MIT Buddy App

Project report submitted

to

MANIPAL ACADEMY OF HIGHER EDUCATION

For Partial Fulfillment of the Requirement for the

Award of the Degree

of

Bachelor of Technology

in

Computer and Communication Engineering

By

Laasya Reddy

Reg. No. 210953170

Kkrishna Saxena

Reg. No. 210953172

Under the guidance of

Vibha Prabhu

Assistant Professor - Senior Scale

Department of I & CT

1. Introduction

The MIT Buddy App project stems from the recognition of the challenges faced by students and administrators in traditional paper-based college environments. Our goal is to revolutionize the college experience by providing a digital platform that enhances efficiency, accessibility, and sustainability. By digitizing various academic and extracurricular activities and its array of user-friendly modules, the app seeks to streamline processes, boost efficiency, and foster greater engagement within the campus community.

From attendance tracking to event registration and access to crucial announcements, the app offers students convenient tools to manage their academic and extracurricular commitments effectively. Additionally, features like the GPA calculator and phone directory further enhance the app's utility, providing students with valuable resources at their fingertips. By harnessing the power of technology, the MIT Buddy App not only simplifies daily tasks but also promotes a culture of efficiency and community involvement, ultimately enhancing the overall college experience for students.

2. Literature Survey

Our literature survey explored existing mobile applications and research papers related to digital solutions in academic settings. We found that while some apps offer similar functionalities, they often lack a focus on sustainability and fail to provide a comprehensive solution. By drawing insights from existing literature, we gained valuable knowledge on best practices and potential pitfalls to avoid in our project.

Our literature survey meticulously examined:

Mobile Applications: We delved into a broad range of student-focused mobile apps to understand the functionalities currently offered. This analysis provided valuable insights into existing solutions for tasks such as class scheduling, event registration, and communication. However, a concerning trend emerged – a lack of such apps in most colleges.

Research Papers: Our research extended to academic publications exploring the implementation of digital solutions in academic environments. These papers offered valuable knowledge on best practices for developing user-friendly and efficient platforms. Additionally, they highlighted potential pitfalls to avoid, such as ensuring seamless integration with existing university systems and fostering user adoption.[1]

3. Problem Definition

Students often face challenges in managing their academic and extracurricular commitments efficiently due to scattered information sources and lack of streamlined communication. The MIT Buddy App addresses this issue by providing a unified platform for students to access essential college-related information and services conveniently. The reliance on paper resources in colleges poses several challenges, including environmental degradation, inefficiencies in information dissemination, and cumbersome resource management. Our project seeks to address these issues by developing a digital platform that reduces paper consumption, enhances organization, and improves overall user experience. By integrating features like attendance tracking, class timetable, GPA calculator, event registration, and important notices, the app aims to enhance student experience and foster community engagement within the college ecosystem.

4. Objective

The main objectives of the MIT Buddy App project are as follows:

1.Enhance Student Experience: The primary objective of the MIT Buddy App is to enhance the overall experience of students by providing a comprehensive platform that simplifies access to academic information, extracurricular activities, and essential services.

2. Streamline Academic Management: The app aims to streamline academic management by offering features such as attendance tracking, class timetable access, and GPA calculator, enabling students to effectively plan their schedules and monitor their academic progress.

3. Foster Community Engagement: Through event registration and timely notices, the app seeks to foster community engagement by encouraging participation in various campus events and ensuring students stay informed about important announcements and activities.

4. Promote Campus Health and Well-being: The inclusion of features like blood requirement alerts from the university hospital promotes campus health and well-being by facilitating timely responses to medical emergencies and encouraging students to contribute to the welfare of the community.

5. Facilitate Access to Resources: The MIT Buddy App aims to facilitate easy access to essential resources by providing a comprehensive phone directory that includes contact information for academic departments, emergency services, and other campus-related entities, thereby enhancing connectivity and support for students.

6.Sustainability Integration: The MIT Buddy App incorporates sustainability principles by minimizing paper usage through digitalization of academic and administrative processes. By reducing reliance on traditional printed materials for notices, timetables, and event registrations, the app promotes eco-friendly practices and contributes to the campus's sustainability goals[5].

5. Methodology

Our methodology encompasses several stages, including requirements gathering, design, implementation and testing:

Requirement Analysis:

1. identify the needs and pain points in managing academic and extracurricular activities for students.
2. Define use cases such as attendance tracking, event registration, GPA calculation, and access to important notices.
3. Develop user stories to illustrate how different types of users (students, faculty, administrators) interact with the app and achieve their goals.

Design Phase:

a. UI/UX Design:

- create an intuitive and visually appealing user interface in Android Studio IDE.

- Design layouts that provide easy access to features like attendance tracking, class timetable, event registration, and notices.

- Create navigation flows that allow users to seamlessly move between different sections of the app.

b. Database Design:

- Define the data schema for Firebase Firestore to efficiently store and manage data related to attendance, notices, timetables, GPA, event registrations, blood requirements, and phone directory.

- Establish collection structures for each type of data and define document fields to store relevant information.

- Implement security rules to restrict access to sensitive data and ensure data privacy.

Development:

A. Frontend Development:

- Implement UI components using Java with Android SDK[2] to create a responsive and interactive user interface.

- Develop activities, fragments, and views to present information and facilitate user interaction.

- Incorporate Java components to handle user input and update UI elements dynamically.

B. Backend Development:

- Develop backend logic using Java with Firebase SDK to handle business logic and server-side functionalities.

- Utilize Firebase Cloud Functions for tasks such as data validation, notifications, and background processing.

- Integrate Firebase Authentication to enable user authentication and authorization for accessing app features.

C. Firebase Integration:

- Integrate Firebase Firestore as the backend database to store and retrieve application data.

- Leverage Firestore's real-time capabilities for dynamic updates to keep information current[4].

- Implement security rules to control access to data and ensure data integrity and confidentiality.

Testing:

a. Unit Testing: Conduct unit tests for frontend and backend components to ensure individual modules function correctly.

b. Integration Testing: Perform integration tests to verify the interaction between frontend and backend components, ensuring seamless communication and data synchronization

By following these steps, the MIT Buddy App can be developed to meet the specific needs of MIT students and provide a seamless and efficient platform for managing academic and extracurricular activities.

6. Modules

Basic Landing page implementation[3](fig 6.1.1). Apart from the landing page, These are the modules the app offers:

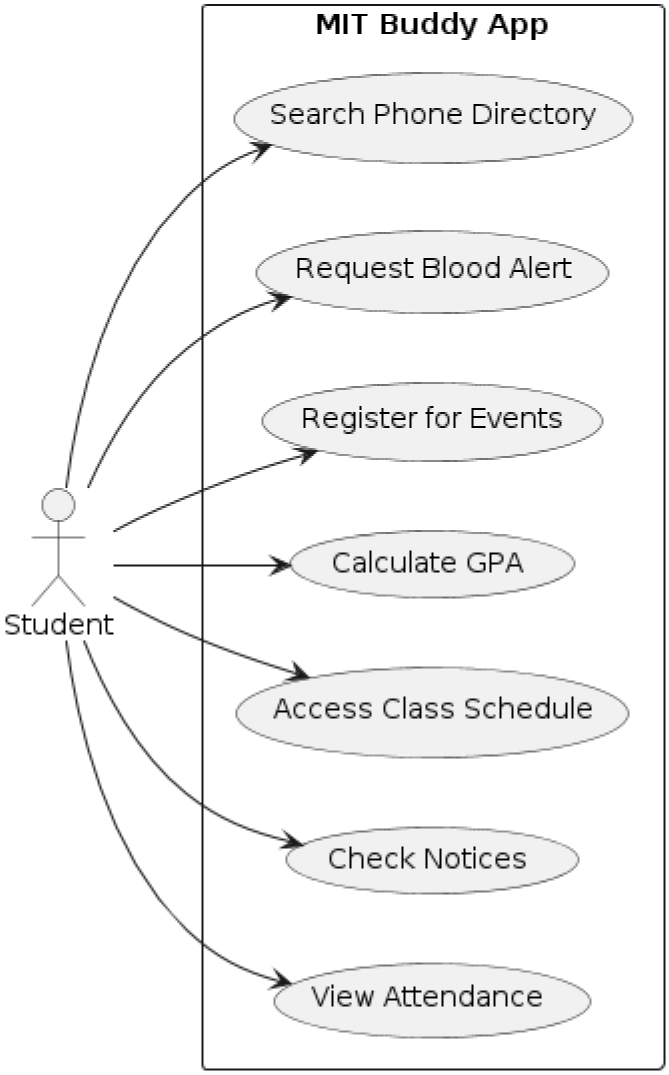


Fig 6.1 Use case diagram for user Student

The student use case diagram depicts interactions with the MIT Buddy app. It enables students to view important notices, track attendance, access class timetables, calculate GPA, register for events, check blood requirements, and access a phone directory. These functionalities empower students to stay informed about institutional updates, manage their academic commitments effectively, plan their schedules with ease, monitor their academic performance, participate in extracurricular activities, contribute to social causes like blood donation, and connect with relevant institutional contacts swiftly. This comprehensive suite of features enhances student engagement, productivity, and overall experience within the academic environment

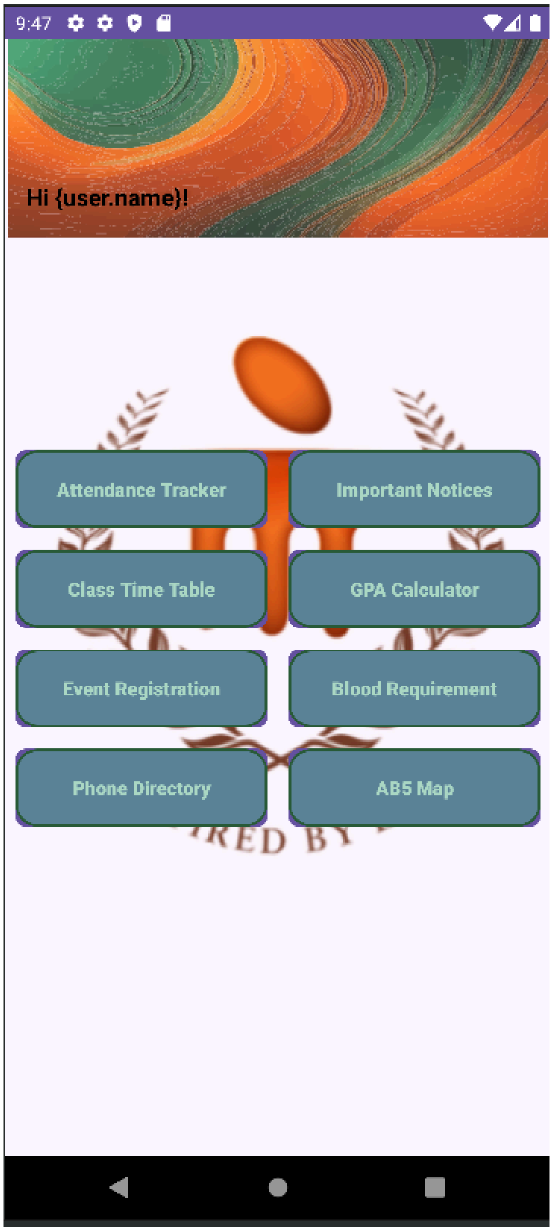
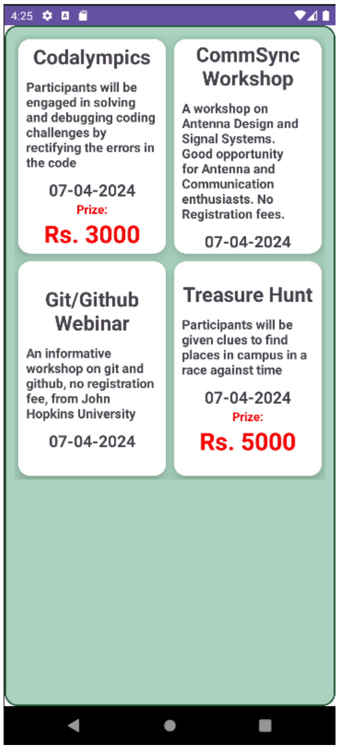


Fig 6.1.1 The homepage

• **Event Registration:** Seamlessly register for various events hosted by clubs, organizations, or the college itself, fostering community engagement and participation. (fig 6.2.1 and fig 6.2.2)



Codalympics

Participants will be engaged in solving and debugging coding challenges by rectifying the errors in the code

07-04-2024

Prize:

Rs. 3000

CommSync Workshop

A workshop on Antenna Design and Signal Systems. Good opportunity for Antenna and Communication enthusiasts. No Registration fees.

07-04-2024

Git/Github Webinar

An informative workshop on git and github, no registration fee, from John Hopkins University

07-04-2024

Treasure Hunt

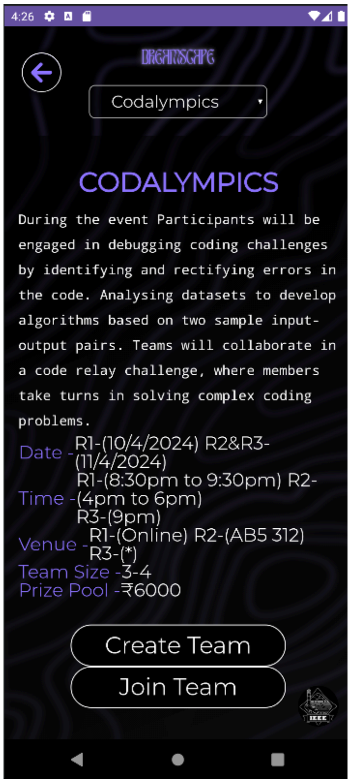
Participants will be given clues to find places in campus in a race against time

07-04-2024

Prize:

Rs. 5000

Fig 6.2.1 The Event Registration module with event cards



Codalympics ▾

CODALYMPICS

During the event Participants will be engaged in debugging coding challenges by identifying and rectifying errors in the code. Analysing datasets to develop algorithms based on two sample input-output pairs. Teams will collaborate in a code relay challenge, where members take turns in solving complex coding problems.

Date - R1-(10/4/2024) R2&R3-(11/4/2024)

Time - R1-(8:30pm to 9:30pm) R2-(4pm to 6pm) R3-(9pm)

Venue - R1-(Online) R2-(AB5 312) R3-(*)

Team Size - 3-4

Prize Pool - ₹6000

Fig 6.2.2 on clicking event cards, it opens registration websites of the event in a Web View

• KMC Blood Requirement Alert: Receive timely alerts about blood requirements at the university hospital, encouraging students to contribute to the well-being of the community. (fig 6.3.1 and fig 6.3.2)

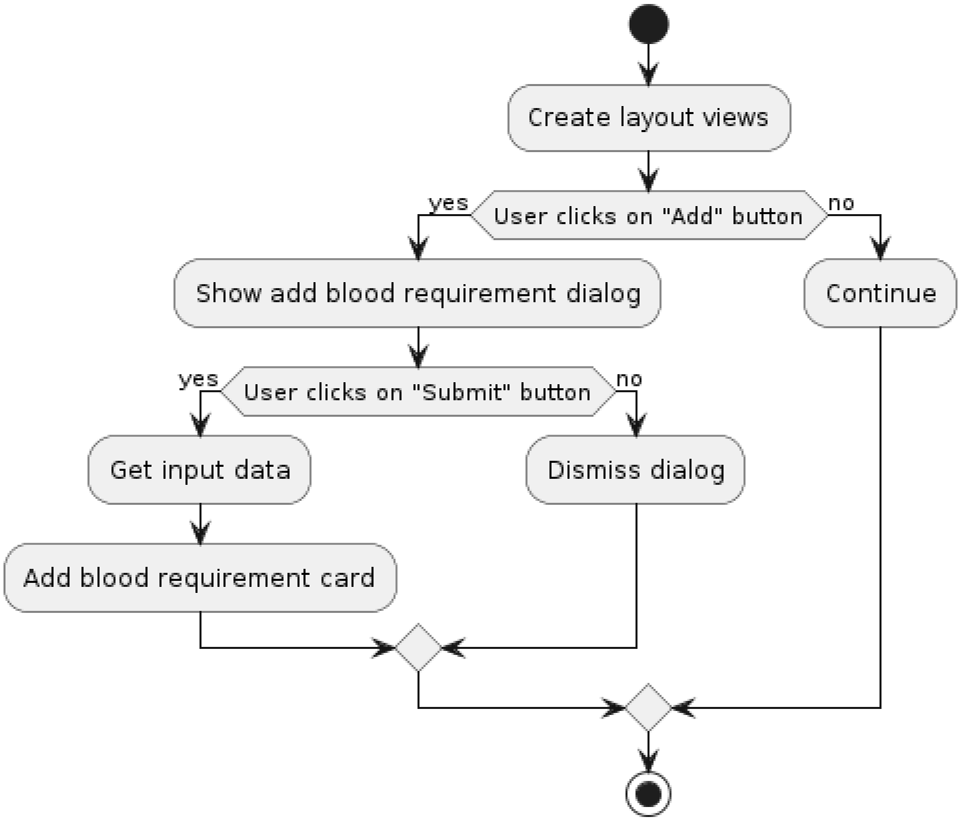
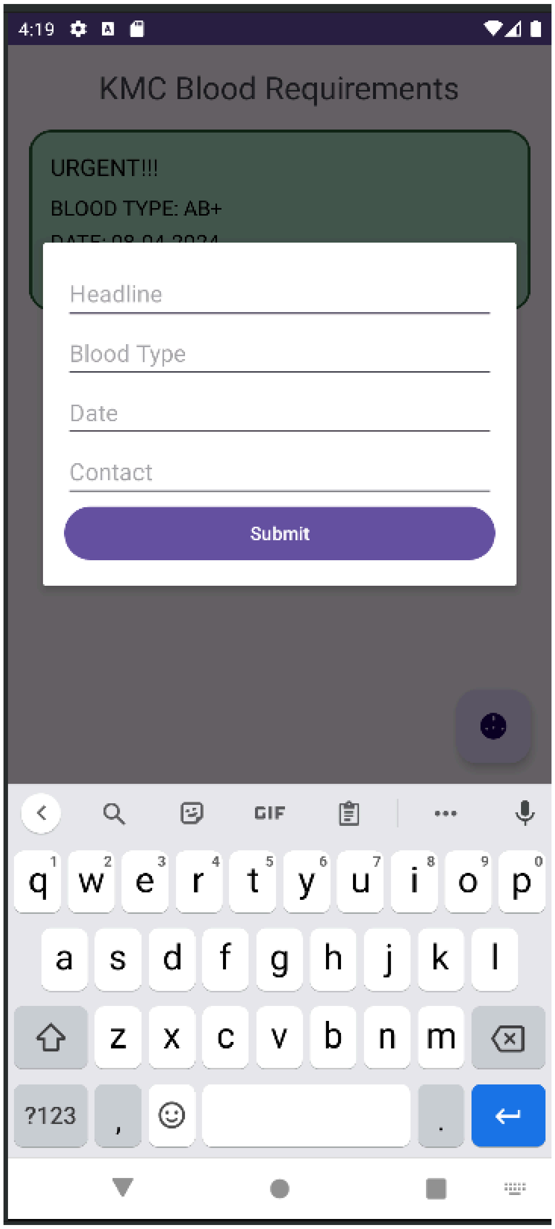

Create layout views
yes
User clicks on "Add" button
no
Show add blood requirement dialog
Continue
yes
User clicks on "Submit" button
no
Get input data
Dismiss dialog
Add blood requirement card

Fig 6.3.1. Activity Diagram of the blood requirement alert module

The Activity Diagram for the Blood Requirement module illustrates the flow of activities within the module. It begins with the user's interaction triggering the display of the main screen, indicating the current blood requirements. Upon user input, the system prompts a dialog for adding new blood requirements. The user then inputs the required information, such as headline, blood type, date, and contact. Once submitted, the system adds this information to the list of blood requirements and updates the display accordingly. This iterative process allows users to continuously add and view blood requirement details within the module.



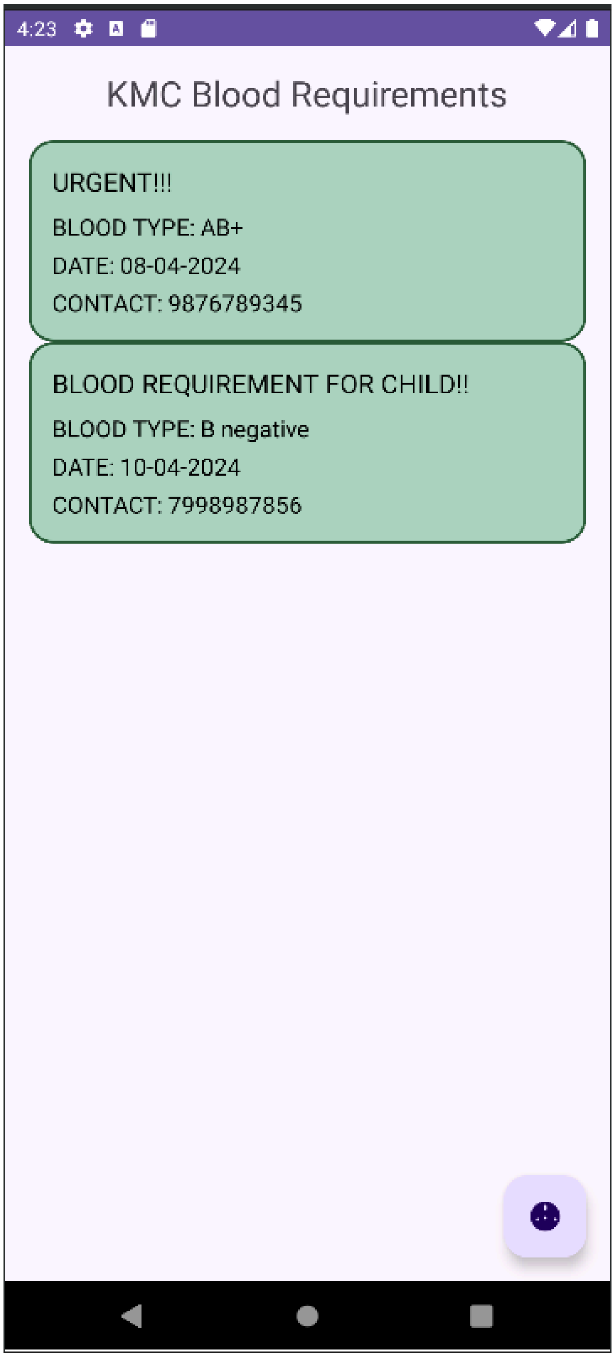
KMC Blood Requirements

URGENT!!!

BLOOD TYPE: AB+

DATE: 08.04.2024

Fig 6.3.1 Blood Requirement Alert : students can add alerts for urgent needs in KMC



KMC Blood Requirements

URGENT!!!

BLOOD TYPE: AB+

DATE: 08-04-2024

CONTACT: 9876789345

BLOOD REQUIREMENT FOR CHILD!!

BLOOD TYPE: B negative

DATE: 10-04-2024

CONTACT: 7998987856

Fig 6.3.2 Alert views

- **Phone Directory:** Access a comprehensive directory that includes contact information for shops, businesses, academic departments, faculties, and emergency services, making it easy to connect with the right resources. (fig 6.4.1, fig 6.4.2 and fig 6.4.3)

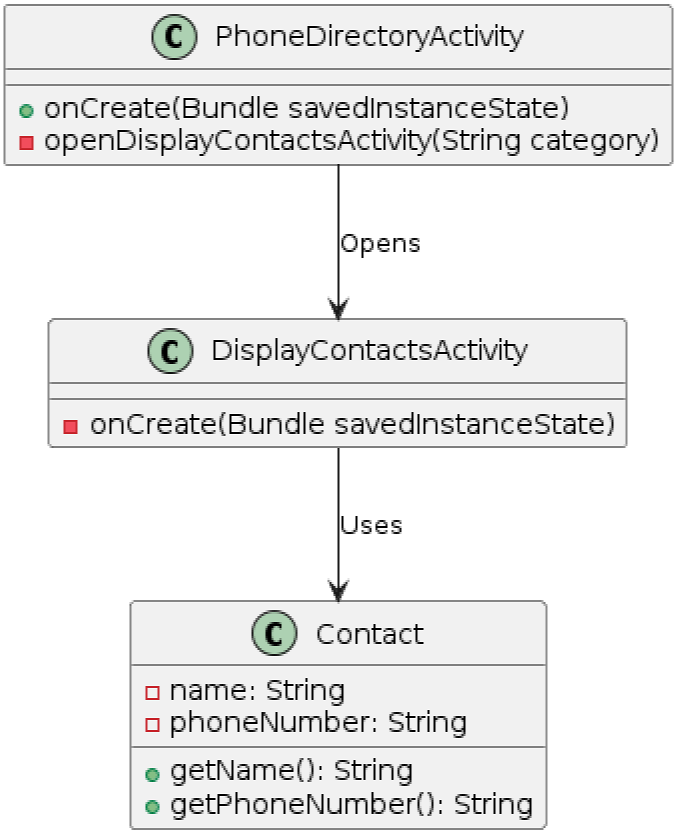

C PhoneDirectoryActivity
+ onCreate(Bundle savedInstanceState)
- openDisplayContactsActivity(String category)
Opens
C DisplayContactsActivity
- onCreate(Bundle savedInstanceState)
Uses
C Contact
- name: String
- phoneNumber: String
+ getName(): String
+ getPhoneNumber(): String

Fig 6.4.1. Class diagram of the phone directory module

The Phone Directory module operates through a hierarchy of classes. At the core is the Contact class, representing individual contacts with attributes for name and number. The PhoneDirectoryActivity class orchestrates the layout and functionality for displaying contact cards, while the DisplayContactsActivity class focuses on presenting contacts based on their category. These classes interact seamlessly, with PhoneDirectoryActivity managing the user interface and interactions, and DisplayContactsActivity handling the display of contacts. This basic structure ensures efficient management and presentation of contact information within the Phone Directory module.

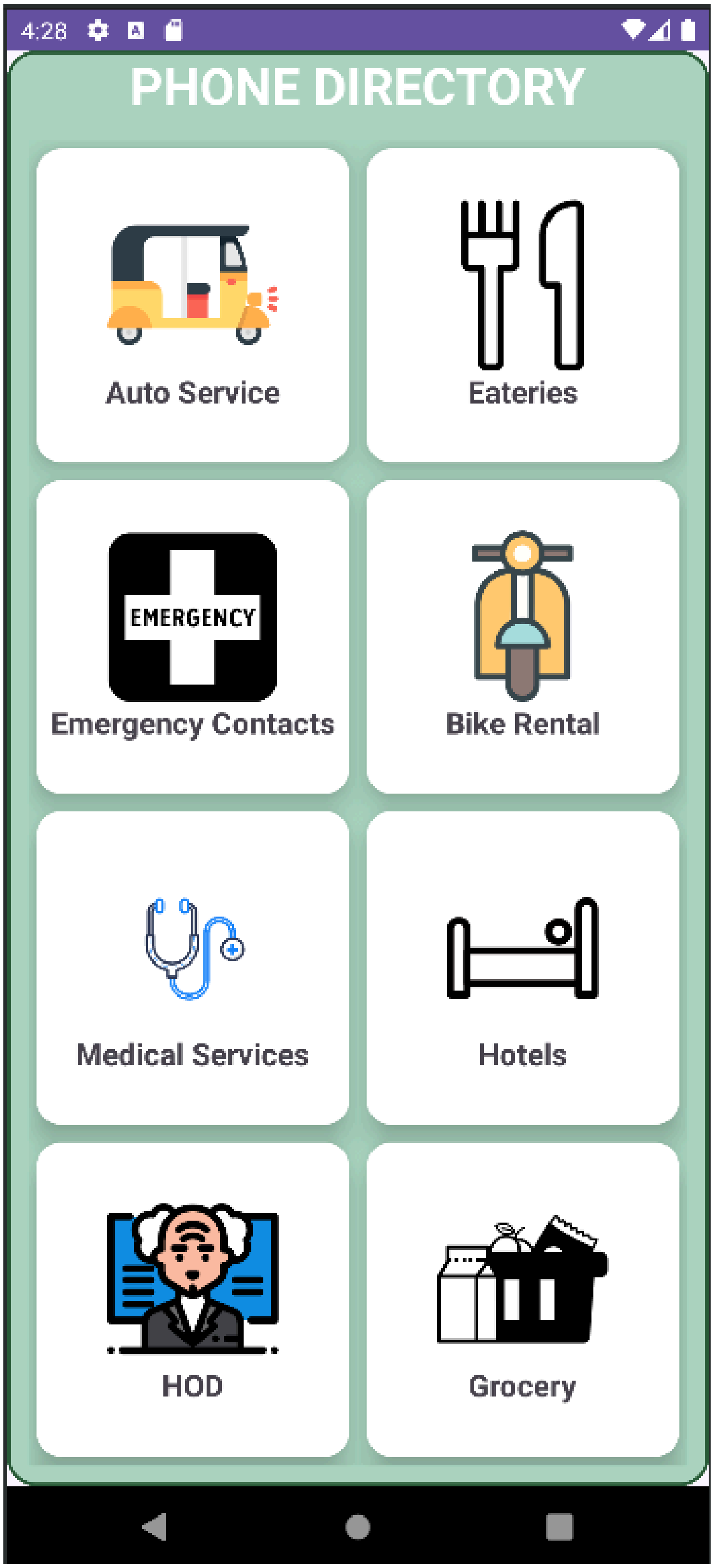
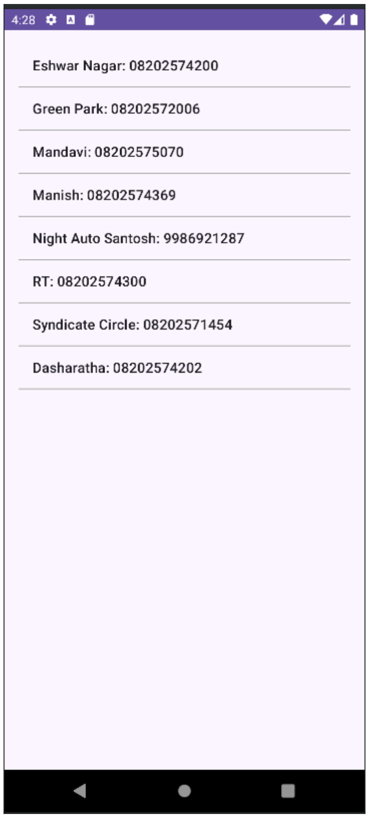


Fig 6.4.2. Each card represents a category of contacts



Eshwar Nagar: 08202574200

Green Park: 08202572006

Mandavi: 08202575070

Manish: 08202574369

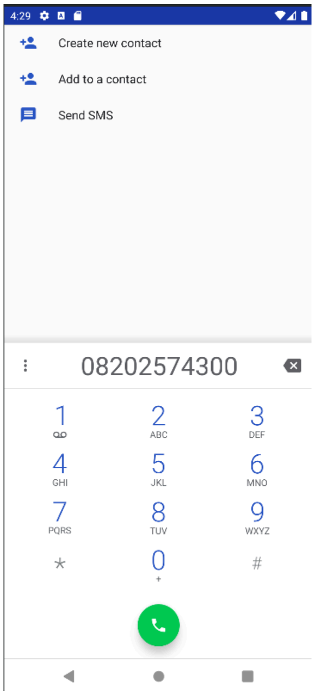
Night Auto Santosh: 9986921287

RT: 08202574300

Syndicate Circle: 08202571454

Dasharatha: 08202574202

Fig 6.4.3 showcases the phone numbers in Auto Services (data is stored in an sqlite database)



08202574300

Fig 6.4.4 clicking the contact in the previous figure opens the call app with the contact.

Few more functionalities:

- **Important Notices:** Stay informed with instant updates on crucial announcements, ensuring you never miss important information from the college administration.

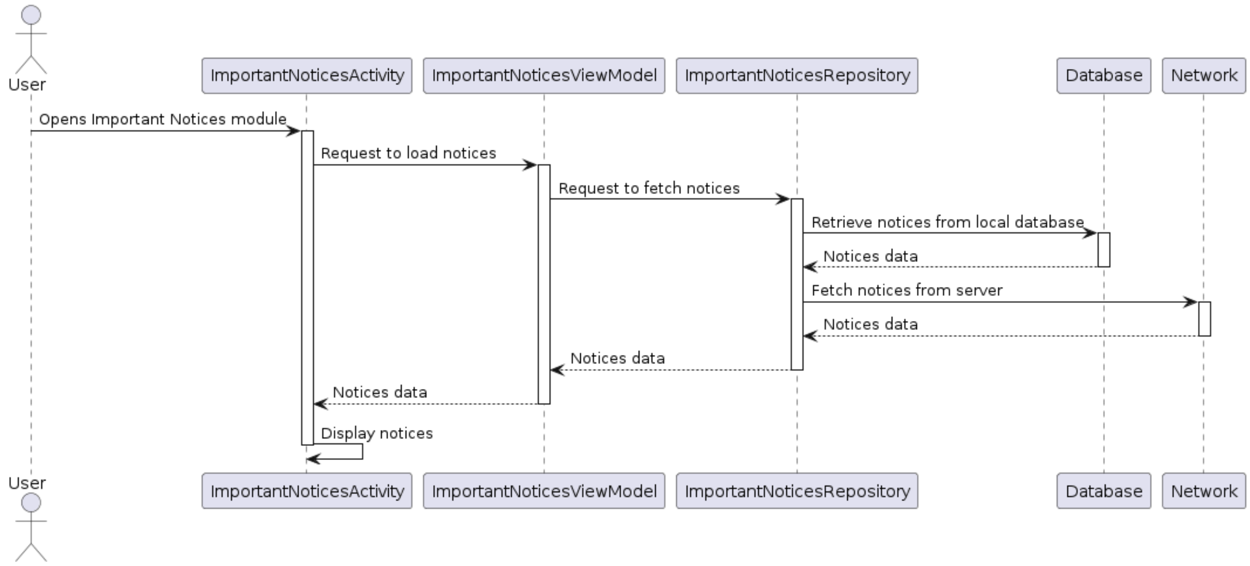


Fig 6.5.1 Sequence diagram demonstrating how the important notice module retrieves information from a repository

The Sequence Diagram for the Important Notices module depicts the interactions between different components or actors involved in the module's functionality. It begins with the user requesting to view important notices. The system retrieves the notices from the database and displays them to the user. Upon selecting a specific notice, the system retrieves detailed information about that notice and presents it to the user. The user can then choose to take further actions, such as saving the notice or dismissing it. Throughout this process, communication occurs between the user interface, database, and user interaction components to facilitate a seamless experience for the user.

- **Attendance Tracking:** Keep track of your attendance easily, helping you stay on top of your academic commitments.

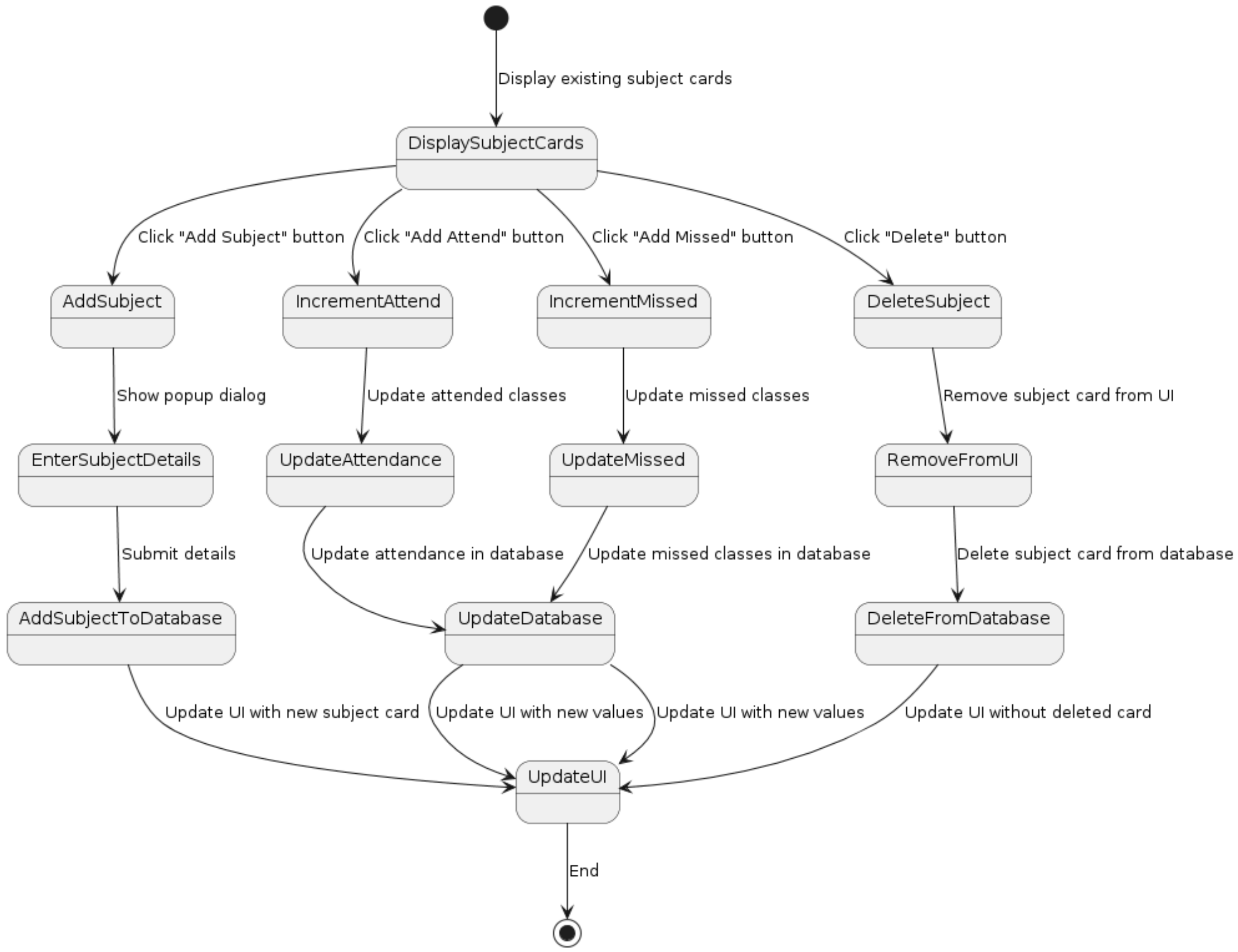

Display existing subject cards
DisplaySubjectCards
Click "Add Subject" button
Click "Add Attend" button
Click "Add Missed" button
Click "Delete" button
AddSubject
IncrementAttend
IncrementMissed
DeleteSubject
Show popup dialog
Update attended classes
Update missed classes
Remove subject card from UI
EnterSubjectDetails
UpdateAttendance
UpdateMissed
RemoveFromUI
Submit details
Update attendance in database
Update missed classes in database
Delete subject card from database
AddSubjectToDatabase
UpdateDatabase
DeleteFromDatabase
Update UI with new subject card
Update UI with new values
Update UI with new values
Update UI without deleted card
UpdateUI
End

Fig 6.6.1 Activity diagram for the attendance tracking module

As shown in the activity diagram (fig 6.6.1) for the Attendance Tracker module outlines the flow of actions within the module. It begins with the user interface displaying existing subject cards retrieved from the database. The user can then choose to add a new subject by clicking the "Add Subject" button, which triggers the display of a popup dialog. Within the dialog, the user enters details such as the subject name, attended classes, missed classes, and total classes required. Upon submission, the system calculates the attendance percentage and updates the database with the new subject card. The user interface is then updated to reflect the addition of the new subject card. Users can also increment attended or missed classes for each subject, which dynamically updates the attendance percentage and database records. Additionally, users have the option to delete subject cards, which removes them from both the user interface and the database. Throughout this process, interactions occur between the user interface components, database, and user input, ensuring seamless functionality and data management within the Attendance Tracker module.

- **Faculty Finder:** Find your teachers at the click of a button. A search functionality to find cabins and profiles of faculty.

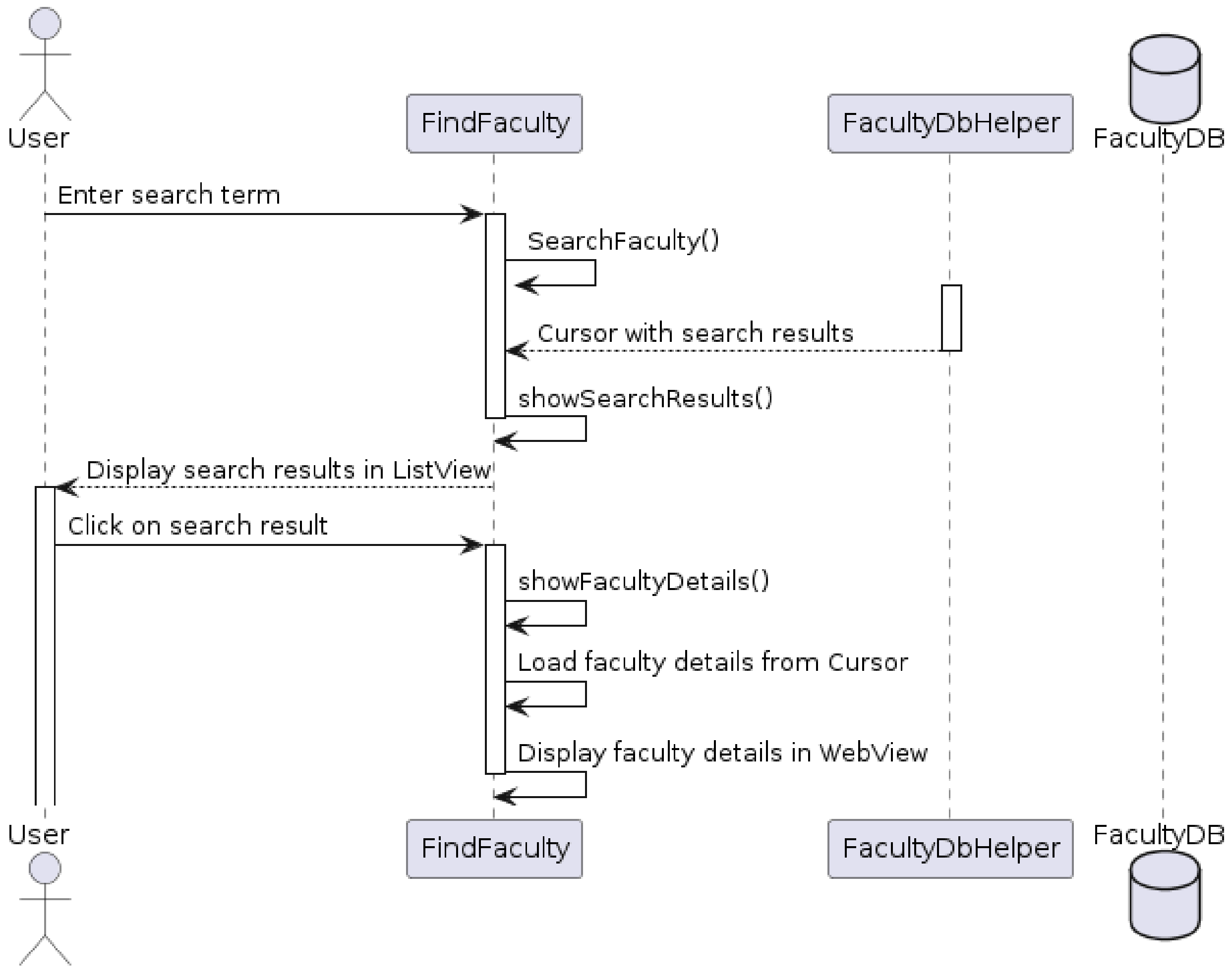


Fig 6.7.1. Sequence diagram for the faculty finder functionality

The sequence diagram fig 6.7.1 illustrates the process of searching for faculty in the `FindFaculty` activity. Initially, the user enters a search term, triggering the `SearchFaculty()` method. This method activates the `FacultyDbHelper`, which queries the database for matching records. Upon retrieving the search results, the `FindFaculty` activity displays them in a ListView through the `showSearchResults()` method. When the user selects a faculty member, `showFacultyDetails()` retrieves the details from the Cursor and displays them in a WebView. This sequence demonstrates the interaction between the user interface, database helper, and data presentation components in searching and displaying faculty information.

- **Class Timetable:** Access your class schedule at your fingertips, making it convenient to plan your day and manage your time effectively.

- **GPA Calculator:** Simplify the process of calculating your GPA, providing you with a quick overview of your academic performance.

7. Scope

The MIT Buddy App project has the potential to benefit various stakeholders, including students, administrators, and the environment[6].

1. **Academic Management:**

- Attendance Tracking: Allow students to monitor their attendance records for each class.

- Class Timetable: Provide students with access to their class schedules, including timings and locations.

- GPA Calculator: Enable students to calculate their GPA based on their grades in different courses.

2. **Communication and Notices:**

- Important Notices: Notify students about important announcements, deadlines, and events from the college administration.

- Event Registration: Allow students to register for various events organized by clubs, organizations, or the college itself.

3. **Campus Health and Well-being:**

- Blood Requirement Alerts: Notify students about urgent blood requirements at the KMC hospital, encouraging voluntary donations and community support.

4. **Information Access:**

- Phone Directory: Provide a comprehensive directory containing contact information for academic departments, faculty, emergency services, and other campus resources.

5. **Expansion and Revenue Generation:**

- Potential for Additional Features: The app's modular design allows for the addition of more student-friendly features in the future.

- Revenue Generation Opportunities: Explore possibilities for generating revenue through paid promotions from brands, clubs, or businesses targeting the student audience.

6. **Scalability and Long-term Viability:**

- Ensure scalability to accommodate future growth and increasing user base.

- Regular updates and maintenance to address user feedback, improve performance, and enhance features.

- Strategic partnerships and collaborations to sustain the app's relevance and impact within the MIT community.

The scope of the MIT Buddy App encompasses a wide range of functionalities aimed at enhancing the college experience for students by providing easy access to academic resources, communication channels, campus services, and potential revenue streams.

8. Conclusion

In conclusion, the MIT Buddy App project represents a significant step towards modernizing college management and promoting sustainability in academic institutions. Through our comprehensive approach to digitalization and user-centric design, we aim to revolutionize the way students interact with their college environment while contributing to environmental conservation efforts.

References

[1] Hodjat Hamidi, Amir Chavoshi, Analysis of the essential factors for the adoption of mobile learning in higher education: A case study of students of the University of Technology, Telematics and Informatics, Volume 35, Issue 4, 2018, Pages 1053-1070, ISSN 0736-5853, https://doi.org/10.1016/j.tele.2017.09.016.
(https://www.sciencedirect.com/science/article/pii/S0736585317302277)

[2] Android Developer Documentation: https://developer.android.com/develop

[3] User Interface (UI) Design for Mobile Apps: https://www.nngroup.com/reports/mobile-website-and-application-usability/

[4] Firebase for Android Developers: https://firebase.google.com/docs/android

[5] Montiel, Ivan & Delgado-Ceballos, Javier & Ortiz-de-Mandojana, Natalia. (2017). Mobile Apps for Sustainability Management Education. Academy of Management Learning & Education. 16. amle.2017.0184. 10.5465/amle.2017.0184.

[6] Rahman, Mahfuzur & Uddin, Md. Rakib & Islam, Md & Asadushjaman, Md & Ahmed, Khandaker & Hossen, Md & Saha, Sonali. (2017). An Android App for University Management System. IOSR Journal of Mobile Computing & Application (IOSR-JMCA). 4. 15-20. 10.9790/0050-04011520.

[7]Lotfi, Chaimaa & Srinivasan, Swetha & Ertz, Myriam & Latrous, Imen. (2021). Web Scraping Techniques and Applications: A Literature Review. 10.52458/978-93-91842-08-6-38.

[8] Mahayuddin, Zainal & Suwadi, Nur & Jenal, Ruzzakiah & Arshad, Hussnain & Adiono, T.. (2018). Implementing smart mobile application to achieve a sustainable campus. International Journal of Supply Chain Management. 7. 154-159.