

## **UniNexus**

A capstone project submitted to faculty of Industry and Energy Technology,  
New Cairo Technological University, in partial fulfillment of the  
requirements for the Degree of Higher Diploma

### **Information and Communications Technology**

#### **Submitted by**

**Ammar Tarek Mohamed  
Mostafa Essam Badr  
Omar Hatem Abdelmonem  
Moaz Osama Gamil  
Abd-ElRahman Mohamed Salah**

**Karim Hazem Ali  
Youssef Salama Salem  
Ziad Mohamed Fawzy  
Youssef Mohamed Ahmed  
Khaled Ibrahim Khodary**

#### **Supervised by**

**Dr. Amany Abdelsamea Saeed**

Information and Communications Technology Department  
Faculty of Industry and Energy Technological, New Cairo Technological University.

**Eng. Hossam Medhat**

Information and Communications Technology Department  
Faculty of Industry and Energy Technology, New Cairo Technological University.

# **STUDENT DECLARATION**

## **Plagiarism**

Plagiarism is a particular form of cheating. Plagiarism must be avoided at all costs and students who break the rules, however innocently, may be penalized. It is your responsibility to ensure that you understand correct referencing practices. As a university level student, you are expected to use appropriate references throughout and keep carefully detailed notes of all your sources of materials for material you have used in your work, including any material downloaded from the Internet. Please consult the relevant unit lecturer or your course tutor if you need any further advice.

## **Student Declaration**

I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.

**Also, I acknowledge that I have received the feedback about my work from the assessor.**

Student <sup>1</sup> signature: -----	Date: / /
Student <sup>2</sup> signature: -----	Date: / /
Student <sup>3</sup> signature: -----	Date: / /
Student <sup>4</sup> signature: -----	Date: / /
Student <sup>5</sup> signature: -----	Date: / /
Student <sup>6</sup> signature: -----	Date: / /
Student <sup>7</sup> signature: -----	Date: / /
Student <sup>8</sup> signature: -----	Date: / /
Student <sup>9</sup> signature: -----	Date: / /
Student <sup>10</sup> signature: -----	Date: / /

## Approval Sheet

### UniNexus

A capstone project submitted to faculty of Industry and Energy Technology, New Cairo Technological University. In partial fulfillment of the requirements for the Degree of Higher Diploma of Information and Communications Technology.

#### Submitted by

- Ammar Tarek Mohamed
- Karim Hazem Ali
- Mostafa Essam Badr
- Moaz Osama Gamil
- Abd-ElRahman Mohamed Salah
- Youssef Salama Salem
- Omar Hatem Abdelmonem
- Ziad Mohamed Fawzy
- Khaled Ibrahim Khodary
- Youssef Mohamed Ahmed

**This capstone project has been approved by the examining committee:**

**Score:**-----

**Name of the Examiners :**

•

**Signature**

-----

•

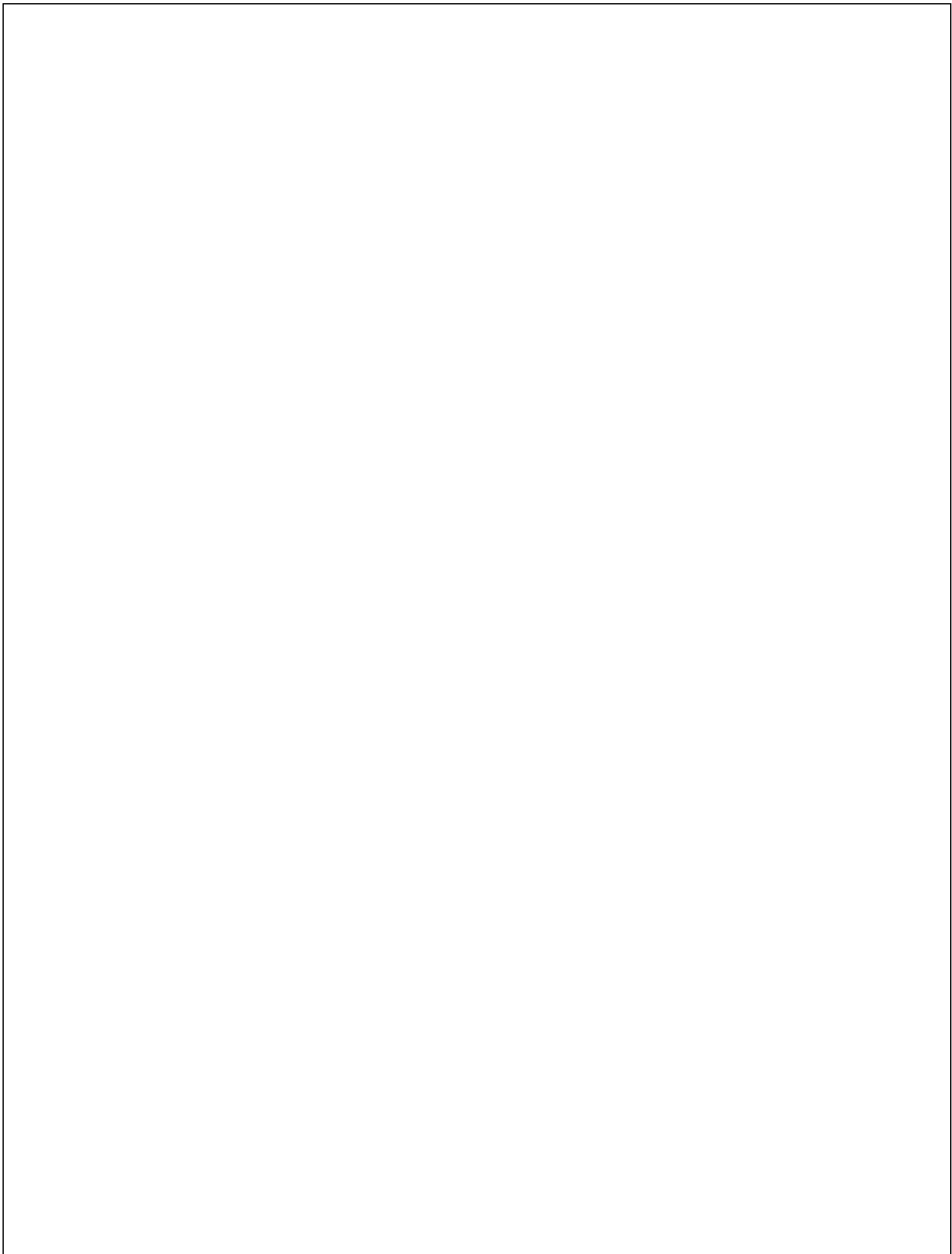
-----



-----

## DEDICATION

*Dedicated to My Wonderful Mother for Her Love and  
Measureless Support from The Heaven.*



## **ACKNOWLEDGMENT**

First of all, I thank my "*God*" for helping me to achieve this work and giving me the ability to finish this thesis in that satisfactory form.

I would like to express my sincere appreciation to my supervisors:

***Dr.Amany*** and ***Eng.Hossam***. I am very grateful for their strong effort, continuous support, and encouragement during the research study in this capstone project. They really influenced my way of thinking and developing the research ideas adopted in this thesis. Really, I can't find the appropriate words to thank them. I am very grateful for their strong effort, continuous support, and encouragement during the research study in this capstone project.

I would like also to express my deepest thanks *all the members* of my colleges for their cooperation during the period I spend with them to prepare this work.

I am extremely grateful to *my family especially my father, mother.*

*Capstone Project Team*

# **ABSTRACT**

The UniNexus application is transforming university life by seamlessly combining multiple essential functions into a single, user-friendly platform. It simplifies daily campus operations and fosters open communication among students, faculty and IT team.

This app, at its core, streamlines access control by providing electronic IDs for students and staff. This system ensures secure access to university facilities, eliminating the need for physical keys or cards and maintaining a precise access log for enhanced security.

The app also simplifies attendance tracking for instructors, removing manual roll calls and boosting administrative efficiency. Its integration into the educational process enables students and educators to focus on learning.

One of its standout features is the ability to report lab malfunctions instantly. It acts as a bridge between instructors and IT teams, expediting issue resolution and minimizing disruptions.

The app's real-time dashboard displays lab availability, optimizing resource management. Users can effortlessly plan their activities based on lab status, promoting more efficient facility usage.

In summary, the UniNexus App is a revolutionary tool that simplifies administrative tasks, streamlines communication and maximizes resource utilization. Embracing this innovation allows universities to create a safer and more connected campus environment, where technology seamlessly integrates with education.

## **Keywords:**

QR-code Scanner, Punch-in, Punch out, Attendance, Chat-bot, Time Table, Request and Halls Error Detection

## Table of Contents

<b>ACKNOWLEDGEMENT.....</b>	<b>I</b>
<b>ABSTRACT.....</b>	<b>II</b>
<b>TABLE OF CONTENTS.....</b>	<b>III</b>
<b>LIST OF FIGURES.....</b>	<b>IV</b>
<b>LIST OF ABBREVIATIONS.....</b>	<b>V</b>
<b>CHAPTER I: INTRODUCTION</b>	
1.1 OVERVIEW.....	11
1.2 Problem statement.....	11
1.3 The Applied Approach.....	12
1.4 Capstone project objective.....	13
1.5 Capstone project Methodology.....	13
1.6 Capstone project Used tools.....	14
1.7 Significance of proposed capstone project .....	15
1.8 Capstone project organization.....	15
<b>CHAPTER II: LITERATURE REVIEW</b>	
2.1 Literature Review.....	17
<b>CHAPTER III: SYSTEM ANALYSIS AND DESIGN</b>	
3.2 Functional Requirements.....	20
3.3 Non-functional Requirements.....	22
3.4 UML Diagrams.....	24
<b>CHAPTER IV: METHODOLOGY</b>	
4.1 Methodology.....	27
4.2 Used Tools.....	28
4.3 Hardware.....	31
4.4 Proposed project.....	
<b>CHAPTER V: GRAPHICAL USER INTERFACE</b>	
<b>CHAPTER VI: CONCLUSIONS AND FUTURE WORK</b>	
6.1 CONCLUSIONS.....	
6.2 FUTURE WORK.....	
<b>REFERENCES.....</b>	<b>34</b>
<b>Appendix (Codes).....</b>	<b>35</b>

## List of Figures

Figure 1	Capstone Objective	13
Figure 2	Capstone Methodology	13
Figure 3	Capstone project Used tools	14
Figure 4	Capstone project Gantt chart	14
Figure 5	Use Case Diagram	24
Figure 6	Class Diagram	25
Figure 7	Sequence Diagram	26
Figure 8	ERD Diagram	26
Figure 9	DFD Diagram	27
Figure 10	Capstone Methodology with explanation blow	29
Figure 11	Raspberry Pi NoIR v2 Camera	33
Figure 12	Double fan kit with heatsink for Raspberry Pi	34
Figure 13	Raspberry Pi 4 Power Supply	34
Figure 14	Raspberry Pi 4 model B 4GB	35

## **LIST OF ABBREVIATIONS**

<b>Abbreviation</b>	<b>Referenced Terms</b>
AI	Artificial Intelligence
QR	Quick Response
ID	Identification
IT	Information Technology
GUI	Graphical User Interface
IOT	Internet Of Things
RFID	Radio Frequency Identification
OS	Operating System
UML	Unified Modeling Language
ERD	Entity Relationship Diagram
DFD	Data Flow Diagram



---

# **CHAPTER I**

---

# **INTRODUCTION**

---

# CHAPTER 1

## INTRODUCTION

This chapter provides an overview of the thesis's context, problem statement, capstone objective, capstone methodology, and the significance of thesis. Finally, the organization of this capstone project is presented.

### 1.1 Overview

UniNexus offers a comprehensive solution for universities seeking to modernize their operations. By streamlining daily tasks and fostering open communication, the app creates a safer, more efficient, and connected campus environment. This project contributes to the field of educational technology by demonstrating the potential of mobile applications to improve university life for students, faculty, and staff.

### 1.2 Problem Formulation

University life can be complex, juggling classes, administrative tasks, and facility access. Traditional methods often rely on physical keys, paper forms, and manual communication, leading to several key challenges:

- Inefficiency: Manual processes for attendance tracking, facility access control, and maintenance reporting are time-consuming and prone to errors.
- Security Concerns: Physical keys and cards can be lost or stolen, compromising security. Traditional access logs are cumbersome and offer limited traceability.
- Communication Silos: Important information about class attendance, lab availability, or maintenance issues can get lost or delayed due to fragmented communication channels.
- Resource Underutilization: Lack of real-time data on lab availability makes it difficult to optimize resource allocation, potentially leading to unused equipment or overcrowded facilities.

UniNexus aims to address these challenges by developing a mobile application that integrates essential university functions into a single platform.

## 1.3 The Applied Approach

UniNexus tackles the formulated problems by employing a multi-pronged approach that leverages mobile technology:

1. **Electronic Access Control:** UniNexus replaces physical keys or cards with electronic IDs on the app. This enhances security by:
  - **Securing Entry:** Granting access only to authorized users with valid credentials.
  - **Detailed Logging:** Maintaining a digital record of entries and exits for improved traceability.
2. **Automated Administrative Tasks:** Streamlining workflows through:
  - **Attendance Tracking:** Instructors can take attendance electronically within the app, eliminating manual roll calls.
  - **Instant Malfunction Reporting:** Students can report lab equipment malfunctions directly through the app, allowing for quicker resolution by IT staff.
3. **Improved Communication:** Fostering a more connected campus environment through:
  - **Centralized Platform:** Providing a single point of access for announcements, updates, and maintenance reports.
  - **Direct Messaging:** Enabling communication between students, faculty, and IT staff within the app.
4. **Real-time Resource Management:** Optimizing lab utilization by:
  - **Lab Availability Dashboard:** Displaying real-time information on available labs, allowing users to plan activities efficiently.

By adopting a mobile app approach, UniNexus offers a convenient, user-friendly platform that integrates these functionalities seamlessly, addressing the inefficiencies, security concerns, communication gaps, and resource underutilization issues identified in the problem formulation.

## 1.4 Capstone Project Objective

This project aimed to develop and evaluate a mobile application, UniNexus, which integrates essential university functions into a user-friendly platform. The core objectives are presented in figure 1 as follows:

- Enhance security through electronic ID access control.
- Streamline administrative tasks like attendance tracking and lab malfunction reporting.
- Improve communication between students, faculty, and IT staff.
- Optimize resource utilization by providing real-time lab availability.



Figure 1: Capstone Objectives

## 1.5 Capstone Project Methodology

User feedback was incorporated through focus groups and surveys at various stages of development as shown in figure 2.

The first stage is project initialization, then project planning, project execution, project monitoring &controlling and finally project closure

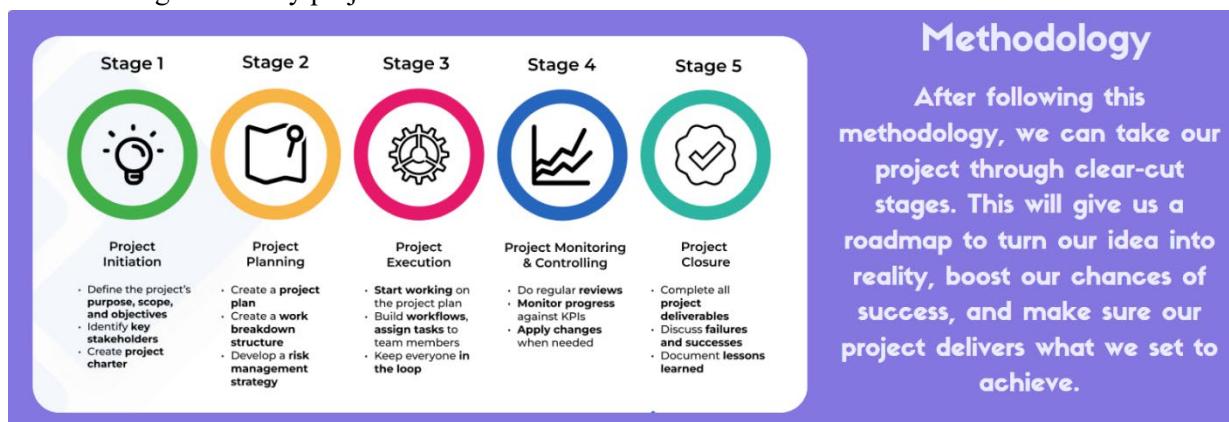


Figure 2: Capstone Methodology

## 1.6 Capstone Project Used tools

The Software used tools are shown in figure 3:  
Android Studio, Java, MySQL, PHP MY Admin,  
FIGMA

The Hardware Used tools is:  
Raspberry pi



Figure 3: Capstone project used tools

## 1.7 Gantt chart



Figure 4: Capstone project Gantt chart

## 1.8 Significance of Proposed Capstone Project

The significance of the UniNexus application lies in its potential to revolutionize university life by:

- **Enhancing Security and Efficiency:** UniNexus replaces physical keys with electronic IDs, improving access control and security. It streamlines administrative tasks like attendance tracking and maintenance reporting, saving time and reducing errors.
- **Optimizing Resource Utilization:** Real-time lab availability displayed on the app allows students to plan their activities effectively and ensures labs are used to their full potential. This eliminates wasted time and optimizes resource allocation.
- **Fostering Communication and Transparency:** The app acts as a central platform for announcements, updates, and maintenance reports. It also facilitates direct communication between students, faculty, and IT staff, promoting a more connected and informed campus environment.
- **Improving User Experience:** UniNexus streamlines daily tasks and eliminates the need for manual processes. This creates a more user-friendly and convenient experience for students, faculty, and IT staff.

## 1.9 Capstone Project Organization

This documentation is divided into 5 chapters:

**Chapter 1:** This chapter provides an overview of the thesis's context, problem statement, capstone objective, capstone methodology, and the significance of thesis. Finally, the organization of this capstone project is presented.

**Chapter 2:** Provides a literature review that provides the papers that works on the same project before

**Chapter 3:** Provides problems and our idea to solve it

**Chapter 4:** Provides tools and hardware used in the capstone project

**Chapter 5:** Provides the GUI

**Chapter 6:** This chapter we summarize the general conclusions and future work of capstone project.



---

## **CHAPTER 2**

# **Literature Review**

---

## CHAPTER 2

### INTRODUCTION

This chapter delves into the existing body of research relevant to the development of the UniNexus mobile application. We explore the literature on various aspects crucial to the app's functionality, including existing solutions for student information systems, mobile learning platforms, and specific functionalities like attendance management or equipment reporting. By examining these previous studies and industry trends, we gain valuable insights and establish a foundation for designing a robust and user-centered mobile application that effectively addresses the needs of the university community.

### Literature Review

#### **University of New South Wales, Sydney, Australia [1]:**

The journey towards the realization of smart campus in a large university. They engaged with various stakeholders to articulate the vision of a smart campus using IoT technologies. They then developed their threelayer system architecture comprising separate layers of sensing, data, and analytics that prevents vertical lock-in and scales easily. Lastly, they executed four pilot use-cases in their campus and revealed insights they obtained, highlighting the benefits that IoT-based continuous and automated data collection brings to decision making in a University campus.

#### **Smart Campus Implementation in University Tun Hussein Onn Malaysia [2]:**

In recent years, smart campus concepts have been introduced and implemented to enrich the quality of life in the campus environment. This was also implemented to reduce the cost of running a campus. Smart campuses can be defined as devices and applications that create new experiences, services and facilitate operational efficiency of a campus. The concept of the smart campus is originally derived from the smart city concepts and experiences where it converted a traditional campus into a smart campus. This concept helps to improve better experiences to the campus community and provide diversification modes of teaching to the students and lecturers through the enhanced learning process. The smart campus implementation also offers campus management with various tools to efficiently manage the campus.

## **University of Birmingham, The Smart Campus A platform for innovation [3]:**

Universities are by their very nature complex. First and foremost, they are educational institutions, where undergraduate and postgraduate students learn and where faculty teach and conduct research. In addition, they function as small cities, complete with facilities teams, energy plants, sports facilities, and other civic attractions. Finally, they are major local employers and act as an important catalyst within local and regional economies. These unique characteristics mean that universities have a large number of stakeholders with which to communicate, and often have highly complex revenue models.

## **QR Code Based Smart Attendance System [4]:**

Among the various types of attendance systems that have been developed, using punch cards, log books, fingerprint systems, barcodes, QR codes and also RFID still cause lots of problems such as providing incorrect information to the users. The purpose of the smartphone based attendance system is to computerize the traditional way of recording attendance and provide an easiest and smart way to track attendance in institutions nowadays, the most common device that have been come into account in marketing and business are smartphone devices. Moreover, it comprises lots of them running Android OS.

# **CHAPTER 3**

## **System Analysis and Design**

## **CHAPTER 3**

# **System Analysis and Design**

### **3.1: Introduction:**

The foundation for any successful system lies in a thorough understanding of its purpose and functionalities. This chapter delves into the analysis and design phases of the UniNexus mobile application.

### **3.2: Functional Requirements:**

#### **Students:**

- **View Class Schedule and Announcements:**
  - Access course schedules with dates, times, locations, and instructor information.
  - View announcements and updates from instructors for each course.
- **Access Course Materials:**
  - Download or stream course materials (syllabi, lectures, presentations) provided by instructors.
  - Access additional resources (readings, assignments) linked within the course materials.
- **Communicate with Faculty and Staff (Messages):**
  - Send direct messages to instructors or staff for questions, clarifications, or reporting issues.
  - Receive messages and replies from instructors and staff.

#### **Mark Attendance (In-Class or Remote):**

- The university uses the app for attendance, students can mark their presence for classes (in-person or remote sessions) through the app.

**Faculty:**

- **Manage Course Information (Syllabus, Materials):**
  - Upload course syllabi, lectures, presentations, and other relevant materials for students to access.
  - Edit and update course information as needed throughout the semester.
- **Record Attendance for Classes:**
  - Mark attendance for students during class sessions using the app (if applicable).
  - Optionally, allow students to self-report attendance through the app.
- **Submit Equipment Malfunction Reports:**
  - Report malfunctioning equipment within labs or classrooms.
  - Provide details on the equipment type, location, and nature of the malfunction.
- **Access Student Rosters and Attendance Reports:**
  - View a list of enrolled students for each course they are teaching.
  - Access attendance reports summarizing student attendance data.
- **Communicate with Students and Staff (Messages):**
  - Send messages to individual students or the entire class for announcements, reminders, or discussions.
  - Send messages to staff members for collaboration or to report issues.

**Staff (IT):**

- **Address Equipment Malfunction Reports:**
  - View submitted equipment malfunction reports from students.
  - Assign technicians or initiate repair procedures based on the reported issues.
  - Communicate with students regarding the repair status.
- **Update Lab Availability Information:**
  - Manage lab availability schedules, indicating open and booked slots.
  - Update information on lab closures, maintenance schedules, or changes in available equipment.
- **Manage Access Control (User Permissions for Building/Lab Access):**
  - Assign or revoke access permissions for students and faculty to specific buildings or labs based on their roles and needs.
- **Communicate with Students and Faculty (Messages):**
  - Send system maintenance updates or important announcements to students and faculty.
  - Respond to messages from students or faculty regarding technical issues or access concerns.

### **3.3: Non-Functional Requirements:**

**Performance:**

- **Load Times:** The app should load quickly on various devices and network connections (Wi-Fi, cellular data). Users shouldn't experience significant delays when opening the app, accessing features, or downloading course materials.
- **Responsiveness:** The app should respond promptly to user actions, like button clicks or scrolling. Minimal lag or wait times are crucial for a smooth user experience.

## **Usability:**

- **Intuitive Interface:** The user interface (UI) should be clear, user-friendly, and easy to navigate for all user roles (students with varying technical skills, faculty, and staff).
- **Accessibility:** The app should be accessible for users with disabilities. This includes features like screen reader compatibility, adjustable text size and color contrast, and support for alternative input methods (voice commands).
- **Offline Functionality (Optional):** Consider enabling certain functionalities (e.g., viewing downloaded course materials) even when users are offline.

## **Security:**

- **User Authentication:** Secure login mechanisms (passwords, multi-factor authentication) should be implemented to prevent unauthorized access to user accounts.
- **Data Encryption:** User data (personally identifiable information, grades) and system data should be encrypted to ensure confidentiality and protect against breaches.
- **Regular Updates:** The app should receive regular security updates to address vulnerabilities and maintain a strong defense against cyber threats.

## **Reliability:**

- **Uptime:** The app should be available for use with minimal downtime. Scheduled maintenance windows should be communicated clearly to users.
- **Data Integrity:** Data accuracy and consistency are critical. The system should ensure that data is stored and transmitted reliably without corruption.

## **Maintainability:**

- **Code Quality:** The app's code should be well-documented, modular, and easy to maintain. This facilitates future updates, bug fixes, and feature enhancements.

**Scalability:** The app's architecture should be scalable to accommodate a growing user base and potentially increased data volume.

### 3.4: UML Diagrams:

- **Use Case Diagram:**

The use case diagram in UML is used to identify the different ways that a user might interact with the system. It describes what the system does and how the actors use it, but not how the system operates internally.

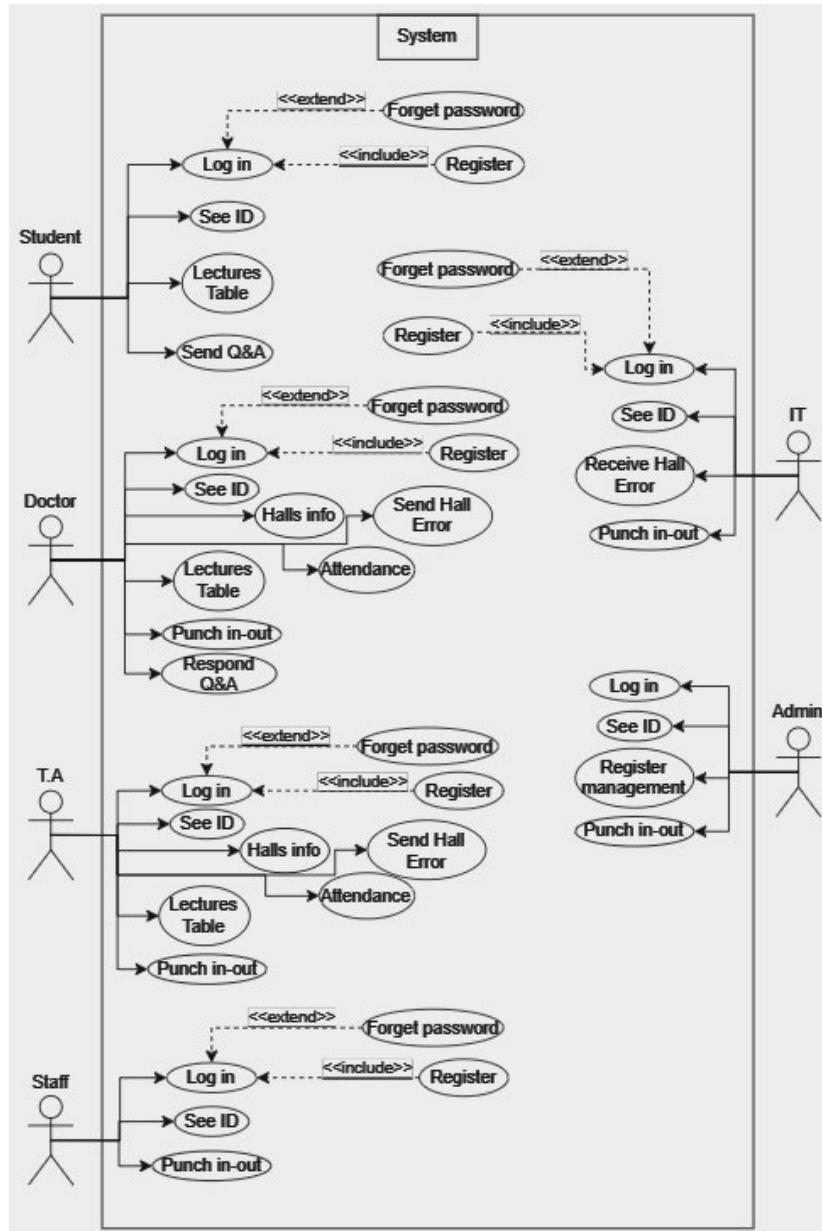
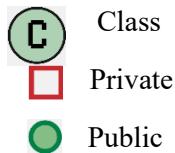


Figure 5: Use Case Diagram

- Class Diagram:

Class diagram 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 6: Class Diagram



- Sequence Diagram:**

The sequence diagram is used to show the interactions between objects in the sequential order that those interactions occur.

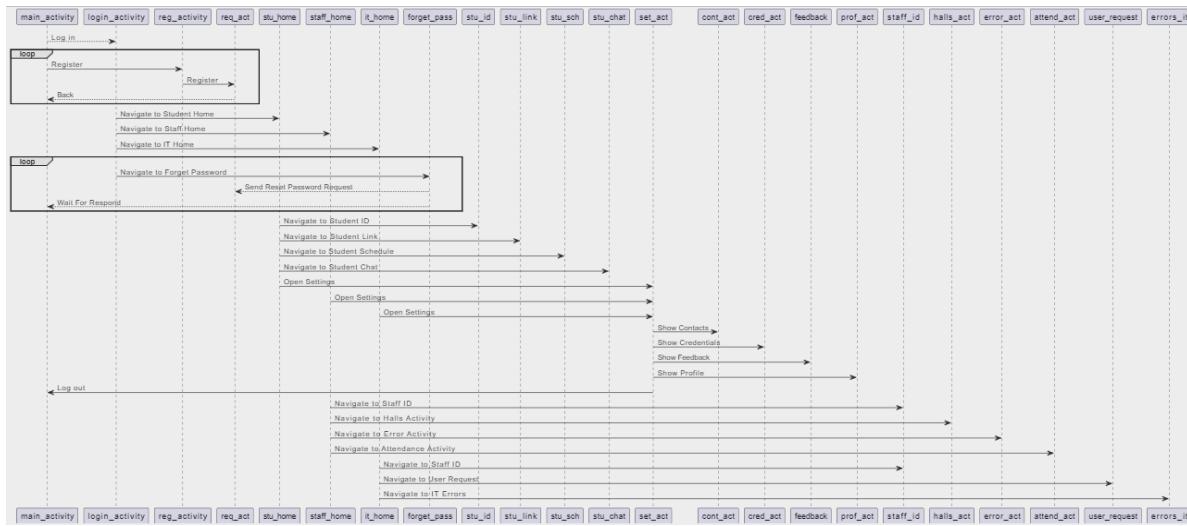


Figure 7: Sequence Diagram

## ERD Diagram:

The ERD diagram shows the relationships between different database tables that is available in our university system as shown figure 5.

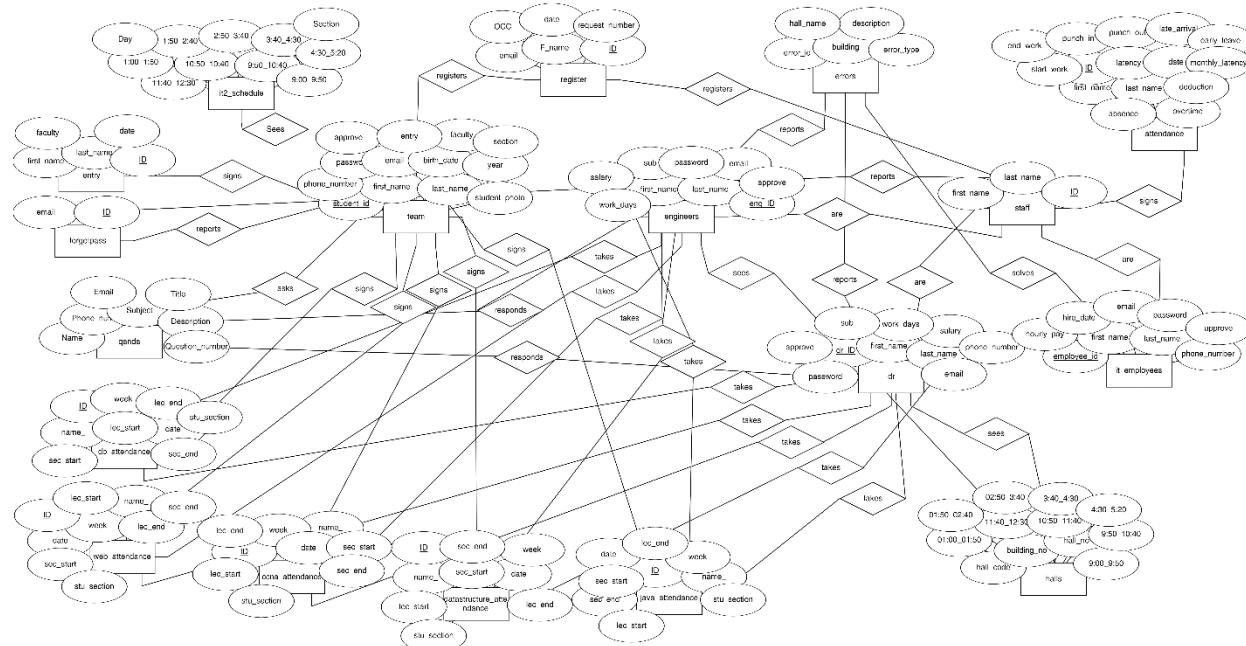


Figure 8: ERD Diagram

## DFD Diagram:

The DFD diagram shows how the system works between users (faculty, staff, and students) to interact with a course management system as shown in figure 6.

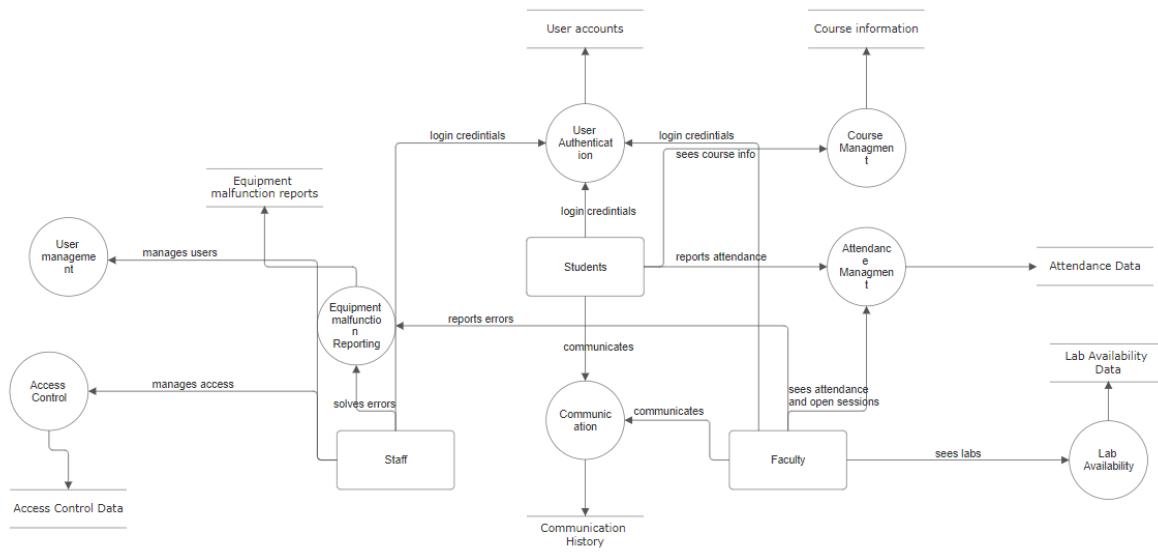


Figure 9: DFD Diagram



---

## **CHAPTER 4**

# **METHODOLOGY**

---

# CHAPTER 4

## INTRODUCTION

Methodology chapter explains how the project was carried out, including the approach, methods, and techniques used to achieve the goals of the project. This chapter describes the steps involved in project management. Used tools and hardware also mentioned in this chapter.

### 4.1: Methodology:

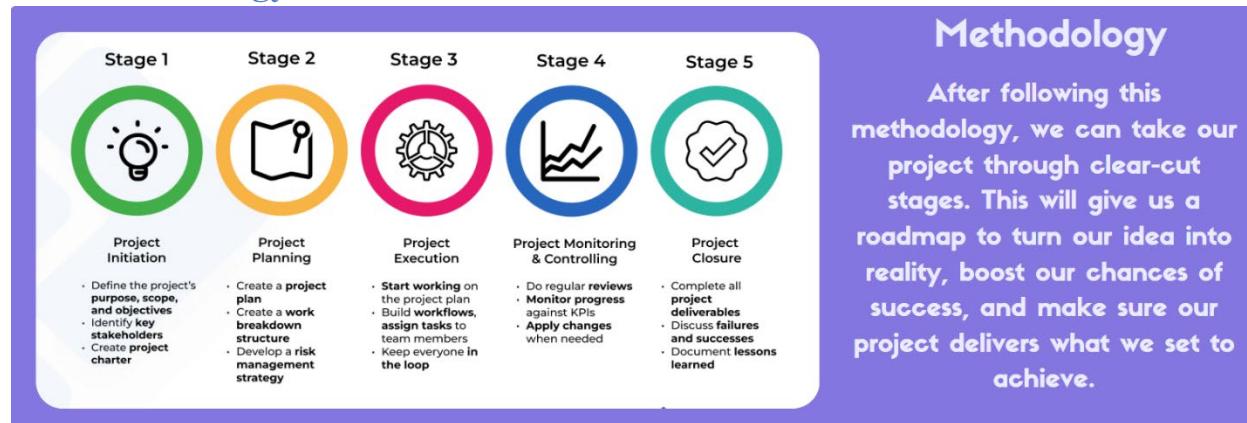


Figure 10: Capstone Methodology with explanation blow

**Stage 1: Project Initiation:** At this stage, the project's objectives, scope, stakeholders, and first supplies are all decided. The purpose of this stage is to confirm that the project is feasible and aligns with the main goals.

**Stage 2: Project Planning:** This stage creates a detailed project plan based on the data gathered during the start stage. The project plan includes details on the project's budget, timetable, resources, scope, and risk management strategy.

**Stage 3: Project Execution:** After the plan is created, the project enters the execution stage. This is the actual area where the project's goals are achieved. To continue the project at this phase, team members must effectively coordinate and communicate with one another.

**Stage 4: Project Monitoring & Controlling:** At this stage, progress is tracked and managed during the execution phase. Regular progress reports, performance evaluations, and any required project plan modifications are all included in this phase.

**Stage 5: Project Closure:** The final stage of the project process is project closure. In this stage, the project deliverables are completed, and the project is formally closed out. Evaluate the project's success against its objectives and identify lessons learned for future projects.

## 4.2: Used Tools:



### Android Studio:

Android Studio is the official Integrated Development Environment (IDE) for building Android apps. Created by Google and based on JetBrains' IntelliJ IDEA, it provides a comprehensive set of tools and features to streamline the development process.

Android Studio offers a user-friendly interface with code completion, syntax highlighting, and debugging tools to help developers write clean and efficient code. It integrates seamlessly with various Android SDK (Software Development Kit) components, allowing developers to leverage pre-built libraries, emulators, and testing frameworks. This makes it easier to create high-quality apps that function optimally on different Android devices. Whether you're a seasoned developer or just starting your journey with Android app development, Android Studio offers a powerful and versatile platform to bring your ideas to life.



### Xampp:

XAMPP (X stands for Cross-Platform, A for Apache, M for MySQL, PP for PHP and Perl) is a free and open-source software package that bundles essential components for running a web server on your local machine. It simplifies the setup process by combining Apache HTTP Server, the popular MySQL database management system, and interpreters for scripting languages like PHP and Perl within a single package.

This all-in-one solution allows developers to create a development environment on their personal computers to test and debug their web applications before deploying them to a live server. XAMPP is particularly beneficial for beginners as it eliminates the need to install and configure each component individually. Its user-friendly interface makes it easy to start, stop, and manage the web server, database, and scripting languages, allowing developers to focus on coding and building their web applications efficiently.



## phpMyAdmin:

phpMyAdmin is a free and open-source web application that acts as a graphical user interface (GUI) for managing MySQL databases. Imagine it as a user-friendly control panel for your MySQL databases. Without phpMyAdmin, interacting with a MySQL database would involve writing complex SQL code directly.

phpMyAdmin offers a user-friendly interface that allows you to perform various database administration tasks visually. These tasks include:

- Creating, modifying, and deleting databases and tables
- Adding, editing, and deleting data within tables
- Executing SQL queries to retrieve, manipulate, and analyze data
- Managing user accounts and access privileges within the database

By providing a visual interface, phpMyAdmin simplifies database management for users who may not be comfortable writing raw SQL code. It empowers developers, content managers, and even non-technical users to interact with and manage their MySQL databases efficiently, enhancing the overall workflow for web application development and maintenance.



MySQL is a powerful and widely used Open-Source Relational Database Management System (RDBMS). Think of it as a digital filing cabinet that efficiently stores and organizes information. Unlike a physical cabinet, MySQL excels at handling large datasets and allows for quick retrieval and manipulation of that data. This makes it ideal for powering websites and web applications that require storing and managing user information, product details, or any other structured data.

MySQL utilizes a structured approach, storing data in tables with rows and columns. This structure allows for efficient searching, sorting, and analysis of the information. Furthermore, MySQL's robust security features ensure data integrity and prevent unauthorized access. Its popularity is further enhanced by its compatibility with various programming languages and its free and open-source nature, making it a cost-effective and versatile solution for developers.



**Figma:**

Figma is a web-based design tool that revolutionizes collaboration for user interface (UI) and user experience (UX) design teams. Unlike traditional design software, Figma operates within your web browser, eliminating the need for individual installations on each team member's device. This fosters real-time collaboration, allowing designers to work on the same project simultaneously, see edits instantly, and discuss changes seamlessly within the platform.

Beyond collaboration, Figma boasts a robust feature set for creating high-fidelity prototypes, designing interactive elements, and managing design systems. Its user-friendly interface makes it accessible for designers of all experience levels, while its cloud-based nature ensures version control and easy access to design assets from anywhere. Whether you're a solo designer or part of a large team, Figma streamlines the design process and empowers effective communication, making it a popular choice for modern design workflows.

## 4.3: Hardware:

### Raspberry Pi NoIR v2 Camera

The Raspberry Pi HD Pi NoIR Camera Board connects to any Raspberry Pi or Compute Module to create HD photos and video. The Pi NoIR (non-infrared) is the same as the standard camera module but without the infrared filter. Therefore, it is excellent for photography and video in the dark.

Without infrared filter

High quality images

High data capacity

8 megapixel fixed focus

Supports 1080p, 720p60 and VGA90

Sony IMX219PQ CMOS Image Sensor

15-pin flat cable

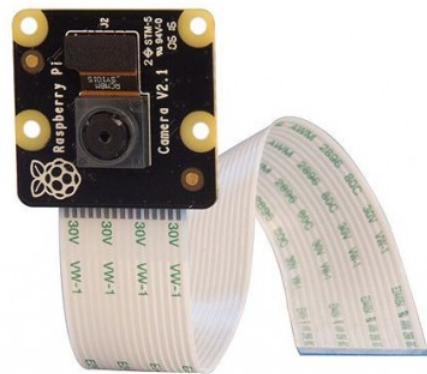


Figure 11: Raspberry Pi NoIR v2 Camera

## Double fan kit with heatsink for Raspberry Pi

This is a radiator kit with two fans compatible with Raspberry Pi 4B. The heat sink is made of high-quality aluminum and the surface is designed with grooves, which is conducive to heat dissipation.

Power: 5V (1.5W)

Fan speed: 7500 RPM

Weight: 12 grams

Compatible with Raspberry Pi 3B/3B+ and 4B

Dimensions: 50x25x13 millimeters



Figure 12: Double fan kit with heatsink for Raspberry Pi

The kit includes:

A large heatsink with 2 fans

2 small heatsinks

## Raspberry Pi 4 Power Supply

This official universal power supply for Raspberry Pi has an integrated USB-C cable and connector. It is valid for the Raspberry 4 Model B in all its versions. This model is especially suitable for the Raspberry Pi 4 if you want to have enough current, since it has a 3A output to connect external devices.

Output: 5.1 V - 3A

Short circuit, overcurrent and overvoltage protection.

Low standby current consumption.

USB-C connector, with approximately 1.5 m cable.

Nominal Power: 15.3W

Class II double insulation



Figure 13: Raspberry Pi 4 Power Supply

## Raspberry Pi 4B 4GB

The Raspberry Pi 4 Model B is a mini desktop computer the size of just over a credit card. All you need to do is connect a monitor, keyboard, and mouse and load the latest Raspberry Pi OS to a MicroSD card. You will have a complete small desktop computer ready to use.

- Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.8GHz
- 1GB, 2GB, 4GB or 8GB LPDDR4-3200 SDRAM (depending on model)
- 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE
- Gigabit Ethernet
- 2 USB 3.0 ports; 2 USB 2.0 ports.
- Raspberry Pi standard 40 pin GPIO header (fully backwards compatible with previous boards)
- 2 × micro-HDMI® ports (up to 4kp60 supported)
- 2-lane MIPI DSI display port
- 2-lane MIPI CSI camera port
- 4-pole stereo audio and composite video port
- H.265 (4kp60 decode), H264 (1080p60 decode, 1080p30 encode)
- OpenGL ES 3.1, Vulkan 1.0
- Micro-SD card slot for loading operating system and data storage
- 5V DC via USB-C connector (minimum 3A\*)
- 5V DC via GPIO header (minimum 3A\*)
- Power over Ethernet (PoE) enabled (requires separate PoE HAT)

Operating temperature: 0 – 50 degrees C ambient



Figure 14: Raspberry Pi 4 model B 4GB

# **CHAPTER 5**

## **Graphical User Interface**

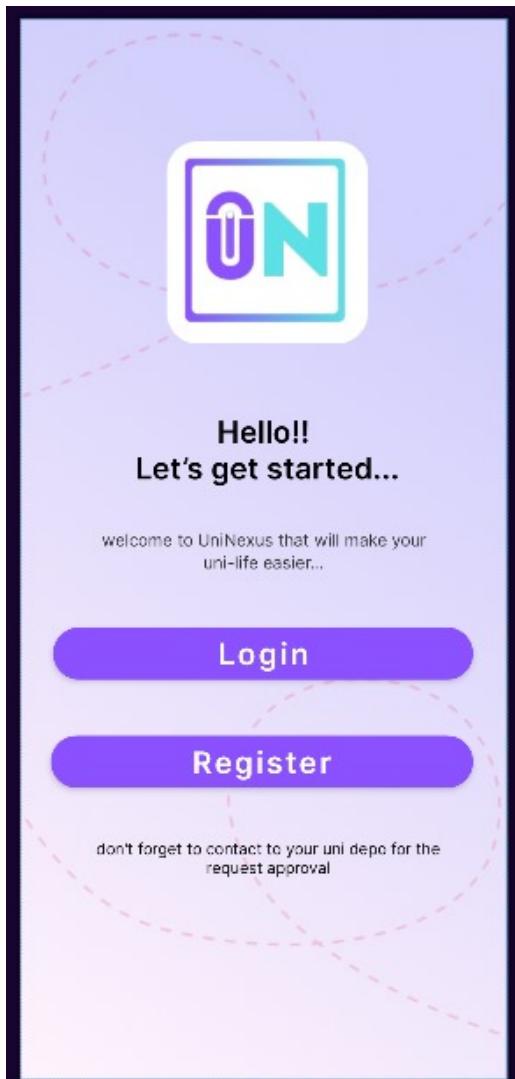
# CHAPTER 5

## INTRODUCTION

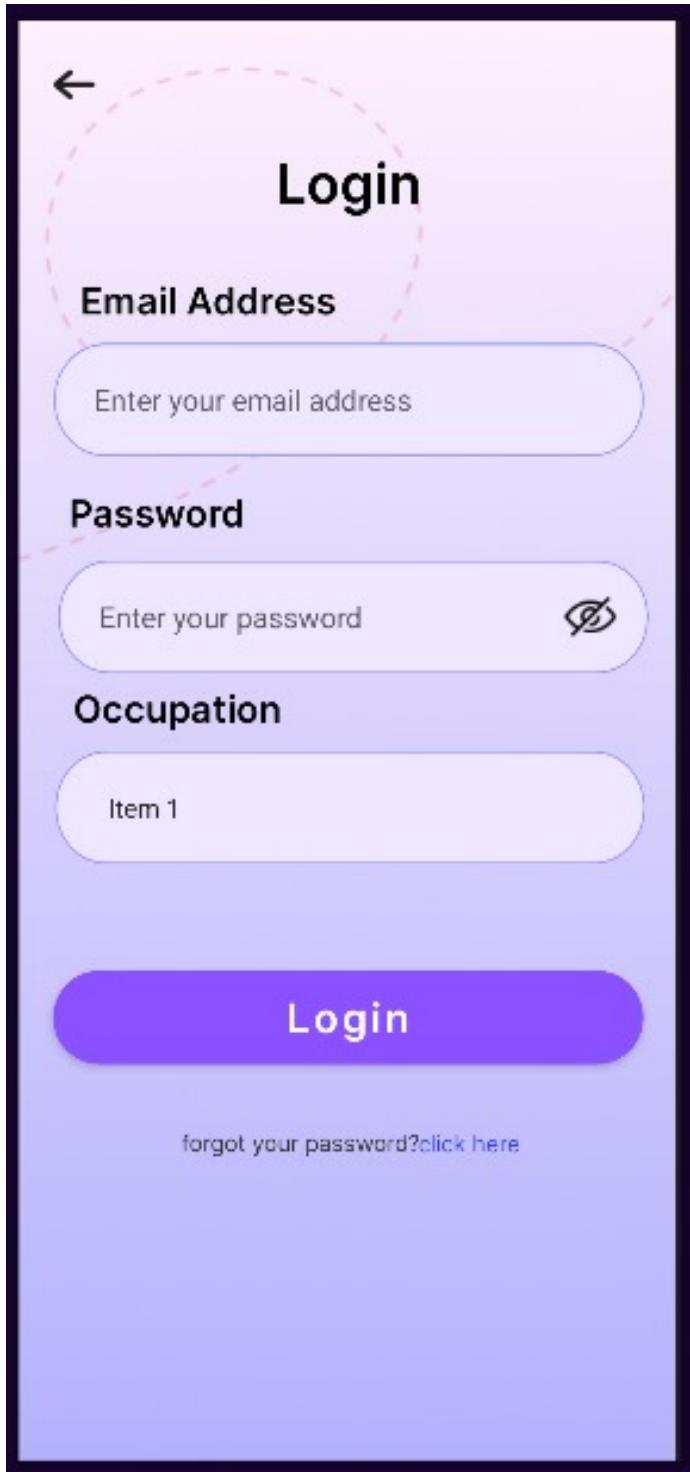
Chapter 5 dives into the design of the UniNexus mobile application's user interface (UI). We'll explore the visual elements, layout, and interaction patterns that users will encounter when navigating the app. This chapter emphasizes creating a user-friendly and intuitive interface that caters to the diverse needs of students, faculty, and staff, ensuring a smooth and efficient user experience for everyone within the university community.

### 5.1 GUI:

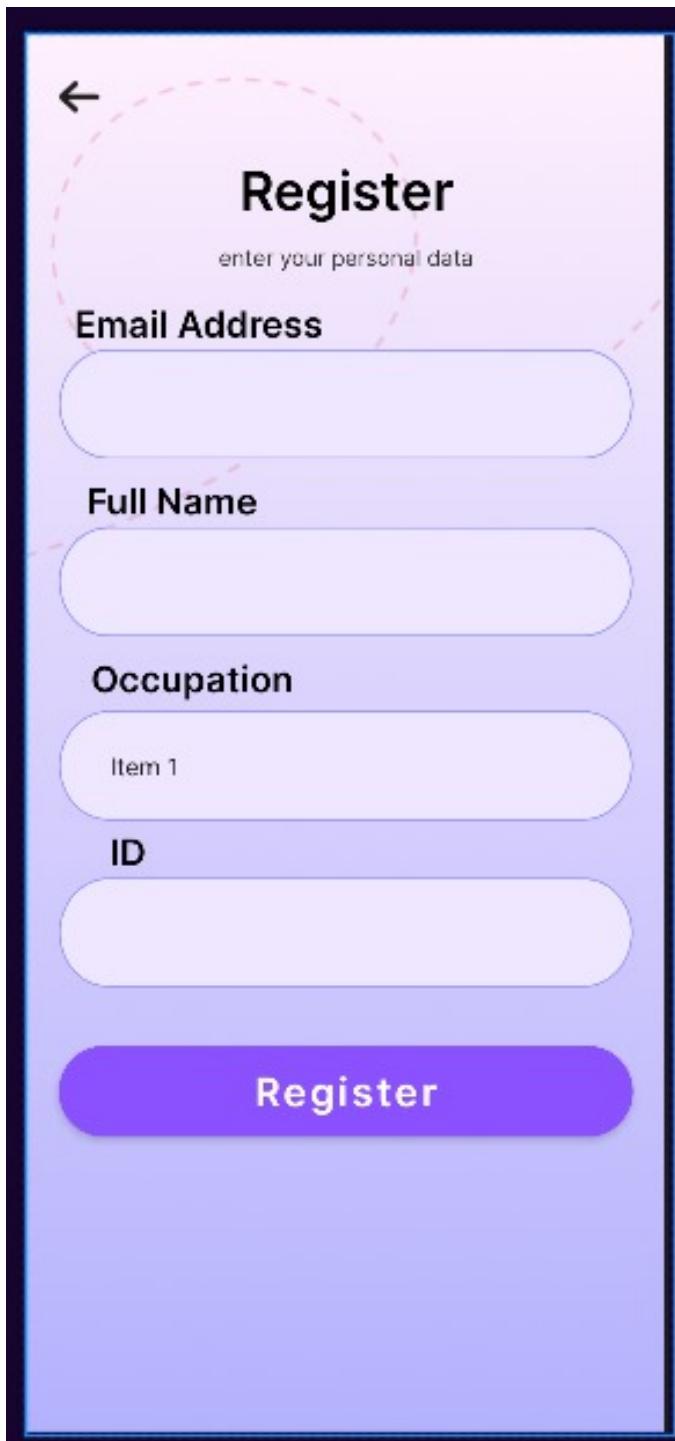
This Frame represents the main page



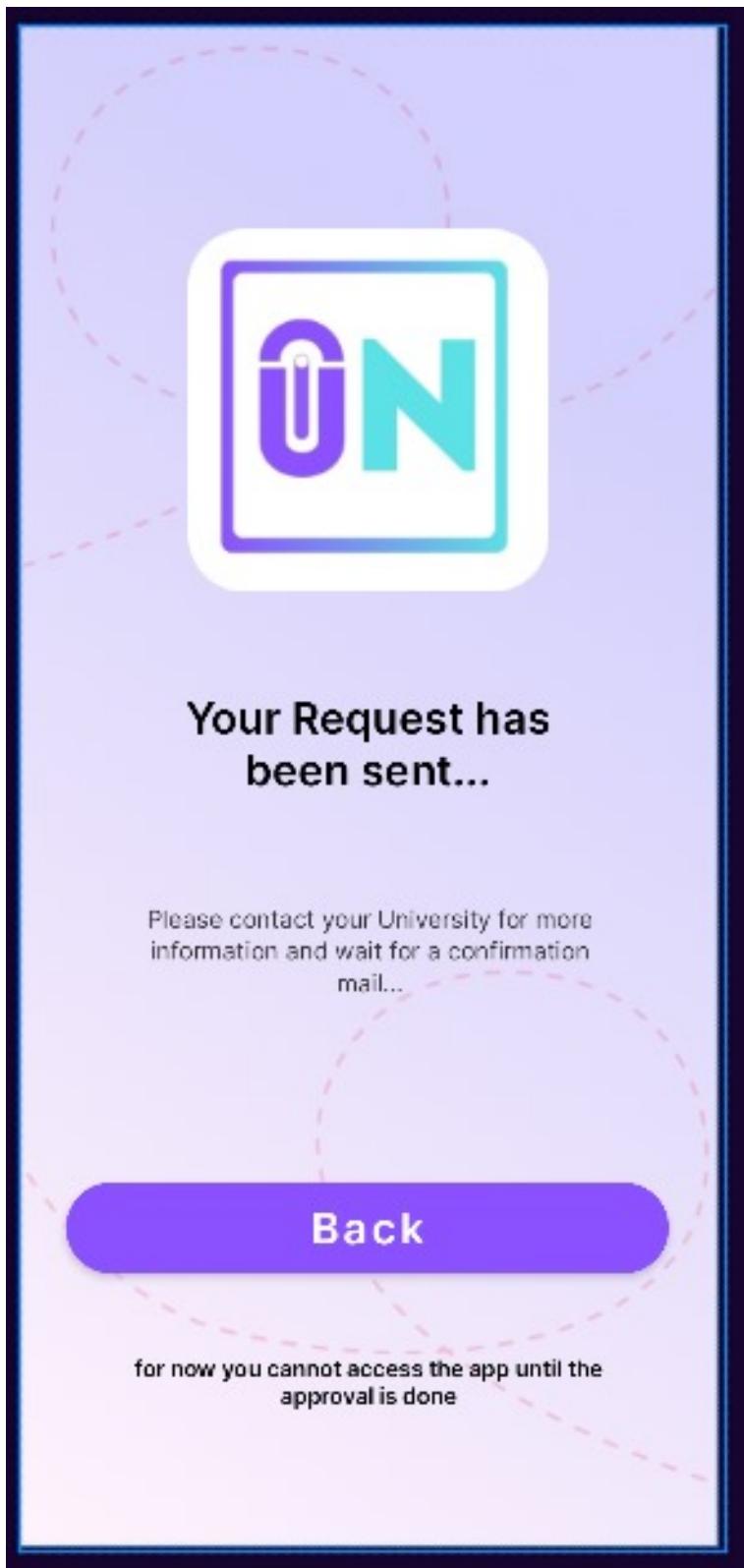
This frame reepresents the login page



This frame represents the register page



This frame represents that the register has been successful and a request is sent and it will redirect the user to the main page



This frame represent forgot password then it will redirect him to the request frame then the home page

←

## Password Reset Request

enter your data to check if the account exists

Email Address

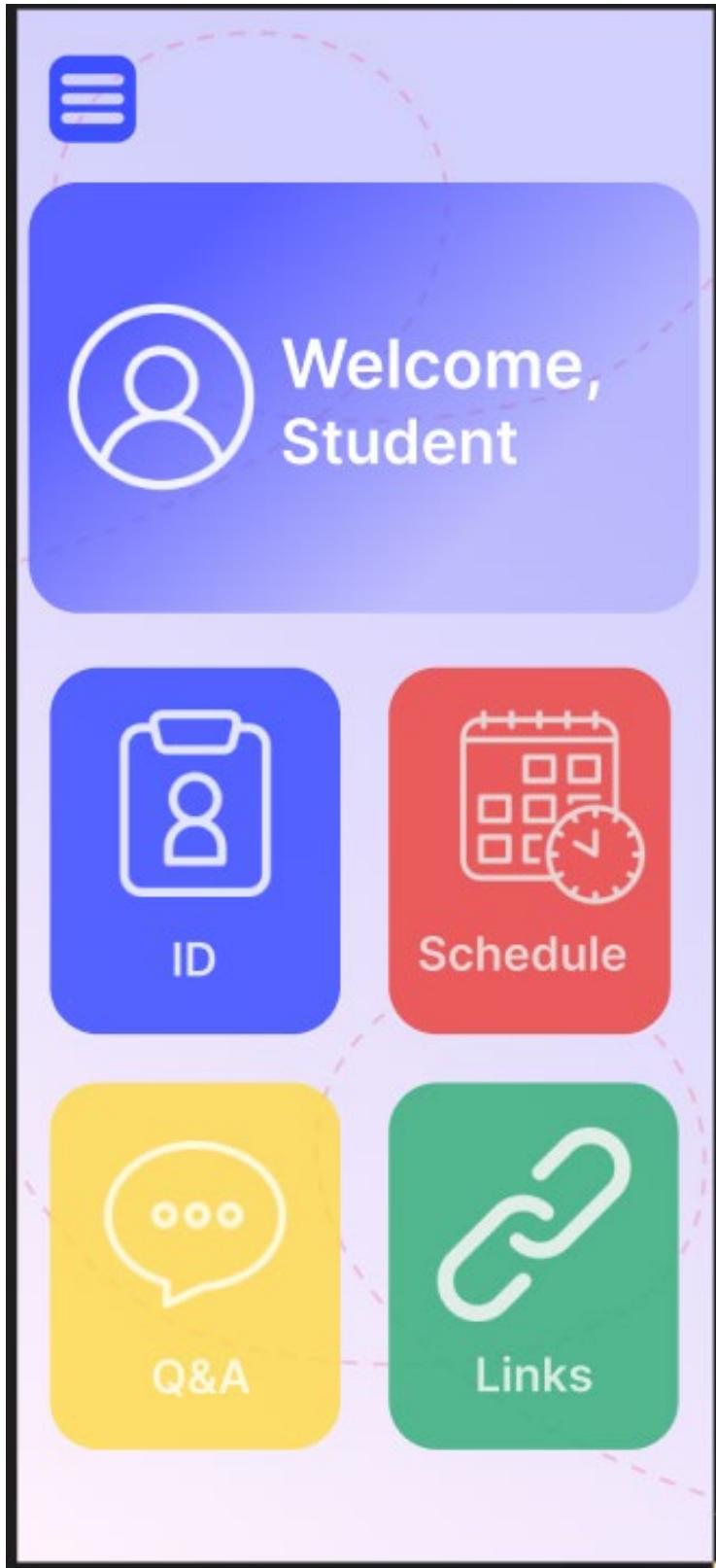
ID

Occupation

Item 1

Submit

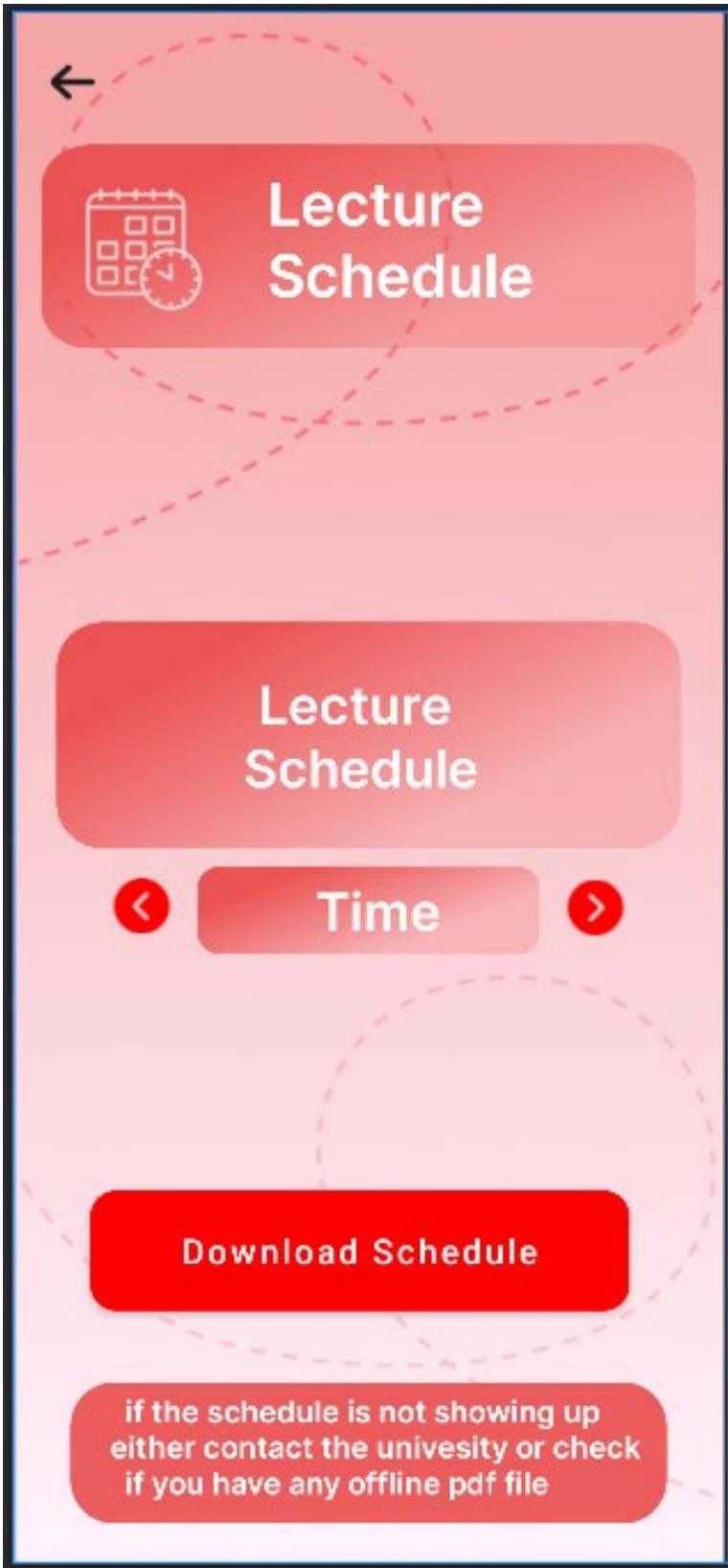
1. This frame represents the Students home page



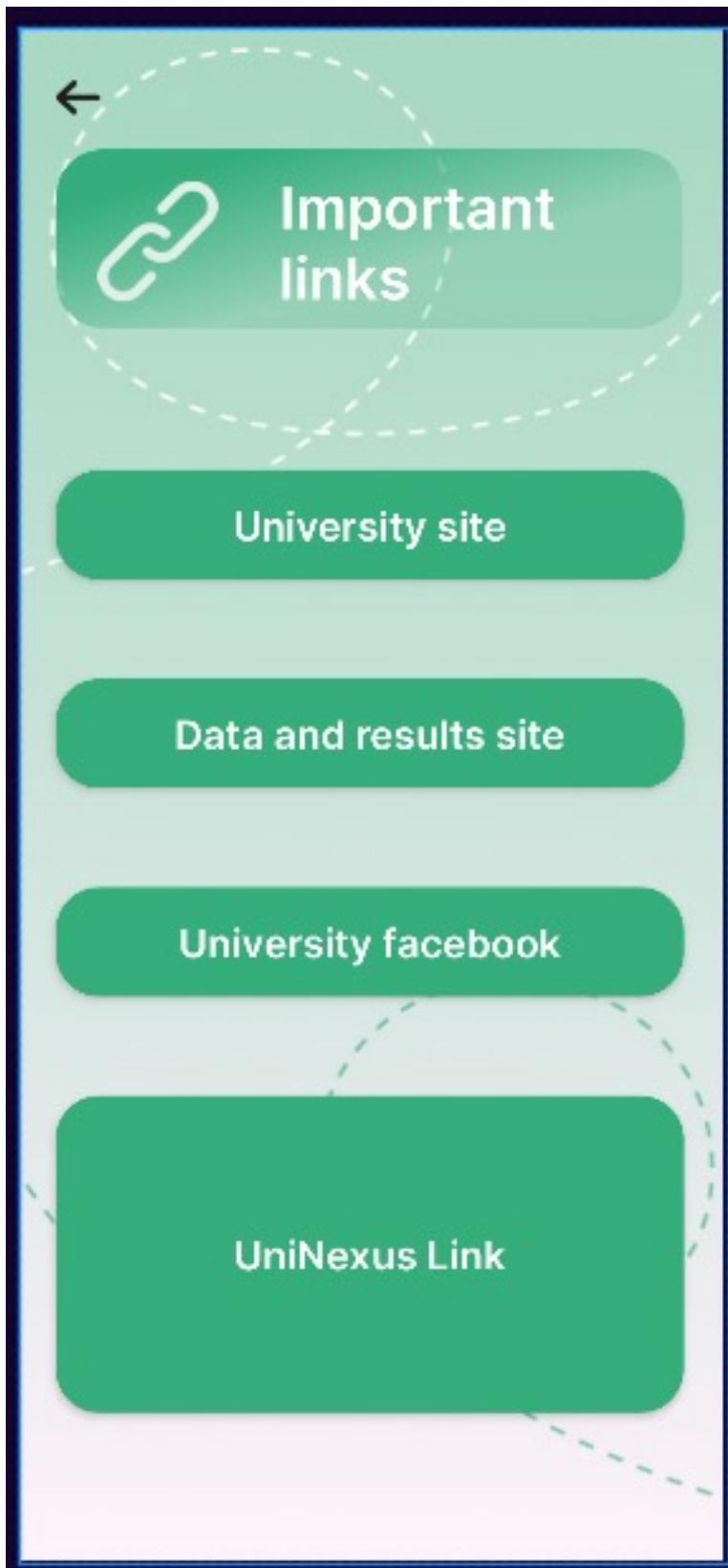
1.1 This frame represents the ID frame



1.2 This frame represents the schedule frame



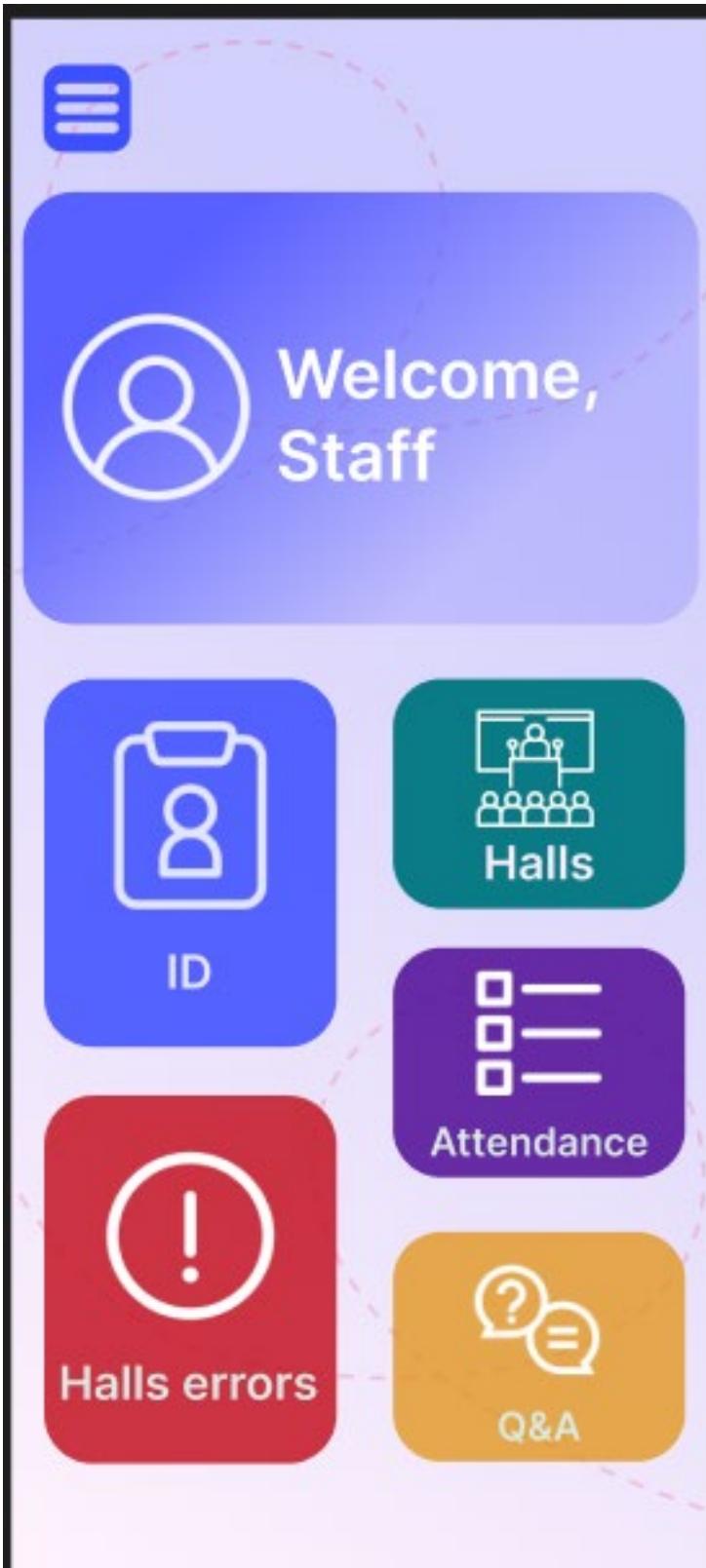
1.3 This frame represents the links frame



1.4 This frame represents the Q&A frame for Students

The diagram illustrates a user interface frame for a Q&A application. It consists of several nested dashed rounded rectangles. The innermost rectangle is light blue and contains a large, dark blue rectangular input field. Above this input field is the text "Subject name". Below it is another text input field labeled "Title". At the bottom of this section is the text "Description", followed by a larger, empty rectangular input field. A central, rounded rectangular button labeled "Submit" is positioned below the "Description" field. The entire form is set against a background divided into three horizontal sections: a yellow top section, a white middle section containing the form, and a pink bottom section.

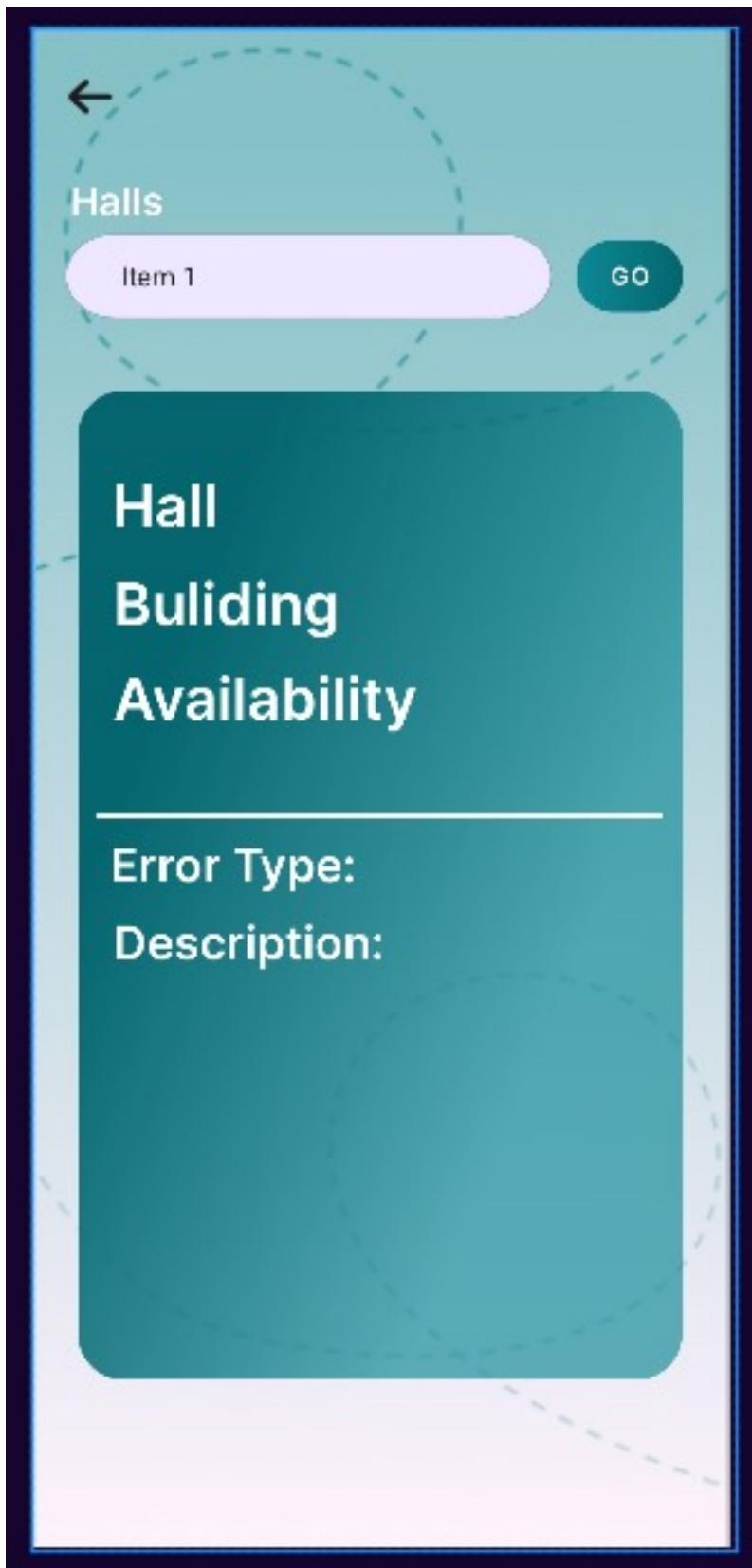
2. This frame represents the Staff home page



2.1 This frame represents the Punch in and Punch out for Staff



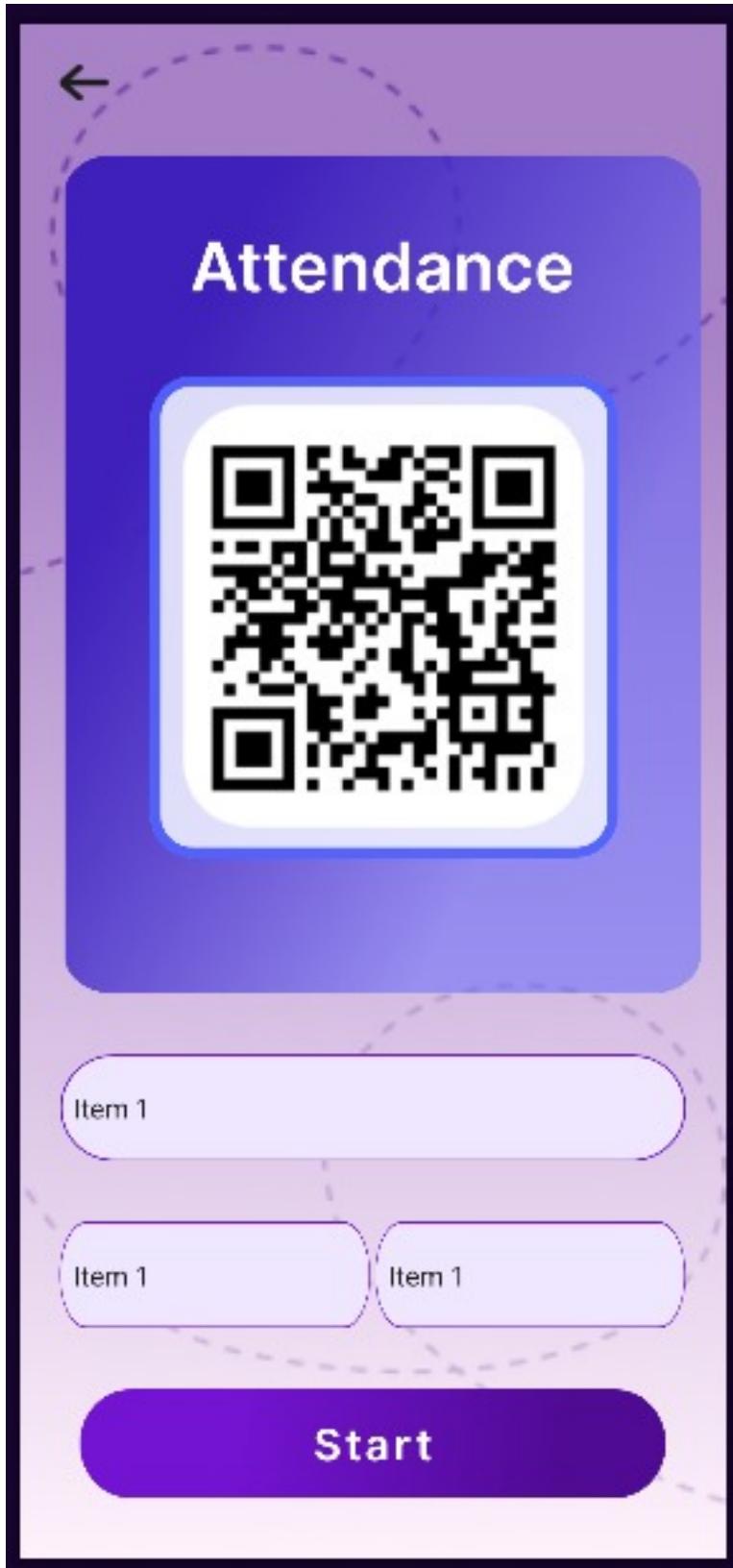
2.2 This frame represents the Halls frame



2.3 This frame represents the Halls Errors frame

The image shows a wireframe of a mobile application interface. The title 'Halls Errors frame' is at the top. A back arrow icon is in the top-left corner. Below are four input fields: 'Hall Name' (placeholder: Item 1), 'Building' (placeholder: Item 1), 'Error Type' (placeholder: Item 1), and 'Description' (a large text area). A red 'Submit' button is at the bottom.

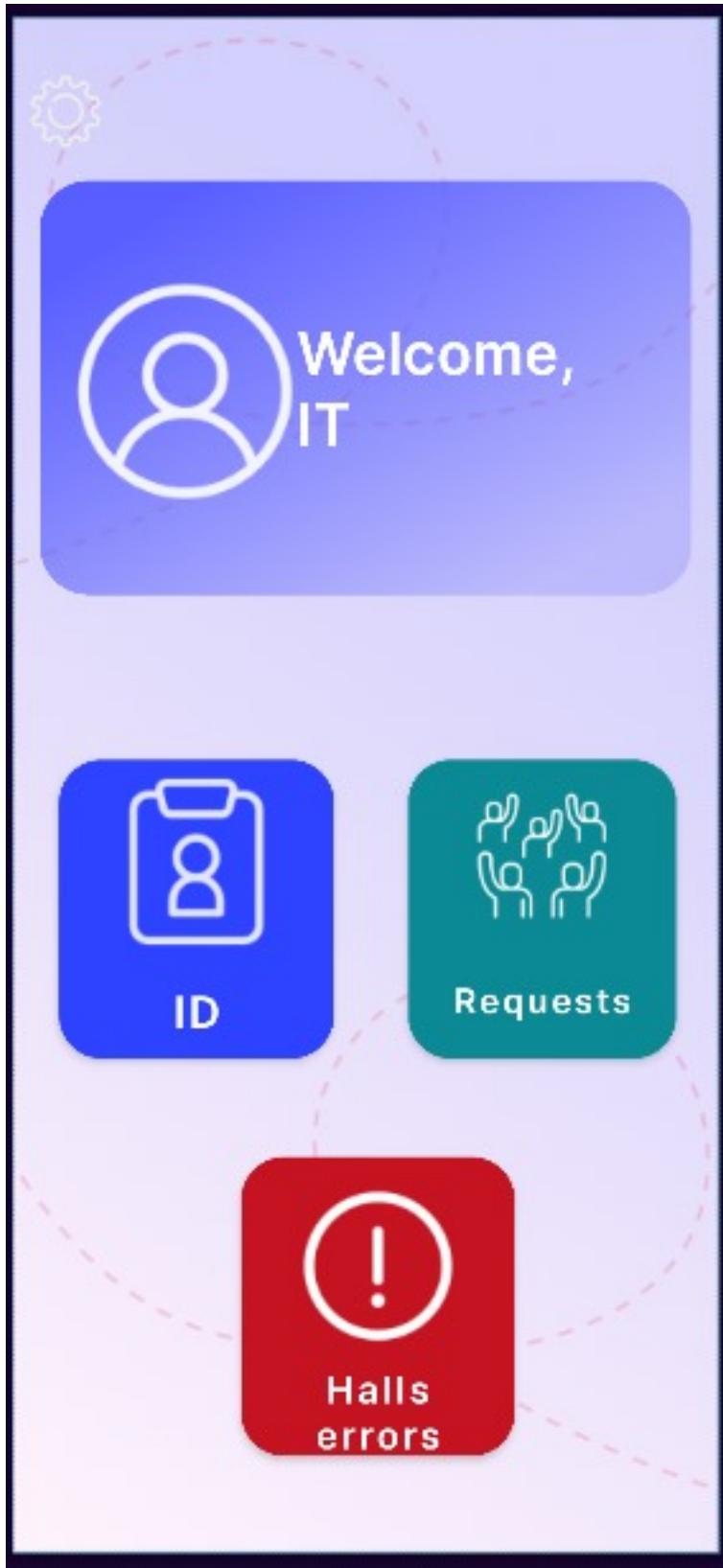
2.4 This frame represents the Attendance frame



2.5 This frame represents the Q&A Frame for Staff



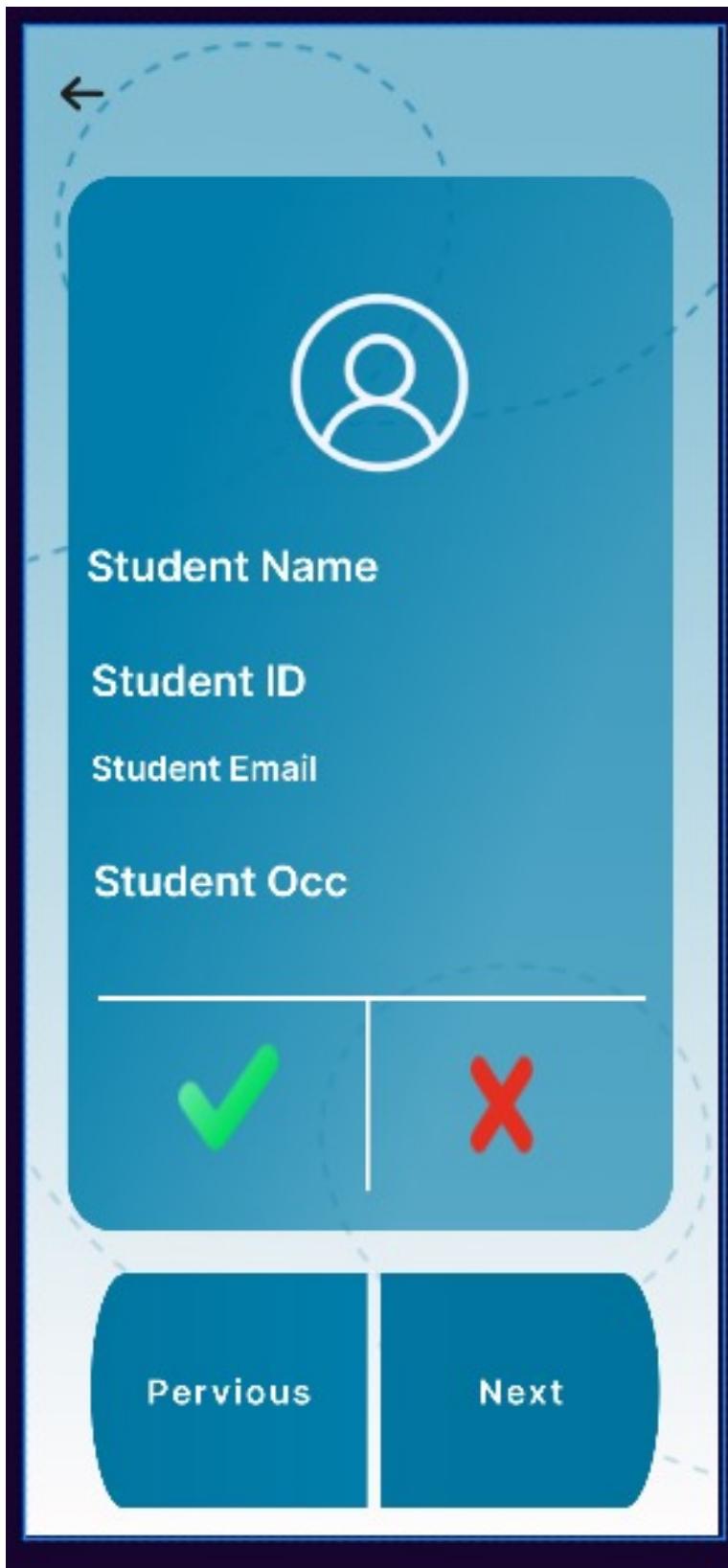
3. This frame represents the IT home page



3.1 This frame represents the Punch in and Punch out for IT



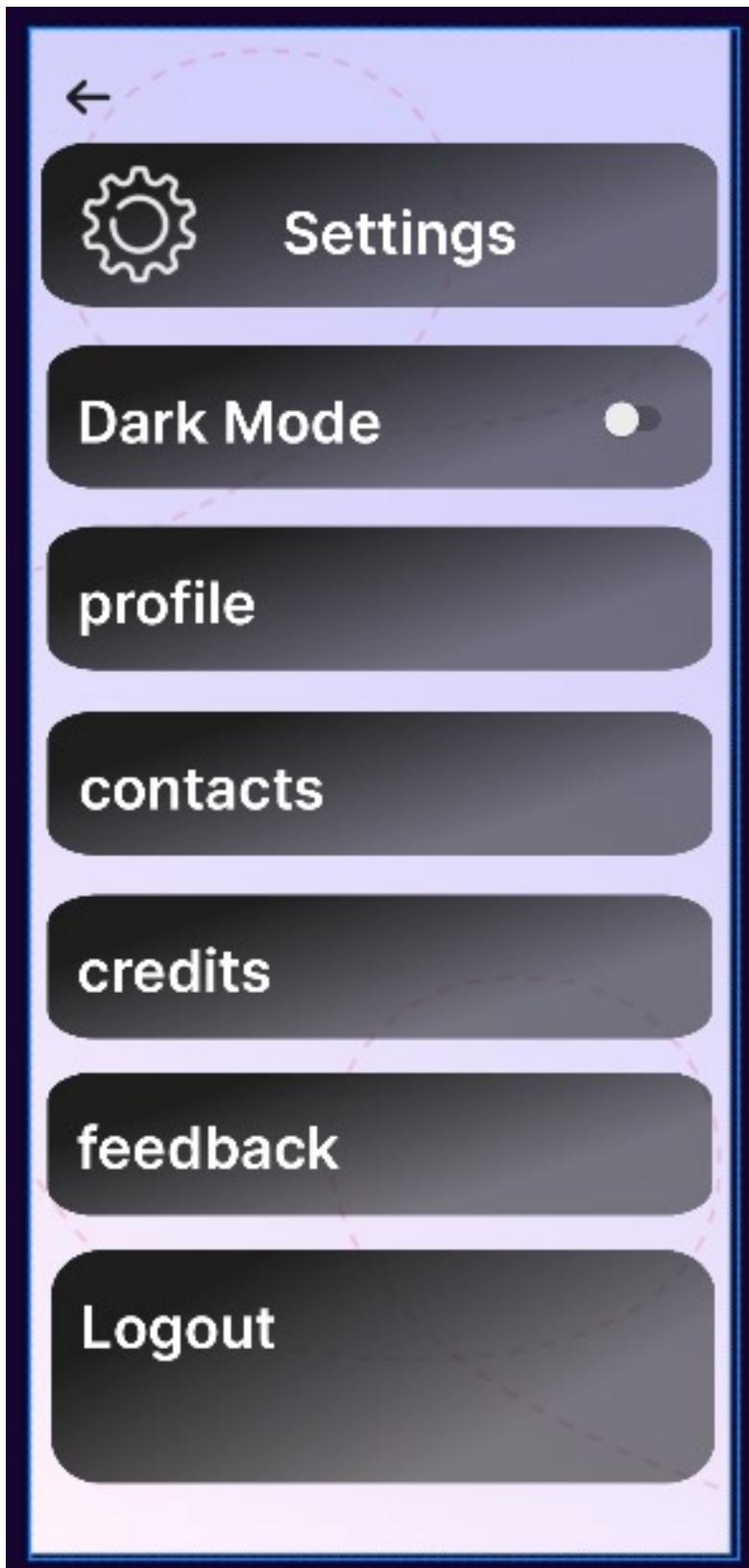
3.2 This frame represents the requests frame



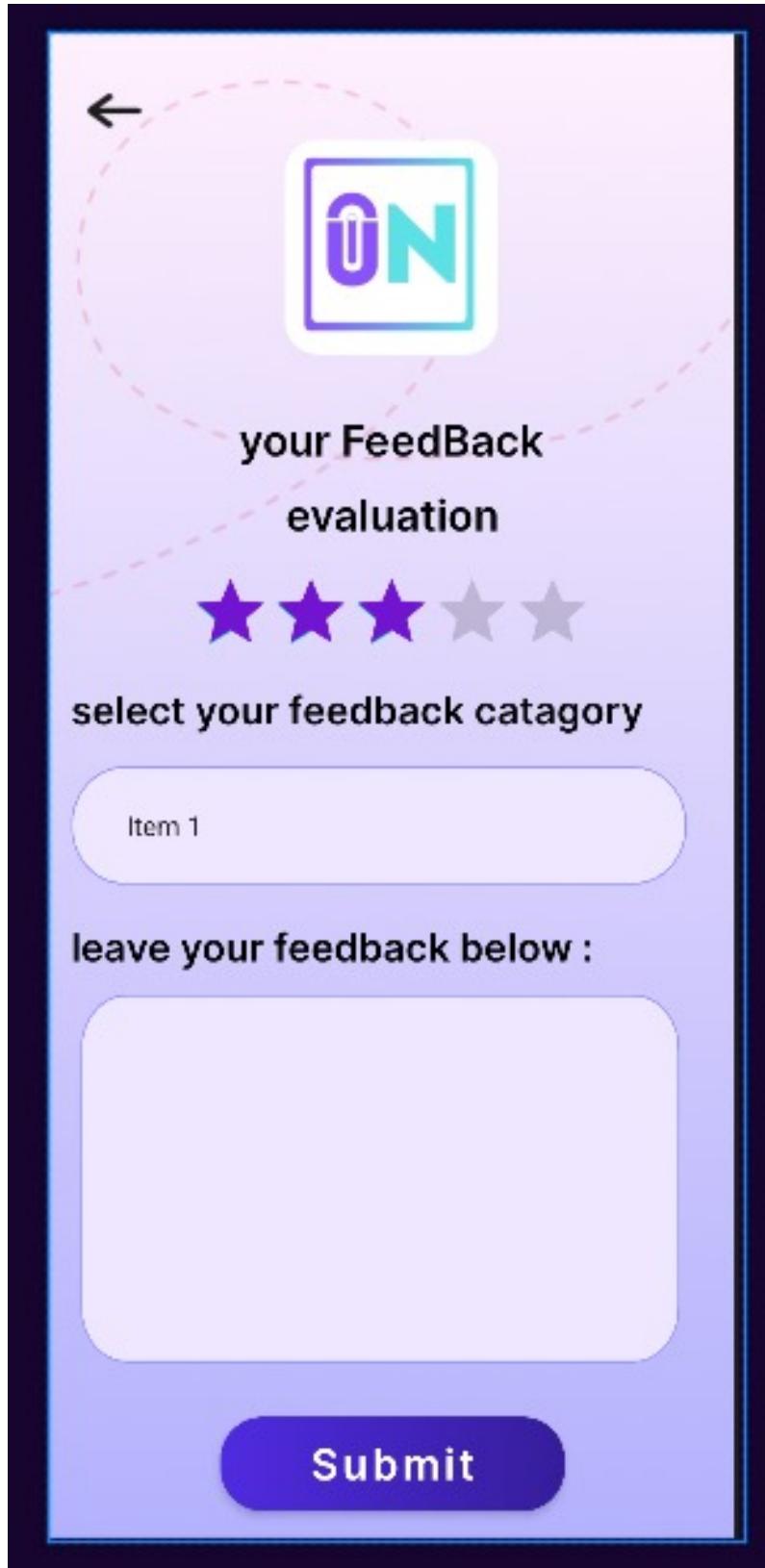
3.3 This frame represents the hall errors for IT



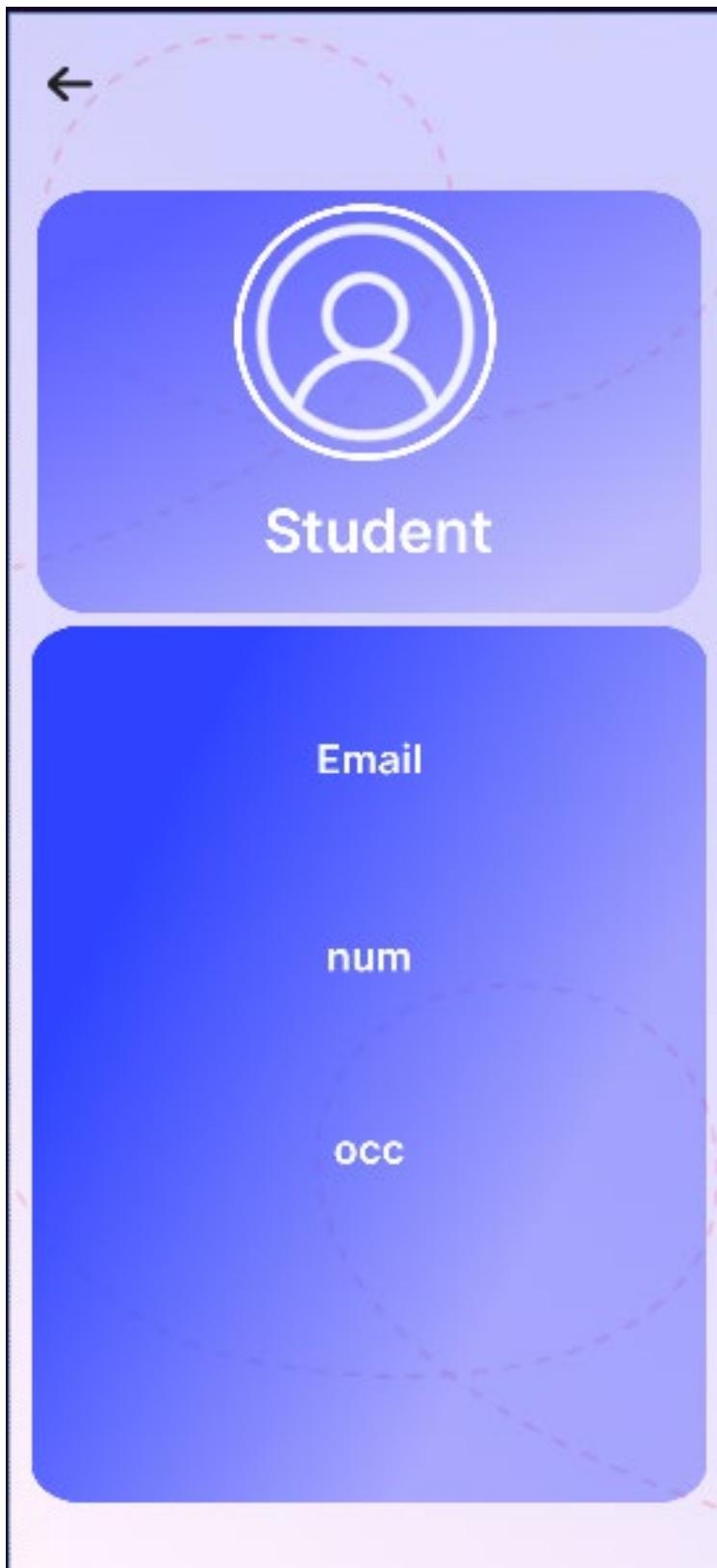
This frame represents the Settings page



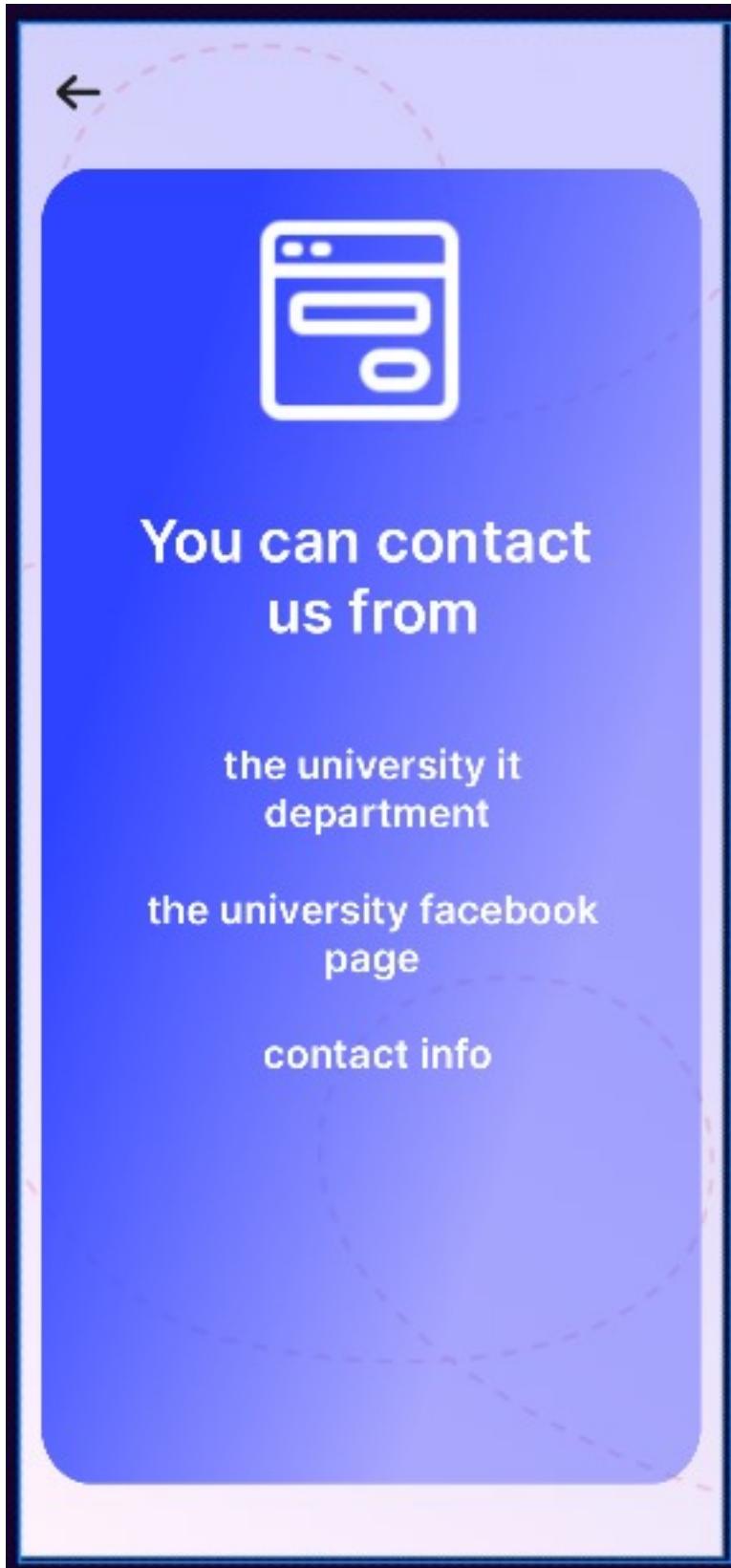
This frame represents the feedback page



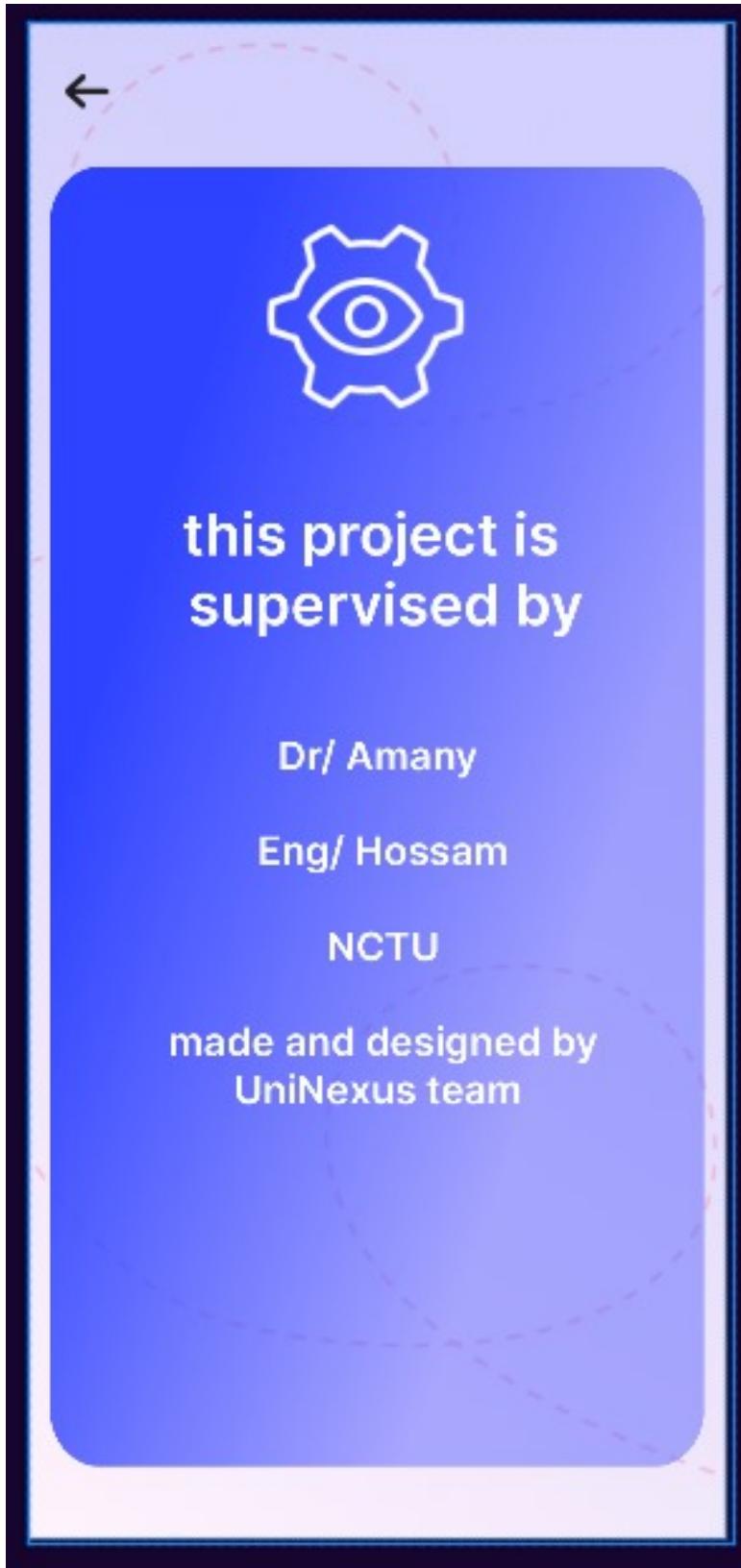
This frame represents the profile page



This frame represents the contacts page



This frame represents the credits page



## **CHAPTER 6**

### **Conclusion And Future Work**

# CHAPTER 6

## INTRODUCTION

Chapter 6 concludes our exploration of the UniNexus mobile application. We'll summarize the key findings and accomplishments throughout the development process, highlighting the app's functionalities and its potential impact on the university community. We'll also discuss potential areas for future development and ongoing enhancements, paving the way for the continued evolution of UniNexus as a valuable tool for students, faculty, and staff.

### 6.1 CONCLUSIONS:

The UniNexus mobile application stands poised to revolutionize the way students, faculty, and staff interact at New Cairo Technological University. This comprehensive project has culminated in the design of a user-centered app that addresses the core needs of the university community.

#### Key Achievements:

- **Streamlined Communication:** UniNexus fosters seamless communication by providing a centralized platform for announcements, messages, and course materials.
- **Enhanced Learning:** The app empowers students with easy access to course materials, attendance records, and the ability to submit attendance (if applicable), promoting a more engaged learning experience.
- **Improved Efficiency:** Faculty can manage course information, record attendance, and communicate with students efficiently, saving valuable time and resources.
- **Streamlined Operations:** Staff can address equipment malfunctions, manage lab availability, and communicate with students and faculty through a centralized platform.

## **6.2 FUTURE WORK:**

UniNexus has the potential to significantly enhance the overall university experience. By fostering better communication, improving access to information, and streamlining processes, the app can contribute to a more efficient, collaborative, and enriching environment for everyone.

Looking ahead, continuous improvement remains paramount. Future iterations of UniNexus could explore features like:

- 1. Expanding Reach:** A native iOS app broadens accessibility and leverages platform strengths.
- 2. Deeper Engagement:** Personalized learning, interactive features, and system integrations enrich the user experience.
- 3. Uninterrupted Flow:** Performance optimizations, offline functionality, and continuous improvement ensure a seamless journey.
- 4. Smart Ecosystem:** UniNexus connects with intelligent devices, creating a personalized and interconnected campus experience.
- 5. Unwavering Security:** Advanced authentication, regular audits, and user education fortify the app's defenses.

## References

- [1] [https://www.researchgate.net/publication/332988829\\_Realizing\\_a\\_Smart\\_University\\_Campus\\_Vision\\_Architecture\\_and\\_Implementation](https://www.researchgate.net/publication/332988829_Realizing_a_Smart_University_Campus_Vision_Architecture_and_Implementation)
- [2] <https://iopscience.iop.org/article/10.1088/1742-6596/1860/1/012008/pdf>
- [3] <https://www.birmingham.ac.uk/documents/university/smart-campus/the-smart-campus.pdf>
- [4] <https://www.academia.edu/download/56635087/1.pdf>

# Appendix

**attend\_act.java:** this code starts the session by generating a unique QR code

```
1 attend_act.java > attend_act
2 package com.example.loginnex;
3
4 import android.content.SharedPreferences;
5 import android.graphics.Bitmap;
6 import android.os.Bundle;
7 import android.view.View;
8 import android.widget.ArrayAdapter;
9 import android.widget.Button;
10 import android.widget.ImageView;
11 import androidx.appcompat.app.AppCompatActivity;
12 import com.google.zxing.BarcodeFormat;
13 import com.google.zxing.MultiFormatWriter;
14 import com.google.zxing.WriterException;
15 import com.google.zxing.common.BitMatrix;
16 import com.journeyapps.barcodescanner BarcodeEncoder;
17 import java.util.ArrayList;
18
19 public class attend_act extends AppCompatActivity {
20
21     private ImageView back;
22     private Spinner time;
23     private Spinner sub;
24     private Spinner mode;
25     private Button start;
26     private ImageView qrcode;
27
28     @Override
29     protected void onCreate(Bundle savedInstanceState) {
30         super.onCreate(savedInstanceState);
31         setContentView(R.layout.attendance);
32         back = findViewById(R.id.backbutton);
33         time = findViewById(R.id.frame1);
34         qrcode = findViewById(R.id.qr);
35         sub = findViewById(R.id.frame2);
36         mode = findViewById(R.id.frame3);
37         start = findViewById(R.id.logbutton);
38
39         SharedPreferences preferences = getSharedPreferences("checkbox" , MODE_PRIVATE);
40         String subj = preferences.getString("subj","");
41         String occ = preferences.getString("Occ","");
42
43         ArrayList<CharSequence> list = new ArrayList<>();
44         list.add(subj); //adds the users sub to the sun spinner
45         ArrayAdapter<CharSequence> adptr = new ArrayAdapter<CharSequence>(this, androidx.appcompat.R.layout.support_simple_spinner_dropdown_item, list);
46         sub.setAdapter(adptr);
47
48         ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(this, R.array.time, androidx.appcompat.R.layout.support_simple_spinner_dropdown_item);
49         time.setAdapter(adapter);
50
51         ArrayAdapter<CharSequence> adaptr = ArrayAdapter.createFromResource(this, R.array.scan, androidx.appcompat.R.layout.support_simple_spinner_dropdown_item);
52         mode.setAdapter(adptr);
53
54
55         start.setOnClickListener(new View.OnClickListener() {
56             @Override
57             public void onClick(View v) {
58                 MultiFormatWriter multiFormatWriter = new MultiFormatWriter();
59                 String clock= time.getSelectedItem().toString();
60                 String sbj= sub.getSelectedItem().toString();
61                 String mde= mode.getSelectedItem().toString();
62                 String qr ="";
63                 if (occ.equals(anObject:"TA"))
64                 {
65                     qr = sbj+" "+clock+" ,sec"+","+"mde;//qr code for the sections
66                 }
67                 if (occ.equals(anObject:"Doctor"))
68                 {
69                     qr = sbj+" "+clock+" ,lec"+","+"mde;// qr code for the lectures
70                 }
71
72                 try {
73                     BitMatrix bitMatrix = multiFormatWriter.encode(qr.toString(), BarcodeFormat.QR_CODE, 300, 300); //encode text to qr code
74
75                     BarcodeEncoder barcodeEncoder = new BarcodeEncoder();
76                     Bitmap bitmap = barcodeEncoder.createBitmap(bitMatrix); //creates the qr code
77
78                     qrcode.setImageBitmap(bitmap); //changes image to the new qr
79
80                 } catch (WriterException e) {
81                     throw new RuntimeException(e);
82                 }
83             }
84         });
85
86         back.setOnClickListener(new View.OnClickListener() {
87             @Override
88             public void onClick(View v) {
89                 onBackPressed();
90             }
91         });
92     }
93 }
```

**cont\_act.java:** this code displays the contacts

```
package com.example.loginnex;

import android.os.Bundle;
import android.view.View;
import android.widget.ImageView;

import androidx.appcompat.app.AppCompatActivity;

public class cont_act extends AppCompatActivity {
    private ImageView btnback;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.contact);
        btnback = findViewById(R.id.backbutton);
        btnback.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                onBackPressed();
            }
        });
    }
}
```

**cred\_act.java:** this code displays the credits

```
package com.example.loginnex;

import android.os.Bundle;
import android.view.View;
import android.widget.ImageView;
import androidx.appcompat.app.AppCompatActivity;

public class cred_act extends AppCompatActivity {
    private ImageView btnback;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.credits);
        btnback = findViewById(R.id.backbutton);
        btnback.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                onBackPressed();
            }
        });
    }
}
```

**doc\_qa.java:** this code takes the message from the database and links it with the sender information

```
1 package com.example.loginnex;
2
3 import android.content.Intent;
4 import android.content.SharedPreferences;
5 import android.net.Uri;
6 import android.os.Bundle;
7 import android.os.Handler;
8 import android.view.View;
9 import android.widget.Button;
10 import android.widget.ImageView;
11 import android.widget.TextView;
12 import android.widget.Toast;
13
14 import androidx.appcompat.app.AppCompatActivity;
15
16 import com.vishnusivadas.advanced_httpurlconnection.PutData;
17
18 public class doc_qa extends AppCompatActivity {
19
20     private ImageView back;
21     private TextView accept;
22     private TextView deny;
23     private Button next;
24     private Button pervious;
25     private TextView name;
26     private TextView phone;
27     private TextView email;
28     private TextView title;
29     private TextView des;
30
31     int renum = 1;
32     int max = 0;
33
34
35     @Override
36     protected void onCreate(Bundle savedInstanceState) {
37         super.onCreate(savedInstanceState);
38         setContentView(R.layout.doc_qa);
39         back = findViewById(R.id.backbutton);
40         accept = findViewById(R.id.respond);
41         deny = findViewById(R.id.decline);
42         next = findViewById(R.id.next);
43         pervious = findViewById(R.id.per);
44         name = findViewById(R.id.stuname);
45         phone = findViewById(R.id.stuphone);
46         email = findViewById(R.id.stuem);
47         title = findViewById(R.id.title);
48         des = findViewById(R.id.desc);
49         Loading dialog = new Loading dialog(doc_qa.this);
```

```

C:/Users/Laptop/Downloads/> doc_qrt.java
50     SharedPreferences preferences = getSharedPreferences("checkbox", MODE_PRIVATE);
51     String sub = preferences.getString("sub", "");
52     String[] reslt = {"",};
53     Handler handler = new Handler();
54     handler.post(new Runnable() {
55         @Override
56         public void run() {
57             loadingdialog.StartLoadingDialog();
58             String[] field = new String[2];
59             field[0] = "renum";
60             field[1] = "sub";
61             String[] data = new String[2];
62             data[0] = String.valueOf(renum); //equal to 1 to get the first row in table for subject
63             data[1] = sub;
64             PutData putata = new PutData("http://192.168.1.38/loginRegister/qarows.php", "POST", field, data); //gets number of row in the table
65             if (putata.startPut()) {
66                 if (putata.onComplete()) {
67                     String top = putata.getResult();
68                     max = Integer.parseInt(top);
69                 }
70             }
71             PutData ptData = new PutData("http://192.168.1.38/loginRegister/qaname.php", "POST", field, data); //gets the name of the user
72             if (ptData.startPut()) {
73                 if (ptData.onComplete()) {
74                     String result = ptData.getResult();
75                     if (!result.equals(anObject: "")) {
76                         name.setText(result);
77                     } else {
78                         name.setText("No Question submitted");
79                     }
80                 }
81             }
82             PutData putData = new PutData("http://192.168.1.38/loginRegister/qaphone.php", "POST", field, data); //gets the ID of the user
83             if (putData.startPut()) {
84                 if (putData.onComplete()) {
85                     String result = putData.getResult();
86                     phone.setText(result);
87                 }
88             }
89             PutData puData = new PutData("http://192.168.1.38/loginRegister/qemail.php", "POST", field, data); //gets the email of the user
90             if (puData.startPut()) {
91                 if (puData.onComplete()) {
92                     result[0] = puData.getResult();
93                     email.setText(result[0]);
94                 }
95             }
96             PutData puDat = new PutData("http://192.168.1.38/loginRegister/qatitle.php", "POST", field, data); //gets the email of the user
97             if (puDat.startPut()) {
98
99                 if (puDat.onComplete()) {
100                     String result = puDat.getResult();
101                     title.setText("Title: "+result);
102                 }
103                 PutData utData = new PutData("http://192.168.1.38/loginRegister/qades.php", "POST", field, data); //gets the desupation of the user
104                 if (utData.startPut()) {
105                     if (utData.onComplete()) {
106                         String result = utData.getResult();
107                         des.setText("Description:\n"+ result);
108                     }
109                 }
110                 loadingdialog.cancelLoadingDialog();
111             }
112         }
113     });
114     next.setOnClickListener(new View.OnClickListener() {
115         @Override
116         public void onClick(View v) {
117             loadingdialog.StartLoadingDialog();
118             if (renum < max) {
119                 renum++; // adds 1 to renum if it's not the last row to go to next row in the table aka the next user
120             }
121             Handler handler = new Handler(); //like the above it gets info about the user to update the app
122             handler.post(new Runnable() {
123                 @Override
124                 public void run() {
125                     String[] field = new String[2];
126                     field[0] = "renum";
127                     field[1] = "sub";
128                     String[] data = new String[2];
129                     data[0] = String.valueOf(renum);
130                     data[1] = sub;
131                     PutData ptData = new PutData("http://192.168.1.38/loginRegister/qaname.php", "POST", field, data); //gets the name of the user
132                     if (ptData.startPut()) {
133                         if (ptData.onComplete()) {
134                             String result = ptData.getResult();
135                             if (!result.equals(anObject: "")) {
136                                 name.setText(result);
137                             }
138                         } else {
139                             name.setText("No Question submitted");
140                         }
141                     }
142                     PutData putData = new PutData("http://192.168.1.38/loginRegister/qaphone.php", "POST", field, data); //gets the ID of the user
143                     if (putData.startPut()) {
144                         if (putData.onComplete()) {
145
146

```

```

147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178 previous.setOnClickListener(new View.OnClickListener() {
179     @Override
180     public void onClick(View v) {
181         loadingdialog.StartLoadingDialog();
182         if (renum != 1) {
183             renum--;//decreases by 1 if it's not already equal to 1 to go back one user
184         }
185         Handler handler = new Handler(); //like the above it gets info about the user to update the app
186         handler.post(new Runnable() {
187             @Override
188             public void run() {
189                 String[] field = new String[2];
190                 field[0] = "renum";
191                 field[1] = "sub";
192                 String[] data = new String[2];
193                 data[0] = String.valueOf(renum);
194                 data[1] = sub;
195
196                 PutData ptData = new PutData("http://192.168.1.38/loginRegister/qaname.php", "POST", field, data); //gets the name of the user
197                 if (ptData.startPut()) {
198                     if (ptData.onComplete()) {
199                         String result = ptData.getResult();
200                         if(!result.equals(anObject:"")){
201                             name.setText(result);
202                         }
203                         else {
204                             name.setText("No Question submited");
205                         }
206                     }
207                     PutData putData = new PutData("http://192.168.1.38/loginRegister/qaphone.php", "POST", field, data); //gets the ID of the user
208                     if (putData.startPut()) {
209                         if (putData.onComplete()) {
210                             String result = putData.getResult();
211                             phone.setText(result);
212                         }
213                     }
214                     PutData puData = new PutData("http://192.168.1.38/loginRegister/qemail.php", "POST", field, data); //gets the email of the user
215                     if (puData.startPut()) {
216                         if (puData.onComplete()) {
217                             result[0] = puData.getResult();
218                             email.setText(result[0]);
219                         }
220                     }
221                     PutData puDat = new PutData("http://192.168.1.38/loginRegister/qatitle.php", "POST", field, data); //gets the email of the user
222                     if (puDat.startPut()) {
223                         if (puDat.onComplete()) {
224                             String result = puDat.getResult();
225                             title.setText("Title: "+result);
226                         }
227                     }
228                     PutData utData = new PutData("http://192.168.1.38/loginRegister/qades.php", "POST", field, data); //gets the desupation of the user
229                     if (utData.startPut()) {
230                         if (utData.onComplete()) {
231                             String result = utData.getResult();
232                             des.setText("Description:\n"+ result);
233                         }
234                     }
235                 });
236                 loadingdialog.CancelLoadingDialog();
237             }
238         });
239         accept.setOnClickListener(new View.OnClickListener() {
240             @Override

```

```

243     public void onClick(View v) {
244         loadingdialog.StartLoadingDialog();
245         Handler handler = new Handler();
246         handler.post(new Runnable() {
247             @Override
248             public void run() {
249                 String[] field = new String[2];
250                 field[0] = "renum";
251                 field[1] = "sub";
252                 String[] data = new String[2];
253                 data[0] = String.valueOf(renum);
254                 data[1] = sub;
255                 Intent mail = new Intent(Intent.ACTION_VIEW);
256                 Uri datas = Uri.parse("mailto:"+result[0]);
257                 mail.setData(datas);
258                 startActivity(mail);
259                 PutData ptData = new PutData("http://192.168.1.38/loginRegister/qact.php", "POST", field, data); //updates the user permission to 1 to be able to access th
260                 if (ptData.startPut()) {
261                     if (ptData.onComplete()) {
262                         String result = ptData.getResult();
263                         if(result.equals(anObject:"accepted")){
264                             Toast.makeText(getApplicationContext(), result, Toast.LENGTH_SHORT).show();
265                         }
266                         Toast.makeText(getApplicationContext(), result, Toast.LENGTH_SHORT).show();
267                     }
268                 }
269                 PutData putata = new PutData("http://192.168.1.38/loginRegister/qarows.php", "POST", field, data); //gets number of row in the table
270                 if (putata.startPut()) {
271                     if (putata.onComplete()) {
272                         String top = putata.getResult();
273                         max = Integer.parseInt(top);
274                     }
275                 }
276                 PutData putDta = new PutData("http://192.168.1.38/loginRegister/qaup.php", "POST", field, data); //reorganize the table
277                 if (putDta.startPut()) {
278                     if (putDta.onComplete()) {
279                         String reut = putDta.getResult();
280                     }
281                 }
282                 PutData pData = new PutData("http://192.168.1.38/loginRegister/qaname.php", "POST", field, data); //gets the name of the user
283                 if (pData.startPut()) {
284                     if (pData.onComplete()) {
285                         String result = pData.getResult();
286                         if(!result.equals(anObject:"")){
287                             name.setText(result);
288                         }
289                     } else {
290                         name.setText("No Question submitted");
291                     }
292                 }
293                 PutData putData = new PutData("http://192.168.1.38/loginRegister/qaphone.php", "POST", field, data); //gets the ID of the user
294                 if (putData.startPut()) {
295                     if (putData.onComplete()) {
296                         String result = putData.getResult();
297                         phone.setText(result);
298                     }
299                 }
300                 PutData puData = new PutData("http://192.168.1.38/loginRegister/qemail.php", "POST", field, data); //gets the email of the user
301                 if (puData.startPut()) {
302                     if (puData.onComplete()) {
303                         result[0] = puData.getResult();
304                         email.setText(result[0]);
305                     }
306                 }
307                 PutData puDat = new PutData("http://192.168.1.38/loginRegister/qatitle.php", "POST", field, data); //gets the email of the user
308                 if (puDat.startPut()) {
309                     if (puDat.onComplete()) {
310                         String result = puDat.getResult();
311                         title.setText("Title: "+result);
312                     }
313                 }
314                 PutData utData = new PutData("http://192.168.1.38/loginRegister/qades.php", "POST", field, data); //gets the desupation of the user
315                 if (utData.startPut()) {
316                     if (utData.onComplete()) {
317                         String result = utData.getResult();
318                         des.setText("Description:\n"+ result);
319                     }
320                 }
321             }
322         });
323     };
324     loadingdialog.CancelloadingDialog();
325 }
326 );
327 );
328 );
329 deny.setOnClickListener(new View.OnClickListener() {
330     @Override
331     public void onClick(View v) {
332         loadingdialog.StartLoadingDialog();
333         Handler handler = new Handler();
334         handler.post(new Runnable() {
335             @Override
336             public void run() {
337                 String[] field = new String[2];
338                 field[0] = "renum";

```

```

338     field[0] = "renum";
339     field[1] = "sub";
340     String[] data = new String[2];
341     data[0] = String.valueOf(renum);
342     data[1] = sub;
343     PutData ptData = new PutData("http://192.168.1.38/loginRegister/qaact.php", "POST", field, data); //updates the user permission to 1 to
344     if (ptData.startPut()) {
345         if (ptData.onComplete()) {
346             String result = ptData.getResult();
347             if(result.equals(anObject:"accepted")) {
348                 Toast.makeText(getApplicationContext(), result, Toast.LENGTH_SHORT).show();
349             }
350             Toast.makeText(getApplicationContext(), result, Toast.LENGTH_SHORT).show();
351         }
352     }
353     PutData putata = new PutData("http://192.168.1.38/loginRegister/qarows.php", "POST", field, data); //gets number of row in the table
354     if (putata.startPut()) {
355         if (putata.onComplete()) {
356             String top = putata.getResult();
357             max = Integer.parseInt(top);
358         }
359     }
360     PutData putDta = new PutData("http://192.168.1.38/loginRegister/qaup.php", "POST", field, data); //reorganize the table
361     if (putDta.startPut()) {
362         if (putDta.onComplete()) {
363             String result = putDta.getResult();
364         }
365     }
366     PutData pData = new PutData("http://192.168.1.38/loginRegister/qaname.php", "POST", field, data); //gets the name of the user
367     if (pData.startPut()) {
368         if (pData.onComplete()) {
369             String result = pData.getResult();
370             if(!result.equals(anObject:"")) {
371                 name.setText(result);
372             }
373         } else {
374             name.setText("No Question submitted");
375         }
376     }
377     PutData putData = new PutData("http://192.168.1.38/loginRegister/qaphone.php", "POST", field, data); //gets the ID of the user
378     if (putData.startPut()) {
379         if (putData.onComplete()) {
380             String result = putData.getResult();
381             phone.setText(result);
382         }
383     }
384 }
385
386 PutData puData = new PutData("http://192.168.1.38/loginRegister/qaemail.php", "POST", field, data); //gets the email of the user
387 if (puData.startPut()) {
388     if (puData.onComplete()) {
389         result[0] = puData.getResult();
390         email.setText(result[0]);
391     }
392 }
393 PutData puDat = new PutData("http://192.168.1.38/loginRegister/qatitle.php", "POST", field, data); //gets the email of the user
394 if (puDat.startPut()) {
395     if (puDat.onComplete()) {
396         String result = puDat.getResult();
397         title.setText("Title: "+result);
398     }
399 }
400 PutData utData = new PutData("http://192.168.1.38/loginRegister/qades.php", "POST", field, data); //gets the desupation of the user
401 if (utData.startPut()) {
402     if (utData.onComplete()) {
403         String result = utData.getResult();
404         des.setText("Description:\n"+ result);
405     }
406 }
407 });
408 loadingdialog.CancelLoadingDialog();
409 }
410 });
411 });
412 back.setOnClickListener(new View.OnClickListener() {
413     @Override
414     public void onClick(View v) {
415         onBackPressed();
416     }
417 });
418 });
419 }
420 }
421

```

## error\_act.java: this code allows the submitting of errors by the staff

```
1 package com.example.loginnex;
2
3 import android.content.Context;
4 import android.net.ConnectivityManager;
5 import android.net.NetworkInfo;
6 import android.os.Bundle;
7 import android.view.View;
8 import android.widget.ArrayAdapter;
9 import android.widget.Button;
10 import android.widget.EditText;
11 import android.widget.ImageView;
12 import android.widget.Spinner;
13 import android.widget.Toast;
14
15 import androidx.appcompat.app.AppCompatActivity;
16
17 import com.vishnusivadas.advanced_httURLConnection.PutData;
18
19 public class error_act extends AppCompatActivity {
20
21     private ImageView back;
22
23     private Spinner hall;
24
25     private Spinner type;
26
27     private Spinner build;
28
29     private EditText des;
30
31     private Button sub;
32
33     @Override
34     protected void onCreate(Bundle savedInstanceState) {
35         super.onCreate(savedInstanceState);
36         setContentView(R.layout.error);
37         back = findViewById(R.id.backbutton);
38         hall = findViewById(R.id.halltext);
39         type = findViewById(R.id.errortext);
40         build = findViewById(R.id.buildtext);
41         des = findViewById(R.id.destext);
42         sub = findViewById(R.id.submit);
43
44         ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(this, R.array.Halls, androidx.appcompat.R.layout.support_simple_spinner_dropdown_item);
45         hall.setAdapter(adapter);
46         ArrayAdapter<CharSequence> adapt = ArrayAdapter.createFromResource(this, R.array.error, androidx.appcompat.R.layout.support_simple_spinner_dropdown_item);
47         type.setAdapter(adapt);
48         ArrayAdapter<CharSequence> adapt1 = ArrayAdapter.createFromResource(this, R.array.Build, androidx.appcompat.R.layout.support_simple_spinner_dropdown_item);
49         build.setAdapter(adapt1);
50
51         sub.setOnClickListener(new View.OnClickListener() {
52             @Override
53             public void onClick(View v) {
54                 ConnectivityManager connectivityManager = (ConnectivityManager) getSystemService(Context.CONNECTIVITY_SERVICE);
55
56                 boolean connected = (connectivityManager.getNetworkInfo(ConnectivityManager.TYPE_MOBILE).getState() == NetworkInfo.State.CONNECTED ||
57                         connectivityManager.getNetworkInfo(ConnectivityManager.TYPE_WIFI).getState() == NetworkInfo.State.CONNECTED);
58
59                 if (connected) {
60                     String name = hall.getSelectedItem().toString(); //takes the data from the edittext , spinners and converts it to string
61                     String error = type.getSelectedItem().toString();
62                     String bld = build.getSelectedItem().toString();
63                     String desc = des.getText().toString();
64                     String[] field = new String[4];
65                     field[0] = "hall";
66                     field[1] = "error";
67                     field[2] = "build";
68                     field[3] = "desc";
69                     String[] data = new String[4];
70                     data[0] = name;
71                     data[1] = error;
72                     data[2] = bld;
73                     data[3] = desc;
74                     PutData ptData = new PutData("http://192.168.1.62/loginRegister/errorsub.php", "POST", field, data); //send info as a new error to the error table
75                     if (ptData.startPut()) {
76                         if (ptData.onComplete()) {
77                             String result = ptData.getResult();
78                             Toast.makeText(getApplicationContext(), result, Toast.LENGTH_SHORT).show();
79                         }
80                     }
81                     PutData ptData1 = new PutData("http://192.168.1.62/loginRegister/errup.php", "POST", field, data); //reorganizes the errors table
82                     if (ptData1.startPut()) {
83                         if (ptData1.onComplete()) {
84                             String result1 = ptData1.getResult();
85                         }
86                     }
87                 }
88             }
89         });
90         back.setOnClickListener(new View.OnClickListener() {
91             @Override
92             public void onClick(View v) {
93                 onBackPressed();
94             }
95         });
96     }
}
```

## errors\_it.java: this code views the available errors

```
1 package com.example.loginnex;
2
3 import android.os.Bundle;
4 import android.os.Handler;
5 import android.view.View;
6 import android.widget.ArrayAdapter;
7 import android.widget.Button;
8 import android.widget.ImageView;
9 import android.widget.Spinner;
10 import android.widget.TextView;
11 import android.widget.Toast;
12
13 import androidx.appcompat.app.AppCompatActivity;
14
15 import com.vishmusivadas.advanced_httpurlconnection.PutData;
16
17 import java.util.ArrayList;
18 import java.util.Arrays;
19
20 public class errors_it extends AppCompatActivity {
21
22     private ImageView back;
23     private Button go;
24     private Spinner hall;
25     private TextView halls;
26     private TextView errortype;
27     private TextView desc;
28     private TextView build;
29     private TextView done;
30
31     int max = 0;
32
33     @Override
34     protected void onCreate(Bundle savedInstanceState) {
35         super.onCreate(savedInstanceState);
36         setContentView(R.layout.error2);
37
38         back = findViewById(R.id.backbutton);
39         hall = findViewById(R.id.hall);
40         halls = findViewById(R.id.haller);
41         go = findViewById(R.id.go);
42         errortype = findViewById(R.id.errortype);
43         desc = findViewById(R.id.desc);
44         build = findViewById(R.id.bulidling);
45         done = findViewById(R.id.done);
46
47         ArrayList<CharSequence> list = new ArrayList<>(); //creates a list to be used a the spinners item
48         list.add(e:"");
49
50         Handler handler = new Handler();
51         handler.post(new Runnable() {
52             @Override
53             public void run() {
54                 String[] field = new String[1];
55                 field[0] = "renum";
56                 String[] data = new String[1];
57                 data[0] = "item";
58                 PutData putata = new PutData("http://192.168.1.62/loginRegister/errors.php", "POST", field, data); //counts the rows in the table
59                 if (putata.startPut()) {
60                     if (putata.onComplete()) {
61                         String top = putata.getResult();
62                         max = Integer.parseInt(top);
63                     }
64                 }
65                 String[] hals = new String[max];
66
67                 for (int i = 1; i <= max; i++) {
68                     int row = i;
69                     String[] fields = new String[1];
70                     fields[0] = "ID";
71                     String[] datas = new String[1];
72                     datas[0] = String.valueOf(row);
73                     PutData putdata = new PutData("http://192.168.1.62/loginRegister/errhall.php", "POST", fields, datas); //gets the halls with errors
74                     if (putdata.startPut()) {
75                         if (putdata.onComplete()) {
76                             String item = putdata.getResult();
77                             if (!Arrays.asList(hals).contains(item)) //checks if this hall was added to the spinner before
78                             {
79                                 hals[i-1] = item;
80                                 list.add(hals[i-1]);
81                             }
82                         }
83                     }
84                 }
85             });
86             ArrayAdapter<CharSequence> adapter = new ArrayAdapter<CharSequence>(this, androidx.appcompat.R.layout.support_simple_spinner_dropdown_item, list); //adds the halls to the spinner
87             hall.setAdapter(adapter);
88
89
90
91
92
93             go.setOnClickListener(new View.OnClickListener() {
94                 @Override
95             
```

```

94     go.setOnClickListener(new View.OnClickListener() {
95         @Override
96         public void onClick(View v) {
97             String hal = hall.getSelectedItem().toString(); //gets the selected hall and converts it to string
98             Handler handler = new Handler();
99             handler.post(new Runnable() {
100                 @Override
101                 public void run() {
102                     halls.setText("Hall "+hal);
103                     String[] field = new String[1];
104                     field[0] = "hall";
105                     String[] data = new String[1];
106                     data[0] = hal;
107                     PutData putDta = new PutData("http://192.168.1.62/loginRegister/errortype.php", "POST", field, data); //gets the error type
108                     if (putDta.startPut()) {
109                         if (putDta.onComplete()) {
110                             String reslt = putDta.getResult();
111                             errtype.setText("Error type: " + reslt);
112                         }
113                     }
114                     PutData ptDta = new PutData("http://192.168.1.62/loginRegister/errordes.php", "POST", field, data); //gets the error description
115                     if (ptDta.startPut()) {
116                         if (ptDta.onComplete()) {
117                             String reslt = ptDta.getResult();
118                             desc.setText("Description:\n" + reslt);
119                         }
120                     }
121                     PutData putata = new PutData("http://192.168.1.62/loginRegister/errbuild.php", "POST", field, data); //gets the halls building
122                     if (putata.startPut()) {
123                         if (putata.onComplete()) {
124                             String reslt = putata.getResult();
125                             build.setText("Building " + reslt);
126                         }
127                     }
128                 }
129             });
130         }
131     });
132 });
133 });
134
done.setOnClickListener(new View.OnClickListener() {
135     @Override
136     public void onClick(View v) {
137         String hal = hall.getSelectedItem().toString();
138         Handler handler = new Handler();

```

```

139         handler.post(new Runnable() {
140             @Override
141             public void run() {
142                 String[] field = new String[1];
143                 field[0] = "hall";
144                 String[] data = new String[1];
145                 data[0] = hal;
146                 PutData putDta = new PutData("http://192.168.1.62/loginRegister/errdone.php", "POST", field, data); //delete selected hall from the table
147                 if (putDta.startPut()) {
148                     if (putDta.onComplete()) {
149                         String reslt = putDta.getResult();
150                         Toast.makeText(getApplicationContext(), reslt, Toast.LENGTH_SHORT).show();
151                     }
152                 }
153                 PutData ptDta = new PutData("http://192.168.1.62/loginRegister/errup.php", "POST", field, data); //reorganizes the table
154                 if (ptDta.startPut()) {
155                     if (ptDta.onComplete()) {
156                         String reut = ptDta.getResult();
157                     }
158                 }
159             }
160         });
161     });
162 });
163
back.setOnClickListener(new View.OnClickListener() {
164     @Override
165     public void onClick(View v) {
166         onBackPressed();
167     }
168 });
169 });
170 });
171 
```

**feedback.java:** this code allows the submission of user's feedback

```
1 package com.example.loginnex;
2
3 import android.content.Context;
4 import android.content.SharedPreferences;
5 import android.net.ConnectivityManager;
6 import android.net.NetworkInfo;
7 import android.os.Bundle;
8 import android.view.View;
9 import android.widget.ArrayAdapter;
10 import android.widget.Button;
11 import android.widget.EditText;
12 import android.widget.ImageView;
13 import android.widget.RatingBar;
14 import android.widget.Spinner;
15 import android.widget.Toast;
16
17 import androidx.appcompat.app.AppCompatActivity;
18
19 import com.vishnusivadas.advanced_httpurlconnection.PutData;
20
21 public class feedback extends AppCompatActivity {
22
23     private ImageView back;
24
25     private Spinner type;
26
27     private EditText des;
28     private RatingBar rate;
29
30     private Button sub;
31
32     @Override
33     protected void onCreate(Bundle savedInstanceState) {
34         super.onCreate(savedInstanceState);
35         setContentView(R.layout.feedback);
36         back = findViewById(R.id.backbutton);
37         type = findViewById(R.id.cat);
38         rate = findViewById(R.id.rate);
39         des = findViewById(R.id.destext);
40         sub = findViewById(R.id.feedback_button);
41
42         SharedPreferences preferences = getSharedPreferences("checkbox", MODE_PRIVATE);
43         ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(this, R.array.cat, androidx.appcompat.R.layout.support_simple_spinner_dropdown_item);
44         type.setAdapter(adapter);
45
46
47         sub.setOnClickListener(new View.OnClickListener() {
48             @Override
49             public void onClick(View v) {

```

```
public void onClick(View v) {
    ConnectivityManager connectivityManager = (ConnectivityManager) getSystemService(Context.CONNECTIVITY_SERVICE);

    boolean connected = (connectivityManager.getNetworkInfo(ConnectivityManager.TYPE_MOBILE).getState() == NetworkInfo.State.CONNECTED ||
    connectivityManager.getNetworkInfo(ConnectivityManager.TYPE_WIFI).getState() == NetworkInfo.State.CONNECTED);
    if (connected) {
        String cat = type.getSelectedItem().toString(); //takes the data from the edittext , spinners and converts it to string
        String desc = des.getText().toString();
        if (!desc.equals("")) {
            String[] field = new String[5];
            field[0] = "Email";
            field[1] = "ID";
            field[2] = "Rating";
            field[3] = "cat";
            field[4] = "feedback";
            String[] data = new String[5];
            data[0] = preferences.getString("email", "");
            data[1] = preferences.getString("ID", "");
            data[2] = String.valueOf(rate.getRating());
            data[3] = cat;
            data[4] = desc;
            PutData pData = new PutData("http://192.168.1.62/loginRegister/feedback.php", "POST", field, data); //send info as a new suggestion to the feedback table
            if (pData.startPut()) {
                if (pData.onComplete()) {
                    String result = pData.getResult();
                    Toast.makeText(getApplicationContext(), result, Toast.LENGTH_SHORT).show();
                }
            }
        } else {
            Toast.makeText(getApplicationContext(), "Internet connection needed", Toast.LENGTH_SHORT).show();
        }
    });
    back.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            onBackPressed();
        }
    });
}

```

## forget\_pass.java: this code allows the submission of forgotten password

```
1 package com.example.loginnex;
2
3 import android.content.Intent;
4 import android.os.Bundle;
5 import android.os.Handler;
6 import android.view.View;
7 import android.widget.ArrayAdapter;
8 import android.widget.Button;
9 import android.widget.EditText;
10 import android.widget.ImageView;
11 import android.widget.Spinner;
12 import android.widget.Toast;
13
14 import androidx.appcompat.app.AppCompatActivity;
15
16 import com.vishnusivadas.advanced_httpurlconnection.PutData;
17
18 public class forget_pass extends AppCompatActivity {
19     private EditText idtext;
20     private EditText email;
21     private Button subbut;
22     private ImageView btnback;
23     private Spinner Occ;
24
25     @Override
26     protected void onCreate(Bundle savedInstanceState) {
27         super.onCreate(savedInstanceState);
28         setContentView(R.layout.forget_act);
29         idtext = findViewById(R.id.IDText);
30         idtext = findViewById(R.id.emailtext);
31         Occ = findViewById(R.id.ha1);
32         subbut = findViewById(R.id.sub);
33         btnback = findViewById(R.id.backbutton);
34
35         ArrayAdapter <CharSequence> adapter = ArrayAdapter.createFromResource(this,R.array.type, androidx.appcompat.R.layout.support_simple_spinner_dropdown_item);
36         Occ.setAdapter(adapter);
37
38         subbut.setOnClickListener(new View.OnClickListener() {
39             @Override
40             public void onClick(View v) {
41                 String item = Occ.getSelectedItem().toString();
42                 if (idtext.length() != 0 && email.length() != 0) {
43                     Handler handler = new Handler();
44                     handler.post(new Runnable() {
45                         @Override
46                         public void run() {
47                             String[] check = new String[1];
48                             check[0] = "ID";
49                             String[] checkd = new String[1];
50
51                             String[] checkd = new String[1];
52                             checkd[0] = String.valueOf(idtext.getText());
53                             if(item.equals(anObject:"Student")){
54                                 PutData putData = new PutData("http://192.168.1.62/loginRegister/check.php", "POST", check, checkd); //checks if student ID is in the table
55                                 if (putData.startPut()) {
56                                     if (putData.onComplete()) {
57                                         String result = putData.getResult();
58                                         if (result.equals(anObject:"Sign up Success")){
59                                             String[] field = new String[2];
60                                             field[0] = "Email";
61                                             field[1] = "ID";
62                                             String[] data = new String[2];
63                                             data[0] = String.valueOf(email.getText());
64                                             data[1] = String.valueOf(idtext.getText());
65                                             PutData ptData = new PutData("http://192.168.1.62/loginRegister/forget_password.php","POST",field,data); //send user info to the table
66                                             if(ptData.startPut()){
67                                                 if(ptData.onComplete()){
68                                                     String result = ptData.getResult();
69                                                     if(result.equals(anObject:"Request Success")){
70                                                         Toast.makeText(getApplicationContext(),result,Toast.LENGTH_SHORT).show();
71                                                         Intent intent = new Intent(getApplicationContext(),req_act.class);
72                                                         startActivity(intent);
73                                                         finish();
74                                                     }
75                                                 }
76                                             }
77                                         }
78                                     }
79                                 else {
80                                     Toast.makeText(getApplicationContext(), result, Toast.LENGTH_SHORT).show();
81                                 }
82                             }
83                         }
84                     }
85                     if(item.equals(anObject:"Doctor")){
86                         PutData putData = new PutData("http://192.168.1.62/loginRegister/check2.php", "POST", check, checkd); //checks if dr ID is in the table
87                         if (putData.startPut()) {
88                             if (putData.onComplete()) {
89                                 String result = putData.getResult();
90                                 if (result.equals(anObject:"Sign up Success")){
91                                     String[] field = new String[2];
```

```

91             String[] field = new String[2];
92             field[0] = "Email";
93             field[1] = "ID";
94             String[] data = new String[2];
95             data[0] = String.valueOf(email.getText());
96             data[1] = String.valueOf(idtext.getText());
97             PutData ptData = new PutData("http://192.168.1.62/loginRegister/forget_password.php", "POST", field, data);
98             if(ptData.startPut()) {
99                 if(ptData.onComplete()){
100                     String result = ptData.getResult();
101                     if(result.equals(anObject:"Request Success")){
102                         Toast.makeText(getApplicationContext(),result,Toast.LENGTH_SHORT).show();
103                         Intent intent = new Intent(getApplicationContext(),req_act.class);
104                         startActivity(intent);
105                         finish();
106                     }
107                 }
108             }
109         }
110     }
111 }
112 else {
113     Toast.makeText(getApplicationContext(), result, Toast.LENGTH_SHORT).show();
114 }
115 }
116 }
117 }
118 }
119 if(item.equals(anObject:"TA")){
120     PutData putData = new PutData("http://192.168.1.62/loginRegister/check3.php", "POST", check, checkd); //checks if TA ID is in the table
121     if (putData.startPut()) {
122         if (putData.onComplete()) {
123             String result = putData.getResult();
124             if (result.equals(anObject:"Sign up Success")) {
125                 String[] field = new String[2];
126                 field[0] = "Email";
127                 field[1] = "ID";
128                 String[] data = new String[2];
129                 data[0] = String.valueOf(email.getText());
130                 data[1] = String.valueOf(idtext.getText());
131                 PutData ptData = new PutData("http://192.168.1.62/loginRegister/forget_password.php", "POST", field, data);
132                 if(ptData.startPut()) {
133                     if(ptData.onComplete()){
134                         String result = ptData.getResult();
135                         if(result.equals(anObject:"Request Success")){
136                             Toast.makeText(getApplicationContext(),result,Toast.LENGTH_SHORT).show();
137                             Intent intent = new Intent(getApplicationContext(),req_act.class);
138                             startActivity(intent);
139                             finish();
140                         }
141                     }
142                 }
143             }
144         }
145     }
146 }
147 else {
148     Toast.makeText(getApplicationContext(), result, Toast.LENGTH_SHORT).show();
149 }
150 }
151 }
152 }
153 if(item.equals(anObject:"Stuff")){
154     PutData putData = new PutData("http://192.168.1.62/loginRegister/check4.php", "POST", check, checkd); //checks if IT ID is in the table
155     if (putData.startPut()) {
156         if (putData.onComplete()) {
157             String result = putData.getResult();
158             if (result.equals(anObject:"Sign up Success")) {
159                 String[] field = new String[2];
160                 field[0] = "Email";
161                 field[1] = "ID";
162                 String[] data = new String[2];
163                 data[0] = String.valueOf(email.getText());
164                 data[1] = String.valueOf(idtext.getText());
165                 PutData ptData = new PutData("http://192.168.1.62/loginRegister/forget_password.php", "POST", field, data);
166                 if(ptData.startPut()) {
167                     if(ptData.onComplete()){
168                         String result = ptData.getResult();
169                         if(result.equals(anObject:"Request Success")){
170                             Toast.makeText(getApplicationContext(),result,Toast.LENGTH_SHORT).show();
171                             Intent intent = new Intent(getApplicationContext(),req_act.class);
172                             startActivity(intent);
173                             finish();
174                         }
175                     }
176                 }
177             }
178         }
179     }

```

```

174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202

```

**halls\_act.java:** this code allows the staff to see the availability and errors of halls

```

1 package com.example.logimnex;
2
3 import android.graphics.Color;
4 import android.os.Bundle;
5 import android.view.View;
6 import android.widget.ArrayAdapter;
7 import android.widget.Button;
8 import android.widget.ImageView;
9 import android.widget.Spinner;
10 import android.widget.TextView;
11
12 import androidx.appcompat.app.AppCompatActivity;
13
14 import com.vishnusivadas.advanced_httpurlconnection.PutData;
15
16
17 public class halls_act extends AppCompatActivity {
18
19     private Spinner halls;
20     private Button gobut;
21     private ImageView btnback;
22     private TextView hallname;
23     private TextView hallbuilding;
24     private TextView hallav;
25     private TextView hallerror;
26     private TextView errode;
27
28     @Override
29     protected void onCreate(Bundle savedInstanceState) {
30         super.onCreate(savedInstanceState);
31         setContentView(R.layout.halls);
32         btnback = findViewById(R.id.backbutton);
33         gobut = findViewById(R.id.Gobut);
34         halls = findViewById(R.id.hall);
35         hallname = findViewById(R.id.hall);
36         hallbuilding = findViewById(R.id.building);
37         hallav = findViewById(R.id.aval);
38         hallerror = findViewById(R.id.erotype);
39         errode = findViewById(R.id.erode);
40
41         ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(this,R.array.Halls, androidx.appcompat.R.layout.support_simple_spinner_dropdown_item);
42         halls.setAdapter(adapter);
43
44         gobut.setOnClickListener(new View.OnClickListener() {
45             @Override
46             public void onClick(View v) {
47                 String item = halls.getSelectedItem().toString();
48                 String[] field= new String[1];
49                 field[0] = "hall";
50             }
51         });
52     }
53
54 }

```

```

49         field[0] = "hall";
50         String[] data = new String[1];
51         data[0] = item;
52         hallname.setText("Hall "+item);
53         PutData ptDta = new PutData("http://192.168.1.62/loginRegister/hallbuild.php", "POST",field,data); //gets halls buildings
54         if (ptDta.startPut()) {
55             if (ptDta.onComplete()) {
56                 String result = ptDta.getResult();
57                 hallbuilding.setText("Building "+result);
58             }
59         }
60         PutData putData = new PutData("http://192.168.1.62/loginRegister/hallava.php", "POST",field,data); //gets halls availability
61         if (putData.startPut()) {
62             if (putData.onComplete()) {
63                 String result = putData.getResult();
64                 if(result.equals(anObject:"1"))           //sees if the hall have people in it
65                 {
66                     hallav.setText("Full");
67                     hallav.setTextColor(Color.parseColor("#FF0000"));
68                 }
69                 if(result.equals(anObject:"0"))           //sees if the hall doesn't have people in it
70                 {
71                     hallav.setText("Empty");
72                     hallav.setTextColor(Color.parseColor("#00FF38"));
73                 }
74             }
75         }
76         PutData putDta = new PutData("http://192.168.1.62/loginRegister/errorstype.php", "POST",field,data);      //gets the error type of the hall if it existed
77         if (putDta.startPut()) {
78             if (putDta.onComplete()) {
79                 String reslt = putDta.getResult();
80                 if(reslt.equals(anObject:""))
81                 {
82                     hallerror.setText("Error type: No Error");
83                 }
84                 else {
85                     hallerror.setText("Error type: " + reslt);
86                 }
87             }
88         }
89         PutData utDta = new PutData("http://192.168.1.62/loginRegister/errordes.php", "POST",field,data);      //gets the error description of the hall if it existed
90         if (utDta.startPut()) {
91             if (utDta.onComplete()) {
92                 String relt = utDta.getResult();
93                 errode.setText("Description:\n "+relt);
94             }
95         }
96     });
97 }
98
99
100
101     btnback.setOnClickListener(new View.OnClickListener() {
102         @Override
103         public void onClick(View v) {
104             onBackPressed();
105         }
106     });
107 }
108
109
110
111 /**
112 */

```

**it\_home.java:** this code forms the home page for IT and admins

```
1 package com.example.loginnex;
2
3 import android.content.Context;
4 import android.content.Intent;
5 import android.content.SharedPreferences;
6 import android.net.ConnectivityManager;
7 import android.net.NetworkInfo;
8 import android.os.Bundle;
9 import android.view.View;
10 import android.widget.Button;
11 import android.widget.TextView;
12 import android.widget.Toast;
13
14 import androidx.appcompat.app.AppCompatActivity;
15
16 public class it_home extends AppCompatActivity {
17     private Button idbut;
18     private Button schbut;
19     private Button chatbut;
20     private Button linkbut;
21     private TextView wel;
22
23     @Override
24     protected void onCreate(Bundle savedInstanceState) {
25         super.onCreate(savedInstanceState);
26         setContentView(R.layout.it_home);
27         wel = findViewById(R.id.welcome_stu);
28         SharedPreferences preferences = getSharedPreferences("checkbox", MODE_PRIVATE); //a private file for the app
29         String fname = preferences.getString("fname", ""); //variable in the file
30         String lname = preferences.getString("lname", ""); //variable in the file
31         String welcome = "Welcome, " + fname + " " + lname;
32         wel.setText(welcome); //changes the welcome text to show users name
33
34     }
35
36     public void idbut(View view) {
37         startActivity(new Intent(it_home.this, staff_id.class));
38     }
39
40     public void reqbut(View view) {
41         ConnectivityManager connectivityManager = (ConnectivityManager) getSystemService(Context.CONNECTIVITY_SERVICE);
42
43         boolean connected = (connectivityManager.getNetworkInfo(ConnectivityManager.TYPE_MOBILE).getState() == NetworkInfo.State.CONNECTED ||
44             connectivityManager.getNetworkInfo(ConnectivityManager.TYPE_WIFI).getState() == NetworkInfo.State.CONNECTED);
45         if (connected) {
46             startActivity(new Intent(it_home.this, user_request.class));
47         } else {
48             Toast.makeText(getApplicationContext(), "Internet connection needed", Toast.LENGTH_SHORT).show();
49
50         }
51     }
52
53     public void errorbut(View view) {
54         ConnectivityManager connectivityManager = (ConnectivityManager) getSystemService(Context.CONNECTIVITY_SERVICE);
55
56         boolean connected = (connectivityManager.getNetworkInfo(ConnectivityManager.TYPE_MOBILE).getState() == NetworkInfo.State.CONNECTED ||
57             connectivityManager.getNetworkInfo(ConnectivityManager.TYPE_WIFI).getState() == NetworkInfo.State.CONNECTED);
58         if (connected) {
59             startActivity(new Intent(it_home.this, errors_it.class));
60         } else {
61             Toast.makeText(getApplicationContext(), "Internet connection needed", Toast.LENGTH_SHORT).show();
62         }
63     }
64
65     public void logoutbut(View view) {
66         SharedPreferences preferences = getSharedPreferences("checkbox", MODE_PRIVATE);
67         SharedPreferences.Editor editor = preferences.edit();
68         editor.putString("remember", "false");
69         editor.putString("Orc", "");
70         editor.putString("email", "");
71         editor.putString("lname", "");
72         editor.putString("fname", "");
73         editor.apply();
74         startActivity(new Intent(it_home.this, main_activity.class));
75     }
76     public void setbut(View view) {
77         startActivity(new Intent(it_home.this, set_act.class));
78     }
79
80     @Override
81     public void onBackPressed() {
82
83     }
84
85 }
```

**login\_activity.java:** this code allows the login to the application

```
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135 } else if (item.equals(anObject:"Doctor")) {//the same as the above
    PutData putData = new PutData("http://192.168.1.62/loginRegister/login2.php", "POST", field, data);
    if (putData.startPut()) {
```



```

264             if (reID.startPut()) {
265                 if (reID.onComplete()) {
266                     String ID = reID.getResult();
267                     PutData refname = new PutData("http://192.168.1.62/loginRegister/refname4.php", "POST", redata, redata2);
268                     if (refname.startPut()) {
269                         if (refname.onComplete()) {
270                             String fname = refname.getResult();
271                             PutData relname = new PutData("http://192.168.1.62/loginRegister/relname4.php", "POST", redata, redata2);
272                             if (relname.startPut()) {
273                                 if (relname.onComplete()) {
274                                     String lname = relname.getResult();
275                                     PutData rephone = new PutData("http://192.168.1.62/loginRegister/rephone2.php", "POST", redata, redata2);
276                                     if (rephone.startPut()) {
277                                         if (rephone.onComplete()) {
278                                             String phone = rephone.getResult();
279                                             Toast.makeText(getApplicationContext(), result, Toast.LENGTH_SHORT).show();
280                                             Sharedpreferences preferences = getSharedPreferences("checkbox", MODE_PRIVATE);
281                                             Sharedpreferences.Editor editor = preferences.edit();
282                                             editor.putString("remember", "true");
283                                             editor.putString("email", email);
284                                             editor.putString("ID", ID);
285                                             editor.putString("lname", lname);
286                                             editor.putString("fname", fname);
287                                             editor.putString("phonenum", phone);
288                                             editor.putString("Occ", item);
289                                             editor.apply();
290                                             Intent intent = new Intent(getApplicationContext(), it_home.class);
291                                             startActivity(intent);
292                                             finish();
293                                         }
294                                     }
295                                 }
296                             }
297                         }
298                     }
299                 }
300             }
301         } else {
302             Toast.makeText(getApplicationContext(), result, Toast.LENGTH_SHORT).show();
303         }
304     }
305 }
306 }

307     });
308 } else {
309     Toast.makeText(getApplicationContext(), "ALL fields required", Toast.LENGTH_SHORT).show();
310 }
311 }
312 }
313 });
314 btnback.setOnClickListener(new View.OnClickListener() {
315     @Override
316     public void onClick(View v) {
317         onBackPressed();
318     }
319 });
320 forget.setOnClickListener(new View.OnClickListener() {
321     @Override
322     public void onClick(View v) {
323         startActivity(new Intent(login_activity.this, forget_pass.class));
324     }
325 });
326

Pass.setOnClickListener(new View.OnClickListener() {
328     boolean isPasswordVisible = false;
329
330     @Override
331     public void onClick(View v) {
332         if (isPasswordVisible) {
333             Pass.setInputType(InputType.TYPE_CLASS_TEXT | InputType.TYPE_TEXT_VARIATION_PASSWORD);
334             Pass.setCompoundDrawablesWithIntrinsicBounds(0, 0, R.drawable.hide, 0);
335         } else {
336             Pass.setInputType(InputType.TYPE_CLASS_TEXT | InputType.TYPE_TEXT_VARIATION_VISIBLE_PASSWORD);
337             Pass.setCompoundDrawablesWithIntrinsicBounds(0, 0, R.drawable.show, 0);
338         }
339         isPasswordVisible = !isPasswordVisible;
340         Pass.setSelection(Pass.getText().length());
341     }
342 });
343
344 }

```

**main\_activity.java:** this code allows to choose between login and register or if you have logged in before it will redirect you to your home page

```
1 package com.example.loginnex;
2
3
4 import android.content.Intent;
5 import android.content.SharedPreferences;
6 import android.os.Bundle;
7 import android.view.View;
8
9 import androidx.appcompat.app.AppCompatActivity;
10
11
12 public class main_activity extends AppCompatActivity {
13
14
15
16
17
18     @Override
19     protected void onCreate(Bundle savedInstanceState) {
20         super.onCreate(savedInstanceState);
21         setContentView(R.layout.main);
22
23         SharedPreferences preferences = getSharedPreferences("checkbox" ,MODE_PRIVATE);
24         String checkbox = preferences.getString("remember","");
25         String user = preferences.getString("Occ","");
26
27         if(checkbox.equals("true"))//checks if user was logged in before
28         {
29             Intent intent;
30             if(user.equals("Student")) {
31                 intent = new Intent(getApplicationContext() , stu_home.class);
32                 startActivity(intent);
33             }
34             else if(user.equals("Staff")) {
35                 intent = new Intent(getApplicationContext() , it_home.class);
36                 startActivity(intent);
37             }
38             else if(user.equals("TA")||user.equals("Doctor")) {
39                 intent = new Intent(getApplicationContext() , staff_home.class);
40                 startActivity(intent);
41             }
42         }
43     }
44
45     public void login(View view) {
46         startActivity(new Intent(main_activity.this , login_activity.class));
47     }
48
49     public void reg(View view) {
50         startActivity(new Intent(main_activity.this , reg_activity.class));
51     }
52
53
54     @Override
55     public void onBackPressed() {
56
57     }
58 //]
59
60
61 }
```

**prof\_act.java:** this code generates the profile page for the user

```
1 package com.example.loginnex;
2
3 import android.content.SharedPreferences;
4 import android.os.Bundle;
5 import android.view.View;
6 import android.widget.ImageView;
7 import android.widget.TextView;
8
9 import androidx.appcompat.app.AppCompatActivity;
10
11 public class prof_act extends AppCompatActivity{
12     private ImageView back;
13     private TextView name;
14     private TextView mail;
15     private TextView phone;
16     private TextView occ;
17     private TextView fac;
18
19
20     @Override
21     protected void onCreate(Bundle savedInstanceState) {
22         super.onCreate(savedInstanceState);
23         setContentView(R.layout.profile_act);
24         name = findViewById(R.id.name);
25         mail = findViewById(R.id.email);
26         phone = findViewById(R.id.phone);
27         back = findViewById(R.id.backbutton);
28         occ = findViewById(R.id.occ);
29         fac = findViewById(R.id.fac);
30
31
32         SharedPreferences preferences = getSharedPreferences("checkbox" ,MODE_PRIVATE);
33         String fname = preferences.getString("fname","");
34         String lname = preferences.getString("lname","");
35         String email_ = preferences.getString("email","");
36         String phone_ = preferences.getString("phonenum","");
37         String occ_ = preferences.getString("Occ","");
38         String fac_ = preferences.getString("faculty","");
39         String id_ = preferences.getString("ID","");
40         String name_ = fname + " " + lname;
41         String case_ = fac_ + " - " + id_;
42         name.setText(name_);
43         mail.setText(email_);
44         phone.setText(phone_);
45         occ.setText(occ_);
46
47         if (occ_.equals("Student")){
48             fac.setText(case_);
49         }else {
50             fac.setText(id_);
51         }
52     }
53
54
55
56
57
58
59
60
61         back.setOnClickListener(new View.OnClickListener() {
62             @Override
63             public void onClick(View v) {
64                 onBackPressed();
65             }
66         });
67     }
68
69     //
70 }
71
```

**Reg\_activity.java:** this code allows the user to register

```

93
94
95
96
97
98 } else if(item.equals(anObject:"Doctor")) {//the same as the above
99 PutData puData = new PutData("http://192.168.1.62/loginRegister/check2.php", "POST", check, checkd);
100 if (puData.startPut()) {
101     if (puData.onComplete()) {
102         String result = puData.getResult();
103         if (result.equals(anObject:"Sign up Success")) {
104             String[] field = new String[4];
105             field[0] = "FName";
106             field[1] = "ID";
107             field[2] = "Email";
108             field[3] = "Occ";
109             String[] data = new String[4];
110             data[0] = String.valueOf(fname.getText());
111             data[1] = String.valueOf(ID.getText());
112             data[2] = String.valueOf(email.getText());
113             data[3] = item;
114             PutData ptData = new PutData("http://192.168.1.62/loginRegister/signup.php", "POST", field, data);
115             if (ptData.startPut()) {
116                 if (ptData.onComplete()) {
117                     String result = ptData.getResult();
118                     if (result.equals(anObject:"Sign Up Success")) {
119                         String[] fields = new String[1];
120                         fields[0] = "renum";
121                         String[] datas = new String[1];
122                         datas[0] = "renum";
123                         PutData putDta = new PutData("http://192.168.1.62/loginRegister/requp.php", "POST", fields, datas);
124                         if (putDta.startPut()) {
125                             if (putDta.onComplete()) {
126                                 String result = putDta.getResult();
127                             }
128                         }
129                         Intent intent = new Intent(getApplicationContext(), req_act.class);
130                         startActivity(intent);
131                         finish();
132                     } else {
133                         Toast.makeText(getApplicationContext(), result, Toast.LENGTH_SHORT).show();
134                     }
135                 }
136             }
137         } else {
138             Toast.makeText(getApplicationContext(), result, Toast.LENGTH_SHORT).show();
139         }
140     }
141 }
142 }
143 }
144 }
145 else if(item.equals(anObject:"TA")) {
146     PutData puData = new PutData("http://192.168.1.62/loginRegister/check3.php", "POST", check, checkd);
147     if (puData.startPut()) {
148         if (puData.onComplete()) {
149             String result = puData.getResult();
150             if (result.equals(anObject:"Sign up Success")) {
151                 String[] field = new String[4];
152                 field[0] = "FName";
153                 field[1] = "ID";
154                 field[2] = "Email";
155                 field[3] = "Occ";
156                 String[] data = new String[4];
157                 data[0] = String.valueOf(fname.getText());
158                 data[1] = String.valueOf(ID.getText());
159                 data[2] = String.valueOf(email.getText());
160                 data[3] = item;
161                 PutData ptData = new PutData("http://192.168.1.62/loginRegister/signup.php", "POST", field, data);
162                 if (ptData.startPut()) {
163                     if (ptData.onComplete()) {
164                         String result = ptData.getResult();
165                         if (result.equals(anObject:"Sign Up Success")) {
166                             String[] fields = new String[1];
167                             fields[0] = "renum";
168                             String[] datas = new String[1];
169                             datas[0] = "renum";
170                             PutData putDta = new PutData("http://192.168.1.62/loginRegister/requp.php", "POST", fields, datas);
171                             if (putDta.startPut()) {
172                                 if (putDta.onComplete()) {
173                                     String result = putDta.getResult();
174                                 }
175                             }
176                             Intent intent = new Intent(getApplicationContext(), req_act.class);
177                             startActivity(intent);
178                             finish();
179                         } else {
180                             Toast.makeText(getApplicationContext(), result, Toast.LENGTH_SHORT).show();
181                         }
182                     }
183                 }
184             }
185         }
186     }
187 }
188 }

```

```

181         }
182     }
183     }
184     }
185     else {
186         Toast.makeText(getApplicationContext(), result, Toast.LENGTH_SHORT).show();
187     }
188 }
189 }
190 }
191 else if(item.equals(anObject:"Stuff")) {
192     PutData puData = new PutData("http://192.168.1.62/loginRegister/check4.php", "POST", check, checkd);
193     if (puData.startPut()) {
194         if (puData.onComplete()) {
195             String result = puData.getResult();
196             if (result.equals(anObject:"Sign up Success")) {
197                 String[] field = new String[4];
198                 field[0] = "FName";
199                 field[1] = "ID";
200                 field[2] = "Email";
201                 field[3] = "Oc";
202                 String[] data = new String[4];
203                 data[0] = String.valueOf(fname.getText());
204                 data[1] = String.valueOf(ID.getText());
205                 data[2] = String.valueOf(email.getText());
206                 data[3] = item;
207                 PutData ptData = new PutData("http://192.168.1.62/loginRegister/signup.php", "POST", field, data);
208                 if (ptData.startPut()) {
209                     if (ptData.onComplete()) {
210                         String result = ptData.getResult();
211                         if (result.equals(anObject:"Sign Up Success")) {
212                             String[] fields = new String[1];
213                             fields[0] = "renum";
214                             String[] datas = new String[1];
215                             datas[0] = "renum";
216                             PutData putDta = new PutData("http://192.168.1.62/loginRegister/requp.php", "POST", fields, datas);
217                             if (putDta.startPut()) {
218                                 if (putDta.onComplete()) {
219                                     String result = putDta.getResult();
220                                 }
221                             }
222                         }
223                         Intent intent = new Intent(getApplicationContext(), req_act.class);
224                         startActivity(intent);
225                         finish();
226                     } else {
227                         Toast.makeText(getApplicationContext(), result, Toast.LENGTH_SHORT).show();
228                     }
229                 }
230             }
231         }
232     }
233     else {
234         Toast.makeText(getApplicationContext(), result, Toast.LENGTH_SHORT).show();
235     }
236 }
237 }
238 }
239 }
240 });
241 }
242 }
243 }
244 }
245 );
246 btnback.setOnClickListener(new View.OnClickListener() {
247     @Override
248     public void onClick(View v) {
249         onBackPressed();
250     }
251 });
252 }
253 }//
254

```

**req\_act.java:** this code returns you to the main activity after registering

```
1 package com.example.loginnex;
2
3
4 import android.content.Intent;
5 import android.os.Bundle;
6 import android.view.View;
7 import android.widget.Button;
8 import androidx.appcompat.app.AppCompatActivity;
9
10
11
12 public class req_act extends AppCompatActivity {
13
14     private Button reg;
15
16
17     @Override
18     protected void onCreate(Bundle savedInstanceState) {
19         super.onCreate(savedInstanceState);
20         setContentView(R.layout.req_act);
21
22         reg = findViewById(R.id.regbutton);
23
24
25     }
26
27
28     public void reg(View view) {
29         startActivity(new Intent(req_act.this, main_activity.class));
30     }
31 }//
```

**Set\_act.java:** this code forms the settings page

```
1 package com.example.loginnex;
2
3 import android.content.Intent;
4 import android.content.SharedPreferences;
5 import android.os.Bundle;
6 import android.view.View;
7 import android.widget.ImageView;
8
9 import androidx.appcompat.app.AppCompatActivity;
10
11 public class set_act extends AppCompatActivity {
12
13     private ImageView btnback;
14     @Override
15     protected void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         setContentView(R.layout.settings);
18         btnback = findViewById(R.id.backbutton);
19         btnback.setOnClickListener(new View.OnClickListener() {
20             @Override
21             public void onClick(View v) {
22                 onBackPressed();
23             }
24         });
25     }
26
27     public void logoutbut(View view) {
28         SharedPreferences preferences = getSharedPreferences("checkbox", MODE_PRIVATE);
29         SharedPreferences.Editor editor = preferences.edit();
30         editor.putString("remember", "false");
31         editor.apply();
32         startActivity(new Intent(set_act.this, main_activity.class));
33     }
34
35     public void contbut(View view) {
36         startActivity(new Intent(set_act.this, cont_act.class));
37     }
38
39     public void credbut(View view) {
40         startActivity(new Intent(set_act.this, cred_act.class));
41     }
42
43     public void probut(View view) {
44         startActivity(new Intent(set_act.this, prof_act.class));
45     }
46
47     public void fedbut(View view) {
48         startActivity(new Intent(set_act.this, feedback.class));
49     }
```

**staff\_home.java:** this code forms the home page for the staff

```
1 package com.example.loginnex;
2
3 import android.content.Context;
4 import android.content.Intent;
5 import android.content.SharedPreferences;
6 import android.net.ConnectivityManager;
7 import android.net.NetworkInfo;
8 import android.os.Bundle;
9 import android.view.View;
10 import android.widget.Button;
11 import android.widget.TextView;
12 import android.widget.Toast;
13
14 import androidx.appcompat.app.AppCompatActivity;
15
16 public class staff_home extends AppCompatActivity {
17     private Button idbut;
18     private Button schbut;
19     private Button chatbut;
20     private Button linkbut;
21     private TextView wel;
22
23     @Override
24     protected void onCreate(Bundle savedInstanceState) {
25         super.onCreate(savedInstanceState);
26         setContentView(R.layout.staff_home);
27         wel = findViewById(R.id.welcome_stu);
28         SharedPreferences preferences = getSharedPreferences("checkbox", MODE_PRIVATE);
29         String fname = preferences.getString("fname", "");
30         String lname = preferences.getString("lname", "");
31         String welcome = "Welcome,\n" + fname + " " + lname;
32         wel.setText(welcome); // changes the welcome to have the user's name
33
34     }
35
36     public void staffidbut(View view) {
37         startActivity(new Intent(staff_home.this, staff_id.class));
38     }
39
40     public void hallsbut(View view) {
41         ConnectivityManager connectivityManager = (ConnectivityManager) getSystemService(Context.CONNECTIVITY_SERVICE);
42
43         boolean connected = (connectivityManager.getNetworkInfo(ConnectivityManager.TYPE_MOBILE).getState() == NetworkInfo.State.CONNECTED ||
44                             connectivityManager.getNetworkInfo(ConnectivityManager.TYPE_WIFI).getState() == NetworkInfo.State.CONNECTED);
45         if (connected) {
46             startActivity(new Intent(staff_home.this, halls_act.class));
47         } else {
48             Toast.makeText(getApplicationContext(), "Internet connection needed", Toast.LENGTH_SHORT).show();
49         }
50     }
51
52
53     public void errorbut(View view) {
54         ConnectivityManager connectivityManager = (ConnectivityManager) getSystemService(Context.CONNECTIVITY_SERVICE);
55
56         boolean connected = (connectivityManager.getNetworkInfo(ConnectivityManager.TYPE_MOBILE).getState() == NetworkInfo.State.CONNECTED ||
57                             connectivityManager.getNetworkInfo(ConnectivityManager.TYPE_WIFI).getState() == NetworkInfo.State.CONNECTED);
58         if (connected) {
59             startActivity(new Intent(staff_home.this, error_act.class));
60         } else {
61             Toast.makeText(getApplicationContext(), "Internet connection needed", Toast.LENGTH_SHORT).show();
62         }
63     }
64
65     public void attenbut(View view) {
66         startActivity(new Intent(staff_home.this, attend_act.class));
67     }
68
69     public void logoutbut(View view) {
70         SharedPreferences preferences = getSharedPreferences("checkbox", MODE_PRIVATE);
71         SharedPreferences.Editor editor = preferences.edit();
72         editor.putString("remember", "false");
73         editor.putString("Occ", "");
74         editor.putString("email", "");
75         editor.putString("lname", "");
76         editor.putString("fname", "");
77         editor.apply();
78         startActivity(new Intent(staff_home.this, main_activity.class));
79     }
80     public void setbut(View view) {
81         startActivity(new Intent(staff_home.this, set_act.class));
82     }
83
84
85     @Override
86     public void onBackPressed() {
87
88     }
89 }
```

**staff\_id.java:** this code generates the QR code for punching in and punching out for the staff

```

1 package com.example.loginnex;
2
3 import android.content.SharedPreferences;
4 import android.graphics.Bitmap;
5 import android.os.Bundle;
6 import android.view.View;
7 import android.widget.Button;
8 import android.widget.ImageView;
9 import android.widget.TextView;
10
11 import androidx.appcompat.app.AppCompatActivity;
12
13 import com.google.zxing.BarcodeFormat;
14 import com.google.zxing.MultiFormatWriter;
15 import com.google.zxing.WriterException;
16 import com.google.zxing.common.BitMatrix;
17 import com.journeyapps.barcodescanner BarcodeEncoder;
18
19 public class staff_id extends AppCompatActivity {
20     private ImageView btnback;
21     private ImageView qrcode;
22     private Button punchmode;
23     private TextView punchtext;
24     private View idcard;
25
26     @Override
27     protected void onCreate(Bundle savedInstanceState) {
28         super.onCreate(savedInstanceState);
29         setContentView(R.layout.id_staff);
30
31         btnback = findViewById(R.id.backbutton);
32         qrcode = findViewById(R.id.QR);
33         punchmode = findViewById(R.id.submit);
34         punchtext = findViewById(R.id.buliding);
35         idcard = findViewById(R.id.idcard);
36
37         SharedPreferences preferences = getSharedPreferences("checkbox", MODE_PRIVATE);
38         String ID = preferences.getString("ID", "");
39         String qr;
40         final String[] mode = {"", "punchin"}; //the mode of qr code
41         qr = ID + mode[0]; //the id and the mode
42         MultiFormatWriter multiFormatWriter = new MultiFormatWriter();
43
44         try {
45             BitMatrix bitMatrix = multiFormatWriter.encode(qr.toString(), BarcodeFormat.QR_CODE, 300, 300); //encodes the qr code
46             BarcodeEncoder barcodeEncoder = new BarcodeEncoder();
47             Bitmap bitmap = barcodeEncoder.createBitmap(bitMatrix); //creates the qr
48
49             qrcode.setImageBitmap(bitmap); //changes the image to the qr code
50
51         } catch (WriterException e) {
52             throw new RuntimeException(e);
53         }
54
55
56         punchmode.setOnClickListener(new View.OnClickListener() {
57             @Override
58             public void onClick(View view) {
59                 if (mode[0].equals(anObject: "", "punchin")) { //sees if the mode is in punch in
60                     String qr;
61                     mode[0] = "", "punchout"; //changes the mode into punchout
62                     qr = ID + mode[0]; //the new qr code
63                     punchtext.setText("Punch Out"); //changes the appearance
64                     punchmode.setText("Punch In");
65                     punchmode.setBackgroundResource(R.drawable.punchoutgrad);
66                     idcard.setBackgroundResource(R.drawable.idpurple);
67                     try {
68                         BitMatrix bitMatrix = multiFormatWriter.encode(qr.toString(), BarcodeFormat.QR_CODE, 300, 300);
69
70                         BarcodeEncoder barcodeEncoder = new BarcodeEncoder();
71                         Bitmap bitmap = barcodeEncoder.createBitmap(bitMatrix);
72
73                         qrcode.setImageBitmap(bitmap);
74
75                     } catch (WriterException e) {
76                         throw new RuntimeException(e);
77                     }
78                 }
79             }
80             else if (mode[0].equals(anObject: "", "punchout")) { //the same as the above but in reverse
81                 String qr;
82                 mode[0] = "", "punchin";
83                 qr = ID + mode[0];
84                 punchtext.setText("Punch In");
85                 punchmode.setText("Punch Out");
86                 punchmode.setBackgroundResource(R.drawable.punchinggrad);
87                 idcard.setBackgroundResource(R.drawable.idblue);
88                 try {
89                     BitMatrix bitMatrix = multiFormatWriter.encode(qr.toString(), BarcodeFormat.QR_CODE, 300, 300);
90
91                     BarcodeEncoder barcodeEncoder = new BarcodeEncoder();
92                     Bitmap bitmap = barcodeEncoder.createBitmap(bitMatrix);
93
94                     qrcode.setImageBitmap(bitmap);
95
96                 } catch (WriterException e) {
97                     throw new RuntimeException(e);
98                 }
99             }
100
101         });
102
103         btnback.setOnClickListener(new View.OnClickListener() {
104             @Override
105             public void onClick(View v) {
106                 onBackPressed();
107             }
108         });
109
110
111
112
113

```

**stu\_home.java:** this code forms the home page for the student

```
1 package com.example.loginnex;
2
3 import android.content.Intent;
4 import android.content.SharedPreferences;
5 import android.os.Bundle;
6 import android.view.MenuItem;
7 import android.view.View;
8 import android.widget.ImageButton;
9 import android.widget.TextView;
10
11 import androidx.appcompat.app.AppCompatActivity;
12
13 public class stu_home extends AppCompatActivity{
14     private ImageButton side;
15     private TextView wel;
16
17
18     @Override
19     protected void onCreate(Bundle savedInstanceState) {
20         super.onCreate(savedInstanceState);
21         setContentView(R.layout.stu_home);
22         wel = findViewById(R.id.welcome_stu);
23         side = findViewById(R.id.menu);
24
25
26         SharedPreferences preferences = getSharedPreferences("checkbox", MODE_PRIVATE);
27         String fname = preferences.getString("fname","");
28         String lname = preferences.getString("lname","");
29         String welcome = "Welcome,\n" + fname + " " + lname;
30         wel.setText(welcome);
31
32
33     }
34     public void idbut(View view) {
35         startActivity(new Intent(stu_home.this , stu_id.class));
36     }
37     public void schbut(View view) {
38         startActivity(new Intent(stu_home.this , stu_sch.class));
39     }
40     public void chatbut(View view) {
41         startActivity(new Intent(stu_home.this , stu_chat.class));
42     }
43     public void linkbut(View view) {
44         startActivity(new Intent(stu_home.this , stu_link.class));
45     }
46     public void logoutbut(View view) {
47         SharedPreferences preferences = getSharedPreferences("checkbox", MODE_PRIVATE);
48         SharedPreferences.Editor editor = preferences.edit();
49         editor.putString("remember", "false");
50
51         editor.putString("Occ", "");
52         editor.putString("email", "");
53         editor.putString("lname", "");
54         editor.putString("fname", "");
55         editor.apply();
56         startActivity(new Intent(stu_home.this, main_activity.class));
57     }
58
59     public void setbut(View view) {
60         startActivity(new Intent(stu_home.this , set_act.class));
61     }
62
63     @Override
64     public void onBackPressed() {
65
66     }
67 }
//
```

**Stu\_id.java:** this code generates a unique QR for each student

```
1 package com.example.loginnex;
2
3 import android.content.SharedPreferences;
4 import android.graphics.Bitmap;
5 import android.os.Bundle;
6 import android.view.View;
7 import android.widget.ImageView;
8
9 import androidx.appcompat.app.AppCompatActivity;
10
11 import com.google.zxing.BarcodeFormat;
12 import com.google.zxing.MultiFormatWriter;
13 import com.google.zxing.WriterException;
14 import com.google.zxing.common.BitMatrix;
15 import com.journeyapps.barcodescanner BarcodeEncoder;
16
17 public class stu_id extends AppCompatActivity {
18     private ImageView btnback;
19     private ImageView qrcode;
20
21     @Override
22     protected void onCreate(Bundle savedInstanceState) {
23         super.onCreate(savedInstanceState);
24         setContentView(R.layout.id_stu);
25
26         btnback = findViewById(R.id.backbutton);
27         qrcode = findViewById(R.id.QR);
28         SharedPreferences preferences = getSharedPreferences("checkbox", MODE_PRIVATE);
29         String ID = preferences.getString("ID","");
30
31         MultiFormatWriter multiFormatWriter = new MultiFormatWriter();
32
33         try {
34             BitMatrix bitMatrix = multiFormatWriter.encode(ID.toString(), BarcodeFormat.QR_CODE, 300,300);
35
36             BarcodeEncoder barcodeEncoder = new BarcodeEncoder();
37             Bitmap bitmap = barcodeEncoder.createBitmap(bitMatrix); //creates the students id
38
39             qrcode.setImageBitmap(bitmap);
40
41         }catch (WriterException e){
42             throw new RuntimeException(e);
43         }
44
45         btnback.setOnClickListener(new View.OnClickListener() {
46             @Override
47             public void onClick(View v) {
48                 onBackPressed();
49             }
50         });
51     }
52 }
```

**stu\_link.java:** this code has the links they are presented in the link page

```
1 package com.example.loginnex;
2
3 import android.content.Intent;
4 import android.net.Uri;
5 import android.os.Bundle;
6 import android.view.View;
7 import android.widget.Button;
8 import android.widget.ImageView;
9
10 import androidx.appcompat.app.AppCompatActivity;
11
12 public class stu_link extends AppCompatActivity {
13     private ImageView btnback;
14     private Button uni;
15     private Button data;
16     private Button face;
17     private Button applink;
18
19     @Override
20     protected void onCreate(Bundle savedInstanceState) {
21         super.onCreate(savedInstanceState);
22         setContentView(R.layout.stu_links);
23
24         btnback = findViewById(R.id.backbutton);
25         uni = findViewById(R.id.uni);
26         data = findViewById(R.id.data);
27         face = findViewById(R.id.face);
28         applink = findViewById(R.id.web);
29
30
31         btnback.setOnClickListener(new View.OnClickListener() {
32             @Override
33             public void onClick(View v) {
34                 onBackPressed();
35             }
36         });
37
38         uni.setOnClickListener(new View.OnClickListener() {
39             @Override
40             public void onClick(View v) {
41                 Uri uri = Uri.parse("https://nctu.edu.eg/?lang=ar");
42                 Intent intent = new Intent(Intent.ACTION_VIEW, uri);
43                 startActivity(intent);
44             }
45         });
46
47         data.setOnClickListener(new View.OnClickListener() {
48             @Override
49             public void onClick(View v) {
50                 Uri uri = Uri.parse("http://193.227.0.21/Nctu/Registration/ed_login.aspx");
51                 Intent intent = new Intent(Intent.ACTION_VIEW, uri);
52                 startActivity(intent);
53             }
54         });
55
56         face.setOnClickListener(new View.OnClickListener() {
57             @Override
58             public void onClick(View v) {
59                 Uri uri = Uri.parse("https://www.facebook.com/share/6k1d5SRcd9uPmXEu/?mibextid=qi20mg");
60                 Intent intent = new Intent(Intent.ACTION_VIEW, uri);
61                 startActivity(intent);
62             }
63         });
64
65         applink.setOnClickListener(new View.OnClickListener() {
66             @Override
67             public void onClick(View v) {
68                 Uri uri = Uri.parse("https://nctu.edu.eg/?lang=ar"); // soon
69                 Intent intent = new Intent(Intent.ACTION_VIEW, uri);
70                 startActivity(intent);
71             }
72         });
73
74
75     } // |
76 }
```

**stu\_qa.java:** this code sends the message sent in the app and puts it in the database

```
1 package com.example.loginnex;
2
3 import android.content.Context;
4 import android.content.SharedPreferences;
5 import android.net.ConnectivityManager;
6 import android.net.NetworkInfo;
7 import android.os.Bundle;
8 import android.view.View;
9 import android.widget.ArrayAdapter;
10 import android.widget.Button;
11 import android.widget.EditText;
12 import android.widget.ImageView;
13 import android.widget.Spinner;
14 import android.widget.Toast;
15
16 import androidx.appcompat.app.AppCompatActivity;
17
18 import com.vishnusivadas.advanced_httpurlconnection.PutData;
19
20 public class stu_qa extends AppCompatActivity {
21
22     private ImageView back;
23
24     private Spinner subj;
25
26     private EditText des;
27     private EditText title;
28
29     private Button sub;
30
31     @Override
32     protected void onCreate(Bundle savedInstanceState) {
33         super.onCreate(savedInstanceState);
34         setContentView(R.layout.stu_qa);
35         back = findViewById(R.id.backbutton);
36         subj = findViewById(R.id.halldtext);
37         title = findViewById(R.id.titlertext);
38         des = findViewById(R.id.destext);
39         sub = findViewById(R.id.submit);
40
41         SharedPreferences preferences = getSharedPreferences("checkbox", MODE_PRIVATE);
42         String fac = preferences.getString("faculty", "");
43         String year = preferences.getString("year", "");
44         String facyear = fac+year;
45         switch(facyear){
46             case "T12":
47                 ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(this, R.array.TT2, androidx.appcompat.R.layout.support_simple_spinner_dropdown_item);
48                 subj.setAdapter(adapter);
49                 break;
50
51             sub.setOnClickListener(new View.OnClickListener() {
52                 @Override
53                 public void onClick(View v) {
54                     ConnectivityManager connectivityManager = (ConnectivityManager) getSystemService(Context.CONNECTIVITY_SERVICE);
55
56                     boolean connected = (connectivityManager.getNetworkInfo(ConnectivityManager.TYPE_MOBILE).getState() == NetworkInfo.State.CONNECTED || connectivityManager.getNetworkInfo(ConnectivityManager.TYPE_WIFI).getState() == NetworkInfo.State.CONNECTED);
57                     if (connected) {
58                         String cat = subj.getSelectedItem().toString(); //takes the data from the edittext , spinners and converts it to string
59                         String desc = des.getText().toString();
60                         String titlestring = title.getText().toString();
61                         String[] fields = new String[1];
62                         fields[0] = "renum";
63                         String[] datas = new String[1];
64                         datas[0] = cat;
65                         String num = "";
66                         int qanumber=0;
67                         PutData putDta = new PutData("http://192.168.1.38/loginRegister/qastart.php", "POST", fields, datas);
68                         if (putDta.startPut()) {
69                             if (putDta.onComplete()) {
70                                 num = putDta.getResult();
71                                 if(num.equals(anObject:""))
72                                     {
73                                         qanumber+=1;
74                                     }
75                                 else {
76                                     qanumber = Integer.parseInt(num)+1;
77                                 }
78                             }
79                         }
80                         if (!desc.equals(anObject:"")) {
81                             String[] field = new String[7];
82                             field[0] = "Email";
83                             field[1] = "phone";
84                             field[2] = "name";
85                             field[3] = "title";
86                             field[4] = "desc";
87                             field[5] = "subj";
88                             field[6] = "qnum";
89                             String[] data = new String[7];
90                             data[0] = preferences.getString("email", "");
91                             data[1] = preferences.getString("phonenum", "");
92                             data[2] = preferences.getString("fname", "")+" "+preferences.getString("lname", "");
93                             data[3] = titlestring;
94                             data[4] = desc;
95                             data[5] = cat;
96                         }
97                     }
98                 }
99             }
100         }
101     }
102 }
```

```
100     data[6] = String.valueOf(qanumber);
101    PutData ptData = new PutData("http://192.168.1.38/loginRegister/qasubmit.php", "POST", field, data); //send info as a new suggestion to the feedback table
102    if (ptData.startPut()) {
103        if (ptData.onComplete()) {
104            String result = ptData.getResult();
105            Toast.makeText(getApplicationContext(), result, Toast.LENGTH_SHORT).show();
106        }
107    }
108 } else {
109     Toast.makeText(getApplicationContext(), "Internet connection needed", Toast.LENGTH_SHORT).show();
110 }
111 }
112 });
113 });
114 });
115 });
116 back.setOnClickListener(new View.OnClickListener() {
117     @Override
118     public void onClick(View v) {
119         onBackPressed();
120     }
121 });
122 }
123 }
124 }
125 }
```

**stu\_sch.java:** this code stores the schedule in the schedule page

```
1 package com.example.loginnex;
2
3 import android.content.Intent;
4 import android.content.SharedPreferences;
5 import android.net.Uri;
6 import android.os.Bundle;
7 import android.view.View;
8 import android.widget.Button;
9 import android.widget.ImageView;
10
11 import androidx.appcompat.app.AppCompatActivity;
12
13 public class stu_sch extends AppCompatActivity {
14     private ImageView btnback;
15     private Button sch;
16
17     @Override
18     protected void onCreate(Bundle savedInstanceState) {
19         super.onCreate(savedInstanceState);
20         setContentView(R.layout.stu_schedule);
21
22         btnback = findViewById(R.id.backbutton);
23         sch = findViewById(R.id.down);
24         SharedPreferences preferences = getSharedPreferences("checkbox" ,MODE_PRIVATE);
25         String faculty = preferences.getString("faculty","");
26         String id = preferences.getString("ID","");
27
28
29
30         btnback.setOnClickListener(new View.OnClickListener() {
31             @Override
32             public void onClick(View v) {
33                 onBackPressed();
34             }
35         });
36
37         sch.setOnClickListener(new View.OnClickListener() {
38             @Override
39             public void onClick(View v) {
40
41                 switch (faculty) {
42                     case "IT":
43                         Uri uri = Uri.parse("https://drive.google.com/file/d/1GbRcD4rbiAG3LxlIocEVSA5WWSi4khFQ/view?usp=sharing");
44                         Intent intent = new Intent(Intent.ACTION_VIEW, uri);
45                         startActivity(intent);
46                         break;
47                     case "MECH":
48                         Uri urmech = Uri.parse("https://drive.google.com/file/d/1J0J8wCz0tAvvSyhsT3IzTcFL93GZ2Qzq/view?usp=sharing");
49                         startActivity(new Intent(Intent.ACTION_VIEW, urmech));
50                         break;
51                     case "RE":
52                         Uri ure = Uri.parse("https://drive.google.com/file/d/1LpgvoMmHZPC1v20zMdqF_Wzr6vnisNPP/view?usp=sharing");
53                         startActivity(new Intent(Intent.ACTION_VIEW, ure));
54                         break;
55                     case "AUTO":
56                         Uri urauto = Uri.parse("https://drive.google.com/file/d/1ufy_0980Ku5-aqknPtgT8dA4p7dhpfdy/view?usp=sharing");
57                         startActivity(new Intent(Intent.ACTION_VIEW, urauto));
58                         break;
59                     case "PETRO":
60                         Uri urpetro = Uri.parse("https://drive.google.com/file/d/1VyJ0ReiCcZlZ6ojLvryRzoQsZLX0oLV/view?usp=sharing");
61                         startActivity(new Intent(Intent.ACTION_VIEW, urpetro));
62                         break;
63                     case "OTH":
64                         Uri uroth = Uri.parse("https://drive.google.com/file/d/10LA12_L3ttIaTsFGNSo4vqfkCKohvNbV/view?usp=sharing");
65                         startActivity(new Intent(Intent.ACTION_VIEW, uroth));
66                         break;
67                 }
68             }
69         });
70     });
71
72     } //}
73 }
74 }
```

**user\_request.java:** this code allows the admin to accept or deny request of registration by the users

```
> Users > Laptop > Documents > unlnex > loginnex > backend > user_request.java > com.example.loginnex
1 package com.example.loginnex;
2
3 import android.os.Bundle;
4 import android.os.Handler;
5 import android.view.View;
6 import android.widget.Button;
7 import android.widget.ImageView;
8 import android.widget.TextView;
9 import android.widget.Toast;
10
11 import androidx.appcompat.app.AppCompatActivity;
12
13 import com.vishnusivadas.advanced_httpurlconnection.PutData;
14
15 public class user_request extends AppCompatActivity {
16
17     private ImageView back;
18     private ImageView accept;
19     private ImageView deny;
20     private Button next;
21     private Button pervious;
22     private TextView name;
23     private TextView ID;
24     private TextView email;
25     private TextView occ;
26
27     int renum = 1;
28     int max = 0;
29
30
31     @Override
32     protected void onCreate(Bundle savedInstanceState) {
33         super.onCreate(savedInstanceState);
34         setContentView(R.layout.requests);
35         back = findViewById(R.id.backbutton);
36         accept = findViewById(R.id.button);
37         deny = findViewById(R.id.button2);
38         next = findViewById(R.id.next);
39         pervious = findViewById(R.id.per);
40         name = findViewById(R.id.stuname);
41         ID = findViewById(R.id.studid);
42         email = findViewById(R.id.stuem);
43         occ = findViewById(R.id.stuocc);
44         Loading_dialog loadingdialog = new Loading_dialog(user_request.this);
45
46
47         Handler handler = new Handler();
48         handler.post(new Runnable() {
49             @Override
```

```

50
51     public void run() {
52         loadingdialog.StartLoadingDialog();
53         String[] field = new String[1];
54         field[0] = "renum";
55         String[] data = new String[1];
56         data[0] = String.valueOf(renum); //equal to 1 to get the first row in table
57         PutData putata = new PutData("http://192.168.1.38/loginRegister/regrows.php", "POST", field, data); //gets number of row in the table
58         if (putata.startPut()) {
59             if (putata.onComplete()) {
60                 String top = putata.getResult();
61                 max = Integer.parseInt(top);
62             }
63         }
64         PutData ptData = new PutData("http://192.168.1.38/loginRegister/reqname.php", "POST", field, data); //gets the name of the user
65         if (ptData.startPut()) {
66             if (ptData.onComplete()) {
67                 String result = ptData.getResult();
68                 if (!result.equals(anObject: "")) {
69                     name.setText(result);
70                 } else {
71                     name.setText("No Request submitted");
72                 }
73             }
74         }
75         PutData putData = new PutData("http://192.168.1.38/loginRegister/reqID.php", "POST", field, data); //gets the ID of the user
76         if (putData.startPut()) {
77             if (putData.onComplete()) {
78                 String result = putData.getResult();
79                 ID.setText(result);
80             }
81         }
82         PutData puData = new PutData("http://192.168.1.38/loginRegister/reqemail.php", "POST", field, data); //gets the email of the user
83         if (puData.startPut()) {
84             if (puData.onComplete()) {
85                 String result = puData.getResult();
86                 email.setText(result);
87             }
88         }
89         PutData utData = new PutData("http://192.168.1.38/loginRegister/reqocc.php", "POST", field, data); //gets the occupation of the user
90         if (utData.startPut()) {
91             if (utData.onComplete()) {
92                 String result = utData.getResult();
93                 occ.setText(result);
94             }
95         }
96         loadingdialog.CancelLoadingDialog();
97     }
98 });
99
100 next.setOnClickListener(new View.OnClickListener() {
101     @Override
102     public void onClick(View v) {
103         loadingdialog.StartLoadingDialog();
104         if (renum < max) {
105             renum++; // adds 1 to renum if it's not the last row to go to next row in the table aka the next user
106         }
107         Handler handler = new Handler(); //like the above it gets info about the user to update the app
108         handler.post(new Runnable() {
109             @Override
110             public void run() {
111                 String[] field = new String[1];
112                 field[0] = "renum";
113                 String[] data = new String[1];
114                 data[0] = String.valueOf(renum);
115                 PutData ptData = new PutData("http://192.168.1.38/loginRegister/reqname.php", "POST", field, data);
116                 if (ptData.startPut()) {
117                     if (ptData.onComplete()) {
118                         String result = ptData.getResult();
119                         if (!result.equals(anObject: "")) {
120                             name.setText(result);
121                         } else {
122                             name.setText("No Request submitted");
123                         }
124                     }
125                 }
126                 PutData putData = new PutData("http://192.168.1.38/loginRegister/reqID.php", "POST", field, data);
127                 if (putData.startPut()) {
128                     if (putData.onComplete()) {
129                         String result = putData.getResult();
130                         ID.setText(result);
131                     }
132                 }
133                 PutData puData = new PutData("http://192.168.1.38/loginRegister/reqemail.php", "POST", field, data);
134                 if (puData.startPut()) {
135                     if (puData.onComplete()) {
136                         String result = puData.getResult();
137                         email.setText(result);
138                     }
139                 }
140             }
141         });
142     }
143 }

```

```

141         PutData utData = new PutData("http://192.168.1.38/loginRegister/reqocc.php", "POST", field, data);
142         if (utData.startPut()) {
143             if (utData.onComplete()) {
144                 String reslt = utData.getResult();
145                 occ.setText(reslt);
146             }
147         }
148     });
149     loadingdialog.CancelLoadingDialog();
150 }
151 });
152 });
153 previous.setOnClickListener(new View.OnClickListener() {
154     @Override
155     public void onClick(View v) {
156         loadingdialog.StartLoadingDialog();
157         if (renum != 1) {
158             renum--;//decreases by 1 if it's not already equal to 1 to go back one user
159         }
160         Handler handler = new Handler(); //like the above it gets info about the user to update the app
161         handler.post(new Runnable() {
162             @Override
163             public void run() {
164                 String[] field = new String[1];
165                 field[0] = "renum";
166                 String[] data = new String[1];
167                 data[0] = String.valueOf(renum);
168                 PutData ptData = new PutData("http://192.168.1.38/loginRegister/reqname.php", "POST", field, data);
169                 if (ptData.startPut()) {
170                     if (ptData.onComplete()) {
171                         String result = ptData.getResult();
172                         if(!result.equals(anObject:""))
173                             name.setText(result);
174                     }
175                 } else {
176                     name.setText("No Request submitted");
177                 }
178             }
179         });
180         PutData putData = new PutData("http://192.168.1.38/loginRegister/reqID.php", "POST", field, data);
181         if (putData.startPut()) {
182             if (putData.onComplete()) {
183                 String result = putData.getResult();
184                 ID.setText(result);
185             }
186         }
187     }
188     PutData puData = new PutData("http://192.168.1.38/loginRegister/reqemail.php", "POST", field, data);
189     if (puData.startPut()) {
190         if (puData.onComplete()) {
191             String result = puData.getResult();
192             email.setText(result);
193         }
194     }
195     PutData utData = new PutData("http://192.168.1.38/loginRegister/reqocc.php", "POST", field, data);
196     if (utData.startPut()) {
197         if (utData.onComplete()) {
198             String result = utData.getResult();
199             occ.setText(result);
200         }
201     }
202   });
203   loadingdialog.CancelloadingDialog();
204 }
205 });
206 });
207 accept.setOnClickListener(new View.OnClickListener() {
208     @Override
209     public void onClick(View v) {
210         loadingdialog.StartLoadingDialog();
211         Handler handler = new Handler();
212         handler.post(new Runnable() {
213             @Override
214             public void run() {
215                 String[] field = new String[1];
216                 field[0] = "renum";
217                 String[] data = new String[1];
218                 data[0] = String.valueOf(renum);
219                 PutData ptData = new PutData("http://192.168.1.38/loginRegister/reqacc.php", "POST", field, data); //updates the user
220                 if (ptData.startPut()) {
221                     if (ptData.onComplete()) {
222                         String result = ptData.getResult();
223                         if(result.equals(anObject:"accepted")){
224                             Toast.makeText(getApplicationContext(), result, Toast.LENGTH_SHORT).show();
225                         }
226                         Toast.makeText(getApplicationContext(), result, Toast.LENGTH_SHORT).show();
227                     }
228                 }
229             }
230         });
231     }
232 });

```



```

311         if (putata.startPut()) {
312             if (putata.onComplete()) {
313                 String top = putata.getResult();
314                 max = Integer.parseInt(top);
315             }
316         }
317         PutData putDta = new PutData("http://192.168.1.38/loginRegister/requp.php", "POST", field, data);//reorganize the table
318         if (putDta.startPut()) {
319             if (putDta.onComplete()) {
320                 String reut = putDta.getResult();
321             }
322         }
323         PutData ptDta = new PutData("http://192.168.1.38/loginRegister/reqname.php", "POST", field, data);//gets a new user info
324         if (ptDta.startPut()) {
325             if (ptDta.onComplete()) {
326                 String reult = ptDta.getResult();
327                 if(!reult.equals(anObject:"")) {
328                     name.setText(reult);
329                 }
330             } else {
331                 name.setText("No Request submited");
332             }
333         }
334         PutData putData = new PutData("http://192.168.1.38/loginRegister/reqID.php", "POST", field, data);
335         if (putData.startPut()) {
336             if (putData.onComplete()) {
337                 String reslt = putData.getResult();
338                 ID.setText(reslt);
339             }
340         }
341         PutData puData = new PutData("http://192.168.1.38/loginRegister/reqemail.php", "POST", field, data);
342         if (puData.startPut()) {
343             if (puData.onComplete()) {
344                 String reslt = puData.getResult();
345                 email.setText(reslt);
346             }
347         }
348         PutData utData = new PutData("http://192.168.1.38/loginRegister/reqocc.php", "POST", field, data);
349         if (utData.startPut()) {
350             if (utData.onComplete()) {
351                 String reslt = utData.getResult();
352                 occ.setText(reslt);
353             }
354         }
355         PutData utData = new PutData("http://192.168.1.38/loginRegister/reqocc.php", "POST", field, data);
356         if (utData.startPut()) {
357             if (utData.onComplete()) {
358                 String reslt = utData.getResult();
359                 occ.setText(reslt);
360             }
361         }
362         back.setOnClickListener(new View.OnClickListener() {
363             @Override
364             public void onClick(View v) {
365                 onBackPressed();
366             }
367         });
368     }
369 }
370
371

```