ANNAPURNA MANAGEMENT SYSTEM

Project report submitted in partial fulfillment of the requirement for
the degree of Bachelor of Technology
in
Computer Science and Engineering/Information Technology
By

Pankhuri Goel (141426)
Raja Jain (141361)

Under the supervision of
Ms. Ruhi Mahajan
to

Department of Computer Science & Engineering and Information Technology

CERTIFICATE

Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

Candidate's Declaration

I hereby declare that the work presented in this report entitled **"Annapurna Management System"** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from August 2017 to May 2017 under the supervision of **Ms. Ruhi Mahajan** (Assistant Professor, Department of Computer Science & Engineering and Information Technology). The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Pankhuri Goel (141426)　　　　Raja Jain (141361)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Ms. Ruhi Mahajan
Assistant Professor
Computer Science & Engineering and Information Technology
Dated:

ACKNOWLEDGEMENT

We take this opportunity to thank God, almighty for having blessed us with his grace and for taking our effort to a culminating success. We extend our sincere thanks to our esteemed guide, **Ms. Ruhi Mahajan** for providing us with the right guidance and advice at crucial times and for showing us the right path. We express our sincere thanks to our respected head of department, **Prof. Dr. Satya Prakash Ghrera**, for allowing us to use the available facilities. We would also like to thank the other members of the faculty. Last but not least, we would like to thank our friends for the support and encouragement they gave us during the course of our work.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The goal of this application is to automate the mess features of the hostel and provide the user and the administrator with an intelligent platform to interact with each other. To use this feature, the User must register and, with account credentials, can log in to the account and connect to the network. Once connected, the user can use the mess features, such as accessing account information, providing feedback and checking the menu. On the other hand, the administrator can log in and verify the student's account details and also know the number of guests who visit the mess that day. This application aims to provide ease both to users and to the administrator.

INTRODUCTION

1.1 INTRODUCTION

The Annapurna management system helps the user to access all the functionality of the mess without having to physically visit it. Allows the administrator to view inventory and access details.

People who want to use any mess functionality can simply log in to the application and have it all in their hands. Use the Internet to update all user queries and pass them on to the administrator.

Using the information provided by all users, the administrator can make decisions and manage mess inventory. The proposed system will provide a better interaction between the administration and the students / employees.

It will manage all the requirements for easy administration. This system will help to structure activities and maintain data transparency where necessary and properly store data for future analysis.

1.2 PROBLEM STATEMENT

We have two mess in our university, which consist of a junior mess (for first-year students) and a senior mess (for the rest of the students). Both mess in the present are handled manually by the administrator's office. The verification of different data is performed manually.

Therefore, there is a lot of tension in the person who is administering the Annapurna system and in this context computer programs are not usually used. This particular project addresses problems in the administration of Annapurna and avoids the problems that occur when they are carried manually.

Identifying the disadvantages of the existing system leads to the design of a computerized system that will be compatible with the existing system, that is easier to use and more oriented to the GUI. We can improve the efficiency of the system, thus overcoming the drawbacks of the existing system.

1.3 OBJECTIVE

The Annapurna management system helps the user access all the mess features without having to physically visit the mess. Allows the administrator to view inventory and login details.The application is free for users.

People who wish to use any mess functionality can simply log in to the application and have everything at their fingertips. Use the Internet to update all queries of users and put them in the hands of the administrator.

Using the information provided by all users, the administrator can make decisions and manage the inventory for the mess.

1.4 METHODOLOGY

Using Android: the world of Android and the Internet of things has changed the pace at which the world normally worked and now we can access multiple systems with a click or just by touching a screen. The online transaction for banks, purchases on e-commerce sites and the reservation of travel tickets is now possible from the smartphone itself. Maintaining speed is the need for time in this fast-paced environment for a student, so what is better than accessing the entire system of mess on your smartphone?
Use of IoT: IoT is a network of objects connected to the Internet that can collect and exchange data using integrated sensors. It allows objects to be detected or controlled remotely through the existing network infrastructure, creating opportunities for direct integration of the physical world into computer-based systems, resulting in better efficiency, accuracy and economic advantage, as well as as a reduction human intervention. It is expected to offer advanced connectivity of devices, systems and services that go beyond machine-to-machine communications and cover a variety of protocols, domains and applications.

LITERATURE SURVEY

2.1 ANDROID

2.1.1 What is android ?

Android is on a very basic level a mobile phone working system. Nevertheless, now we are composed into PDAs, touch sheets or TVs, even automobiles or netbooks. The working system was made by the start-up of a comparable name, controlled by Google since 2005.

2.1.2 SPECIFICATIONS

This working structure relies upon Linux version 2.6, so it has a strong system focus, which infers that all system limits and controllers are accumulated into one square of code.

• Architecture:

Android contains five levels:

- The Linux 2.6 piece that fuses profitable drivers that allow Wi-Fi or Bluetooth.

- The library written in C and C ++ that gives a bigger sum handiness, for instance, a HTML engine or a database (SQLite).

- A runtime circumstance for applications in perspective of a virtual machine, made for inefficient machines, for instance, telephones. The goal is to make an understanding of JAVA into machine tongue appreciated by Android.

- A JAVA structure that allows the applications that continue running in the virtual machine to mastermind and take an interest.
- User applications written in Java (web program, contact administrator, et cetera.)

• Current Version

Today Android is in its eighth frame, Android 8.0. Each variation is proposed to a tiny bit at a time review the nonappearance of APIs, upgrade the UI and incorporate value. The latest variation incorporates segments like HTML5 reinforce in the program, grants multi-touch or brings another contact API, which portrays a database for regulating contacts.

ANDROID
All About Android – Android Versions with List Names and Features

Figure 1: Android version

2.1.3 Characteristic of market

• Competitors

- The guideline contender is iPhone OS. It is essentially to fight with Apple that Android was made.

- Palm OS devices in PDA.

- Blackberry: which amass has an indistinct name from phones.

- Windows Mobile: which PDA equipment and PDAs.

• Market share

Android's bit of the pie has continued climbing since its initiation and is most likely going to continue ascending since it is bolstered by genuine players, for instance, HTC, Sony Ericsson, Samsung, LG, Motorola, Dell and Acer. The essential remote appeared on the date of Android in October 2008.

2.1.4 Why Android is better ?

• Applications

- Google applications

Android most of the time includes many Google applications, such as Gmail, YouTube or Maps. These applications are provided with the machine most of the time, except in some cases, such as some Android phones in which the provider has replaced Google applications with their own applications.

-Widgets
With Android, you can utilize gadgets that are little devices that you can get all the more regularly data. These gadgets are straightforwardly noticeable in the primary window.

- Android Market

This is an online programming store for buying applications. Designers who have made applications. You can add them to the store and these clients can download these applications, which can be both free and paid.

• Multitasking
Android permits multitasking as in numerous applications can be run at the same time. With Task Manager you can see all the running errands and effortlessly switch between them.

• SDK

An advancement pack has been made accessible to everybody. Therefore, any engineer can make your applications or change the Android stage. This pack contains an arrangement of libraries, intense troubleshooting and improvement devices, a phone emulator, nitty gritty documentation, every now and again made inquiries and instructional exercises.

• Modifiability:
This allows everyone to use, enhance or transform Android features, for example, transforming the interface according to uses, to turn the platform into a real system built into Linux.

2.1.5 Basics of creating Application

Activity :

A movement is a UI that enables the client to communicate with the screen to perform activities. For instance, a content informing application may have a movement that shows a rundown of contacts to send messages. Once the contact is chosen, the movement may send data to a moment action that could be utilized to send the message to the contact.

At the point when an application is propelled, what it indicates is the aftereffect of a movement. At the code level, to make an assignment, you have to make a class that broadens the Activity class. A movement has an onCreate () strategy required. It is the principle technique To associate with the program, through the movement, there must be something that is appeared, that is the reason the action contains what is called sees.

View :

The view is the fundamental part for the UI segments. A view possesses a rectangular territory on the screen. View is the base class for gadgets, which is utilized to make intelligent UI segments (catches, content fields, and so on.). There are diverse sorts of perspectives, for example, a ListView can just show an intelligent rundown of what you need to see, while a WebView enables you to see a site page. As expressed already, a view possesses a rectangular region on the screen. To compose these rectangles on the screen, there is a content record written in XML for each extraordinary screen.

Xml:

Xml means Extensible Markup Language. Android provides a simple XML vocabulary that matches the View classes and subclasses. The goal of using the Android XML vocabulary is to quickly design the user interface projects and screen elements they contain, just like creating HTML Web pages with a series of nested elements.

Intent :

A movement can, obviously, begin another, regardless of whether it isn't, you will require an exceptional question called Intention. An endeavor is the essential depiction of a task that will be performed. You can begin an undertaking, send a broadcastIntent to any influenced segment of BroadcastReceiver and speak with a foundation benefit. An endeavor is to interface the code in various applications. It can be considered as the connection between the exercises. You can add some data to an endeavor, on account of a called objectpackage, which is added to the endeavor because of the technique:

Intent.putExtras(Bundle objetbunble);

Android Manifest :

The AndroidManifest.xml record is required for all Android applications and must have this name in its root registry. In the show you can locate the fundamental data about the application for the Android framework, data that the framework must have before you can run any of the application codes. This is the thing that you can discover in the Android show:

- The name of the Java bundle for the application. The bundle name goes about as an exceptional identifier for the application.

- The depiction of the parts of the application: the exercises, the administrations, the beneficiaries of the communicates and the substance suppliers that make up the application and under what conditions they can be begun.

- The procedures that will have the segments of the application.

- The authorizations the application must need to get to ensured parts of the API and connect with different applications.

- The consents that others must need to communicate with the segments of the application.

- The rundown of Instrument classes that give profiles and other data, for example,

the application is running. These announcements are available in the show just amid the advancement and testing of the application; They are evacuated before the application is distributed.

- The base level of the Android API required by the application.

- The rundown of libraries with which the application must be associated.

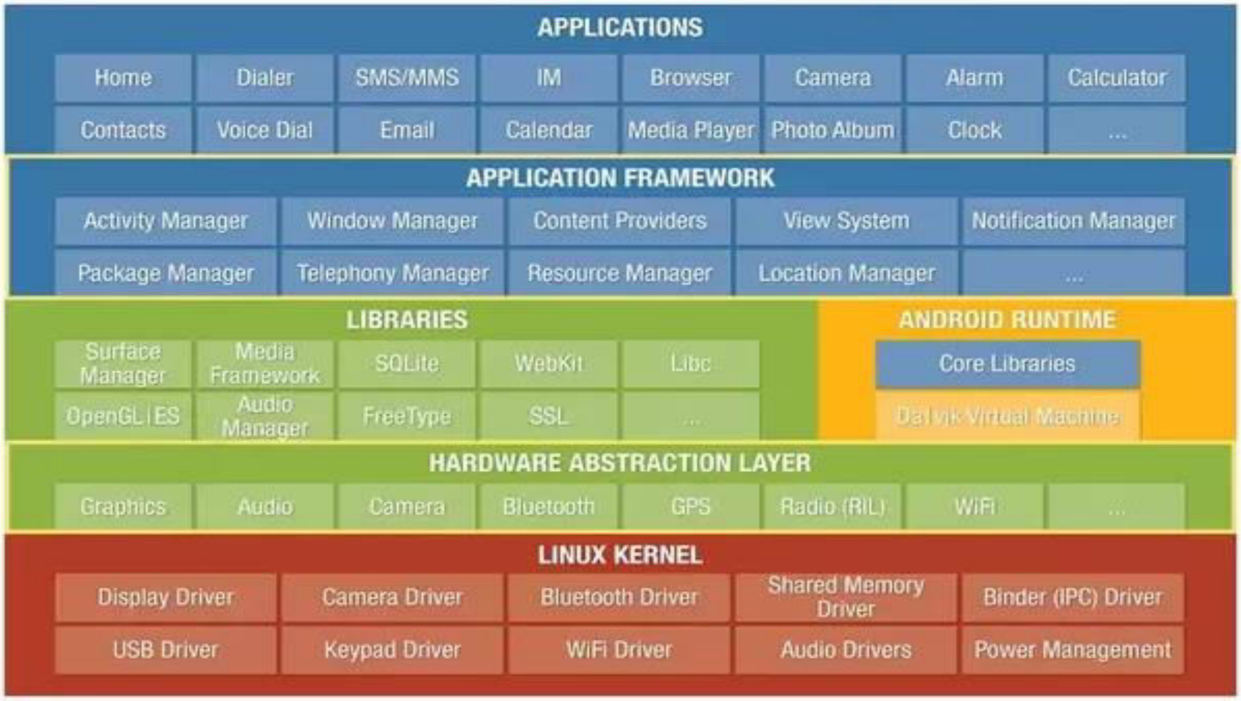


Figure 2: Different layers of Android

2.1.6 Connecting to Database

Using database in Android is different. In fact, there are functions ready to manipulate sqlite, which is the database integrated into Android. To use a database, we will create a class called Helper. This class will allow us to manipulate the database of any other class that instantiated the Helper object. This class has very specific elements and methods. The first specific objects: a SQLite database and a class called openHelper that we will also create.

```
private db SQLiteDatabase;
OpenHelper openHelper = new OpenHelper (this.context);
this.db = openHelper.getWritableDatabase ();
```

The OpenHelper class extends SQLiteOpenHelper. This class is used to create a table or multiple tables in a database and fill the table when it is created, all in the onCreate () method. This class updates the version of the table with the onUpgrade () method. The OnCreate method will be called only once, after the table has been created, this method will no longer be published. For the Helper class, you can add all the methods used to select, add, modify, or delete entries in the table.

2.2 Internet of Things (IoT)

2.2.1 Background

Our homes are progressively the space of savvy coolers, self-ruling robots like Roombas and 3D printers. As of late, we began looking forward with driverless autos and individual utilization rambles. The "modern Internet" may have existed previously, yet now new companies like FitBit and Withings are taking the old referred to gadgets, for example, meters and scales, and influencing them to shrewd by including microcontrollers and Internet associations. Sensors and systems are not any more the area of fly motors or trains; the thought started to saturate our every day lives. Each new intriguing item appears to have a system association, regardless of whether Wi-Fi, Bluetooth, Z-Wave, ZigBee or even a fundamental USB association with a PC. Everything has a sensor, and separate gadgets like a cell phone and a wake up timer are loaded with them. The convergence of these patterns: the blend of equipment, programming, systems, information and knowledge is considerably more critical than any current advancement in the realm of innovation and data innovation. This is the Internet of things. IoT formally assumed control Big Data as the most promoted new innovation.

Plans of action have changed to give benefits as opposed to items. New theories can be made to enable the purchaser to have essential equipment and availability: if the Internet is accessible on a gadget, it promptly turns into an entry associated with a considerably bigger world with a relatively boundless arrangement of potential outcomes. The APIs (Application Programming Interface) make the modules accessible with the goal that everybody can without much of a stretch exploit them. Open source tasks can make the code and learning accessible to anybody. A blast of cloud benefits straightforwardly from Microsoft Azure for Enterprise to Amazon Web Services can significantly decrease the capital expected to begin another product organization. These same comparable improvements now, it might be said, are achieving the physical world. For instance, GitHub enables simple access to the experience of open source ventures facilitated and accessible online in the product area. This time goes into the physical space with gadgets like Arduino. The APIs enable engineers to make programming on prior stages, for example, the Facebook API. In the equipment world, the IFTTT benefit ("If This, Then That") expands this type of development and cooperation to our physical gadgets.

Tom Tunguz, an investor on redpoint, maybe best communicates it when he expresses: "The genuine guarantee of the Internet of Things isn't just to join a huge number of gadgets, similarly as the genuine advancement of the web was not the association of numerous PCs on the system. The genuine but then hidden capability of the IoT is the change of plans of action, enabling organizations to offer items in ways that are totally new and better, profiting both the organization and the client. "

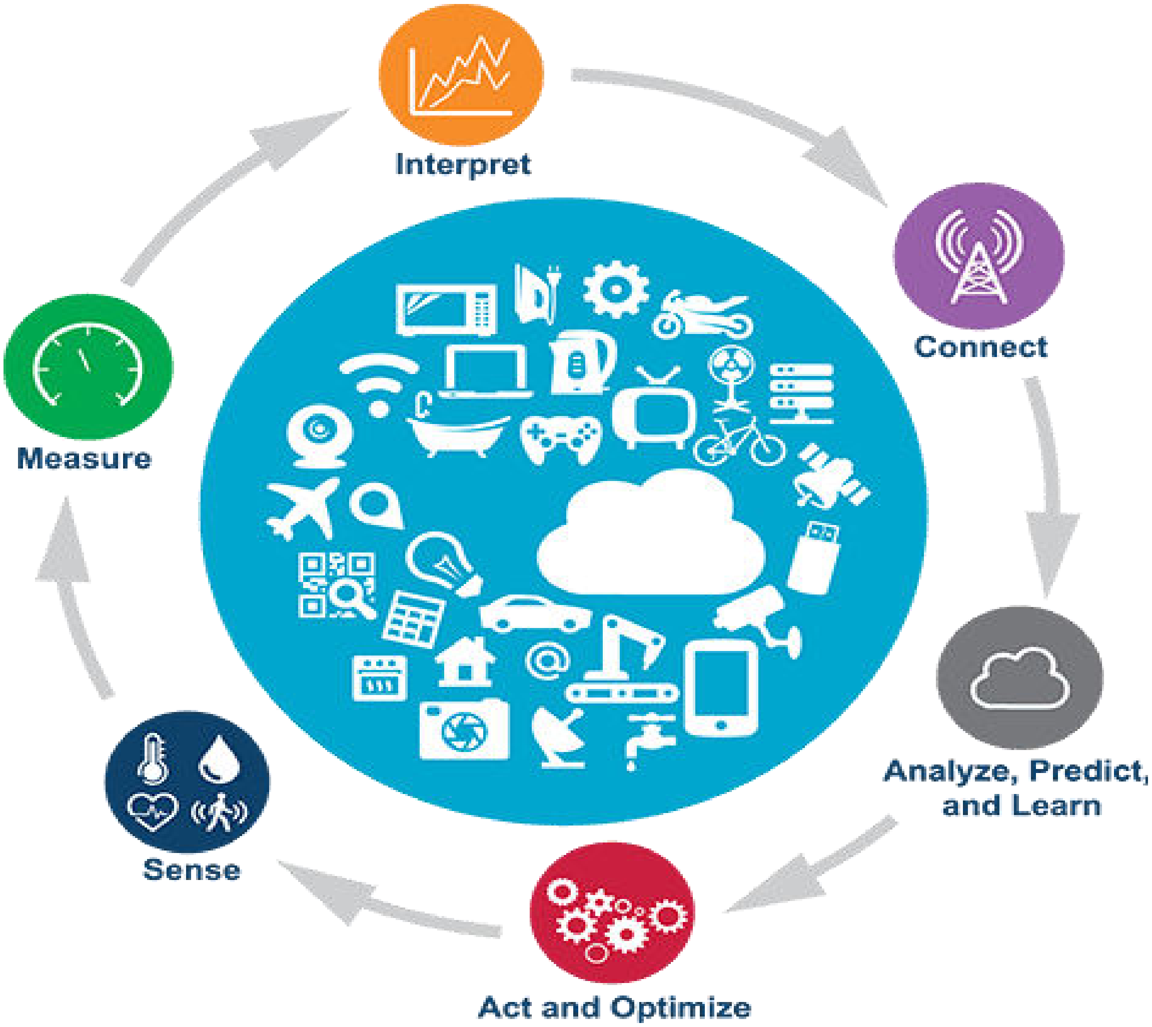


Fig 3: IoT Technology

2.2.2 Role of Software in Innovation and Solutions to Problems

Equipment that has no relationship with the product is presently starting to converge with it. Consider a substantial mechanical organization that produces candy machines. Your machines are transported like huge amounts of metal and plastic. These are machines that are on the motivation and constitute the fundamental foundation of our lives. Its physical plan, its equipment, have been consummated for a long time and, in numerous angles, improved ideally. A large number of these gadgets have achieved an immersion level in their fundamental plan and rationality and have not been altered for quite a long time. The cost of new upgrades in materials and physical outline isn't just restrictive, yet it is viewed as pointless since it "does the activity". Presently, with the IoT, knowledge in routine code in programming and on the Internet is simply beginning to touch these machines. Remember that these developments can be connected retroactively to existing frameworks with minor advancement and financial changes to be coordinated with existing microcontrollers on board. This was the point of convergence of my exploration in this undertaking, and offers a radical new arrangement of enhancements, and in addition a more affordable way to deal with existing highlights. What does this mean for advancement? Considered as basic bits of metal and LEDs/lights, it is hard to see huge upgrades in gadgets, for example, road lights. In any case, consider that the light has been associated in a system. Perhaps it could fill in as a hotspot for the city's Wi-Fi foundation, it could show warnings about activity conditions, delays in broad daylight transport, send climate refreshes, enable clients to explore with the reference points or what you need. You could likewise call the police on the off chance that you distinguish sounds that compare to an auto collision or a human weep for help, strengthened by machine learning calculations. It's difficult to envision junk jars as something that can be progressed. In any case, consider a trash canister: you could caution the city lobby if the rubbish was lost amid the evacuation. On the off chance that it is void, you can advise the pickup truck and it isn't important to require investment to check the compartment. This demonstrates it isn't important to change the fundamental equipment, nor does the item class must be a totally new cross breed gadget (like shopper rambles) so that the IoT favorable circumstances come to the "dumb" gadgets each day.These thoughts require the show of a candy machine, a refuse can or a lamppost as a hub in a product framework. Thus, organizations that influence them to need to consider themselves not as organizations that model steel and model plastic, however as organizations associated with programming. This hierarchical change is going on, and the organizations that don't comprehend it will be deserted, most importantly, the change has been perceived by the automaker Ford. Gotten some information about his remarks on a turn off known as a product organization, Ford's CEO Mark Fields stated, "As we push ahead, I need to be known as an assembling, innovation and assembling organization, on the grounds that our vehicles turn out to be a piece of Internet of Things and customers share their information with us, we need to have the capacity to utilize this information to enable them to enhance their lives and furthermore make some plans of action that will enable us to get an execution, that is the place we are going, our First of all, the approach is to stop ourselves, that is the reason you see us making things like Ford Smart Mobility, LLC, making FordPass, truly supposing uniquely in contrast to the end shopper viewpoint, considering encounters, and after that about how innovation, equipment and programming offer this, unexpectedly, who knows what Apple will do? Our working speculation ought to be: they are doing china, will have an incredible UI, will associate flawlessly to the cloud. This spurs us much more as an organization, to make certain to lead the organization in a considerable lot of these distinctive territories. "Similarly as Apple comprehended and promoted the change that was going to the telephones and gadgets industry, and was in ready to achieve the energy of Nokia and BlackBerry proprietors, the individuals who comprehend and acknowledge this change can be more imaginative, more inventive and more compelling than a considerable lot of the organizations that, today, we expectedly call "innovation organizations".

2.2.3 How Sensors, Data and Communication Intersect Problem Solving to Create Value

Basically, the Internet of Things hoists programming over the level of a gadget. As it were, one of the essential standards of Web 2.0 is taking the material world. Programming is nearer than any time in recent memory to equipment, yet it isn't associated with particular equipment. It isn't astonishing that an indoor regulator, a cooler, an auto, a watch or a couple of shoes is "shrewd" (as in it has a microcontroller that capacities as an ongoing control and input system). . Genuine knowledge and esteem originate from the crossing point of every single associated gadget and those of various sorts of programming made to comprehend crude information. To abuse knowledge, there are two approaches to associate with information. One, the way I picked in my task, is to utilize open gauges, broadly accessible programming dialects and strategies for correspondence autonomous of the gadget and the stage. The information is imparted in a standard arrangement, accessible to any product designer to be utilized. The other mainstream elective practiced by equipment organizations is to make items with express APIs. Similarly as telephone creators have made working frameworks and open APIs to enable engineers to discover better approaches to utilize them and fabricate biological communities around their items, giving them consequently esteem and thinking to possess those gadgets, equipment makers are opening APIs in wise gadgets too. Google Maps gives an intriguing case of when a similar programming is running and uses distinctive gadgets. In spite of the fact that Google Maps appears to put a basic guide on the screen of your cell phone and track its situation on the screen, the administration gets to a great degree exact constant activity information; these don't originate from air and satellite connections; originate from a huge number of Android telephones, every one of which has area benefits that illuminate Google's servers. On the off chance that the telephones are moving, movement is streaming. On the off chance that the telephones on a specific course are moving especially gradually, it means that blockage out and about. The telephone indicates activity conditions, as well as signs movement conditions, as a major aspect of a significantly bigger framework. In the event that the setting is GPS and maps, home lighting, wellness tracker or other, they all offer the Common denominator of programming frameworks that can possibly chip away at numerous gadgets. Savvy lights are an intriguing case, since they can exist alone, yet an ever increasing number of lights turn out to be substantially more effective in a framework with other lights or gadgets. The individual lights can be controlled by a program that additionally peruses the movement locators scattered all through the house, killing the lights on and keeping in mind that moving around the house.

Numerous different frameworks like this have the likelihood of existing. Autos, microwaves, ventilation systems and different gadgets that devour a great deal of vitality can speak with a shrewd network and upgrade the utilization of vitality. You can program the inclination for execution just when power is less expensive. When you have this capacity, it is consistent that the utility furnishes you with marked down rates to move your vitality utilization into hours of less action. Why dry garments amid the day when power is rare? Controlling the manual activity of a gadget to make utilization of off-crest hours is, best case scenario, difficult and likely illogical. Notwithstanding, sending charges over the Internet to a shrewd hair dryer (or programming to be totally free) can be comprehended by partaking in a system of gadgets that works over the level of any hub in that system.

SYSTEM DEVELOPMENT

3.1 Analysis

Use Case Diagram:

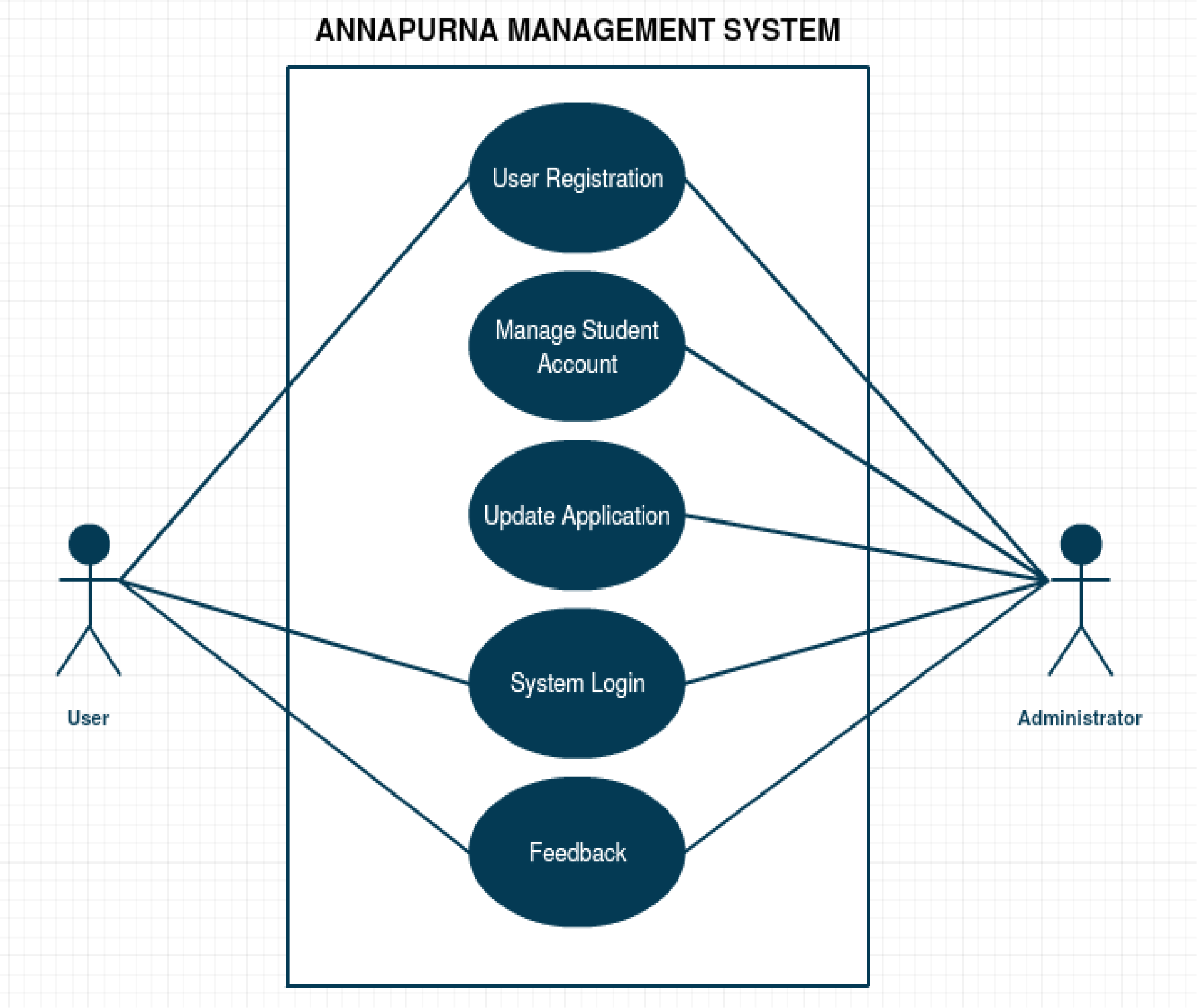


Figure 4: Use case diagram

Activity Diagram of Application:

- For User

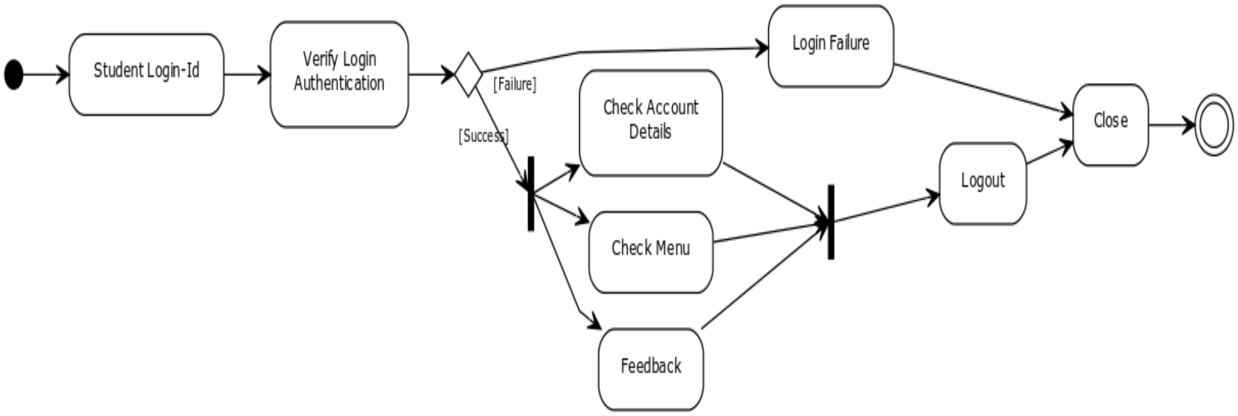


Figure 5: Activity diagram for user

- For Admin

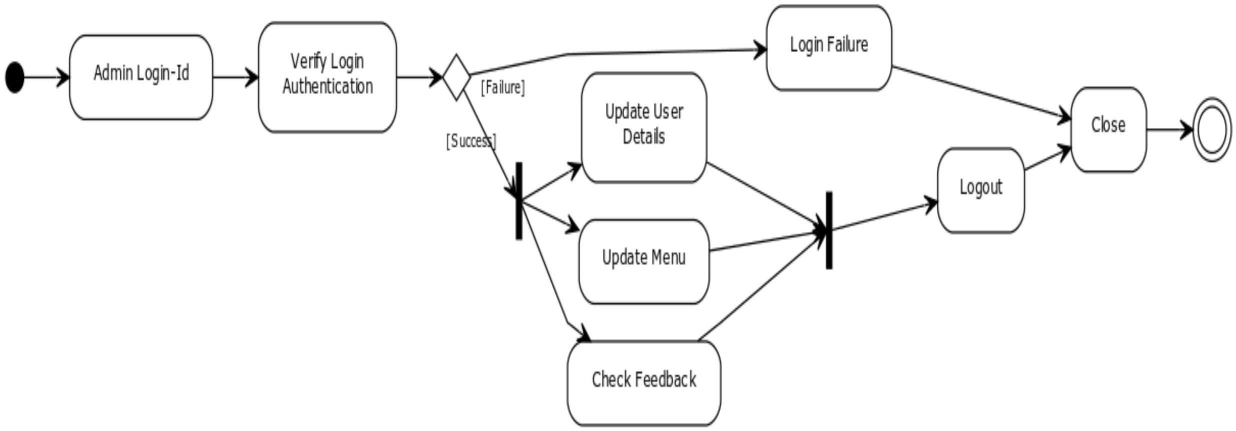


Figure 6: Activity diagram for admin

Activity Diagram of Hardware:

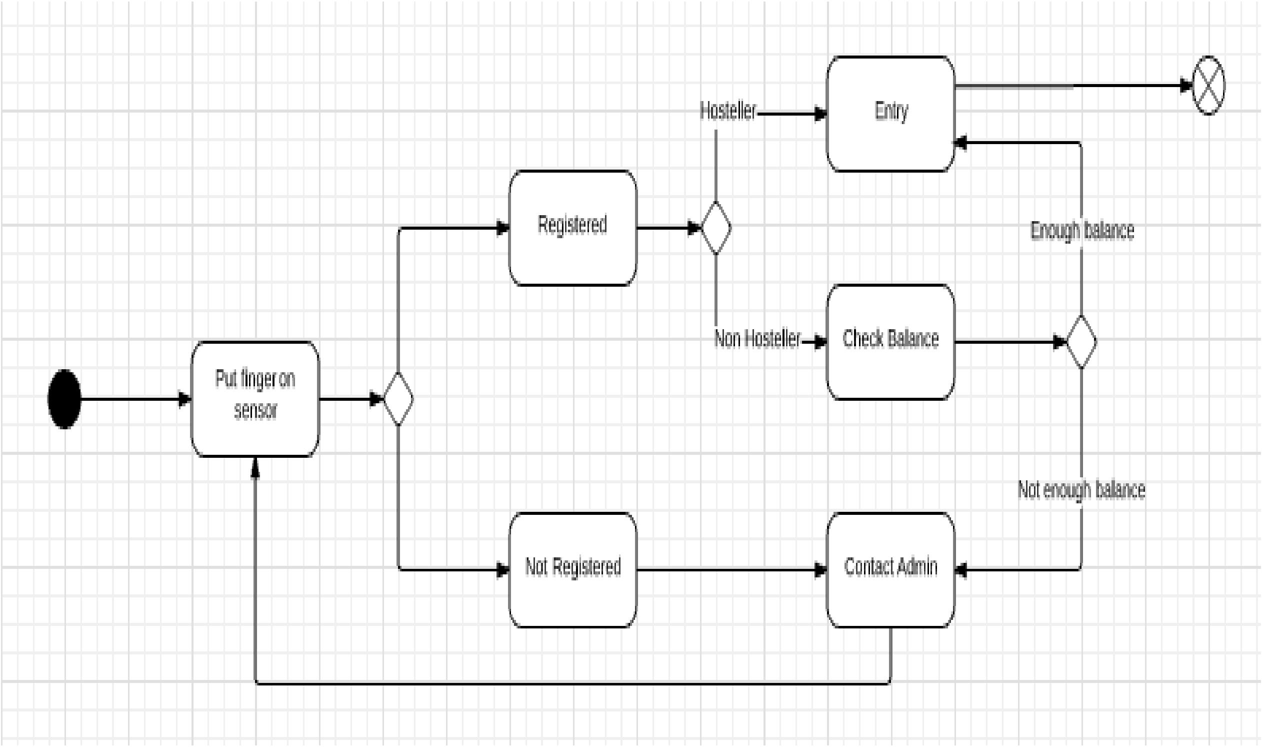


Figure 7: Activity diagram for hardware

Flow Chart:

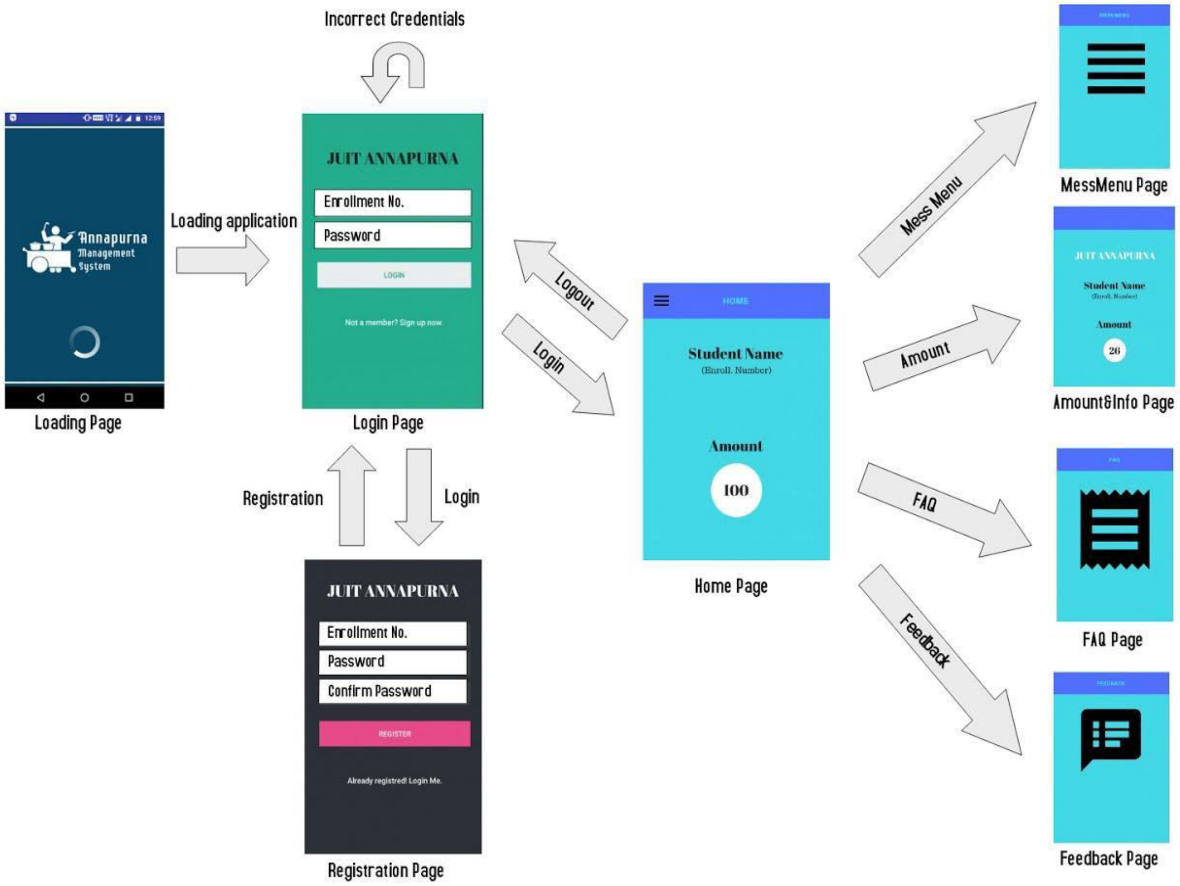


Figure 8: Flow chart

3.2 Development

3.2.1 Overview

As a reason for my deliverable IoT module and usage of the arrangement, I have assessed the framework in parts. It is enticing to consider the diminishment of grating to manufacture physical items like an innovative upset, yet there is nothing in the creation space that is really new in itself. What is diverse is that the instruments are considerably more open. Hardly any individuals could assess microcontrollers and make an undertaking of old programming instruments; They were exceedingly specific and had troublesome UIs that required extensive preparing. Since the universe of novices has gotten another existence with IoT space and the multiplication of microcontrollers out in the open idea, makers must make instruments that are extremely proficient, simple to utilize and cheap. Anybody has the assets accessible to start to comprehend the rudiments of how microcontrollers function with devices like the economically accessible Raspberry Pi and a significant number of the programming/IDE apparatuses that permit programming in dialects that bolster downloading code in a microcontroller are free. This gives a presentation indicate a decreased obstruction enter the field for the normal buyer, novice or understudy. Device providers are discovering that the capability of the purchaser is the new expert capability; Professionals now need a similar convenience requested by buyers. So, we can never totally dispense with erosion. This present reality has limits that you basically can not discover in programming improvement. In any case, the sort of programming that comprehends and deals with the breaking points of this present reality is accessible for singular models. Arranging encoding conditions 18 that can delineate, distinguish clashes, and discover approaches to stay away from them is presently a matter of running a workstation installer. While the experience is accessible in modules, instead of in take-away or inflexible take-away bundles (in itself a mind blowing change in the scene of individual and customer electronic gadgets), fans and understudies may not begin doing alone. processors or finish items with here and now encounter. Genuine experience is more vital than any time in recent memory, however the capacity to pick and pick the profundity at which to investigate one part of the item, (for example, code) leaves the learning gained to others, so you can center around the bigger picture is a change freeing.

3.2.2 Competing Standards for IoT Connectivity

A genuine IoT network, not only an arrangement of gadgets that know how to speak with each other. Amid the youth of the Internet, where the idea of standard and open conventions was set up out of the blue, we got that involvement with the time so anybody could construct equipment that could cooperate with whatever else. Presently we underestimate it, however the worldwide system would be a totally better place on the off chance that we required a program to utilize interpersonal organizations, another to browse email and another to watch motion pictures. It is conceivable that these programs have never been keep running on a similar equipment. In late decades, our gadgets worked this way: each had a particular reason and just that reason, and couldn't be utilized for whatever else. Our TVs were simply TVs, proposed to get, decipher and demonstrate a flag, and just that. The models existed, however the gauges were specific to the point that they didn't wind up institutionalizing much. A container would never do what another case did. Standard conventions change over gadgets into stages. While the system conventions that at last arranged the Internet in the place we see today overwhelmed, we started to understand that there was no exceptional need particular gadgets, and that utilizing a program that could incorporate numerous conventions and perform numerous assignments was to be valuable for content makers and buyers. Right now, the IoT is in a condition of progress, where gadgets become speedier than the measures executed for them. In the event that you have a cell phone that distinguishes and sends the closeness of your home to an indoor regulator, might you want to build the warming in your home when it arrives? Do you need lights that match the music that is playing? Do you have a candy machine that you might want to send you a business report made that day as it approaches? A gadget can identify logical data and transmit it to an application on a server in the cloud. Be that as it may, regardless of how charming the open doors are, none of this can occur without standard. We have numerous gadgets in our homes that we might want to see implanted in the IoT area, yet we would not have any desire to be secured up one provider and purchase every one of our gadgets to make sure they can interface. We need to exploit different offers or get the best out of the considerable number of universes of every producer that is best to create a specific sort of gadget. In reality as we know it where each maker distributes their correspondence and property conventions and anticipates that the champ will take everything, there will unavoidably be a war of measures and disorder in space: contrary gadgets from various providers who don't cooperate. The Z-Wave site characterizes the standard as "a remote innovation that enables shrewd gadgets to speak with each other". Residential items, for example, lights, bolts and indoor regulators are made "brilliant" when Z-Wave availability is included inside the item configuration, enabling them to impart and play out the coveted capacity. "While the two benchmarks are expected to offer advantages and an expansive arrangement of items that are good with them, for the motivations behind my venture, I chose to utilize Wi-Fi just as a standard, on the grounds that different norms don't have just unbending tenets of utilization, they likewise require licenses and an underlying speculation to begin utilizing the innovation.

Wi-Fi is the most suitable in light of the fact that:

• Does not require enrollment in an organization together or installment of a permit expense. The Wi-Fi is totally open.
• It is broadly accessible.
• Wi-Fi is as of now coordinated into relatively every buyer gadget we see today, paying little respect to the maker.
• It is a standard officially shared by a few sorts of gadgets in the IoT space.

• It is sensible to accept that wherever or home as of now contains a Wi-Fi framework to be utilized, dissimilar to the presence of a ZigBee or Z-Wave item.

• Simplicity is the key and Wi-Fi disentangles the association strategies and the devices used to fabricate the undertaking.

3.2.3 Hardware and Software Used, Justifications for Use

3.2.3.1 Software:

- Android Studio

It is the authority coordinated improvement condition (IDE) for the Google Android working framework, worked with the IDEA JetBrains Intellij programming and composed particularly for the advancement of Android. It is accessible for download on Windows, MacOS and Linux working frameworks. It is a trade for Android Eclipse improvement instruments (ADTs) as the essential IDE for creating local Android applications.

AS is uniquely intended for the improvement of Android and to quicken the advancement procedure of Android applications and make it less difficult is the primary target of this IDE.

Gradle joining: Android Studio utilizes the quickly developing Gradle creation framework that is so coordinated and Gradle is a better than average apparatus. On the off chance that you've chosen to run with Eclipse, say you should take a gander at Gradle's highlights and attempt to check whether it fits your undertaking. On the off chance that you need to run with Android Studio, you ought not stress over keeping the Gradle framework since it's decent. Shroud utilizes Apache Ant as its primary arrangement framework, which is a to a great degree hearty XML-based form framework and numerous Java designers definitely know it.

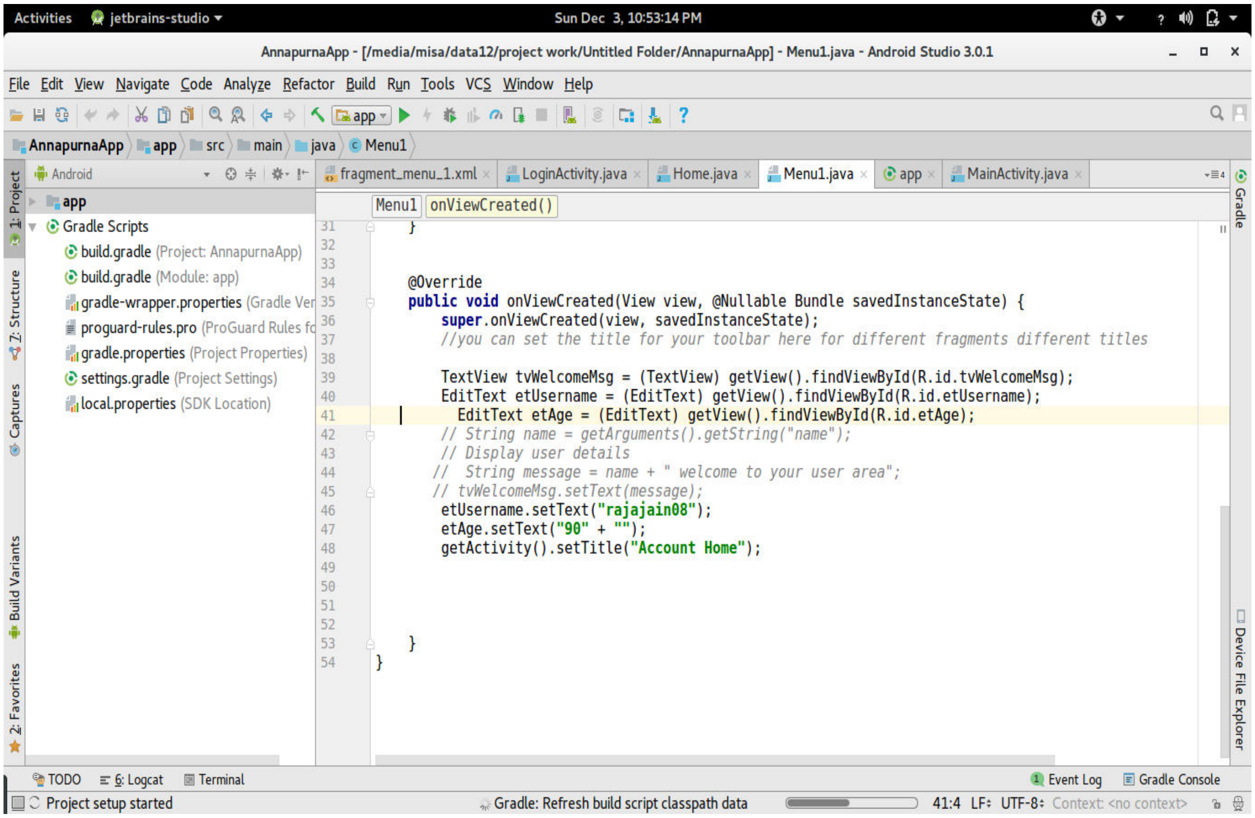
Propelled code finishing: Both Android Studio and Eclipse have the programmed culmination run of the mill of Java code. In any case, by and large, we have discovered that the culmination of the code is in reality preferable in AS over in contrast with Eclipse, which at times appears somewhat puzzled and does not give exact outcomes more often than not. Remember that the additional time you spend as a developer in the code, the more you will assess the fulfillment of the code.

(UI): we know the interface and the Eclipse inconsistencies extremely well, it's huge and somewhat overpowering, yet we need to confront it on the grounds that most IDEs are overpowering when you utilize them out of the blue. Subsequently, thinking about this, it has developed that the devices and menu things in Android Studio tend to take me where we need it to be somewhat less demanding and easy than their Eclipse partners. Also, AS was composed particularly for Android, while Eclipse was intended for numerous utilization IDE that can be utilized with any dialect and stage.

Task Organization: Although both IDEs work contrastingly to enable you to oversee and sort out your undertakings, yet when you need to deal with numerous activities in Eclipse, you have to combine them in a workspace. While trying to change to another workspace, you have to pick the way, after which Eclipse has been restarted and this dependably appears to be awkward. Moreover. Then again, Android Studio utilizes the modules to oversee and sort out the code modules that have their own Gradle arrangement documents, which implies that you can set up your own particular conditions. In examination, AS appears to be more regular, however in the event that you've been utilizing Eclipse for some time, it will set aside some opportunity to get accustomed to it.

Framework strength: Eclipse is basically Java-based programming and a bigger IDE than Android Studio, so you require a significantly bigger measure of RAM with a high CPU speed to work appropriately. Inability to conform to this foundation makes Eclipse crash anomalous and quit reacting. Then again, Android Studio is currently being propelled with not very many blunders and offers a more steady execution ensure than Eclipse and even the framework needs are lower. AS is quick, while it takes 1 or 2 minutes to make dispatch variants of complex ventures in Eclipse, yet in AS you can run a similar task in 30 seconds.

Intuitive: Android Studio has a GUI (graphical UI), yet Eclipse does not. Be that as it may, intuitive isn't fundamental for developers, who are not exceptionally worried about the visual components of their applications. A designer must have a point by point comprehension of Visual Basic, with the goal that the engineer can utilize the intuitive capacity fittingly. It's another element of Android Studio, however its nonattendance status in Eclipse isn't vital.

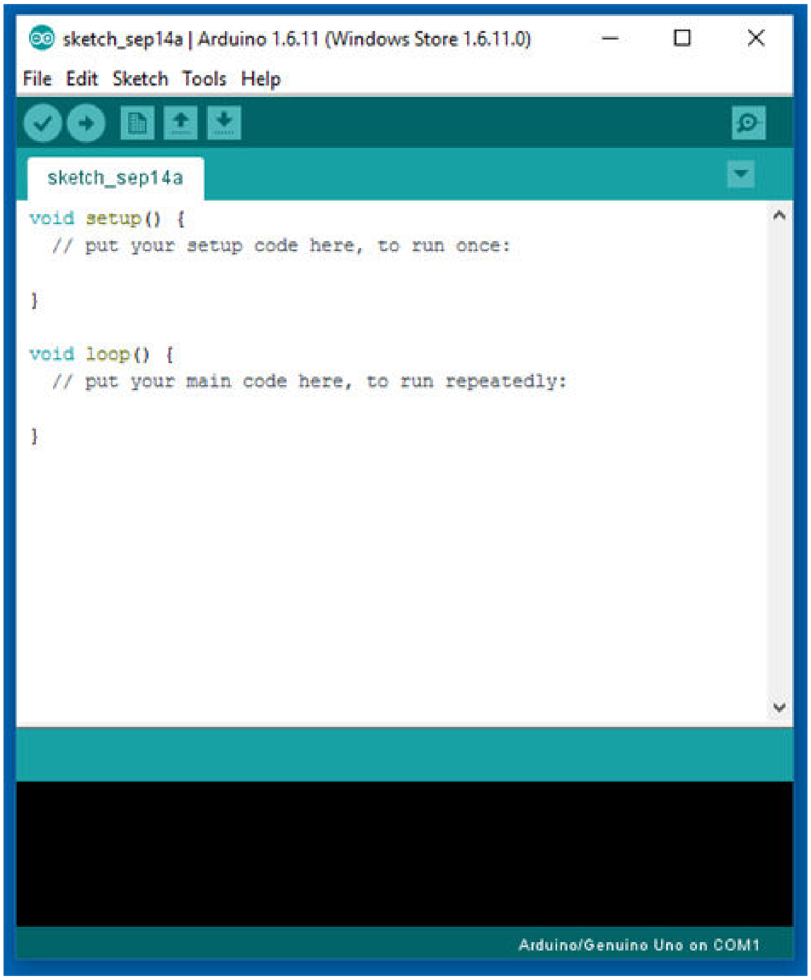

Activities
jetbrains-studio
Sun Dec 3, 10:53:14 PM
AnnapurnaApp - [/media/misa/data12/project work/Untitled Folder/AnnapurnaApp] - Menu1.java - Android Studio 3.0.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
app
AnnapurnaApp app src main java Menu1
Android
fragment_menu_1.xml
LoginActivity.java
Home.java
Menu1.java
app
MainActivity.java
Menu1 onViewCreated()
app
Gradle Scripts
build.gradle (Project: AnnapurnaApp)
build.gradle (Module: app)
gradle-wrapper.properties (Gradle Ver
proguard-rules.pro (ProGuard Rules fo
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)
1: Project
7: Structure
Captures
Build Variants
2: Favorites
Gradle
Device File Explorer
TODO
6: Logcat
Terminal
Event Log
Gradle Console
Project setup started
Gradle: Refresh build script classpath data
41:4 LF÷ UTF-8÷ Context: <no context>

```
    }


    @Override
    public void onViewCreated(View view, @Nullable Bundle savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);
        //you can set the title for your toolbar here for different fragments different titles

        TextView tvWelcomeMsg = (TextView) getView().findViewById(R.id.tvWelcomeMsg);
        EditText etUsername = (EditText) getView().findViewById(R.id.etUsername);
          EditText etAge = (EditText) getView().findViewById(R.id.etAge);
        // String name = getArguments().getString("name");
        // Display user details
       //  String message = name + " welcome to your user area";
       // tvWelcomeMsg.setText(message);
        etUsername.setText("rajajain08");
        etAge.setText("90" + "");
        getActivity().setTitle("Account Home");




    }
}
```

Figure 9: Android Studio

- **Arduino IDE**

Arduino Integrated Development Environment, or Arduino Software (IDE), contains a content manager for composing codes, a message zone, a content comfort, a toolbar with catches for normal capacities and a progression of menus. It interfaces with the Arduino and Genuino equipment to stack programs and speak with them. Projects composed utilizing the Arduino programming (IDE) are called outlines. These portrayals are composed in the content tool and spared with the .ino expansion. The supervisor has the usefulness to slice/glue and to seek/supplant content. The message region gives remarks when sparing and sending out and furthermore indicates mistakes. The comfort shows the content issued by the Arduino programming (IDE), including complete blunder messages and other data. The lower right corner of the window demonstrates the designed card and the serial port. The catches in the toolbar enable you to check and load programs, make, open and spare portrays and open the screen in arrangement.



sketch_sep14a | Arduino 1.6.11 (Windows Store 1.6.11.0)
File Edit Sketch Tools Help
sketch_sep14a
Arduino/Genuino Uno on COM1

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Figure 10: Arduino IDE

3.2.3.2 Hardware

- **Arduino Uno**

Arduino Uno R3 is a microcontroller board in light of a removable AMP ATmega328 dual-pack microcontroller (DIP). It has 20 advanced information/yield pins (of which 6 can be utilized as PWM yields and 6 can be utilized as simple data sources). Projects can be stacked from the simple to-utilize Arduino programming. Arduino has a colossal help group, which makes it a simple method to begin working with incorporated electronic parts. The R3 is the third and last correction of Arduino Uno. Arduino likewise rearranges the way toward working with microcontrollers, however offers a few points of interest for educators, understudies and fans keen on different frameworks:

Temperate: Arduino sheets are moderately cheap contrasted with other microcontroller stages. The more affordable variant of the Arduino module can be gathered by hand, and even the pre-amassed Arduino modules cost not exactly $ 50.

Multiplatform: the Arduino programming (IDE) takes a shot at Windows, Macintosh OSX and Linux working frameworks. Most microcontroller frameworks are restricted to Windows.

Basic and clear programming condition: the Arduino (IDE) programming is anything but difficult to use for fledglings, however sufficiently adaptable to enable propelled clients to exploit it. For instructors, it is advantageously in light of the Processing programming condition, so understudies figuring out how to program in that condition will be acquainted with the activity of the Arduino IDE.

Open source and extensible programming: the Arduino programming is distributed as an open source apparatus, accessible for expansion by master software engineers. The dialect can be extended through C ++ libraries and individuals who need to comprehend specialized points of interest can make the jump from Arduino to the AVR C programming dialect on which it is based. Similarly, you can include the AVR-C code specifically to your Arduino programs, on the off chance that you wish.

Open source and extensible equipment: the plans of Arduino boards are distributed under a Creative Commons permit, so master circuit architects can make their own particular variant of the module, extend it and enhance it. Indeed, even moderately unpracticed clients can make the module variant to see how it functions and spare cash.

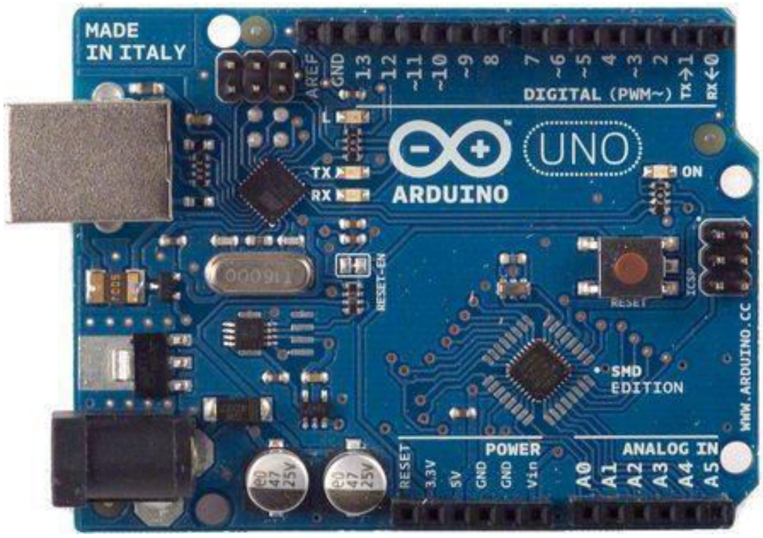


Figure 11: Arduino UNO

- **Optical Fingerprint Reader R303A:**

Unique mark handling incorporates two sections: unique mark enrollment and finger impression coordinating (correspondence can be 1: 1 or 1: N). While enlisting, the client must embed the finger twice. The framework will process the two-finger pictures, produce a finger display in view of the aftereffects of the procedure and store the model. Amid the correspondence, the client embeds the finger into the optical sensor and the framework creates a finger model and contrasts it and the finger library models. For 1: 1 correspondence, the framework will contrast the live finger and the particular model assigned in the Module; for the inquiry framework 1: N or hunt, the framework will look the whole finger library for the relating finger. In the two conditions, the framework will restore the relating result, achievement or disappointment.

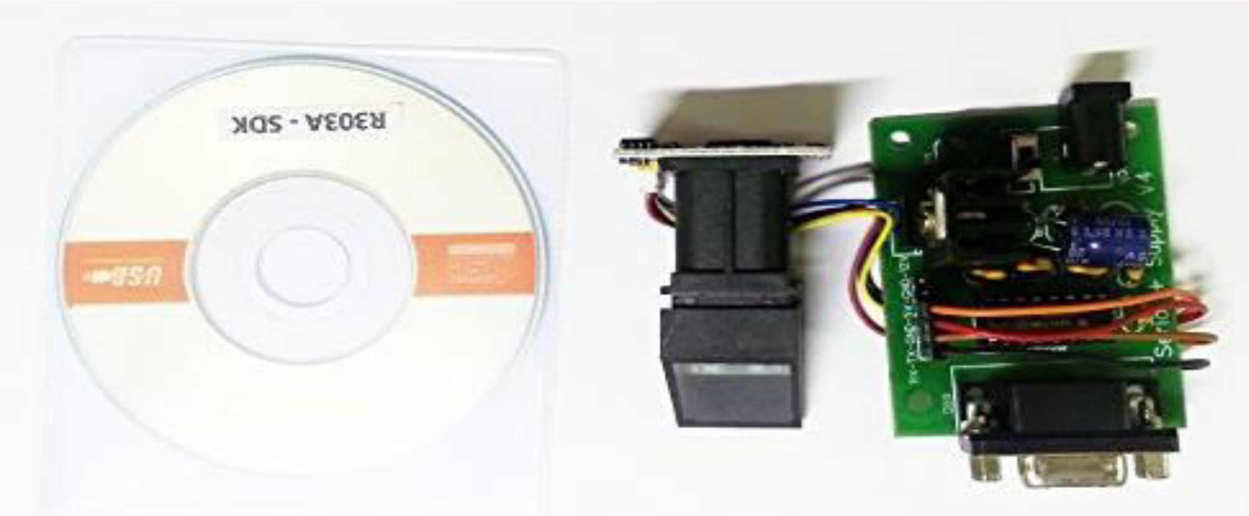
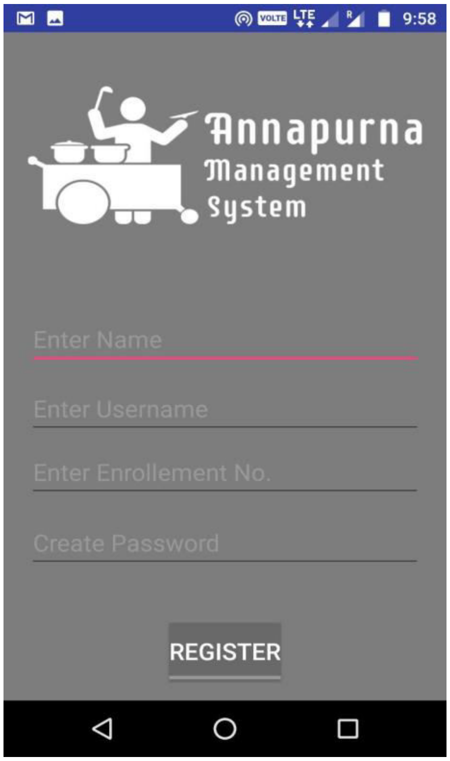


Figure 12: Fingerprint Sensor

Power	DC 3.6V-6.0V	**Interface**	UART(TTL logical level)/ USB 1.1
Working current	Typical: 100mA Peak: 150mA	**Matching Mode**	1:1 and 1:N
Baud rate	(9600*N)bps, N=1～12 (default N=6）	**Character file size**	256 bytes
Image acquiring time	<0.5s	**Template size**	512 bytes
Storage capacity	120/ 375/ 880	**Security level**	5 (1, 2, 3, 4, 5(highest))
FAR	<0.001%	**FRR**	<0.1%
Average searching time	< 0.8s (1:880)	**Window dimension**	18mm*22mm
Working environment	Temp: -10℃- +40℃	**Storage environment**	Temp: -40℃- +85℃
	RH: 40%-85%		RH: <85%
Outline Dimention	Split type	Module: 42*38*7mm Sensor:56*20*21.5mm	
	Integral type	54.5*20.6*23.8mm	

Table 1: Specifications of fingerprint sensor

3.3 Experimental



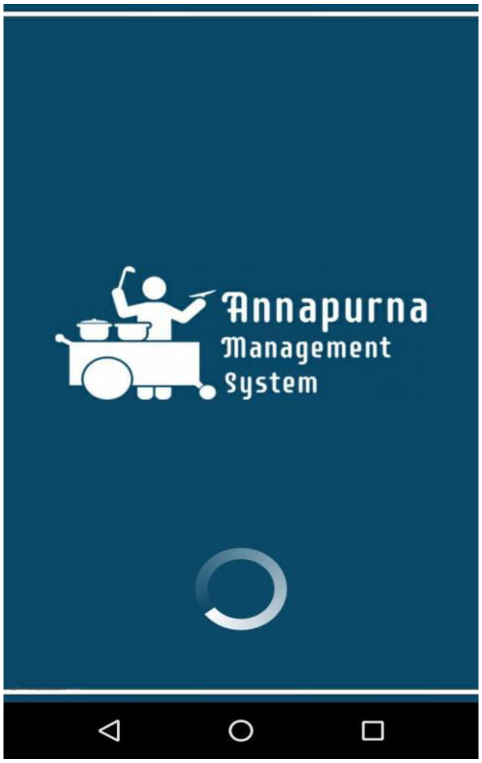


Fig 13: Register page

Fig 14: App loading page

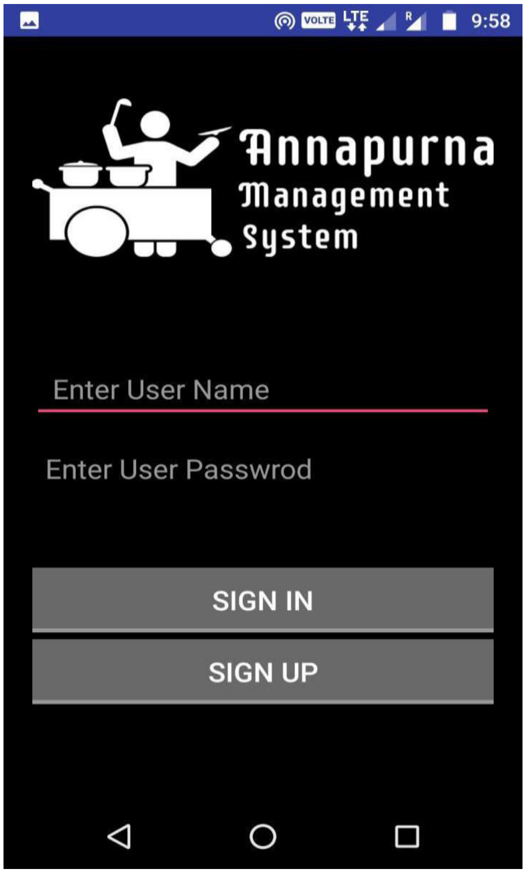
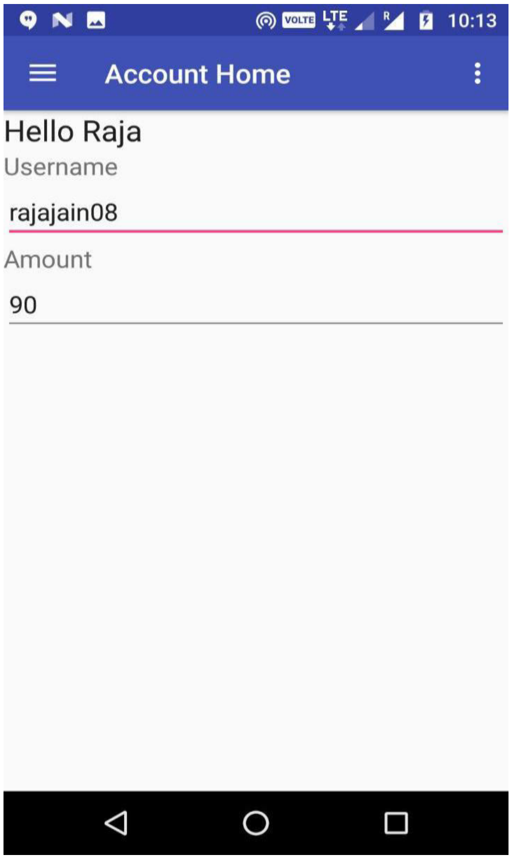


Fig 15: Sign in page



Hello Raja

Username

rajajain08

Amount

90

Fig 16: Home Page

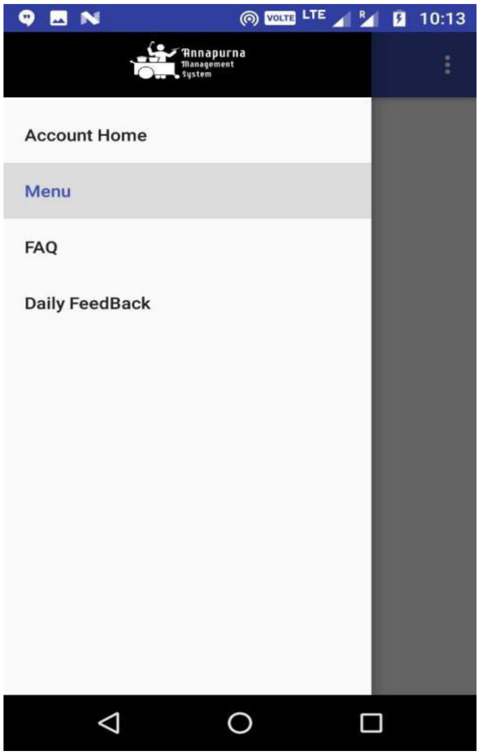


Fig 17: Navigation Drawer

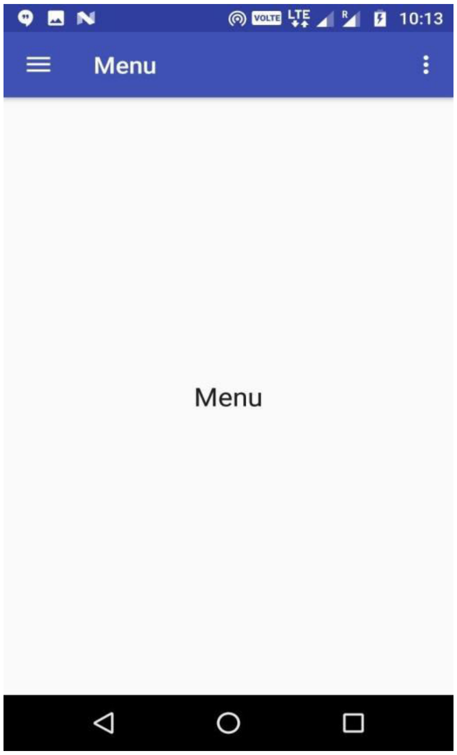
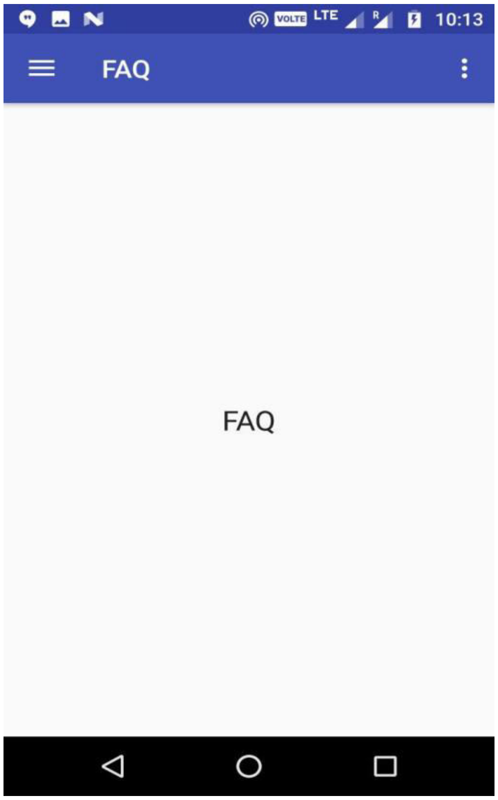
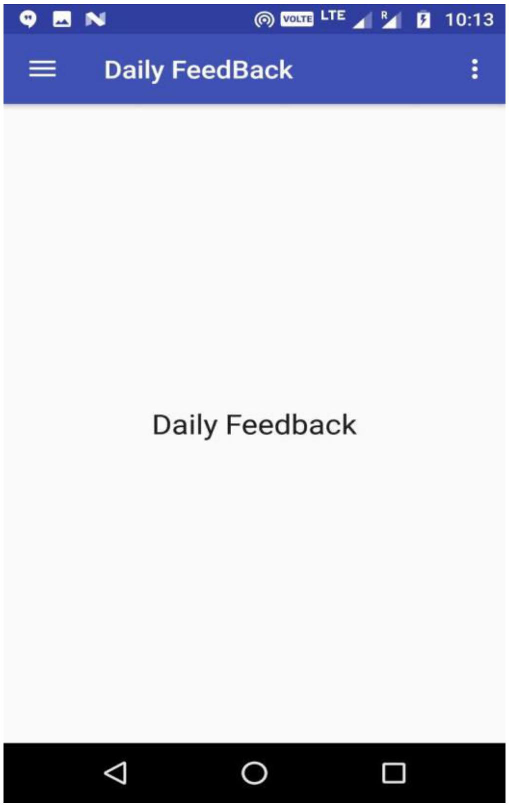


Fig 18: Menu page



FAQ

Fig 19: FAQ page

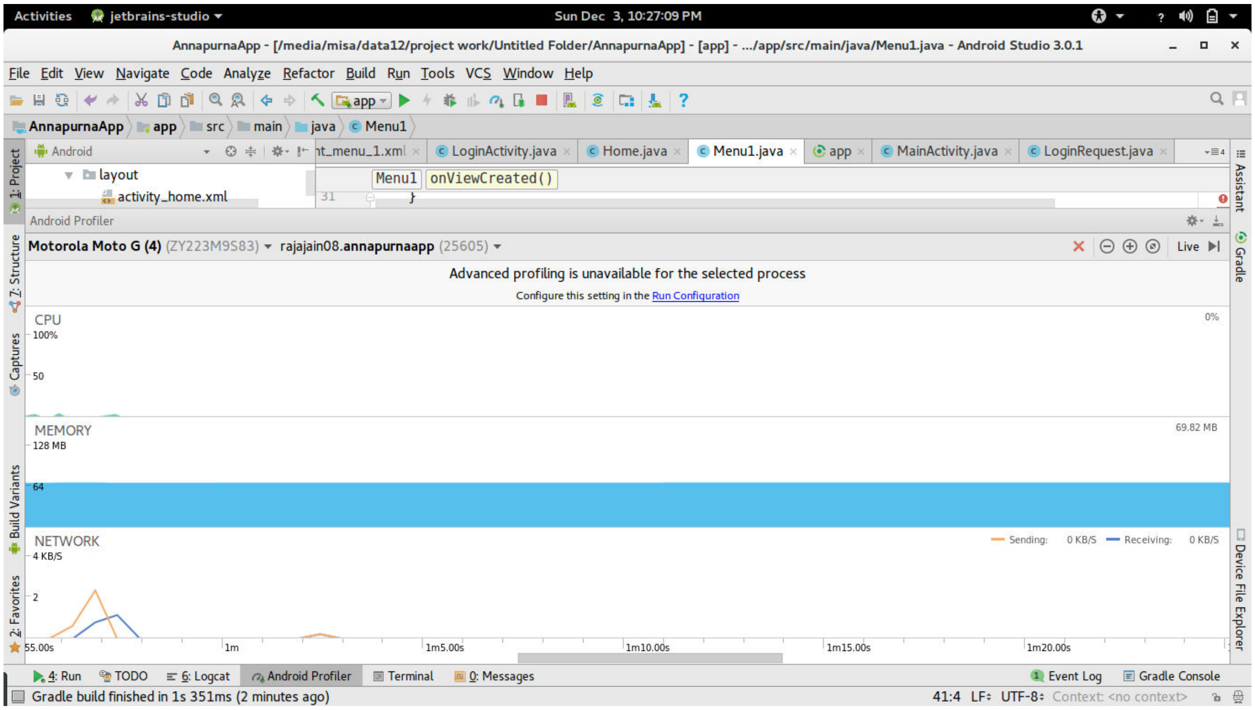


Daily Feedback

Fig 20: Feedback page

PERFORMANCE ANALYSIS

Following screenshots gives the performance of the Android application when different options were choosen.



AnnapurnaApp - [/media/misa/data12/project work/Untitled Folder/AnnapurnaApp] - [app] - .../app/src/main/java/Menu1.java - Android Studio 3.0.1

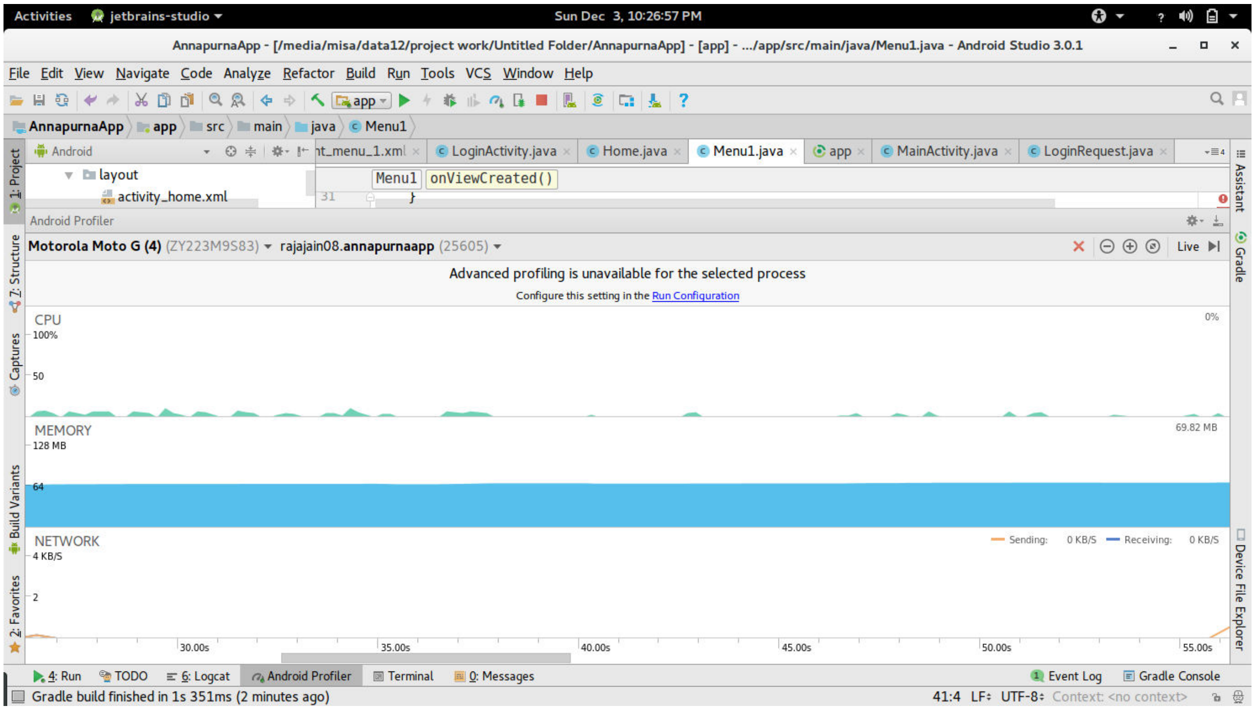
Motorola Moto G (4) (ZY223M9S83) rajajain08.annapurnaapp (25605)

Advanced profiling is unavailable for the selected process

Configure this setting in the Run Configuration

Gradle build finished in 1s 351ms (2 minutes ago)

Figure 21: Screenshot 1 of performance analysis

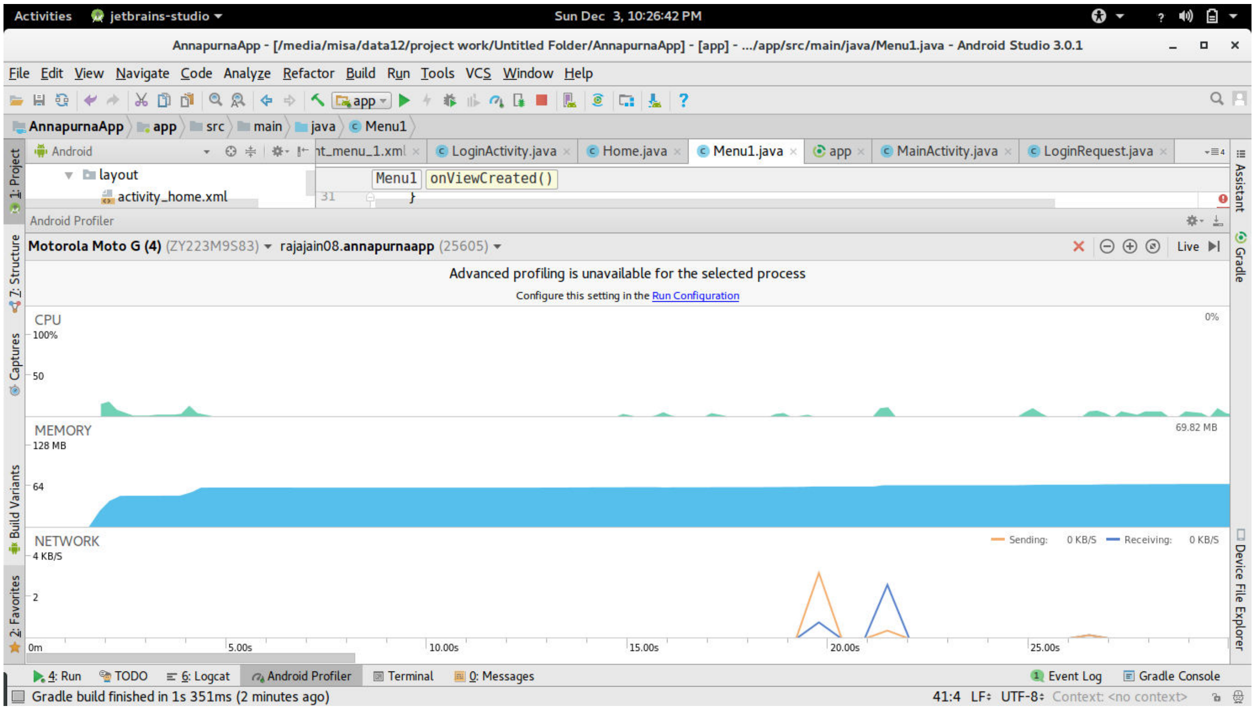

Activities
jetbrains-studio
Sun Dec 3, 10:26:57 PM
AnnapurnaApp - [/media/misa/data12/project work/Untitled Folder/AnnapurnaApp] - [app] - .../app/src/main/java/Menu1.java - Android Studio 3.0.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
app
AnnapurnaApp app src main java Menu1
Android
layout
activity_home.xml
nt_menu_1.xml LoginActivity.java Home.java Menu1.java app MainActivity.java LoginRequest.java
Menu1 onViewCreated()
Android Profiler
Motorola Moto G (4) (ZY223M9S83) rajajain08.annapurnaapp (25605)
Live
CPU
100%
50
0%
MEMORY
128 MB
64
69.82 MB
NETWORK
4 KB/S
2
Sending: 0 KB/S Receiving: 0 KB/S
30.00s
35.00s
40.00s
45.00s
50.00s
55.00s
4: Run TODO 6: Logcat Android Profiler Terminal 0: Messages Event Log Gradle Console
41:4 LF: UTF-8: Context: <no context>
1: Project 7: Structure Captures Build Variants 2: Favorites Assistant Gradle Device File Explorer

Advanced profiling is unavailable for the selected process

Configure this setting in the Run Configuration

Gradle build finished in 1s 351ms (2 minutes ago)

Figure 22: Screenshot 2 of performance analysis


Activities
jetbrains-studio
Sun Dec 3, 10:26:42 PM
AnnapurnaApp - [/media/misa/data12/project work/Untitled Folder/AnnapurnaApp] - [app] - .../app/src/main/java/Menu1.java - Android Studio 3.0.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
app
AnnapurnaApp app src main java Menu1
Android
layout
activity_home.xml
nt_menu_1.xml LoginActivity.java Home.java Menu1.java app MainActivity.java LoginRequest.java
Menu1 onViewCreated()
Android Profiler
Motorola Moto G (4) (ZY223M9S83) rajajain08.annapurnaapp (25605)
Live
CPU
100%
50
0%
MEMORY
128 MB
64
69.82 MB
NETWORK
4 KB/S
2
Sending: 0 KB/S Receiving: 0 KB/S
0m
5.00s
10.00s
15.00s
20.00s
25.00s
4: Run TODO 6: Logcat Android Profiler Terminal 0: Messages Event Log Gradle Console
41:4 LF: UTF-8: Context: <no context>
1: Project 7: Structure Captures Build Variants 2: Favorites Assistant Gradle Device File Explorer

Advanced profiling is unavailable for the selected process

Configure this setting in the Run Configuration

Gradle build finished in 1s 351ms (2 minutes ago)

Figure 23: Screenshot 3 of performance analysis

CONCLUSION

5.1 Conclusion

To finish up the portrayal of the undertaking: The task, created, depends on the detail of the client prerequisites and on the investigation of the current framework, with adaptability for future changes. The task delivers issues identified with the administration of a sanctuary and keeps away from the issues that happen when they are transported physically.

Recognizing the impediments of the current framework prompts the plan of a mechanized framework that will be perfect with the current framework with the framework that is simpler to utilize and more arranged to the GUI.

The goal of this framework is to offer verified access and an entire data gateway about the disarray of the safe houses. In this framework, clients can download the application on their cell phone and after that entrance the messiness capacities. Remarks can be given to the chairman. They can likewise get to your record data.

Likewise, this product will decrease work, in this way lessening the capital contributed. In this way, the product is made with extra advantages contrasted with the current ones. Notwithstanding, different viewpoints will be produced to enhance execution and proficiency.

5.2 Future Work

- Including features for other hostel activities, such as daily attendance, entry to campus for hostellers and non-hostellers.
- Pay the fee online by credit card, debit card or even wallet like paytm, phone.

- Add more functionality in our application, such as the nutritional value of food in the menu.
- Switch from Android to another operating system like iOS, Blackberry.
- Including functionalities for the other mess activities like inventory management.
- Online meal booking.

REFERENCES

[1] developer.android.com, ‘Android layouts, designs, animation’. Online. Available: https://developer.android.com/design/index.html

[2] askubuntu.com, ‘Solution for ubuntu related issue’. Online. Available: https://askubuntu.com/questions/55312/how-to-save-php-file-in-opt-lampp-htdocs-without-going-to-terminal-and-type-sud

[3] Wikipedia.org, ‘Related content’. Online. Available: https://en.wikipedia.org/wiki/Android_version_history

[4] stackoverflow.com, ‘Android related issues’. Online. Available: https://stackoverflow.com/questions/31433687/android-gradle-apache-httpclient-does-not-exist

[5] stackoverflow.com, ‘Database related issues’. Online. Available: https://stackoverflow.com/questions/23230729/warning-mysql-connect-access-denied-for-user-rootlocalhost-using-passw