Table of Contents

[**Android Application** 2](#_Toc26892532)

[First Sprint 3](#_Toc26892533)

[Second Sprint 4](#_Toc26892534)

[**Group Dynamics** 5](#_Toc26892535)

[**UX Challenges** 6](#_Toc26892536)

[**Current System** 7](#_Toc26892537)

[**Marketing & Pricing Strategy** 17](#_Toc26892537)

[**Platform Selection** 18](#_Toc26892538)

[**Things I’ve learned/experienced** 19](#_Toc26892539)

[**References:** 20](#_Toc26892540)

# **Android Application**

As we all know technology increase day by day thus with advances in new technologies, mobile devices have fully grown in quality to become one amongst the foremost helpful and common shopper devices. Currently, Cell phones / mobile devices square measure important a part of fashionable life. Several of the United States of America got to create a decision or send a message at any time from anyplace.

Agile software development is one of the easiest and most effective ways to turn a business idea into a software solution. Agile is a term used to describe software development methods that use continuous planning, learning, development, team collaboration, evolution development, and early delivery. It encourages flexible responses to change.

In the final week I was only fixing bugs, writing my part in the coursework documentation and testing the application for any possible problems. I’ve also demoed the final app to the module convenor and had a final sprint retrospective with the team.

In the final week I was only fixing bugs, writing my part in the coursework documentation and testing the application for any possible problems. I’ve also demoed the final app to the module convenor and had a final sprint retrospective with the team.

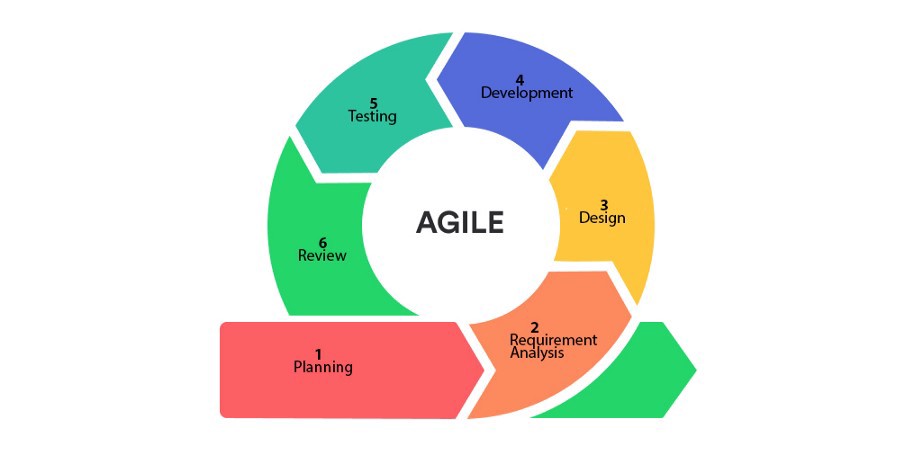


Figure 1

As a team we were looking for a way to develop flexible software such as Agile SCRUM, which is repetitive and growing at the same time because the needs of our project and team capabilities vary.

The SCRUM Agile approach, the actors who contribute to the project are: Product Owner, Scrum Master and Team. The Product Owner in our case is a module facilitator, and I took on the role of Scrum Master and contributed to the team workload.

We agreed to divide the project development into two parts, the first part should be about achieving a (Minimal Viable Product) that contains all the functions except do / listing / selection / rating provided, view photo / video, list. user products, user ratings list, while on the other hand it contains non-existent requirements and has UI and UX improvements such as selecting a category with tabs, adding a second way to add a product etc.

As we can observe from Figure 1, the software life cycle (SDLC) is integrated into each sprint iteration. Sprint is a working area made by a team in a short period of time between one and four weeks. Due to the complexity of the project we agreed to have four weeks of sprints and complete the (Minimal Viable Product) deadline in one sprint and the rest of the work and progress in the second sprint.

## First Sprint

As I said above the first sprint started with the planning phase by creating new tasks and sub-task left behind to determine who and what would contribute to the minimal viable product. This meant that we divided the work into key components of SDLC Agile such as extracting the functional requirements, project management, project implementation and product design and evaluation.

With the requirements analysis we collected and extracted the MVP operating requirements and also consulted the Product Owner (module compiler) about clarifying some of the ambiguity in the work document to be performed.

After that we are fully committed to working on the Use Case Diagram, application map, wireframes and class diagrams for minimal viable product.

## While the team was still working on the design phase I started implementing the project in Android Studio by following the UML classroom drawing schemes. Also, as soon as I completed a major functionality of the project, the team reviewed the component behavior and added bugs to the backlog. Critical bugs had some significant development for the entire project, as it meant application work properly. The little bugs were left when we had some time to rest or postpone fixing them, if we didn’t have time, for a second sprint.

## Second Sprint

The second sprint began with the transfer of uncomplete bugs to a new backlog and the creation of a new program in the Basecamp android application. These included UI and UX enhancements, enhanced existing functionality such as adding security to Firebase collections, and adding the remaining capabilities functions offered, viewing a photo / video, user product list, user ratings list.

Due to time constraints I have started fixing minor bugs that were passed in the first sprint and the team repaired and finalized the operating requirements list, phone frames, application case diagram and class drawing.

After completing the first three phases of the (SDLC) Agile. First phase began to apply the remaining requirements, and the team began working on the curriculum document. By the time I finished work the team had to check it out and add new bugs to the backlog. I soon encountered some serious bugs, leaving small ones to complete the sprint. After completing all the requirements and stabilizing the app I started upgrading the UI and UX and added new features like checking terms and conditions, resetting password password etc.

Last week I was just fixing bugs, writing my part in the study documents and testing the application for any potential problems. I also demoed the last application from the module compiler and got the final sprint retrospective with the team.

# **Group Dynamics**

The Figure 2 shows the Tuckman’s Stages of Group Development applied to our team. “Tuckman (1965) written five stages in which a team goes: forming, storming, norming, performing, and adjourning” (O'Connell & Cuthbertson, 2009, p. 21) .



Figure 2

# 

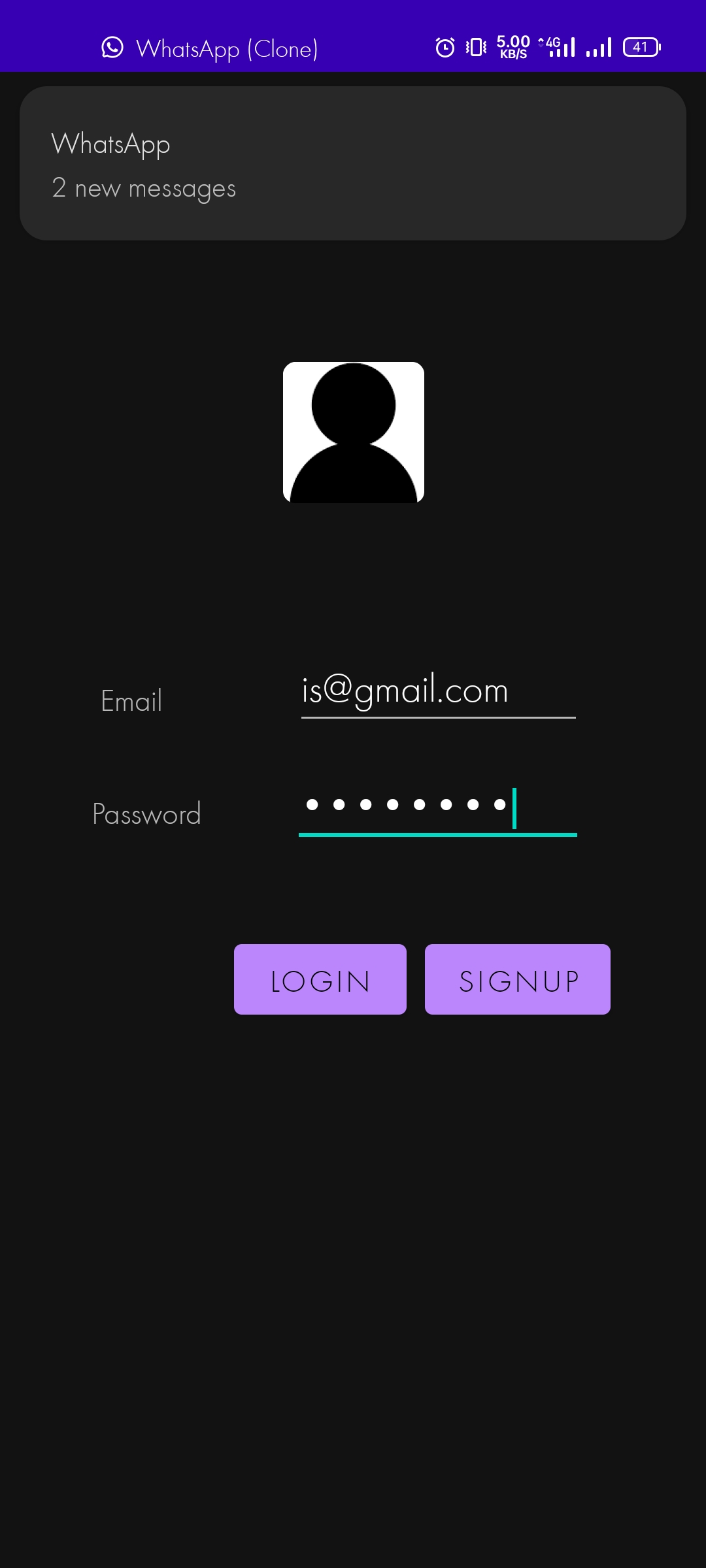
# **UX Challenges**

|  |  |
| --- | --- |
| **Challenge** | **Solutions** |
| Adaptable views for any type of screen | -We use Constraint Layout for better adjustment of our application in any device  -UI elements with adaptable sizes |
| Symmetry | -Using a consistent layout all over the project  -Aligning the UI elements to the factor of dimension rule |
| Keeping content to a minimum | - Divide tasks into several pages to divide information into smaller sizes    - Using Edit-Text. |
| Easy to understand actions | -Using contrasting colours to show opposing actions |
| For Anonymous | -We can use firebase for checking user not repeating the same data |

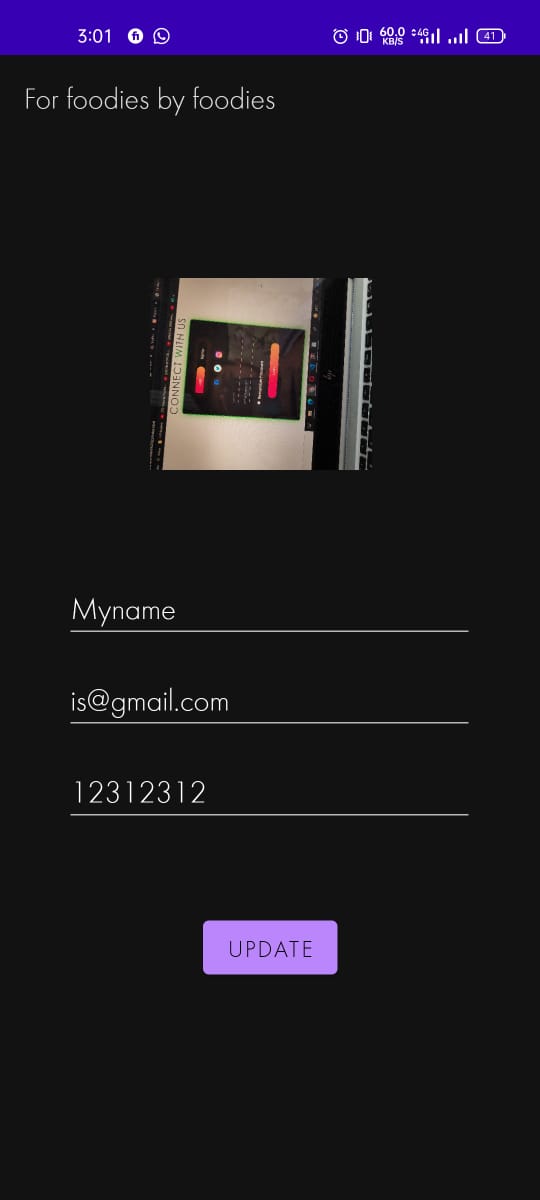
Table 1

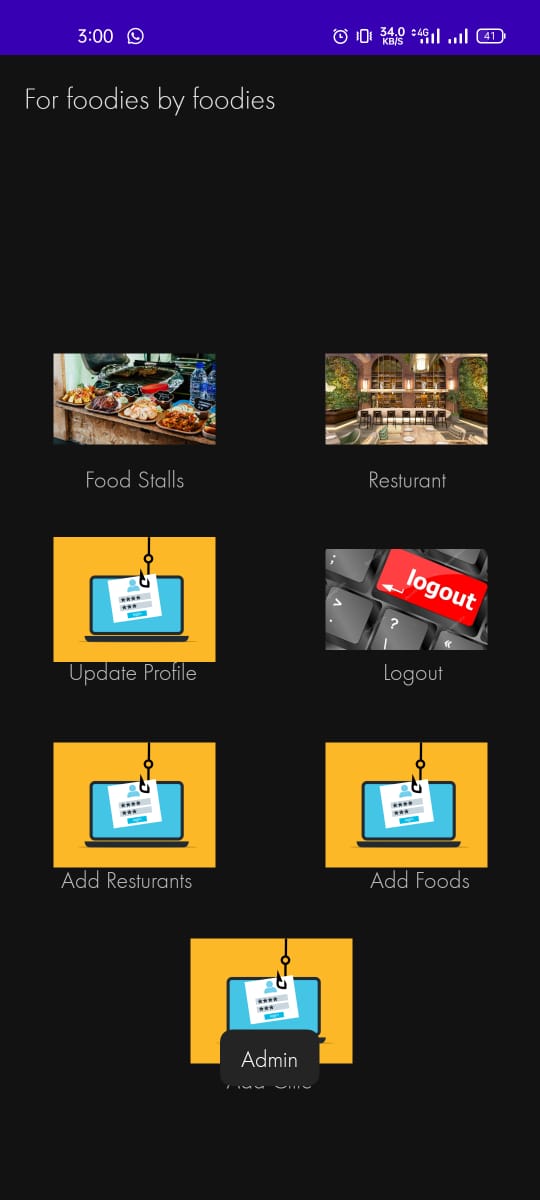
# **Current System Of Admin**

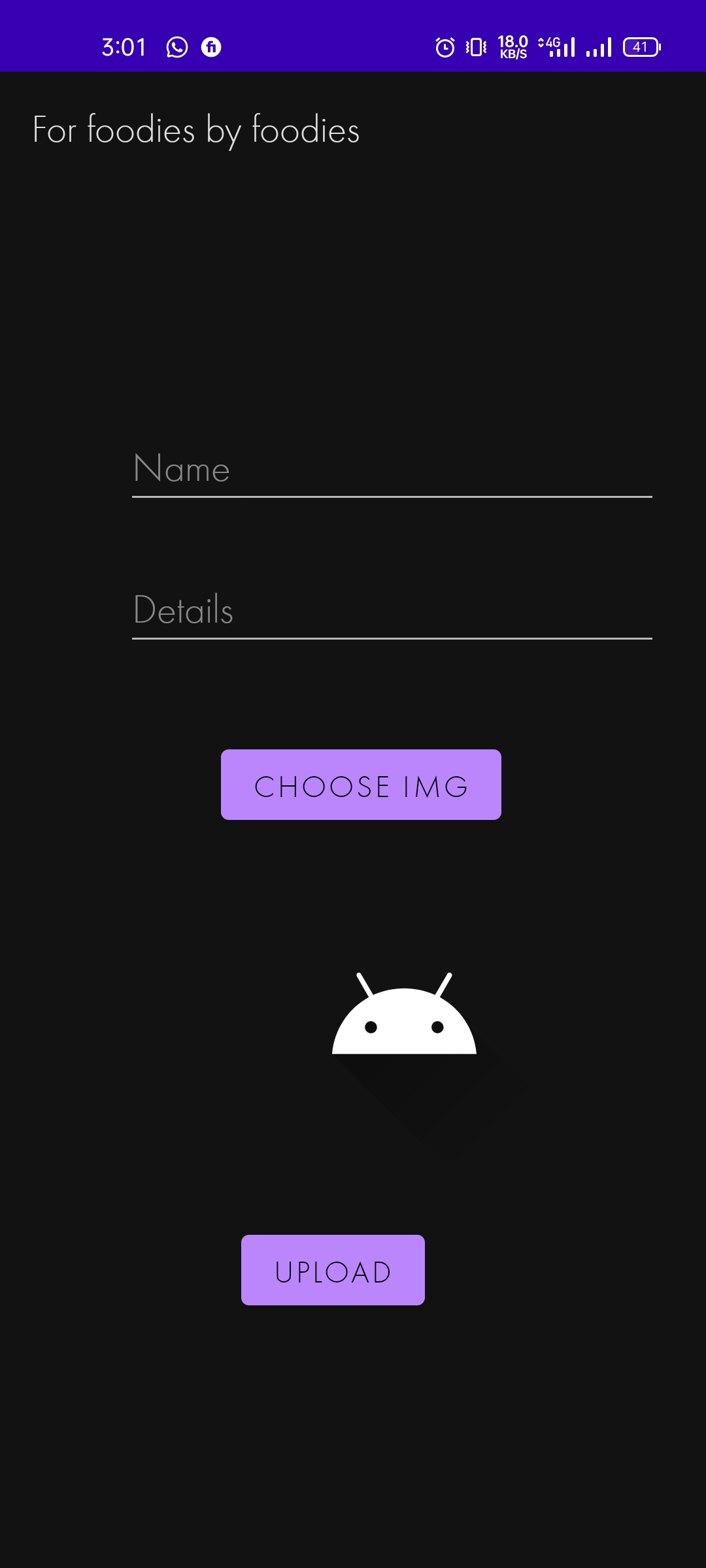
Admin have all type of authority he or she can add any user ,critical user, food , restaurant as well as update profile while on the other hand admin can delete any user ,critical user, food , restaurant. Here I attached the screenshot of admin dashboard.

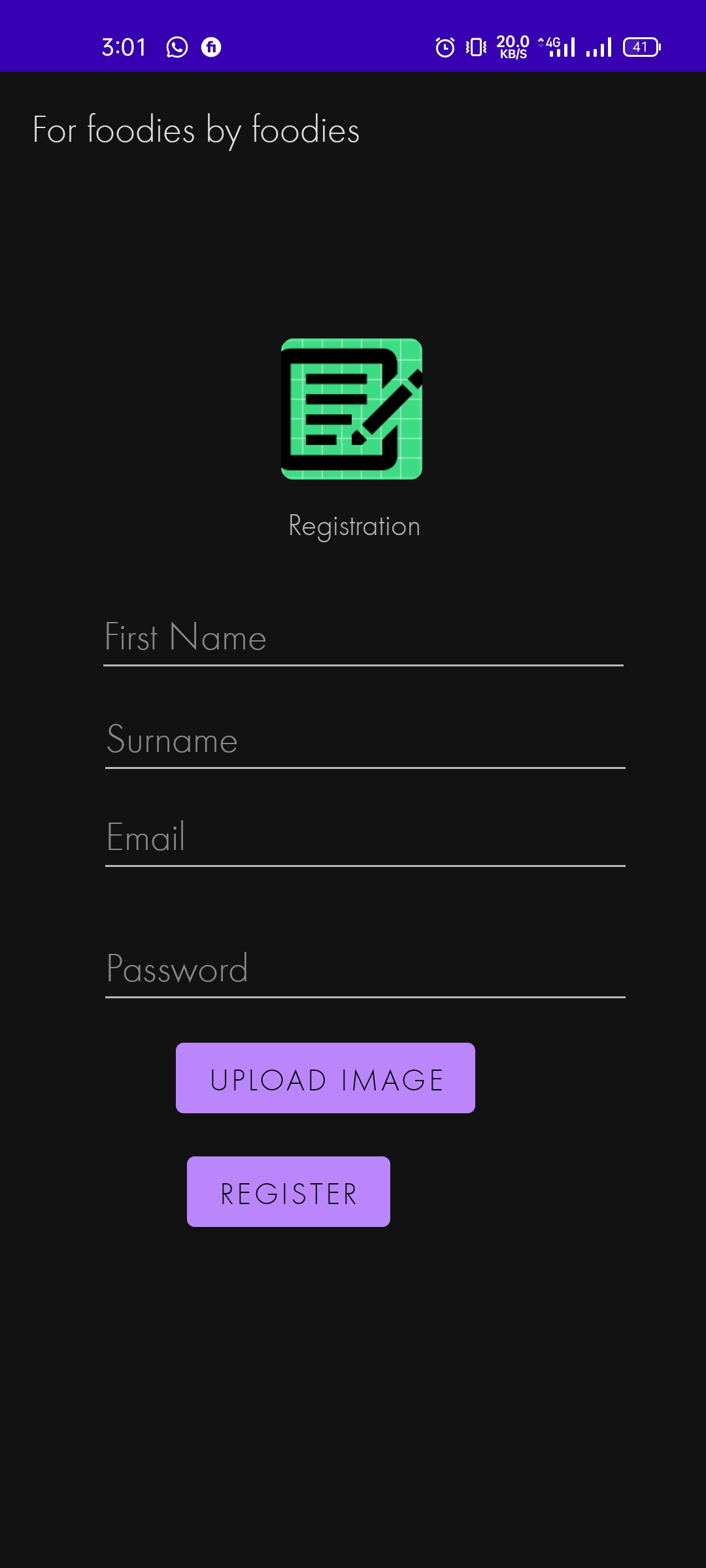






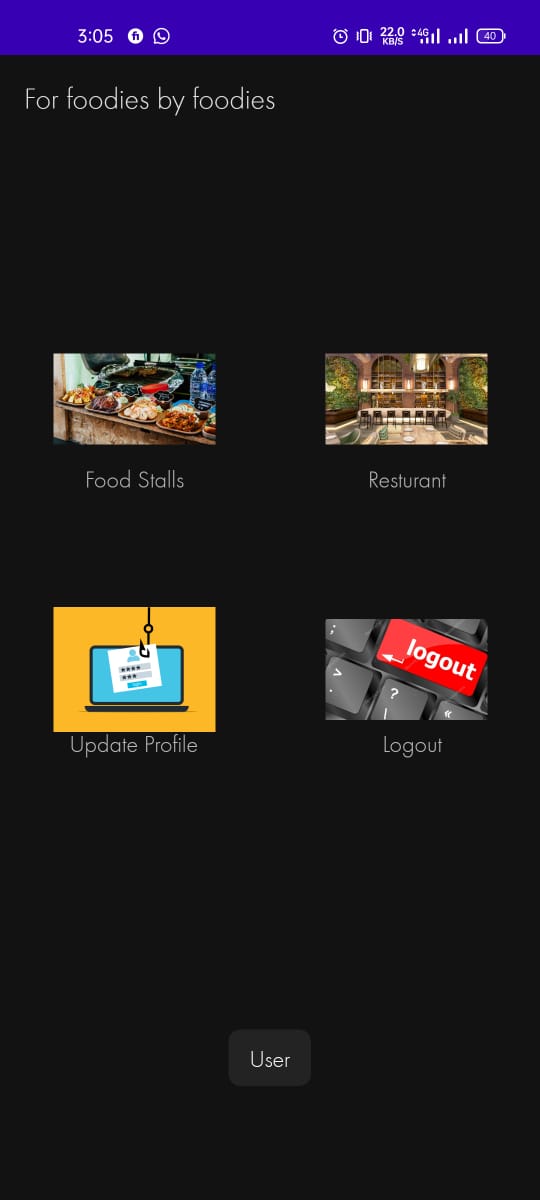






# **Current System Of User**

It is the dashboard for user where user can see food stalls and restaurant. User can reserve the restaurant through this application by going to open-hotel web page. Also he or she can update their profile.







# **Current System Of Critical**

Here is the dashboard of critical user who is just like a admin he has all type of authority except add critical user he can’t add critical user.

# **C:\Users\basit\Downloads\WhatsApp Image 2021-11-25 at 3.20.10 PM (1).jpeg**

# **Marketing & Pricing Strategy**

A great marketing strategy behind our project is that the app allows anyone to view the restaurant. This method will obviously attract customers to sign up, as anonymous users can see what is offered in the app, and if they want, they have the option to register and see restaurants.

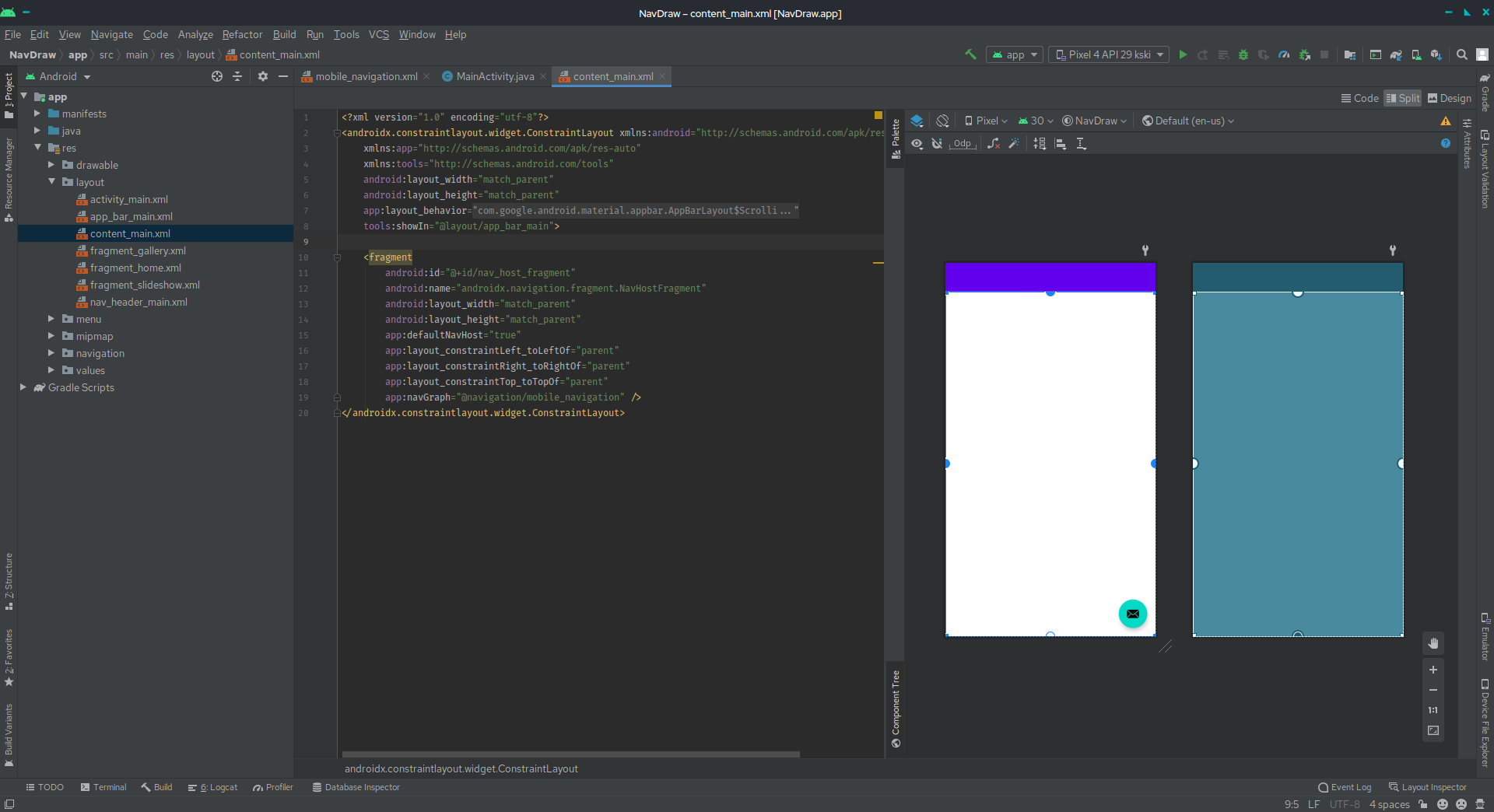
I'm not affiliated with the database, because most customers will contain anonymous and registered users. Technically the ads will be placed after a pre-set number of browsed products in the Main-Activity view, without any necessary change, as the ad will appear as a mere object.

If we reach the goal of having tens of thousands of products that work every day, the cost will increase significantly as a result of our policy of allowing anonymous users to view things. This is because some registered users generate views and anonymous and logged-in users view objects, which means we can have millions of views each day.

To reach this level of readable websites we need to add other revenue streams. Another way is to introduce multiple types of accounts from basic to advanced. Basic accounts will still have ads displayed between products and a limited number of products included, but some types of accounts will pay weekly / monthly / annual packages with different benefits such as having an unlimited number of product inserts or advertising capabilities.

# **Platform Selection**

Android Studio is the official integrated development site (IDE) for Android app development. Based on IntelliJ IDEA, an integrated Java software development platform, and includes its own code editing and developer tools (designing and java). To support system development within the Android app, Android Studio uses a Gradle-based building program, emulator. Every project in Android Studio has one or more methods with source code and application files. These modules include Android app modules, Library modules, and Google App Engine modules. Android Studio uses the Instant Push feature to press code and app changes into the operating system. The code editor assists the developer with coding and provides coding, retrieval and analysis. Applications built into Android Studio and then compiled in APK format for submission to Google Play Store. So, in conclusion I would always choose a native app development solution for the long run, because I’m building an app to last and to easily adapt to the never-ending platform changes.



# **Things I’ve learned/experienced**

Although I have experience in software development I still had to learn all about android from integrating it into the project to using the client side and setting up framework security on the site. This experience made me think that I should start building my own operating system and be independent of my company.

While On the other hand, I have never managed a beginner team before and taking this challenge has made me realize, I have leadership skills. I really enjoyed taking action, directing and forming a team to have a successful outcome.

# **References:**

https://www.javatpoint.com/android-studio

https://www.tutorialspoint.com/android/android\_studio.htm

https://zenkit.com/en/blog/agile-methodology-an-overview/

Available at: https://www.scrumalliance.org/about-scrum/team

[https://www.wrike.com/project-management-guide/faq/what-is-agile-methodology

Available at: https://www.w3schools.in/sdlc-tutorial/agile-model/