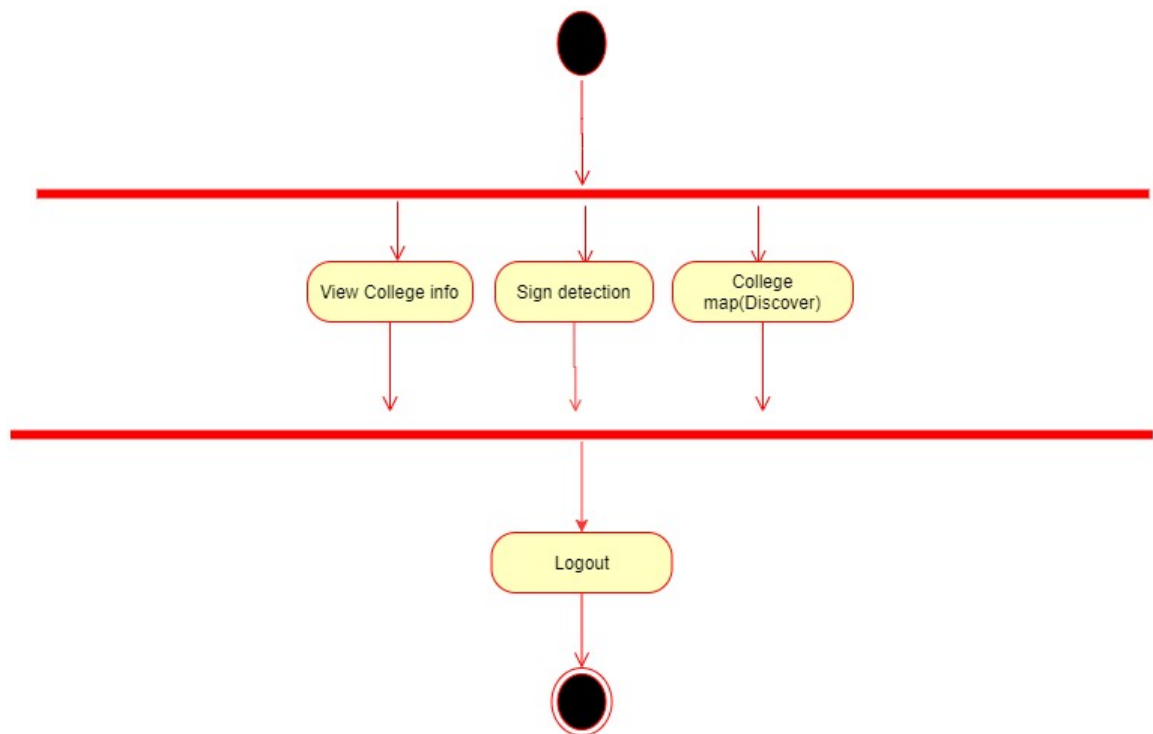


Figure 3.10: Guest Activity Diagram

This is the guest activity diagram of college app which show the flow of guest activity, first guest open the app directly without login activity but he can access limited functionalities such as View college info , open college map and sign detection .

State Machine Diagram

- A State (notation: rounded-rectangle a.k.a. "roundangle") represents a condition or situation during the life of an object during which it satisfies some condition, performs some activity, or waits for some event.
- An State Machine diagram is a dynamic behavioral diagram that shows the sequences of States that an object or an interaction go through during its lifetime in response to
- Events (a.k.a. "Triggers"), which may result in side-effects (Actions).
- The purpose of State Machine diagrams is to specify dynamic system behaviors for time-critical, mission-critical objects

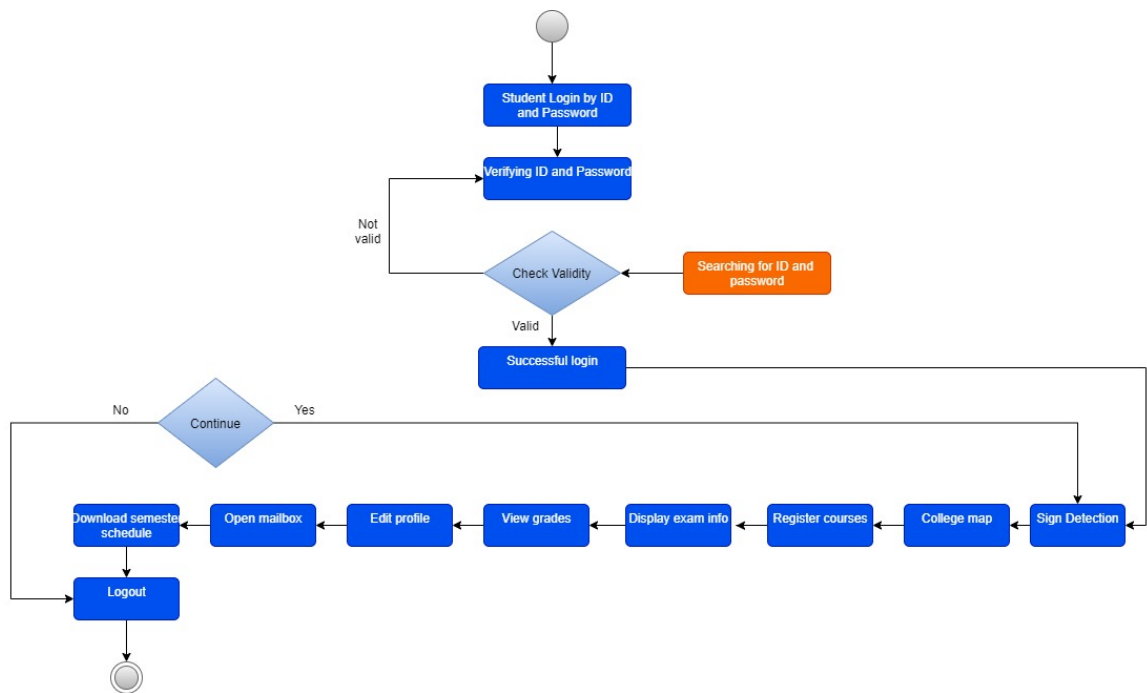


Figure 3.11: Student State machine Diagram

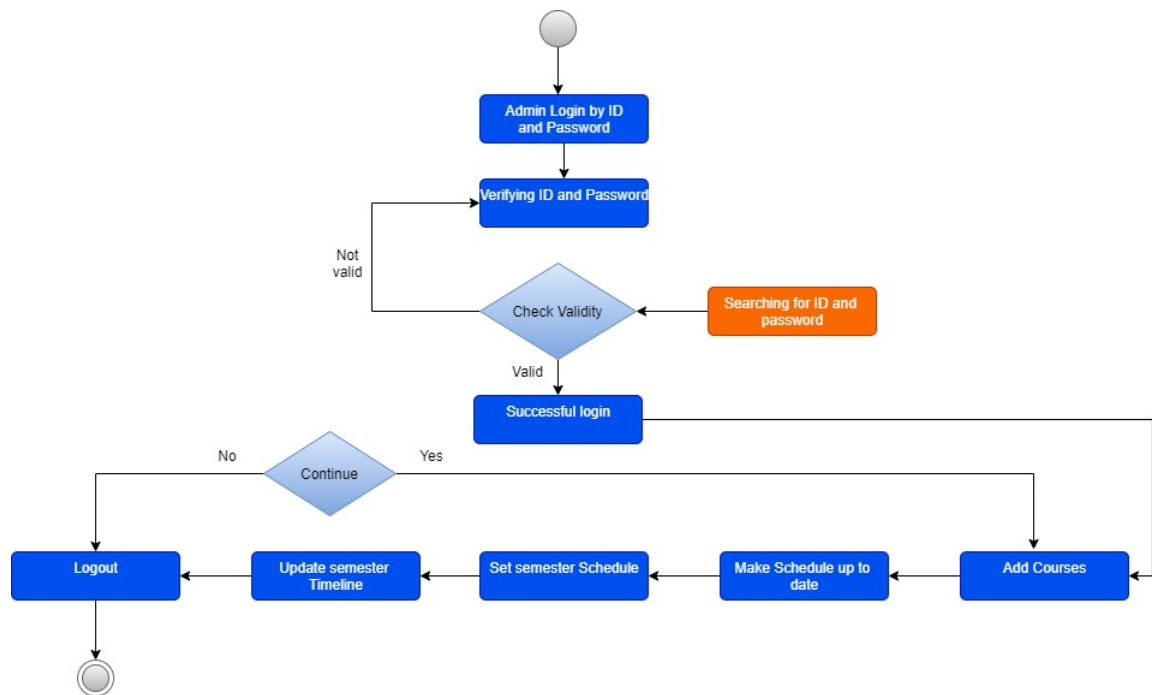


Figure 3.12: Admin State machine Diagram

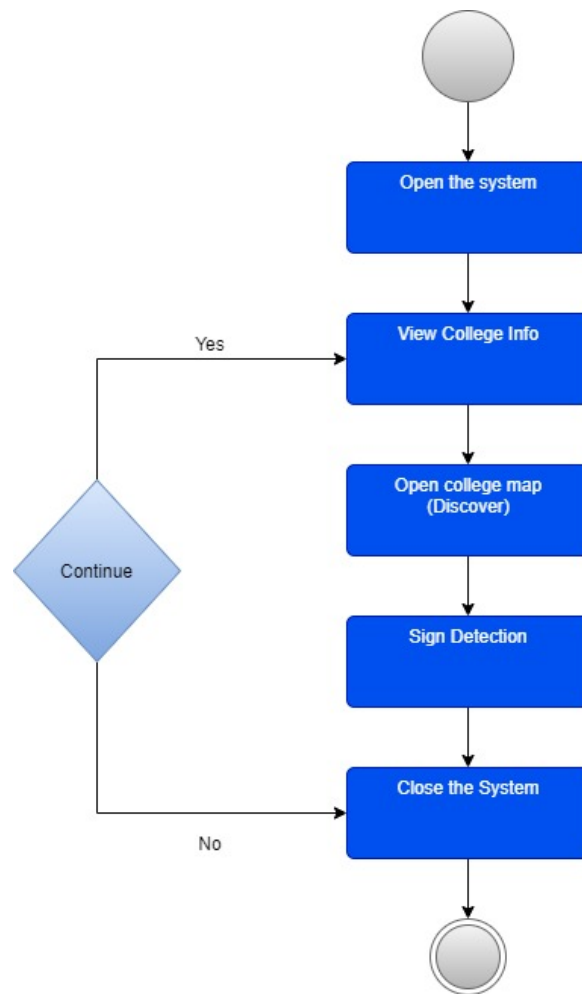


Figure 3.13: Class Diagram

Sequence Diagram

UML Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when.

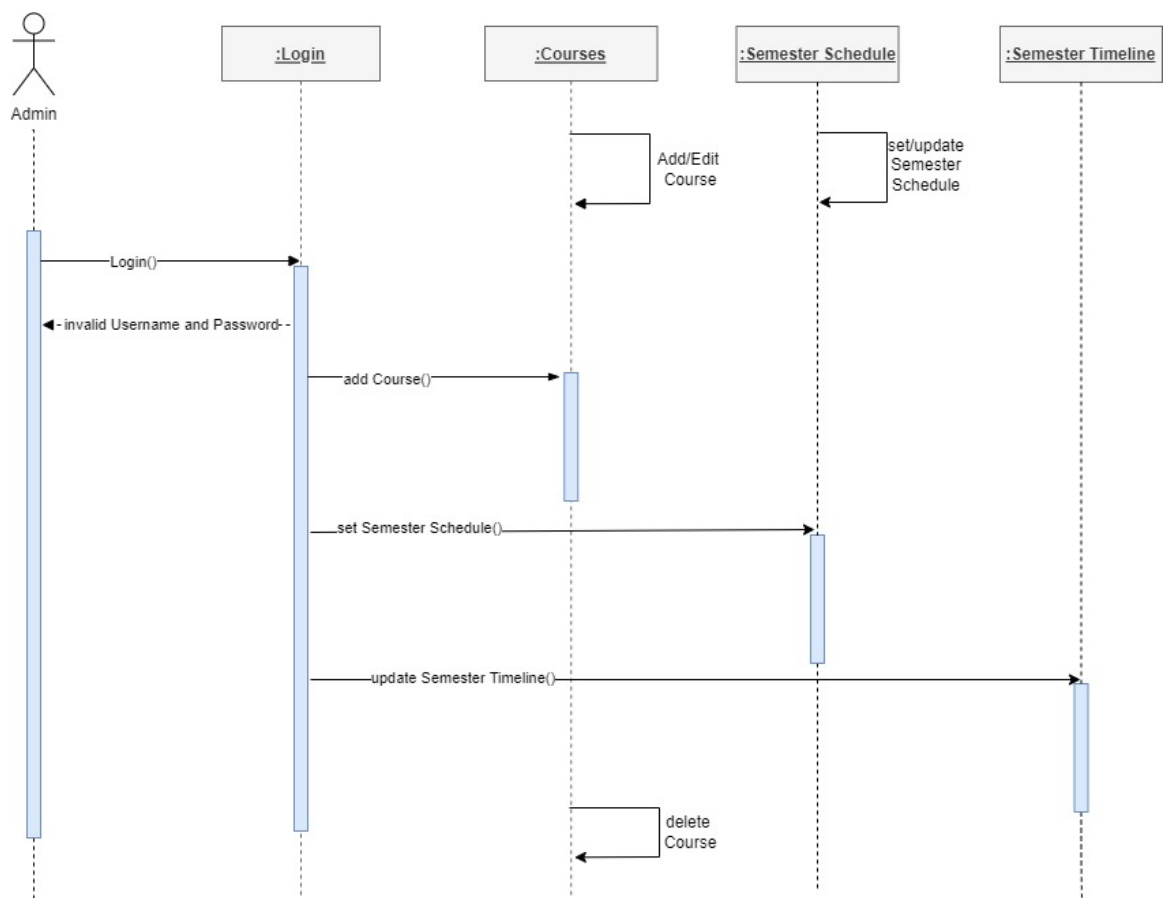


Figure 3.14: Admin Sequence Diagram

The administrator must login to the system so that he/she can use it. The system checks the username and password and if they are invalid, the system will give the administrator a message and cannot login. If the administrator login successfully to the system so he/she can manage courses, manage semester schedule and update semester timeline.

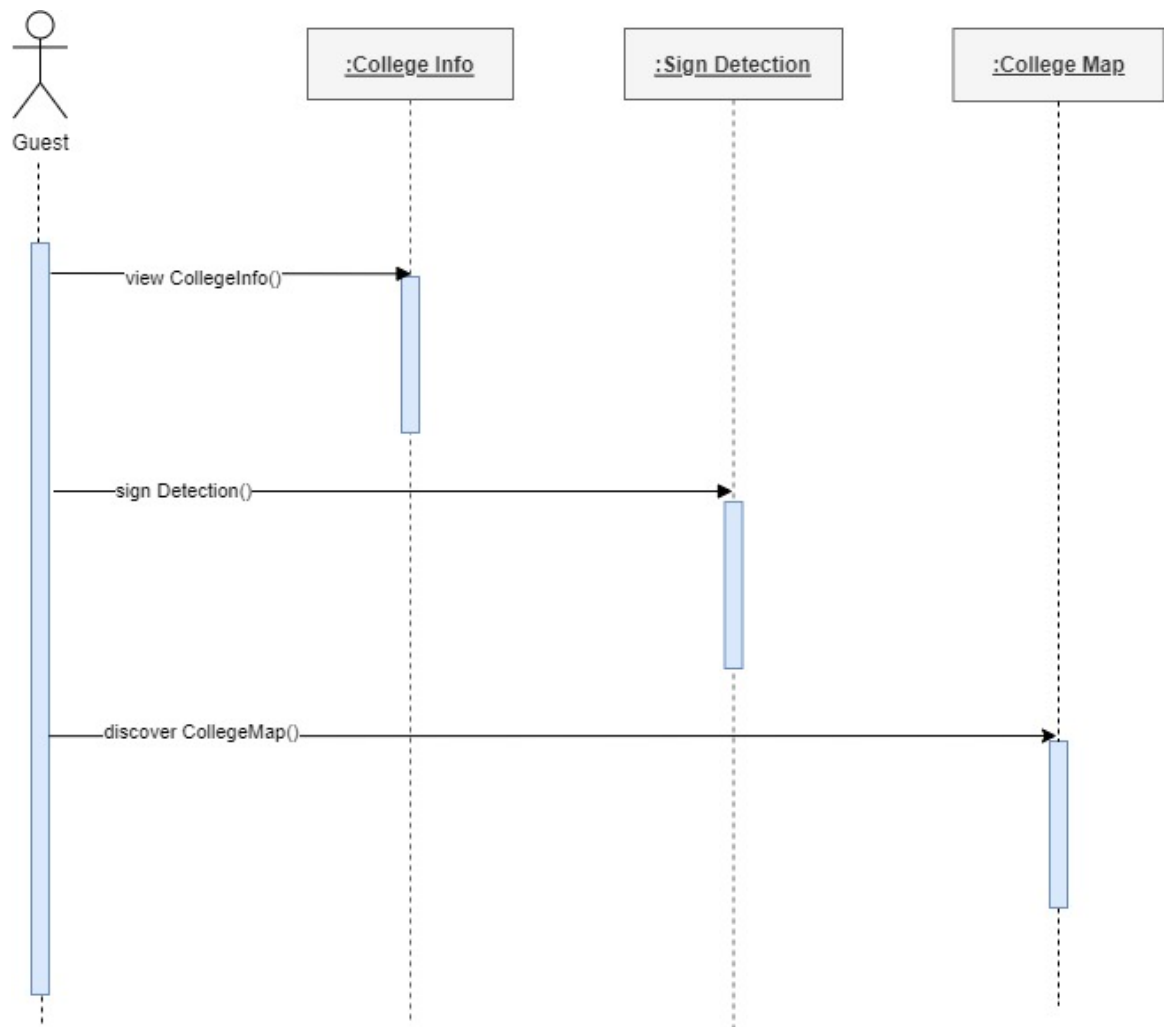


Figure 3.15: Guest Sequence Diagram

The guest don't need to login to system to use it. The guest can use the AR feature, he/she can view college, detect doctor's signs and rooms and halls' signs and the guest can discover college map.

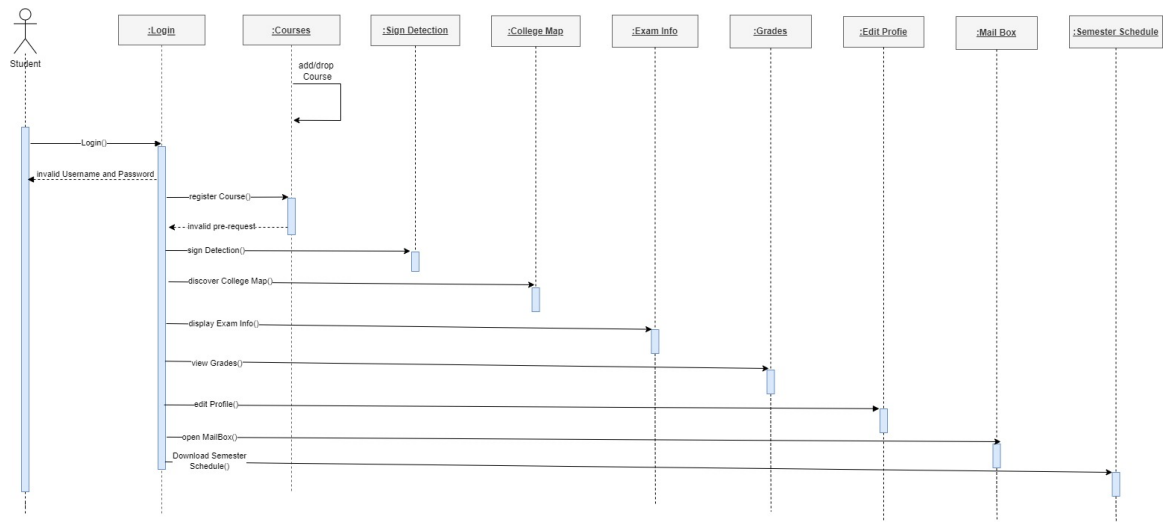


Figure 3.16: Student Sequence Diagram

The student must login to use it. The system checks the username and password and if they are invalid, the system will give the student a message and cannot login. If the student login successfully to the system so he/she can manage his/her courses, view exams information, view grades, he/she can manage his/her profile, open mail box, download semester schedule and use AR feature like detect sign or discover college map.

Communication Diagram

Communication diagrams model the interactions between objects in sequence. They describe both the static structure and the dynamic behavior of a system. Communication diagram can help us to identify potential objects and operations for the College system.

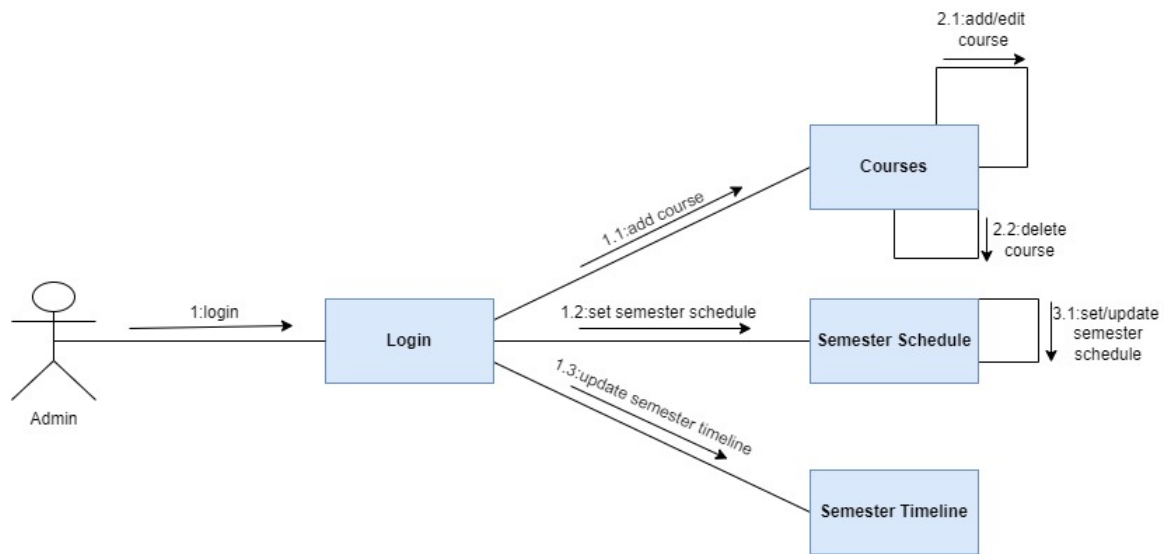


Figure 3.17: Admin Communication Diagram

communication diagram shows the interact between objects to make **admin** manage the system. the admin send a request to login to system and then the admin send a request to manages courses, manage semesters and update semester timeline.

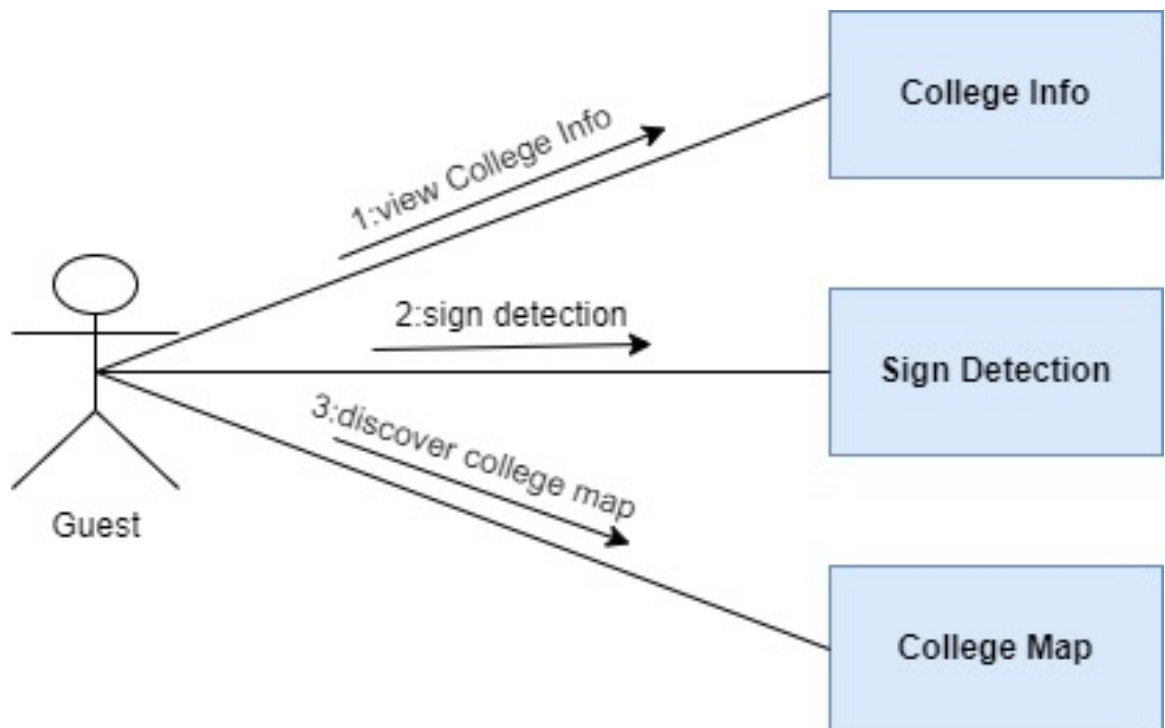


Figure 3.18: Guest Communication Diagram

communication diagram shows the interact between objects to make **the guest** can visit our system and benefits from it. the guest can send a request to the system so he/she can view college information, detect signs and discover college map.

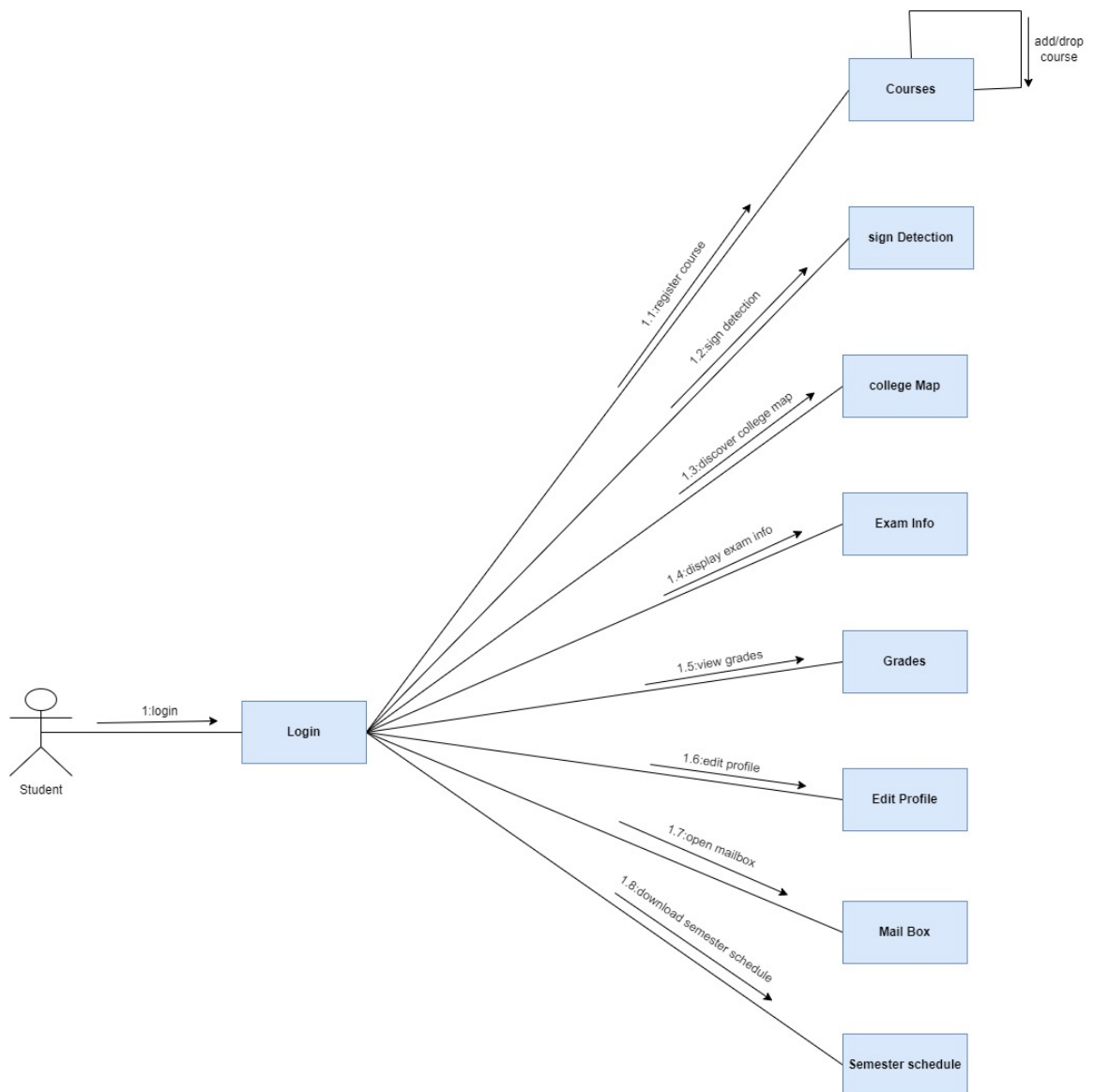


Figure 3.19: Student Communication Diagram

communication diagram shows the interact between objects to make **the student** can use the system and benefits from it. the student can send a request to the system to login and then he/she can send requests to manage courses, profile and semester schedule and can update semester timetable, to know exams information, view grades, open mailbox and download semester schedule. the student can send a request to use AR features to discover college.

Chapter 4

Implementation

4.1 Steps and Problems

The First Part

When we have the Idea of making **Augmented reality** Application, we start to search for the best technique to use, we found out that AR core from google and unity AR is the best.

We preferred using unity Engine because it is easy to create and understand and we studied C in internet Application Course in College so we start to search for Augmented reality in unity and we discovered **3 ways** to made a such our project:

Unity AR-Kit, Vuforia Engine and Manomotion model.

So we tried **Manomotion** and it is a preTraining model that detect hand and has a big dataset, all we need that select which hand move do what.but eventually it was not the best choice to complete the project because of leaks of functionality .

That when we read an article about the latest version of **Vuforia** when they add a pre-Training model that detect object. So we tried it and we find out it was easy to learn and creating a great things with it like filters and Bank simulation on Bank Card and that what we were looking for, we want to simulate College website when we target our ID Cards.

When we start develop the Application we start with touching the ID card with hand to interact with Augmented buttons which appeared on it but we cancel it, it was hard to use and containing a tons of bugs when the hand cover a part of the Card. So we switch to touching the same augmented button but this time from phone screen.

We find out it is more efficient and easy to use. and we containing our development using this technique.that was the part of developing our Augmented Reality environment.

We created the system UI and the buttons did action so the part of coding was called, we start to code the action using **Vuforia and unity built-in libraries** which made the Back-end developing more easy.

We did not implement too much functions.

Now our front and back end interact with each other.

It was the time to use a real data from database.

We start a search for who we could connect **mySql** database to unity.

Unity is using VisualStudio so MySql is not easy to connect like SQL database. So we find out that we had to had a library called **-MySql.Data-** in our project.

We found it it was not that hard. but the problem that the version of it was not the same of our compiling version is was higher so the hard part was to find the perfect version for unity compiler to compile the library. when we add the correct version it worked smoothly and mySql was not hard to learn we learned the insert, select and inner join to get the right data from our database in a couple of days.

Finally we have a full project with intractable component. so now we have to find a good domain to push our database to is, we tried a free hosts and it was not working in a good way, at least we buy a domain from **Godaddy** website until we finished the Project discussion.

The Second Part

was a feature to made a student life in college easier so we made an Augmented reality environment that contains a signs from the college like laps names, doctor's signs and halls sings . We used some of them and made this detectable in our project so that we highlighted the place information, for example if the student use the app and put the camera to the doctor sign he will be able to press a button to know the official hours for this doctor.

We were confused between two options we made the information flowing beside the Sign or appears on the phone as an UI, we choose the second one to make it easy to read to the student.

The last model that unfortunately we could not complete it, we were going to model the

college as a 3D model and simulate the walls and laps to find them if they student get lost or he did not know the way to any lap or hall, that model would take like month of continuance work and effort so we could not made it so we replace it with some description and wide picture of the hall or lap using flutter.

At last but not least, the cooperation is the key of making a good product , our communication and working together made the project we all do our best to make it as real as possible, we are student and we do not have the access to see how the database interact with each other or how it had been structured, we use our imagination to made it like that.

Flutter Challenge

At the beginning of this project, we asked ourselves what's the best choice to implement this mobile app? there were many answers to this question like an android native, iOS native, react native, xamarin, and flutter so the answer to this was a flutter. as Flutter is written in Dart language, which allows compiling a code quicker than JavaScript which react-native based on it. and it was unacceptable to make a college app in one platform like android or iOS only and flutter has its own widget so we do not need to xamarin it's a XAML based so flutter was the best choice for our application.

First, we design our application to display mainly things for guests like dean message and departments and so on, and especially the AR Module for students. The challenge was how to connect flutter with Unity AR Module? we tried to make it through by putting the module in the assets Folder in the flutter project but there is a problem that is the flutter doesn't support unity files so this try failed After a lot of search and search, we found a package in pub.dev.

It's flutter unitywidget but as usual, it doesn't work, actually, its owner didn't know how it works. as he put an incorrect example. it was had a many of opened issues in his GitHub Repo about this example.

but in the final, we found a new package called flutter unity it's working very well and we tested it it's a perfect package but there is a problem. it has an open issue that it accepts only one AR Module and we have at least three AR Module so we should have 3 flutter Apps and 3 projects. it's impossible. so we decided that we will make all of the AR Modules in only one AR Module it was a big challenge to do it, but we did.

There was another solution we had, it was modifying the source code of this package. actually, we did it but in the debug mode the performance was very bad as the application has three AR modules in 3D and the app size changed from 72MB to 390MB it's a very bad for mobile Application so we canceled the solution.

Map Feature

We wanted to make a map of the halls and labs in the college. We try to make this feature by Google maps, but we can not because google maps give us general API. How to go to college, but walking around needs licenses from the college. We try to make this feature by here SDK server, but it allows us to use once for free and then paid for it because this server for commercial use only. We try to make it by short path algorithm, but we can not draw a 3D model of the college and it can not work efficiently. Finally we decided to do it by put 360 degree images of the halls and labs.

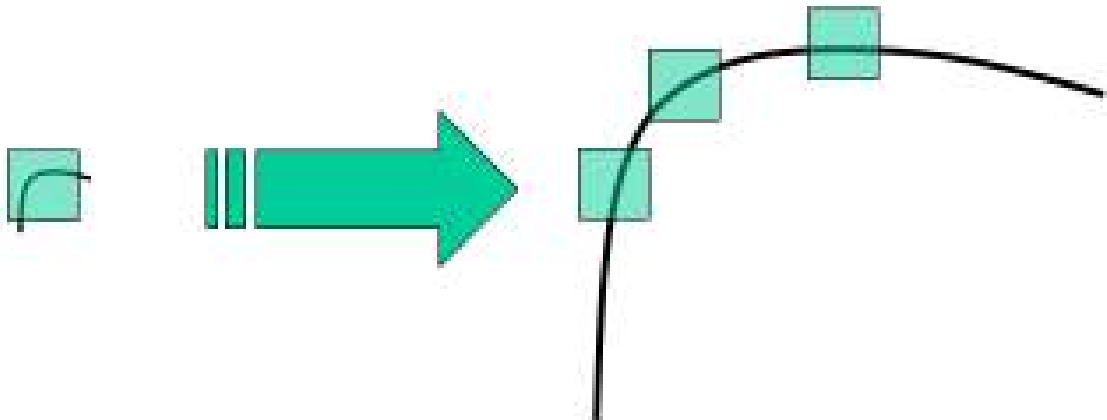
4.2 Vuforia's Algorithm

Scale-invariant feature transform

The scale-invariant feature transform (SIFT) is a computer vision algorithm to detect, describe, and match local features in images, invented by David Lowe in 1999.[1] Applications include object recognition, robotic mapping and navigation, image stitching, 3D modeling, gesture recognition, video tracking, individual identification of wildlife and match moving.

Theory

In last couple of chapters, we saw some corner detectors like Harris etc. They are rotation-invariant, which means, even if the image is rotated, we can find the same corners. It is obvious because corners remain corners in rotated image also. But what about scaling? A corner may not be a corner if the image is scaled. For example, check a simple image below. A corner in a small image within a small window is flat when it is zoomed in the same window. So Harris corner is not scale invariant.

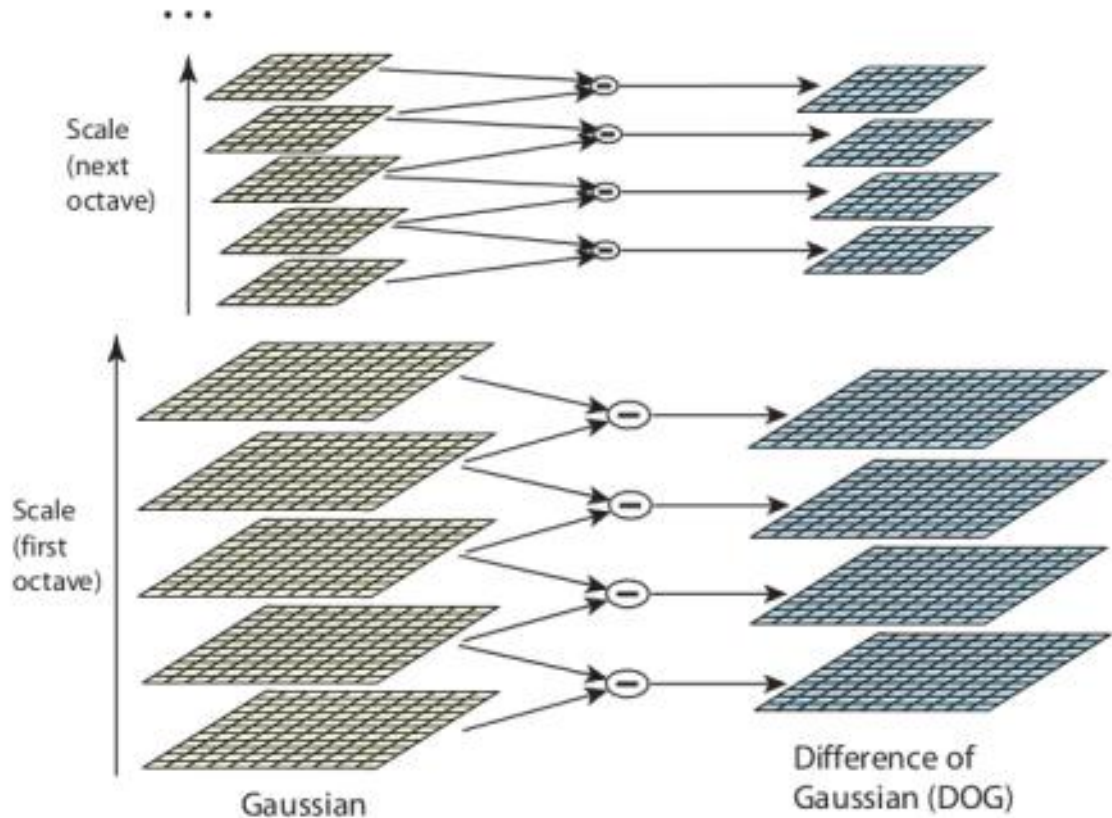


In 2004, D.Lowe, University of British Columbia, came up with a new algorithm, Scale Invariant Feature Transform (SIFT) in his paper, Distinctive Image Features from Scale-Invariant Keypoints, which extract keypoints and compute its descriptors. *(This paper is easy to understand and considered to be best material available on SIFT. This explanation is just a short summary of this paper)*.

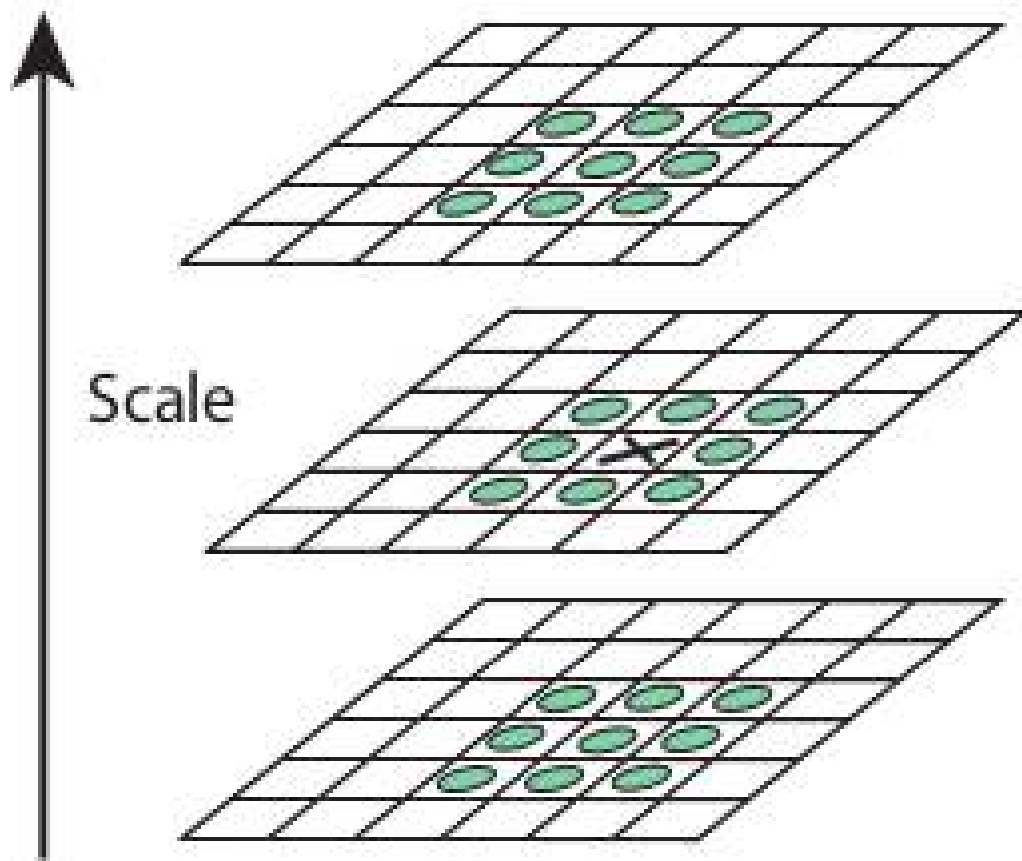
There are mainly four steps involved in SIFT algorithm. We will see them one-by-one.

1. Scale-space Extrema Detection

From the image above, it is obvious that we can't use the same window to detect keypoints with different scale. It is OK with small corner. But to detect larger corners we need larger windows. For this, scale-space filtering is used. In it, Laplacian of Gaussian is found for the image with various σ values. LoG acts as a blob detector which detects blobs in various sizes due to change in σ . In short, σ acts as a scaling parameter. For eg, in the above image, gaussian kernel with low σ gives high value for small corner while gaussian kernel with high σ fits well for larger corner. So, we can find the local maxima across the scale and space which gives us a list of (x,y) values which means there is a potential keypoint at (x,y) at σ scale. But this LoG is a little costly, so SIFT algorithm uses Difference of Gaussians which is an approximation of LoG. Difference of Gaussian is obtained as the difference of Gaussian blurring of an image with two different σ , let it be σ and $k\sigma$. This process is done for different octaves of the image in Gaussian Pyramid. It is represented in below image:



Once this DoG are found, images are searched for local extrema over scale and space. For eg, one pixel in an image is compared with its 8 neighbours as well as 9 pixels in next scale and 9 pixels in previous scales. If it is a local extrema, it is a potential keypoint. It basically means that keypoint is best represented in that scale. It is shown in below image:



Regarding different parameters, the paper gives some empirical data which can be summarized as, number of octaves = 4, number of scale levels = 5, initial = 1.6, $k=2$ etc as optimal values.

2. Keypoint Localization Once potential keypoints locations are found, they have to be refined to get more accurate results. They used Taylor series expansion of scale space to get more accurate location of extrema, and if the intensity at this extrema is less than a threshold value (0.03 as per the paper), it is rejected. This threshold is called `contrastThreshold` in OpenCV. DoG has higher response for edges, so edges also need to be removed. For this, a concept similar to Harris corner detector is used. They used a 2×2 Hessian matrix (H) to compute the principal curvature. We know from Harris corner detector that for edges, one eigen value is larger than the other. So here they used a simple function, If this ratio is greater than a threshold, called `edgeThreshold` in OpenCV, that keypoint is discarded. It is given as 10 in paper.

So it eliminates any low-contrast keypoints and edge keypoints and what remains is strong interest points.

3. Orientation Assignment Now an orientation is assigned to each keypoint to achieve invariance to image rotation. A neighbourhood is taken around the keypoint location depending on the scale, and the gradient magnitude and direction is calculated in that region. An orientation histogram with 36 bins covering 360 degrees is created (It is weighted by gradient magnitude and gaussian-weighted circular window with equal to 1.5 times the scale of keypoint). The highest peak in the histogram is taken and any peak above 80 of it is also considered to calculate the orientation. It creates keypoints with same location and scale, but different directions. It contribute to stability of matching.

4. Keypoint Descriptor Now keypoint descriptor is created. A 16x16 neighbourhood around the keypoint is taken. It is divided into 16 sub-blocks of 4x4 size. For each sub-block, 8 bin orientation histogram is created. So a total of 128 bin values are available. It is represented as a vector to form keypoint descriptor. In addition to this, several measures are taken to achieve robustness against illumination changes, rotation etc.

5. Keypoint Matching Keypoints between two images are matched by identifying their nearest neighbours. But in some cases, the second closest-match may be very near to the first. It may happen due to noise or some other reasons. In that case, ratio of closest-distance to second-closest distance is taken. If it is greater than 0.8, they are rejected. It eliminates around 90 of false matches while discards only 5 correct matches, as per the paper.

This is a summary of SIFT algorithm. For more details and understanding, reading the original paper is highly recommended.

The first goal of a marker detection process is to find the outlines of potential markers, and then to deduce locations of marker's corners in the image. In addition, detection system needs to confirm that it really is a marker and decipher its identity. Finally, the system calculates the pose using the information from the detected marker location.

The basic marker detection procedure consists of the following steps:

1. Image acquisition x Acquisition of an intensity image.
2. pre-processing low level image processing line detection/line fitting Detection of the corners of the marker.
3. Detection of potential markers and discard of obvious non-markers Fast rejection of obvious non-markers and fast acceptance test for potential markers.

4. Identification and decoding of markers template matching (template markers) Decoding (data markers).

5. Calculation of the marker pose estimation of marker pose.

Iterative pose calculation for accurate pose.

The image acquisition step is actually a separate process; it just provides the image for the marker detection process. Marker detection pipelines may differ from this template. Usually Euclidean metrics is used, but another kind of metrics could be applied as well. In normalised cross-correlation the dissimilarity value is Where \bar{x} is mean of the marker's pixel values and \bar{y} is the template's mean pixel value. As a system needs to match the detected marker against each template four times, it is clear that the larger the set of markers the system uses, Therefore, template markers are inefficient in practice if the system requires a large set of markers. By using smart algorithms and other sensors such as accelerometers and gyroscopes the device can keep the augmented elements aligned with the image of the real world.

$$D = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2} \sqrt{\sum_i (y_i - \bar{y})^2}},$$

Figure 4.1: Scale-invariant equation

4.3 Main Functionality and Results

```
public MySqlConnection makeConn()
{
    MySqlConnectionStringBuilder ConnSTR = new MySqlConnectionStringBuilder();

    ConnSTR.Server = "50.63.140.108";
    ConnSTR.Port = 3306;
    ConnSTR.UserID = "kher";
    ConnSTR.Password = "00000";
    ConnSTR.Database = "GP";

    try
    {
        MySqlConnection Conn = new MySqlConnection(ConnSTR.ToString());
        Conn.Open();
        Debug.Log("connected");
    } catch (Exception ex)
    {
        debug.log(ex);
    }

    return Conn;
}
```

Figure 4.2: Connect to database Function

We declare the ID number and then manage the connection string and try it to handle any expected error.


```
public void GetStudentRegisteredCourses(GameObject ID)
{
    ConnectionToDB myConn = new ConnectionToDB();
    MySqlConnection con = myConn.makeConn();

    string Q = "Select * from RegisteredCourses where STD_ID = " + ID.name + "";
    Debug.Log(Q);

    MySqlCommand cmd = new MySqlCommand(Q, con);

    MySqlDataReader x = cmd.ExecuteReader();

    while (x.Read())
    {
        Cname.text += x.GetString("CoureName") + "\n \n";
        Chours.text += x.GetString("CreditHours") + "\n \n";
        Ctime.text += x.GetString("LectureTime") + "\n \n";
        Clecturer.text += x.GetString("Lecturer") + "\n \n";
    }

    con.Close();
}
```

Figure 4.3: Show registered courses Function

Using our class ConnctionToDB's function -myConn- we created a connection to database then we read from it the information we need and pass them to new variables to show them to user.

```
public void IsRegistered(GameObject ID)
{
    ConnectionToDB myConn = new ConnectionToDB();
    MySqlConnection con = myConn.makeConn();

    string Q = "SELECT * from RegisteredCourses where STD_ID = " + ID.name + "";

    MySqlCommand cmd = new MySqlCommand(Q, con);

    MySqlDataReader x = cmd.ExecuteReader();
    Debug.Log("This the the Q : " + Q);

    while (x.Read())
    {
        if (x.Read())
        {
            errAlert.SetActive(true);
            Debug.Log("true");
        }
        else if (!x.Read())
        {
            errAlert.SetActive(false);
            Debug.Log("false");
        }
    }

    con.Close();
}
```

Figure 4.4: Check Registered courses Function

First we check if user registered a courses before there is no way to register twice. if not, he will access the page and start to select the courses , whenever he select a course it will be pushed to a list then we insert the list by looping on her until it is empty.

```
public void Logout()  
{  
    SceneManager.LoadScene("ModelsManager");  
}
```

Figure 4.5: Logout Function

login will check out the data that he entered if it is a real data in data base and then responded depended on the results.

```
public void Login(GameObject ID)
{
    ConnectionToDB myConn = new ConnectionToDB();
    MySqlConnection con = myConn.makeConn();

    string UserName = null;
    string Password = null;

    string q = "select * from student where STD_ID = '" + ID.name + "'";

    MySqlCommand cmd = new MySqlCommand(q, con);

    MySqlDataReader x = cmd.ExecuteReader();

    while (x.Read())
    {
        UserName = x.GetString("STD_username");
        Password = x.GetString("STD_password");
    }

    if (SuserName.text == UserName && Spassword.text == Password)
    {
        SceneManager.LoadScene("IDdetectionModel");
        ErrorMsg.SetActive(false);
    }
    else
    {
        ErrorMsg.SetActive(true);
    }
}
```

Figure 4.6: Login Function

logout is simple it made the user back to first page and end his session so he has to login again to access the page.

```
public void GetStudentInfo(GameObject ID)
{
    ConnectionToDB myConn = new ConnectionToDB();
    MySqlConnection con = myConn.makeConn();

    string q = "select * from student where STD_ID = '" + ID.name + "'";

    MySqlCommand cmd = new MySqlCommand(q, con);

    MySqlDataReader x = cmd.ExecuteReader();

    while (x.Read())
    {
        Sname.text = x.GetString("STD_name");
        SID.text = x.GetString("STD_ID");
    }

    con.Close();
}
```

Figure 4.7: Exam info Function
Regular select and store data from database.

```
public void GetStudentInfo(GameObject ID)
{
    ConnectionToDB myConn = new ConnectionToDB();
    MySqlConnection con = myConn.makeConn();

    string q = "select * from student where STD_ID = '" + ID.name + "'";

    MySqlCommand cmd = new MySqlCommand(q, con);

    MySqlDataReader x = cmd.ExecuteReader();

    while (x.Read())
    {
        username.text = x.GetString("STD_username");
        password.text = x.GetString("STD_password");
        nationalID.text = x.GetString("STD_nationalId");
        Sname.text = x.GetString("STD_name");
        Email.text = x.GetString("STD_email");
        EmCalls.text = x.GetString("STD_emergencycontact");
        Address.text = x.GetString("STD_address");
    }

    con.Close();
}
```

Figure 4.8: Show profile Function

```
public void UpdateStudentInfo(GameObject ID)
{
    ConnectionToDB myConn = new ConnectionToDB();
    MySqlConnection con = myConn.makeConn();
    string q = "UPDATE student " +
        "set STD_password = '" + password.text + "'," +
        "STD_email = '" + Email.text + "'," +
        "STD_address = '" + Address.text + "'," +
        "STD_emergencycontact = '" + EmCalls.text + "'" +
        "where STD_ID = '" + ID.name + "'";
    Debug.Log(q);

    MySqlCommand cmd = new MySqlCommand(q, con);
    cmd.ExecuteNonQuery();
}
```

Figure 4.9: Update profile Function

Show and edit profile are both in one page. Some information couldn't be edited. We let the user change the values that can be changed than we update the query in database.

```
public void GetStdInfo(GameObject ID)
{
    ConnectionToDB myConn = new ConnectionToDB();
    MySqlConnection con = myConn.makeConn();

    string q = "select * from student where STD_ID = '" + ID.name + "'";

    MySqlCommand cmd = new MySqlCommand(q, con);

    MySqlDataReader x = cmd.ExecuteReader();

    while (x.Read())
    {
        Sname.text = x.GetString("STD_name");
        Sid.text = x.GetInt32("STD_ID").ToString();
        Sgpa.text = x.GetFloat("STD_GPA").ToString();
        Smajor.text = x.GetString("Main_specialization");
        Shours.text = x.GetInt32("STD_hours").ToString();
    }
}
```

Figure 4.10: Show basic Information Function


```

public void GetStudentTranscript(GameObject ID)
{
    ConnectionToDB myConn = new ConnectionToDB();
    MySqlConnection con = myConn.makeConn();

    string Q = "SELECT courses.name As Cname, academic_record.status, academic_record.total, academic_record.grade " +
        "FROM academic_record " +
        "INNER JOIN courses " +
        "on academic_record.course = courses.code " +
        "where students = '" + ID.name + "'";
    Debug.Log(Q);

    MySqlCommand cmd = new MySqlCommand(Q, con);

    MySqlDataReader x = cmd.ExecuteReader();

    while (x.Read())
    {
        Cname.text += x.GetString("Cname") + "\n\n";
        Cstatue.text += x.GetString("status") + "\n\n";
        Ctotaldegree.text += x.GetString("total") + "\n\n";
        Cgrade.text += x.GetString("grade") + "\n\n";

        Debug.Log("this is Q inside loop = '" + Cname.text + "' and '" + Cstatue.text + "' and '" + Ctotaldegree.text + "' and '" + Cgrade.text + "'");
    }

    Debug.Log("this is Q = '" + Cname.text + "' and '" + Ctotaldegree.text + "' and '" + Cgrade.text + "'");
    con.Close();
}

```

Figure 4.11: Show Transcript Function

First function in Figure 4.9 will show the user his basic information like name and GPA and total hours as a header the ta table. then we merge two tables from database to show the right information that the user should know it and a simple query that select and show data from database.



Figure 4.12: College Logo

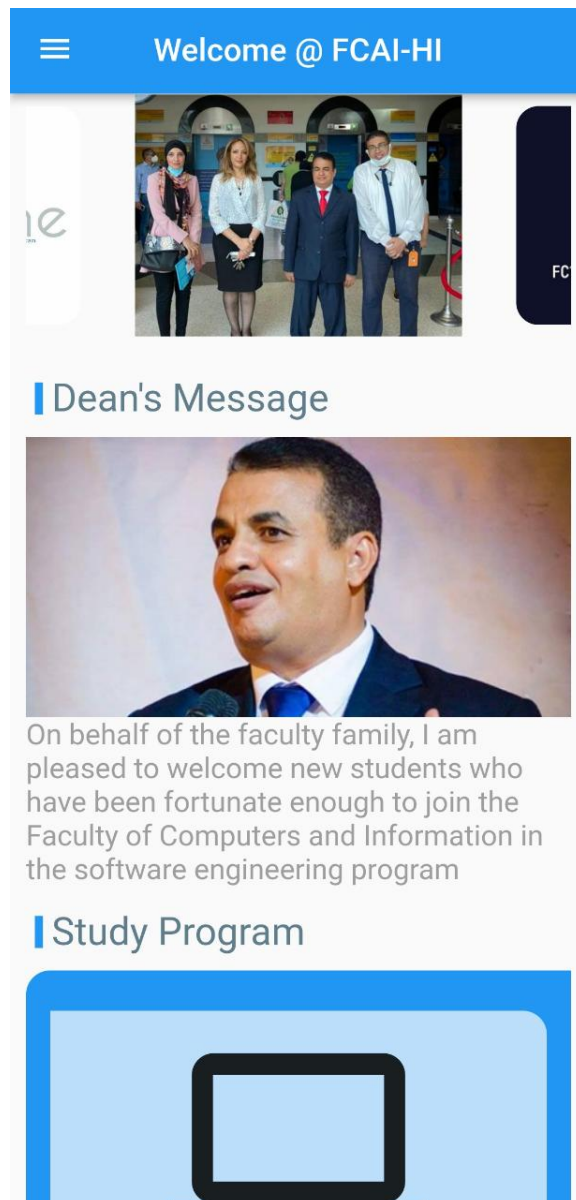


Figure 4.13: Home

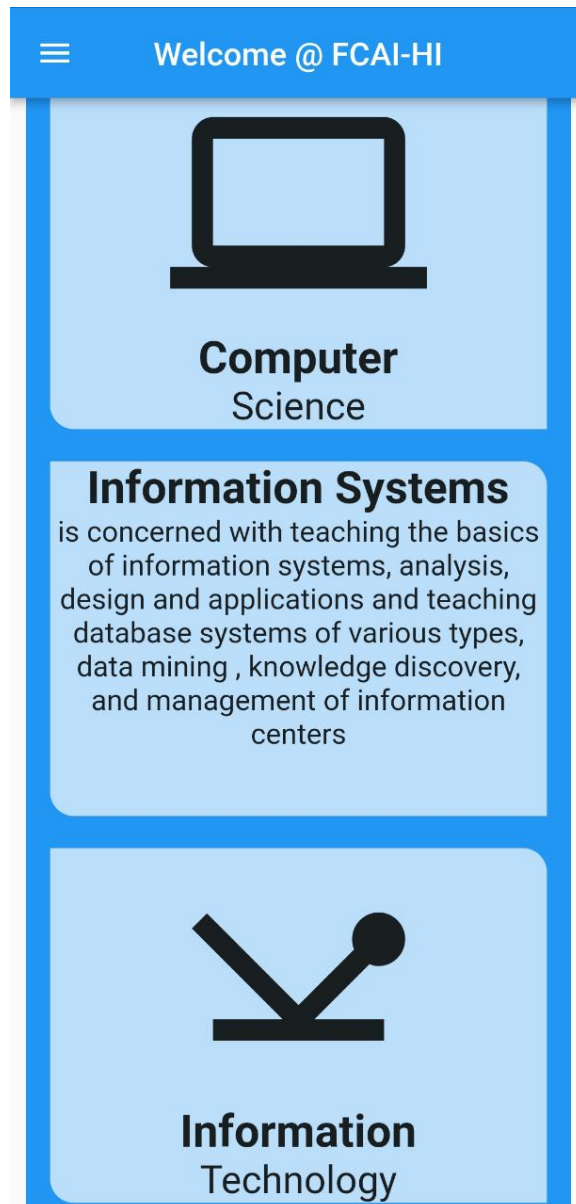


Figure 4.14: Home

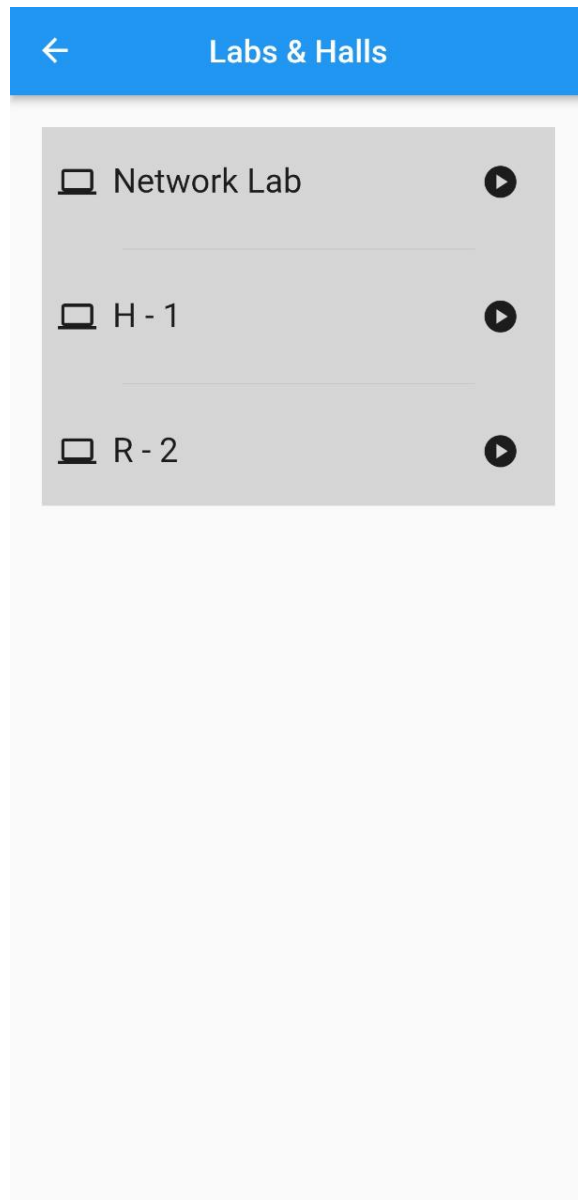


Figure 4.15: Labs and Halls

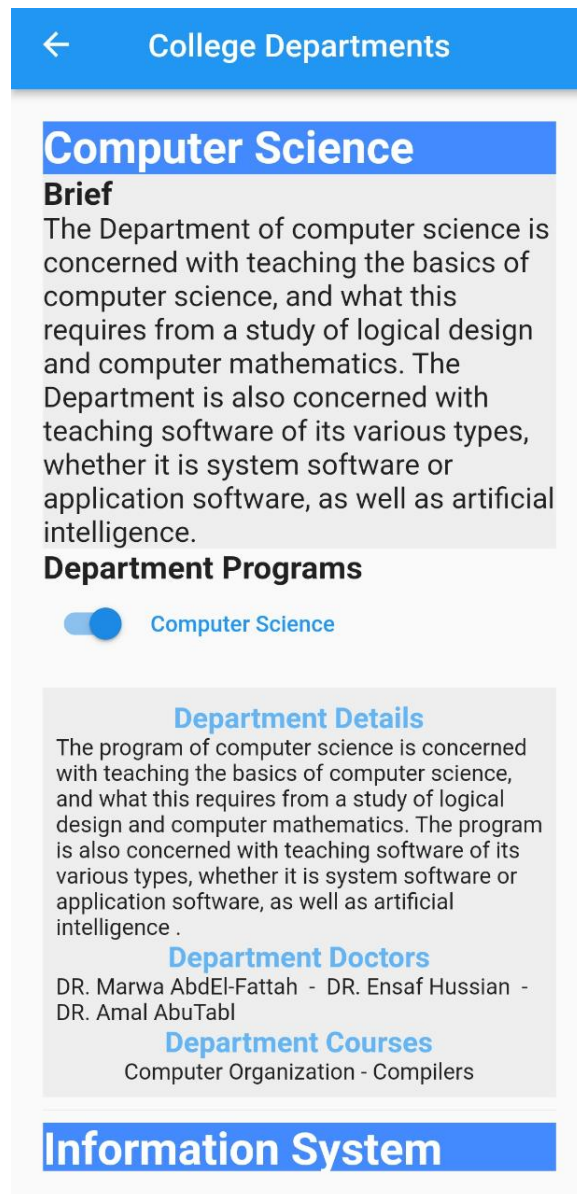


Figure 4.16: Information about College Departments

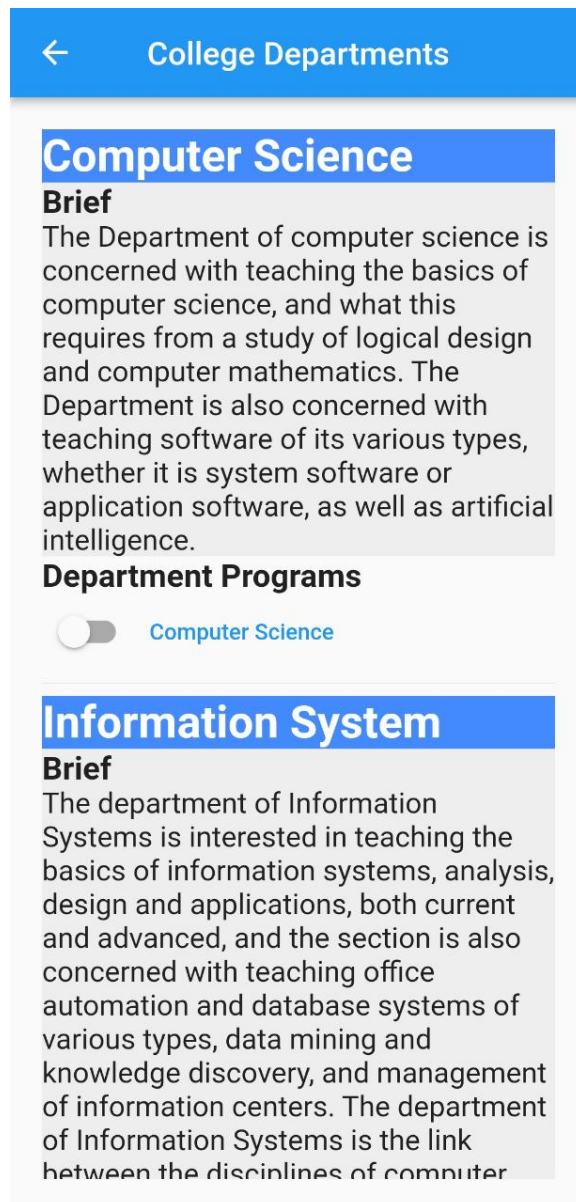


Figure 4.17: Information about College Departments

Welcome To Student Panel

Profile



Discover



Figure 4.18: Welcome To Student Panel
Student choose if he or she want to use the system or discover with AR.

Please Login To continue

Back

Student Profile 

UserName

Password

Login

Figure 4.19: Login To Profile
Student should put his or her Username and password.

Please Login To continue

Back

Student Profile 

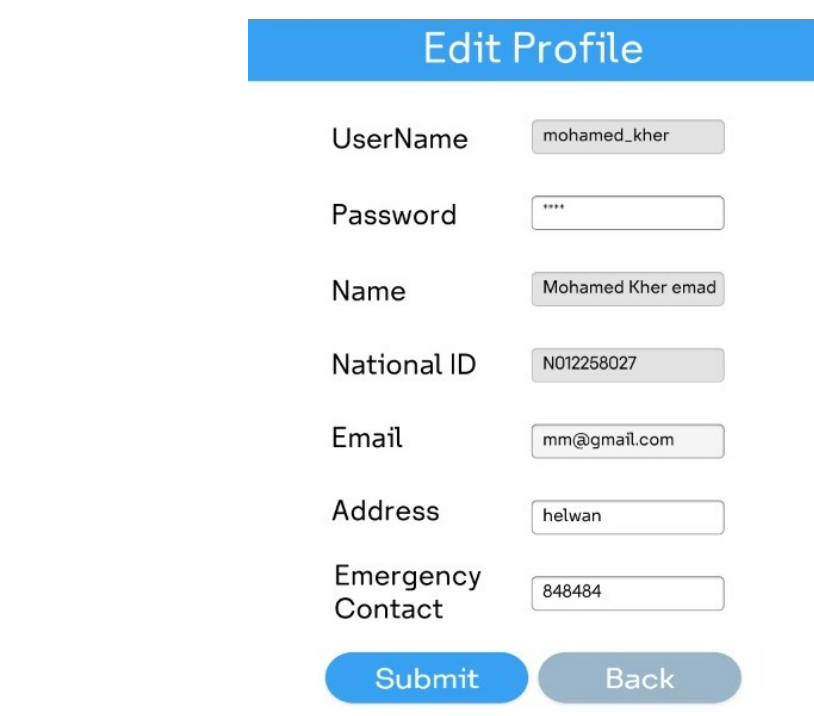
dhdgdvdhdv

Login

Your Username Or Password is incorrect , please if you forgot your login info contact with students Affairs as soon as possible

Figure 4.20: Check Username and Password

The System will check the user name and password, if there are incorrect the system notify the user, if there are correct the student login successfully to the system.



The image shows a web form titled "Edit Profile" with a blue header. The form contains several input fields for personal information, each with a label on the left and a text input box on the right. The fields are: Username (mohamed_kher), Password (masked with four asterisks), Name (Mohamed Kher emad), National ID (N012258027), Email (mm@gmail.com), Address (helwan), and Emergency Contact (848484). At the bottom of the form are two buttons: a blue "Submit" button and a grey "Back" button.

Edit Profile	
UserName	mohamed_kher
Password	****
Name	Mohamed Kher emad
National ID	N012258027
Email	mm@gmail.com
Address	helwan
Emergency Contact	848484
Submit Back	

Figure 4.21: Edit Profile
Student can edit his or her personal information.

Student Name : Mohamed Kher emad		Total GPA : 2.31	
Student ID : 20180727	Major : Computer Science	Total Hours : 102	
Course Name	Status	Total Degree	Grade
Professional Ethics	PASS	84	B+
English-1	PASS	66	C
English-2	PASS	61	D+
Human Rights	PASS	68	C
Fundamentals of Economics	PASS	72	C+
Pattern Recognition	PASS	56	D
Database System-2	PASS	82	B+
Computer Graphics-1	PASS	54	D
Multimedia	PASS	67	C
Concepts of Programming	PASS	50	D
Computer Organization	FAIL	48	F
Compilers	FAIL	49	F
Probability and Statistics			
Back			

Figure 4.22: Show Transcript
Student can view his or her transcript to see grades,gpa and more.

Register your courses

Course Name	Credit Hours	Time	Lecturer	
CS313 Programming-3	3	sunday 8:10AM	Lecturer	<input checked="" type="checkbox"/>
MA 214 Mathematics -3	3	sunday 12:2AM	Lecturer	<input checked="" type="checkbox"/>
IS 321Project Management	3	monday 12:2AM	Lecturer	<input checked="" type="checkbox"/>
CS 471 Parallel Processing	3	tuesday 2:4AM	Lecturer	<input type="checkbox"/>
IT 322 Computer Networks - 2	3	tuesday 10:12AM	Lecturer	<input type="checkbox"/>

SubmitBack

Figure 4.23: Register your Courses

Student can register courses for every semester and submit the registration form.

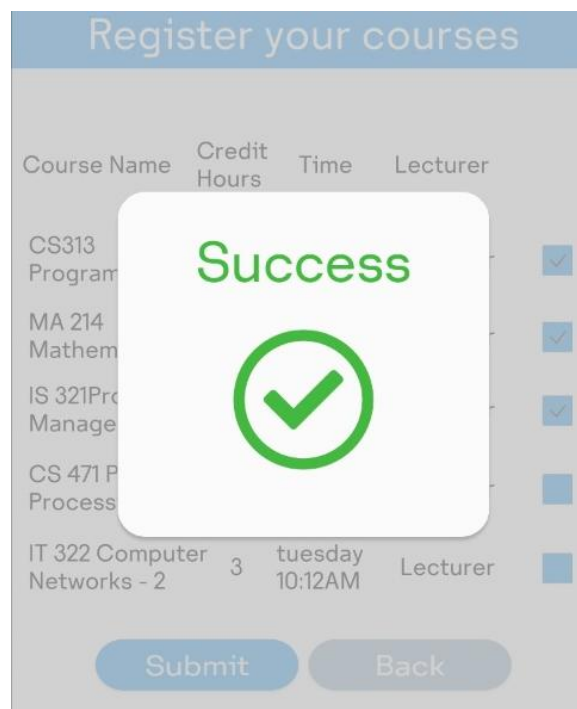


Figure 4.24: Successful Registration

The system will check if the student did not register before, the registration success.

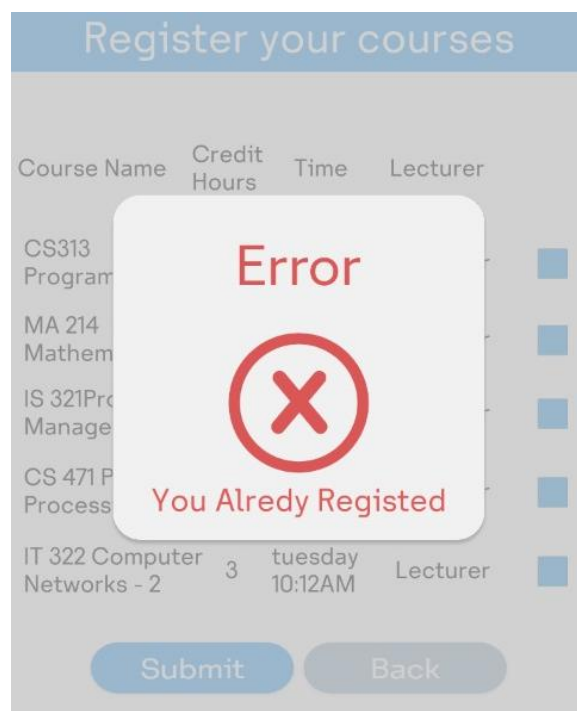


Figure 4.25: Fail Registration
The system found the student registered already.



Course Name	Credit Hours	Time	Lecturer
IS 321Project Management	3	monday 12:2AM	Lecturer
MA 214 Mathematics -3	3	sunday 12:2AM	Lecturer
CS313 Programming-3	3	sunday 8:10AM	Lecturer

Back

Figure 4.26: Show Registered Courses
Student can view his or her registered courses.

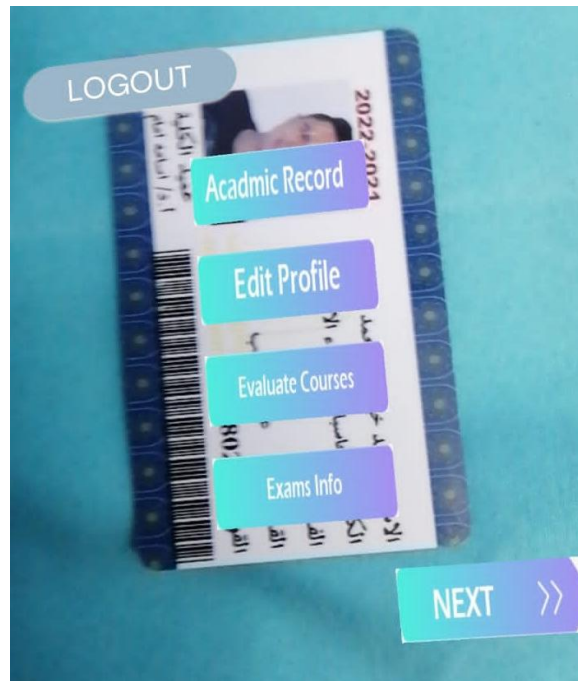


Figure 4.27: ID detection

Student can use his or her personal id to show information through it.



Figure 4.28: Detect Sign

Student can discover the signs in the college.

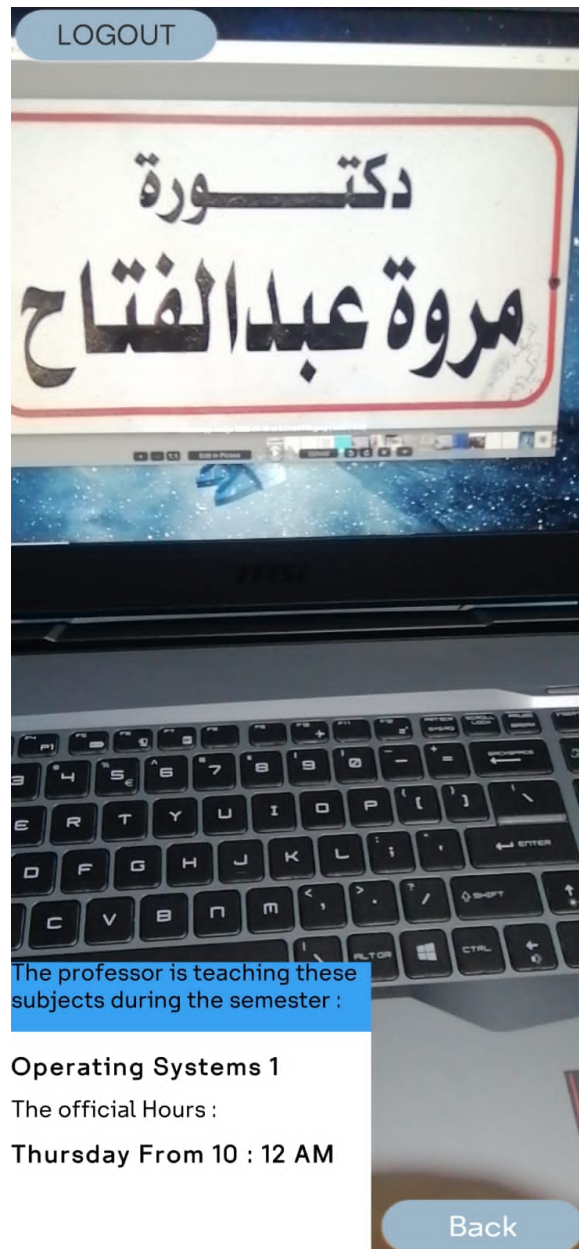


Figure 4.29: Show Sign Information
Student can view the information about each sign in the college.

Chapter 5

Conclusions and Future work

5.1 Conclusion

And in the end, we hope to have achieved what we wanted.

We want to make the use of the college system much easier and add fun to it by using AR technology. Our system help students, administrator and guests, guests are doctors ,teacher assistants , graduated student , parents of students , newcomers and anyone who wanted to know anything about the college such as college history ,College Departments , college achievements , master fields information and faculty Members. Our system use new method with back mobile camera.

5.2 Future Work

We want to improve the non functional requirements like performance, security and usability. we want to add more AR features to the system. we hope to make better map of the college That we couldn't do. Help students by let the system make perfect time table without gaps So we don't waste students time.

Appendix A

Collected materials from

For developing Augmented Reality model :

<https://docs.unity3d.com/Manual/index.html>

<https://library.vuforia.com/sites/default/files/references/unity/classVuforia11VirtualButtonBehaviour>

<https://learn.unity.com/tutorial/recorded-video-session-creating-ar-content-with-vuforia>

<https://www.youtube.com/c/LorantCsonka>

<https://www.scnsoft.com/case-studies/development-of-a-unity-3d-plugin-for-an-ar-content-creation-app-in-a-digital-signage-solution>

https://docs.opencv.org/4.x/da/df5/tutorial_py_intro.html

For developing UI in unity :

<https://www.youtube.com/c/Tarodev>

<https://www.youtube.com/c/CocoCode>

<https://docs.unity3d.com/Manual/index.html>

To import MySQL into project :

<https://www.red-gate.com/simple-talk/development/dotnet-development/using-mysql-unity/>

<https://devepaper.com/unity-connection-to-mysql-database-latest-details/>

https://www.dllme.com/dll/files/mysql_data.dll.html

<https://dev.mysql.com/doc/>

Searching for answers :

<https://stackoverflow.com/>

<https://support.unity.com/hc/en-us>

<https://developer.vuforia.com/support>

<https://www.facebook.com/groups/2415642461786858>