PROJECT REPORT

ON

Intellect Based Discount Application

BY

Vaibhav Nandkumar Bhatambrekar

Savitribai Phule Pune University

Master of Computer Application

MES's

Institute of Management and Career Courses (IMCC)

Seat No:11589

2017-2020

क्रियेवीण वाचाळता व्यर्थ आहे
महाराष्ट्र एज्युकेशन सोसायटी, पुणे
स्थापना १८६०

Maharashtra Education Society's

INSTITUTE OF MANAGEMENT AND CAREER COURSES (IMCC)

(Recognized by Savitribai Phule Pune University & Approved by AICTE)

131, Mayur Colony, Kothrud, Pune 411 038.
Tel. +91-20-25466271, 25463453 • E-mail: director.imcc@mespune.in

DR. SANTOSH DESHPANDE
Director
Web Site: https://imcc.mespune.in

Ref. No: MCA/Project/023/2020-21

Date : 14/09/2020

CERTIFICATE

This is to certify that the Project Report entitled ***"Intellect based discount application" is*** prepared by ***Vaibhav Nandkumar Bhatambrekar ,*** a student of **M.C.A.** Course for the Academic Year 2019-20 at M.E.Society's Institute of Management & Career Courses (IMCC), Pune - 41038. M.C.A Course is affiliated to Savitribai Phule Pune University.

To the best of our knowledge, this is original study done by the said student and important sources used by him have been duly acknowledged in this report.

The report is submitted in partial fulfillment of M.C.A. Course for the Academic Year 2019-20 as per the rules & prescribed guidelines of Savitribai Phule Pune University.

Dr. Ravindra Vaidya

HOD, Department of MCA, IMCC

Dr. Santosh Deshpande

Director, IMCC

ACKNOWLEDGEMENT

A successful project work is result of a well-organized and coordinated teamwork. So, at the completion of the project I feel obliged to extend my gratitude towards all those who made valuable contribution throughout the execution of my project

“Intellect Based Discount Application”.

I would like to extend my thanks and gratitude to my project guide **Mrs. Darshana Yadav** (Assistant Professor, IMCC) – Internal Guide and **Mr. Abhay Damle** – External Guide for their valuable guidance and timely assistance throughout the development of this project.

I would also like to extend my thanks and gratitude to **Dr. Santosh Deshpande** (Director, IMCC), **Dr. Ravindra Vaidya** (HOD, IMCC) and **Dr. Manasi Bhate** (Head – Training and Placement, IMCC) for their constant help and support.

-Vaibhav Bhatambrekar

INDEX

Sr. No.	Name of Topic	Page No.
1	**Chapter 1: Introduction**	
	1.1 Existing System and Need for System	1
	1.2 Scope of Work	5
	1.3 Operating Environment – Hardware and Software	8
	1.4 Detail Description of Technology Used	10
2	**Chapter 2: Proposed system**	
	2.1 Proposed System	18
	2.2 Objectives of System	22
	2.3 User Requirements	25
3	**Chapter 3: Analysis & Design**	
	3.1 Object Diagram	27
	3.2 Class Diagram	28

	3.3 Use Case Diagrams	29
	3.4 Activity Diagrams	31
	3.5 Sequence Diagrams	33
	3.6 Entity Relationship Diagram	35
	3.7 Module Hierarchy Diagram	36
	3.8 Component Diagram	37
	3.9 Deployment Diagram	38
	3.10 Module Specifications	39
	3.13 User Interface Design	43
	3.14 Data Dictionary	55
	3.15 Table specifications	57
	3.16 Test Procedures and Implementation	63
4	**CHAPTER 4: USER MANUAL**	
	4.1 User Manual	77
	4.2 Operations Manual / Menu Explanation	94
	4.3 Program Specifications / Flow Charts	96
5	**Drawbacks and Limitations**	101

6	**Proposed Enhancements**	102
7	**Conclusions**	103
8	**Bibliography**	
9	**ANNEXURES:**	
	ANNEXURE 1: USER INTERFACE SCREENS	
	ANNEXURE 2: OUTPUT REPORTS WITH DATA	
	ANNEXURE 3: SAMPLE PROGRAM CODE	

CHAPTER 1

INTRODUCTION

1.2. EXISTING SYSTEM AND NEED FOR SYSTEM:

EXISTING SYSTEM:

The current system is manual. If customer want to shop mobile phone and customer want a discount on that then customer go to the shopping mall or we can say the wholesale shop to get the discount.

Sometimes there is no discount or sometime there is flat discount which were around 5 to 10 % and to get that 5 to 10 % discount customer have to do lot of things like search on the internet that which shopkeeper is providing the large discount.

Due to that the customer is wasting his/her time and the shopkeeper who provide the less discount will get less number of customers and he will get less profit and it effect on his business. So, it is not ideal for the customer as well as the shopkeeper.

The customer approaches a shop and select the product and then demands the shopkeeper for the discount, further the shopkeeper decide whether to give discount or not then the customer will negotiate the shopkeeper to give the discount and hence after all this process the customer will may or may not get discount.

In the existing system, the customer will get the direct discount or get the flat discount which is decided by the shopkeeper or the company, he will not get the major discount over the product.

Due to that the shopkeeper sale is not increasing and the market is not growing up. So, for that we have to discover the solution. Because in the existing system no one is happy, neither the shopkeeper or the customer.

The customer has another way of shopping that is online shopping. In which the customer will get the flat discount and he can compare over the various online shopping sites.

But that has also a limited discount of 10 to 15%. And for that discount it has a limited period so it is difficult to get that discount.

Another point is that there are number of customers who even don’t know about the offers and they are unable to get the discount on the product.

So, in the conclusion part we can say that in the existing system the customer is only benefited through the direct discount. So, for this existing system we have to develop an application which gives an effective discount.

NEED FOR THE SYSTEM:

1) In the existing system the customer is bound to flat discount or the particular amount of discount
2) In the existing system the customer physically approaches to the shop then sometimes the shop is closed so the customer has to return back.
3) In the existing system the discount offers are limited.

4) In the existing system the customer only gets 5 to 10% discount, customer will not able to get more discount.
5) In this system the shopkeeper which gives less discount has gets less number of customers and this will impact his/her business.

6) In the existing system the customer approaches the shop and then customer select mobile but as the mobile price is very high, he demands the shopkeeper to give the discount. But the shopkeeper can't give the discount, So the customer negotiates the shopkeeper to give the discount, then after all this shopkeeper may or may not give the discount, so in this situation neither the customer is happy or the shopkeeper.
7) In the existing system the customer has to negotiate to shopkeeper every time to give the discount, so that case the shopkeeper cannot able to maintain his/her customers.
8) So, there is loss of shopkeeper as well as the customer.

1.3 SCOPE OF WORK

This system is being developed to get the intellect-based discount to the customer

This benefit to the customer that he can't bound to particular or the flat discount, the customer can earn his own discount.

There are several modules or the users in the system.

1) Firstly, the customer has to download the application from the play store

2) Then customer has to register on the application with the email id and the phone number.

3) Then the customer has to login into the application.

4) After the login, user will get the list of products.

5) Then after that user will select one of the products.

6) After selecting the product user will get the quiz.

7) Then the user will get the questions based upon the complexity level of questions and the price of the product.

8) As the higher price of product, the complexity of questions will be high.

9) As the user will give as more number of correct answers the user will get more and more discount.

10) Then the user will have to choose the correct option for the question within the limited time.

11)if the answer is correct then the user will get the discount based on the complexity level of the question.

12) after that the user will get the next question.

13) if the user selected answer is wrong then user will redirect to the page and he will ger the discount based on the previous correct question.

14) if the user has given the wrong answer of first question then he will not get any discount.

15) If the user can guess all the correct answers then user can able to get the large amount of discount on the mobile up to 80%.

16) After user answers the all or number of questions then customer redirected to the quiz score page, where customer will know how much discount he/she availed during the quiz.

17) Also, in quiz score page customer will know much discount price he/she offered/availed, and the price of the product after the discount.

18) Then after, to get that product/mobile customer will get the code on his/her mobile.

19) Then customer have to show that code to shopkeeper, so that shopkeeper can give the discount to the customer.

20) After that customer goes to the counter and collect the mobile.

21) So, this application is beneficial to the customer as well as shopkeeper/vender.

22) Because customer can get as much as discount, he/she wants, and the shopkeeper can get as much as number of customers he/she wants.

1.4 OPERATING ENVIRONMENT – HARDWARE AND SOFTWARE

HARDWARE REQUIREMENTS

- CPU: - Intel (I3 AND ABOVE)
- RAM: - 8 GB
- HARD DISK: - 100 GB
- DISPLAY- LCD Monitor.

SOFTWARE REQUIREMENTS

Front End Tools: -

- Operating system (Windows, Linux).
- Android Studio.
- An Emulator to run android application.
- Latest JDK installed on the system.

Back End Tools: -

- Microsoft SQL Server.
- Microsoft SQL Server management studio.

Client side: -

- An Android mobile having version above 5.0.
- An internet connection.

1.5 DETAIL DESCRIPTION OF TECHNOLOGY USED: -

ANDROID STUDIO:

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA . On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

- A flexible Gradle-based build system.
- A fast and feature-rich emulator.
- A unified environment where you can develop for all Android devices.
- Apply Changes to push code and resource changes to your running app without restarting your app.
- Code templates and GitHub integration to help you build common app features and import sample code.

- Extensive testing tools and frameworks.
- Lint tools to catch performance, usability, version compatibility, and other problems.
- C++ and NDK support.
- Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine.

JAVA:

Java is a high-level programming language originally developed by Sun Microsystems and released in 1995. Java runs on a variety of platforms, such as Windows, Mac OS, and the various versions of UNIX.

Java is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Software Development Domain. I will list down some of the key advantages of learning Java Programming:

- **Object Oriented** − In Java, everything is an Object. Java can be easily extended since it is based on the Object model
- **Platform Independent** − Unlike many other programming languages including C and C++, when Java is compiled, it is not compiled into platform specific machine, rather into platform independent byte code. This byte code is distributed over the web and interpreted by the Virtual Machine (JVM) on whichever platform it is being run on.
- **Simple** − Java is designed to be easy to learn. If you understand the basic concept of OOP Java, it would be easy to master.
- **Secure** − With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.
- **Architecture-neutral** − Java compiler generates an architecture-neutral object file format, which makes the

compiled code executable on many processors, with the presence of Java runtime system.

- **Portable** − Being architecture-neutral and having no implementation dependent aspects of the specification makes Java portable. Compiler in Java is written in ANSI C with a clean portability boundary, which is a POSIX subset.
- **Robust** − Java makes an effort to eliminate error prone situations by emphasizing mainly on compile time error checking and runtime checking.
- **Multithreaded** − With Java's multithreaded feature it is possible to write programs that can perform many tasks simultaneously. This design feature allows the developers to construct interactive applications that can run smoothly.
- **Interpreted** − Java byte code is translated on the fly to native machine instructions and is not stored anywhere. The development process is more rapid and analytical since the linking is an incremental and light-weight process.

- **High Performance** − With the use of Just-In-Time compilers, Java enables high performance.
- **Distributed** − Java is designed for the distributed environment of the internet.
- **Dynamic** − Java is considered to be more dynamic than C or C++ since it is designed to adapt to an evolving environment. Java programs can carry extensive amount of run-time information that can be used to verify and resolve accesses to objects on run-time

SQL SERVER:

SQL Server is a relational database management system (RDBMS) developed by Microsoft. It is primarily designed and developed to compete with MySQL and Oracle database.

SQL Server supports ANSI SQL, which is the standard SQL (Structured Query Language) language. However, SQL Server comes with its own implementation of the SQL language, T-SQL (Transact-SQL).

T-SQL is a Microsoft propriety Language known as **Transact-SQL.** It provides further capabilities of declaring variable, exception handling, stored procedure, etc.

SQL Server Management Studio (SSMS) is the main interface tool for SQL Server, and it supports both 32-bit and 64-bit environments.

Microsoft and Sybase released version 1.0 in 1989. However, the partnership between these two ended in the early 1990s. Microsoft maintained ownership rights to the name SQL Server. Since the 1990s, subsequent versions of SQL Server have been released including SQL Server 2000, 2005, 2008, 2012, 2014, 2016 and 2017.

Key Components and Services of SQL Server

Database Engine: This component handle storage, Rapid transaction Processing, and Securing Data.

SQL Server: This service starts, stops, pauses, and continues an instance of Microsoft SQL Server. Executable name is sqlservr.exe.

SQL Server Agent: It performs the role of Task Scheduler. It can be triggered by any event or as per demand. Executable name is sqlagent.exe.

SQL Server Browser: This listens to the incoming request and connects to the desired SQL server instance. Executable name is sqlbrowser.exe.

SQL Server Full-Text Search: This lets user running full-text queries against Character data in SQL Tables. Executable name is fdlauncher.exe.

SQL Server VSS Writer: This allows backup and restoration of data files when the SQL server is not running. Executable name is sqlwriter.exe.

SQL Server Analysis Services (SSAS): Provide Data analysis, Data mining and Machine Learning capabilities. SQL server is integrated with R and Python language for advanced analytics. Executable name is msmdsrv.exe.

SQL Server Reporting Services (SSRS): Provides reporting features and decision-making capabilities. It includes integration with Hadoop. Executable name is ReportingServicesService.exe

SQL Server Integration Services (SSIS): Provided Extract-Transform and Load capabilities of the different type of data from one source to another. It can be view as converting raw information into useful information. Executable name is MsDtsSrvr

CHAPTER 2

PROPOSED SYSTEM

2.1 PROPOSED SYSTEM

1) The goal of this system is to provide the customer discount based on his/her intellect.
2) The shopkeeper now able to extend discount facility through this application.
3) Now the concept of flat discount will be eliminated

There are three modules or the users.

Login/Registration:

1) For the registration purpose, customer have to register in the application
2) For that, customer require an email id/ mobile number and he /she have to set a password
3) After that customer can login into the application using an email id/mobile number and password.
4) After successful login customer will redirect to dashboard page.

Customer:

1) Firstly, the customer has to download the application from the play store.

2) Then customer has to register on the application with the email id and the phone number.

3) Then the customer has to login into the application.

4) After the login, user will get the list of products.

5) Then after that user will select one of the products.

6) After selecting the product user will get the quiz.

7) Then the user will get the questions based upon the complexity level of questions and the price of the product.

8) As the higher price of product, the complexity of questions will be high.

9) As the user will give the more number of correct answers the user will get more and more discount.

10) Then the user will have to choose the correct option for the question within the limited time.

11) As customer give the correct answer, for each correct question customer can able to view the discount that he got through the question.

12) Every question has a parametrized time slot based on the complexity level of the question.

13)if the answer is correct then the user will get the discount based on the complexity level of the question

14) After that the user will get the next question.

15) If the user selected answer is wrong then user will redirect to the page and he will ger the discount based on the previous correct question

16) If the user has given the wrong answer of first question then he will not get any discount.

17) If the user can guess all the correct answers then user can able to get the large amount of discount on the mobile up to 80%.

18) The customer has two options either via online shopping or through the direct counter shopping.

Admin:

1) The admin is the user in the system who has all the access in the system.

2) The admin can add the shopkeepers/vender.

3) The admin can also add the product(mobile) in the system.

4) The admin can add the questions into the database, and according to that questions he can also add the options to that questions

5) The admin has the access to delete the product.

6) The admin can also add the users based on their access levels.

7) The admin can also delete the users of the system.

8) The admin can delete the shopkeeper or the vender.

9) The admin can also view the list of products according to the shopkeeper.

10) The admin can also check the number of active users.

11) The admin can update product price and also, he can update the product description.

2.2 OBJECTIVES OF SYSTEM

The intellect-based discount application will be having different users and objectives of the system are categorized according to type of user.

Customer:

- Customer should be able to register and login into the application
- Immediately after the login, the customer dashboard will display on the screen.
- The customer can able to view all the details, that how much amount of discount he/she gained.
- The customer can able to view the list of products.
- The customer can view details of the product and also the maximum discount that he/she can avail.
- The customer can also play the quiz to get the discount on mobile
- The customer can get up to 80% discount on the mobile.

- The customer can purchase the mobile, but he/she has to pay only final price which is calculated using discount price deducted from original price.
- The customer can get the mobile from billing counter or through the online shopping.

Admin:

- Admin is the user in the system which were having all the access in the system
- Admin can view the list of customers as well as shopkeepers.
- Admin can add/update/delete the customers as well as shopkeepers.
- Admin can also add/update/delete the products in the application.
- Admin can also add the questions in the quiz
- Admin can also set the complexity level of questions, so that according to that the discount will be offered.

- Admin can also add the access level to customer and as well as shopkeeper.
- Admin can also add the question category into the system
- Admin can view the list of users in the system.
- Admin can generate bill using the registered email id of the user.
- Admin is the super user of the application.

2.3 User Requirements

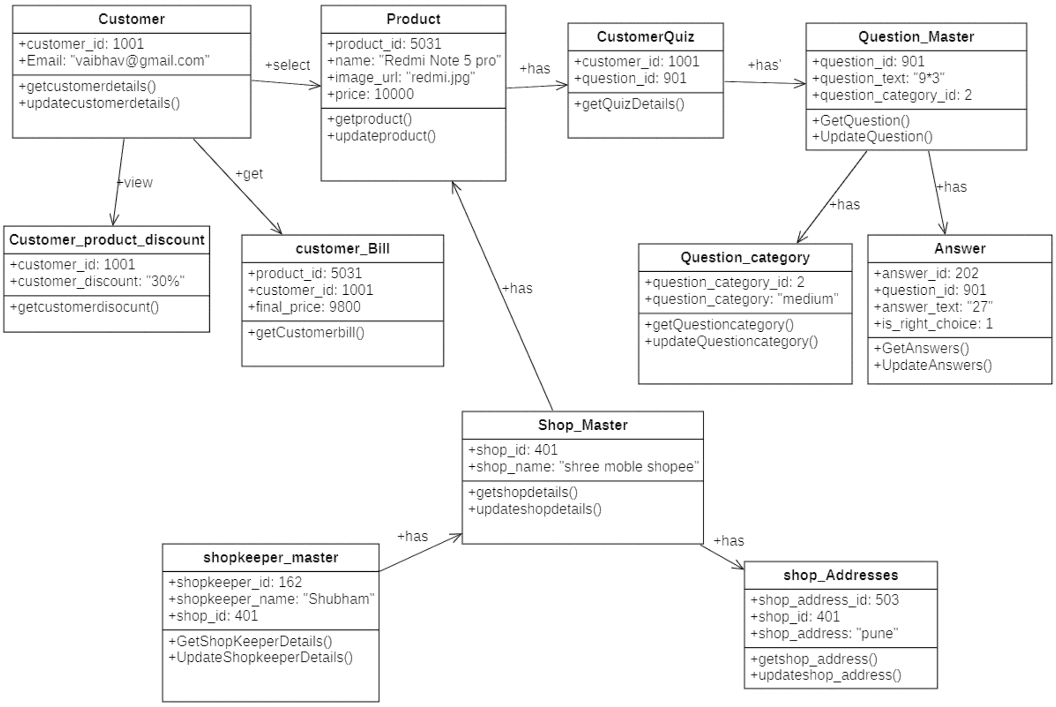
- Any user can able to login into the system using the email id/mobile number and password.
- Customer can able to get the discount based on his/her intellect.
- Customer can play the quiz and he/she can get more and more discount as he/she can guess the correct answers.
- Now the customer not bound through only flat discount.
- Customer has two purchase options in the application
- Either customer can choose online shopping or he can choose billing counter in the shop.
- There is a shopkeeper in this application. Which give discount to the customer.
- The shopkeeper can able to view the list of the products (mobile phones).
- The shopkeeper can able to view the details of the each and every product.
- The shopkeeper can able to update the product price and product description.

- The shopkeeper can able to view the discount on the product (mobile phones).
- There is an admin user also which is having the full access into the application
- Admin can able to view the list of users and list of shopkeepers.
- Admin can able to view the questions in the application.
- Admin can update the questions in the application.

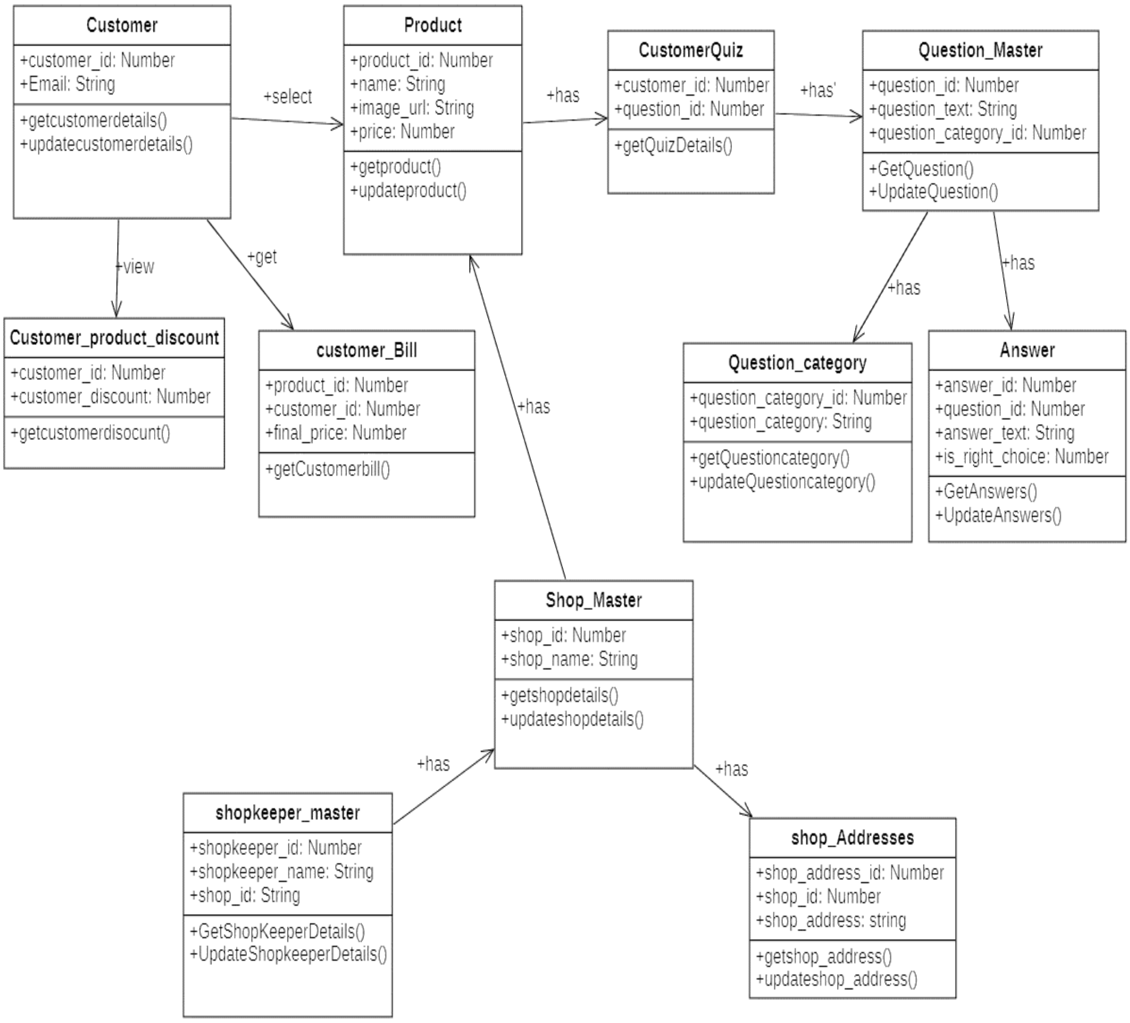
CHAPTER 3

ANALYSIS AND DESIGN

3.1 Object Diagram:

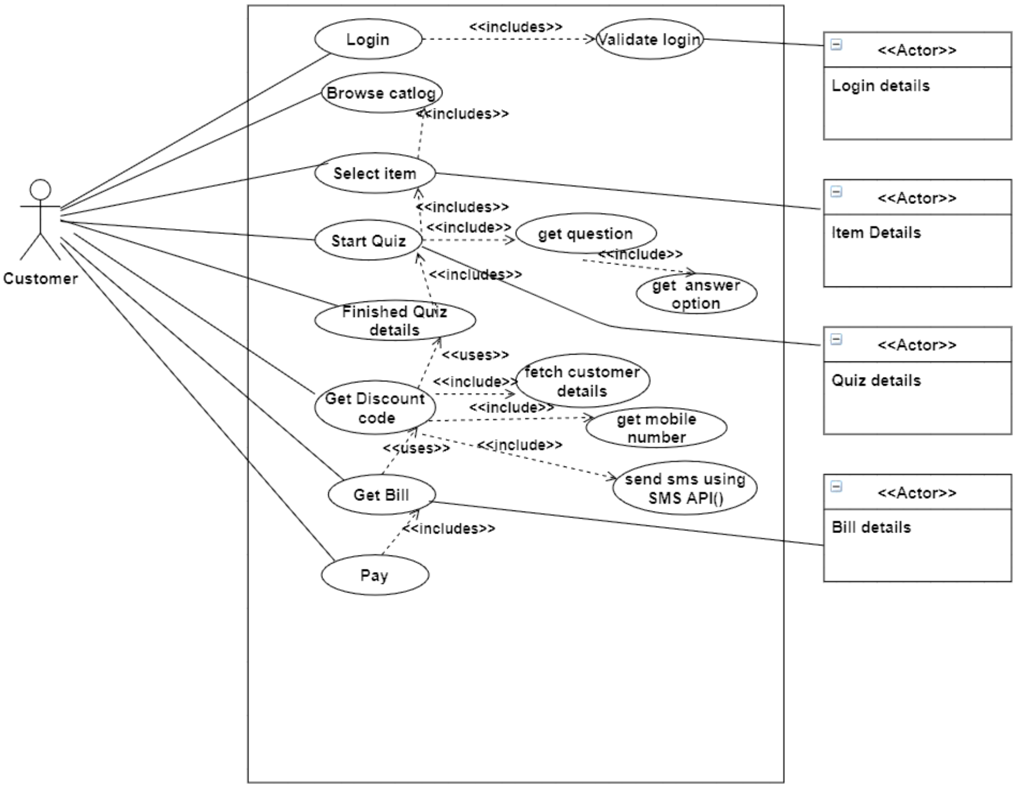


3.2 Class Diagram

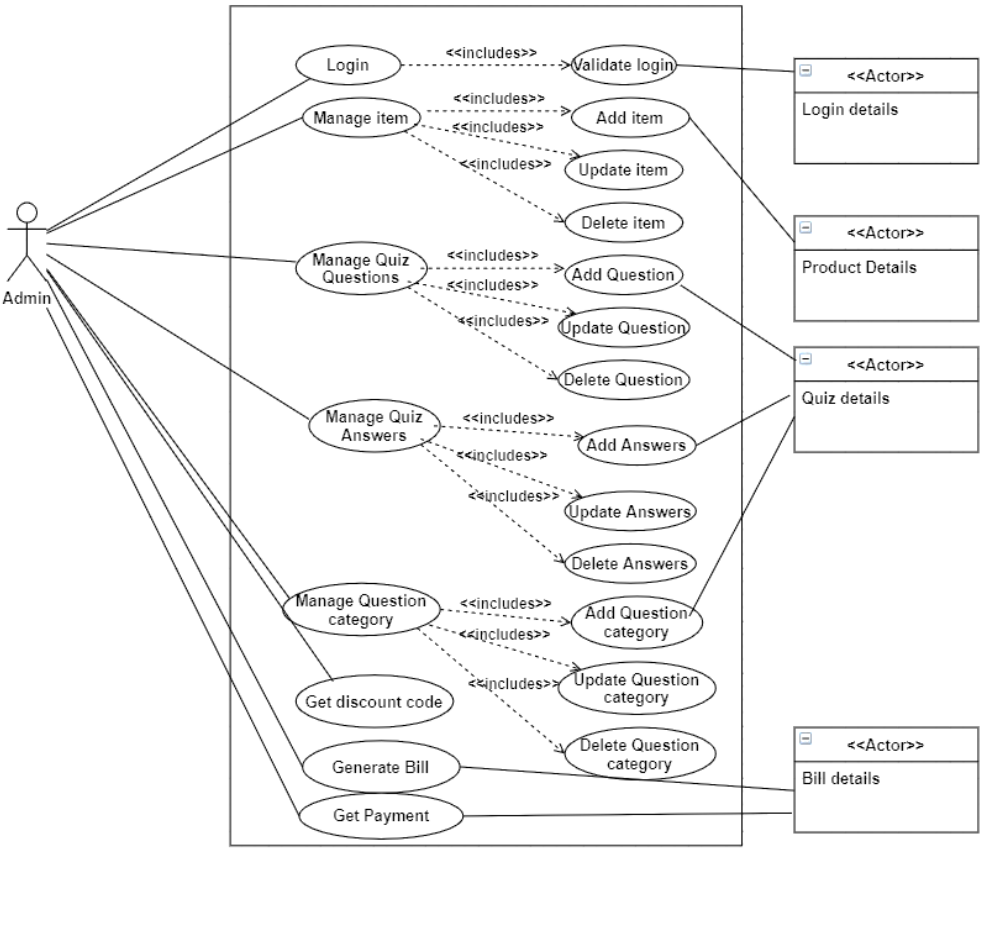


3.3 Use Case Diagram:

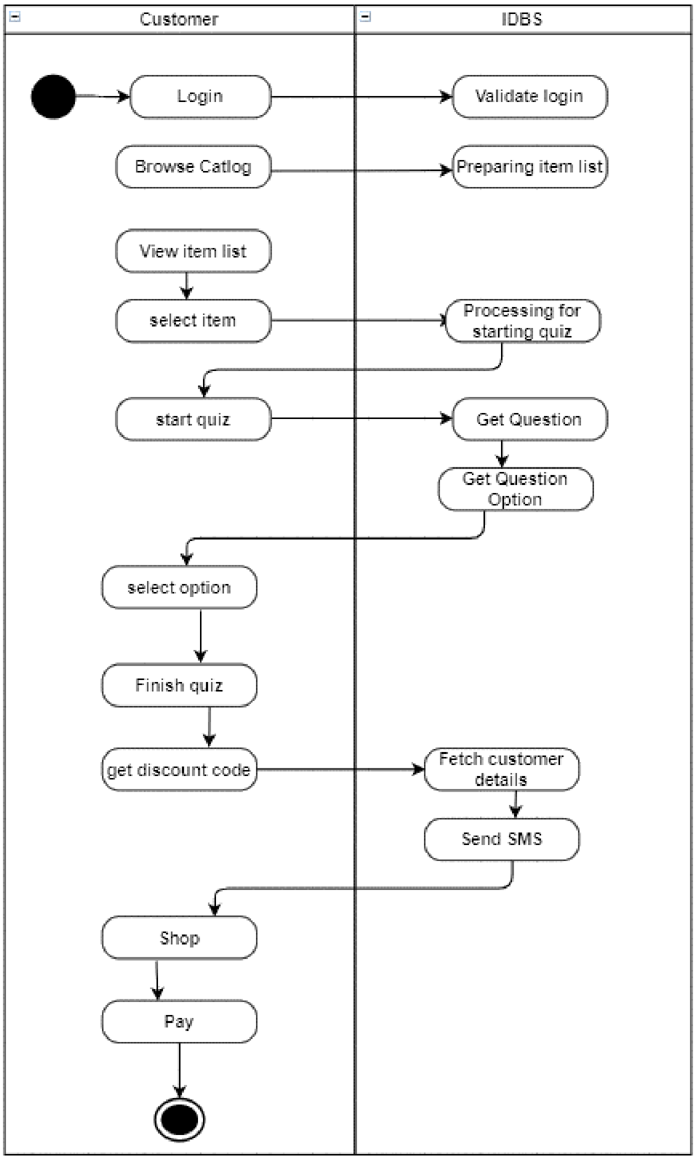
Customer User:



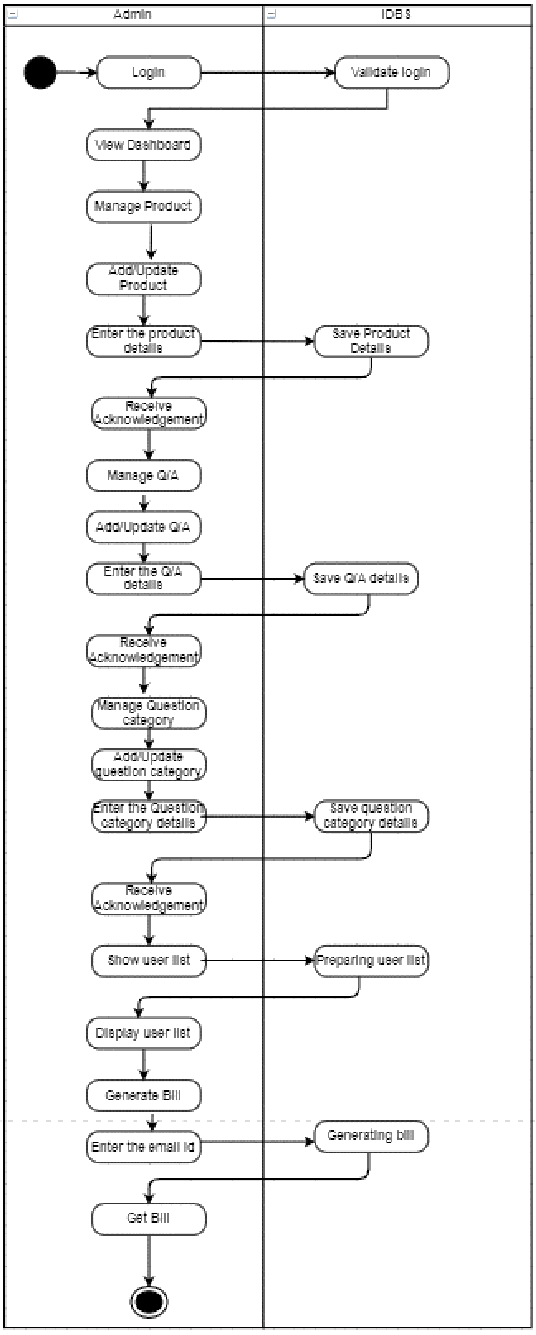
Admin User:


Admin
Login
<<includes>>
Validate login
Manage item
<<includes>>
Add item
<<includes>>
Update item
<<includes>>
Delete item
Manage Quiz Questions
<<includes>>
Add Question
<<includes>>
Update Question
<<includes>>
Delete Question
Manage Quiz Answers
<<includes>>
Add Answers
<<includes>>
Update Answers
<<includes>>
Delete Answers
Manage Question category
<<includes>>
Add Question category
<<includes>>
Update Question category
<<includes>>
Delete Question category
Get discount code
Generate Bill
Get Payment
<<Actor>>
Login details
<<Actor>>
Product Details
<<Actor>>
Quiz details
<<Actor>>
Bill details

3.4 Activity Diagram:

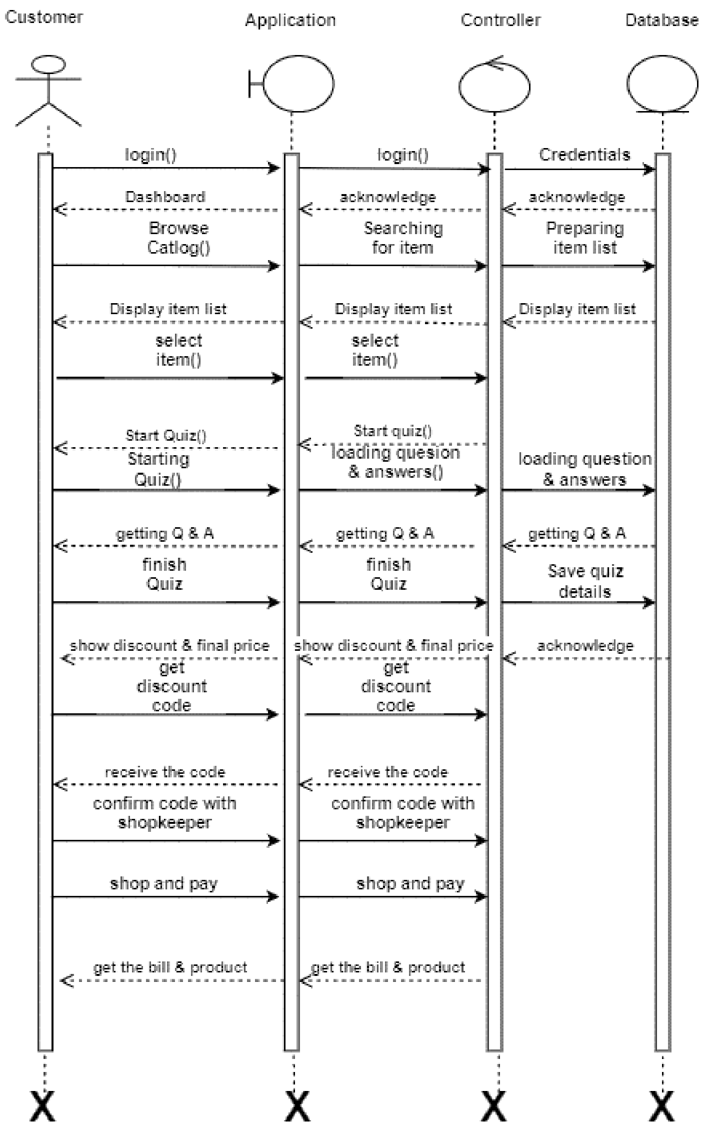


Admin Activity Diagram:



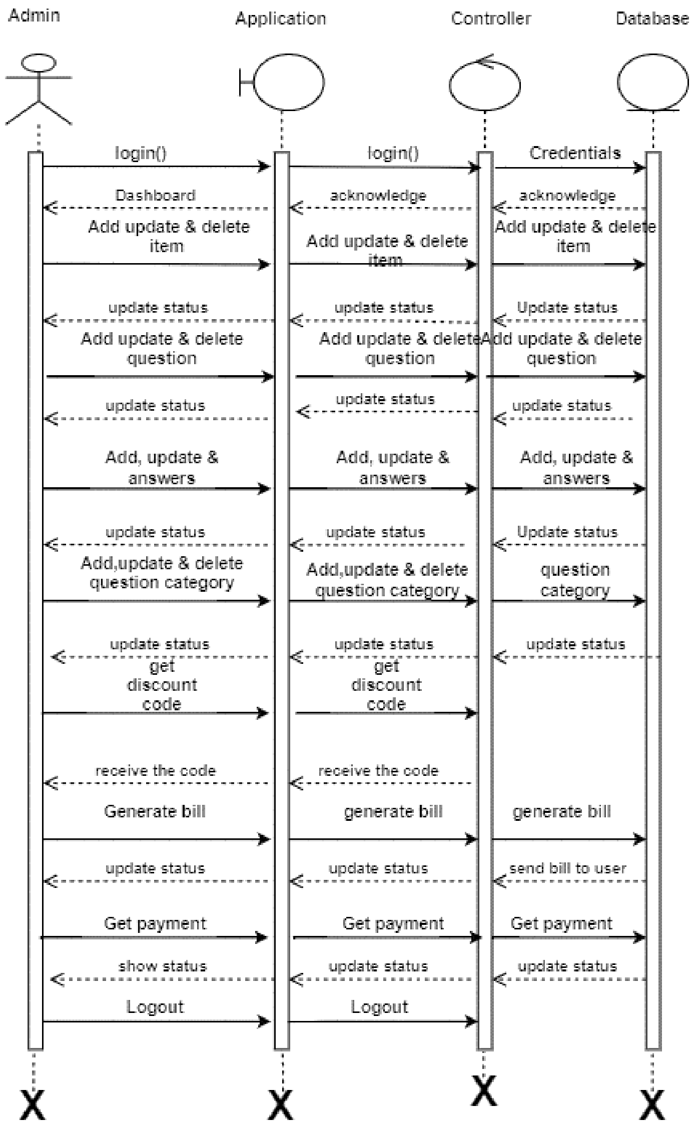
3.5 Sequence Diagram:

Customer User:

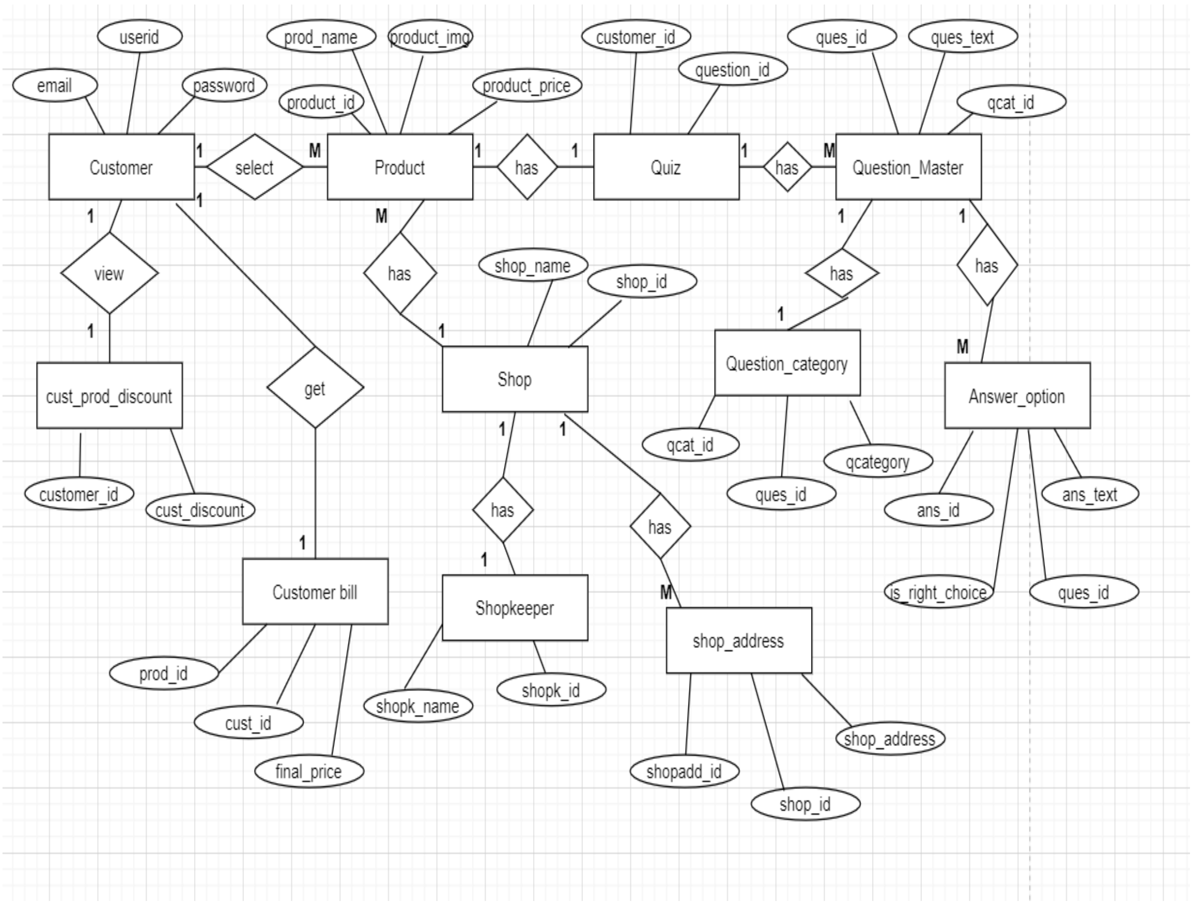


Sequence Diagram:

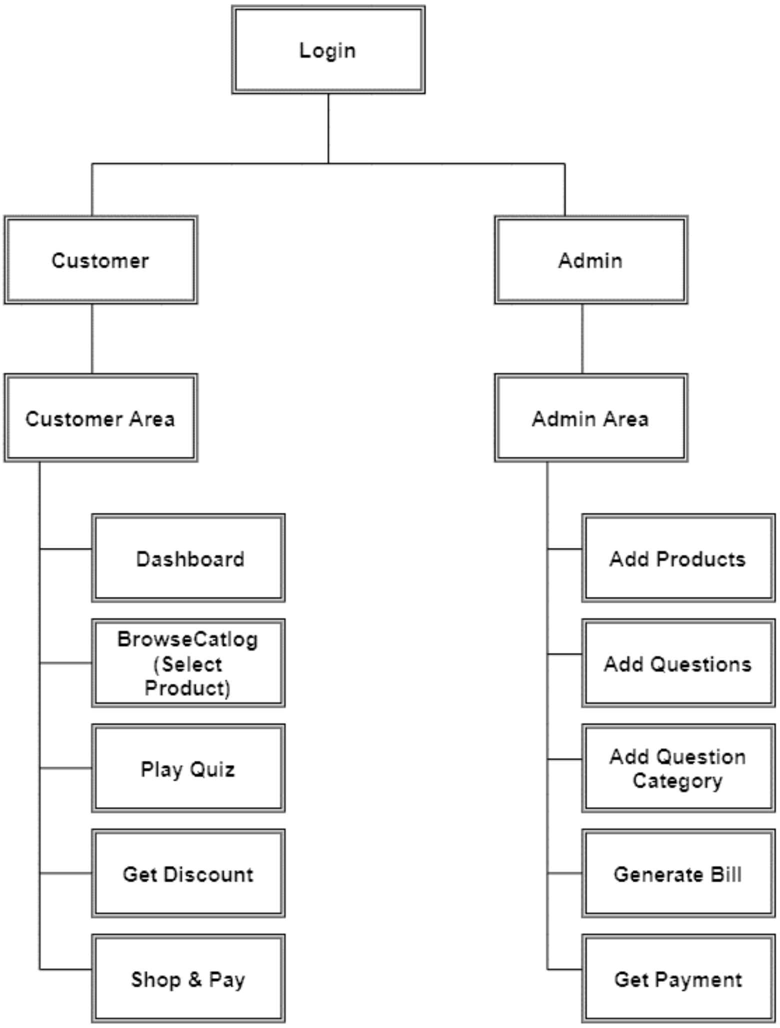
Admin User:



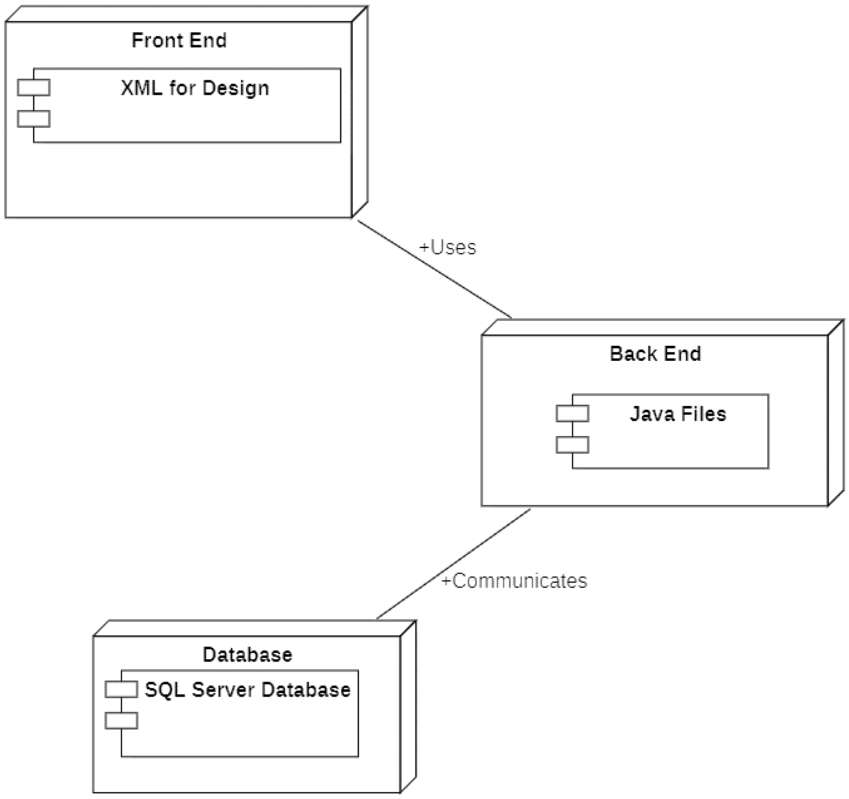
3.6 Entity Relationship Diagram:



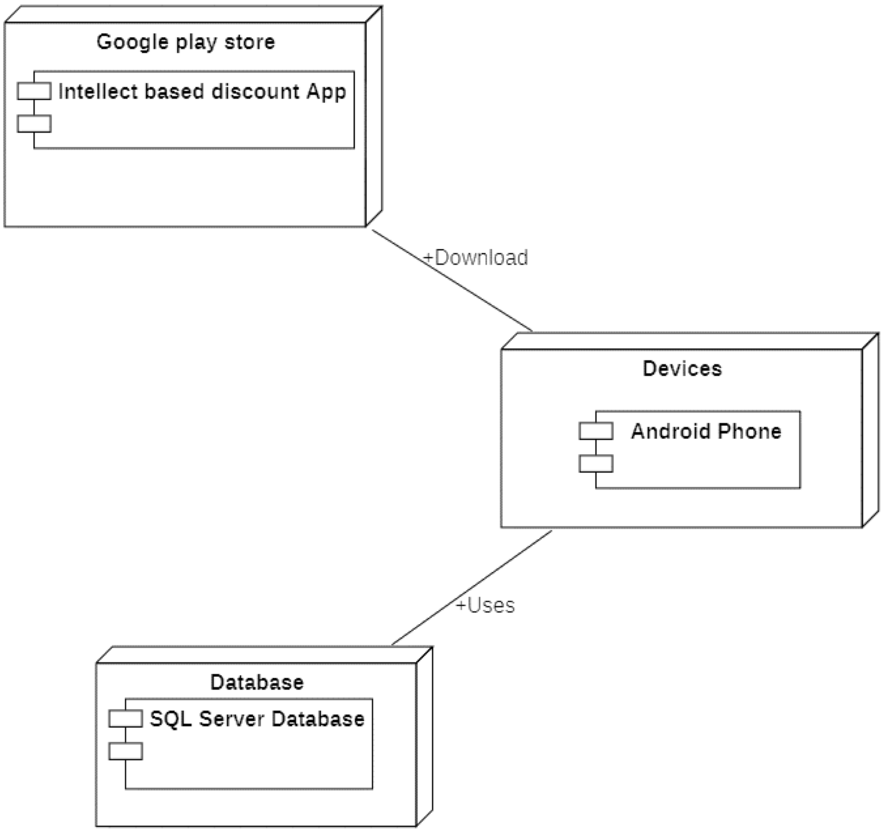
3.7 Module Hierarchy Diagram:



3.8 Component Diagram:



3.9 Deployment Diagram:



3.10 Module Specifications:

The system is divided into two modules. Below are the module specifications.

Customer:

1) Firstly, the customer has to download the application from the play store.

2) Then customer has to register on the application with the email id and the phone number.

3) Then the customer has to login into the application.

4) After the login, user will get the list of products.

5) Then after that user will select one of the products.

6) After selecting the product user will get the quiz.

7) Then the user will get the questions based upon the complexity level of questions and the price of the product.

8) As the higher price of product, the complexity of questions will be high.

9) As the user will give the more number of correct answers the user will get more and more discount.

10) Then the user will have to choose the correct option for the question within the limited time.

11) As customer give the correct answer, for each correct question customer can able to view the discount that he got through the question.

12) Every question has a parametrized time slot based on the complexity level of the question.

13)if the answer is correct then the user will get the discount based on the complexity level of the question

14) After that the user will get the next question.

15) If the user selected answer is wrong then user will redirect to the page and he will ger the discount based on the previous correct question

16) If the user has given the wrong answer of first question then he will not get any discount.

17) If the user can guess all the correct answers then user can able to get the large amount of discount on the mobile up to 80%.

18) The customer has two options either via online shopping or through the direct counter shopping.

Admin:

1) The admin is the user in the system who has all the access in the system.

2) The admin can add the shopkeepers/vender.

3) The admin can also add the product(mobile) in the system.

4) The admin can add the questions into the database, and according to that questions he can also add the options to that questions

5) The admin has the access to delete the product.

6) The admin can also add the users based on their access levels.

7) The admin can also delete the users of the system.

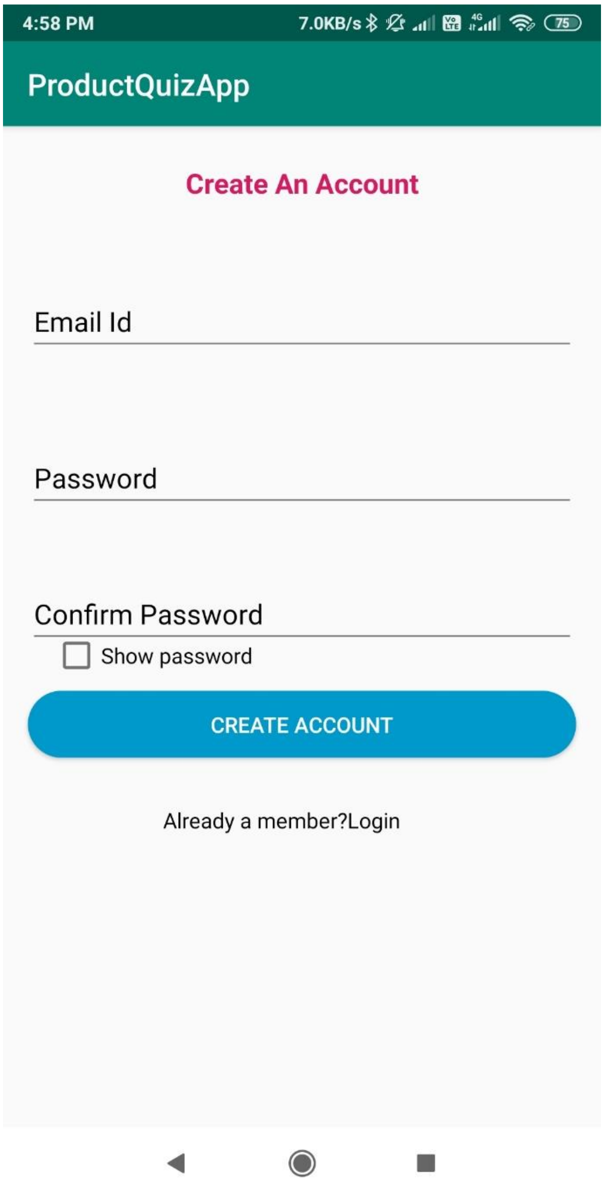
8) The admin can delete the shopkeeper or the vender.

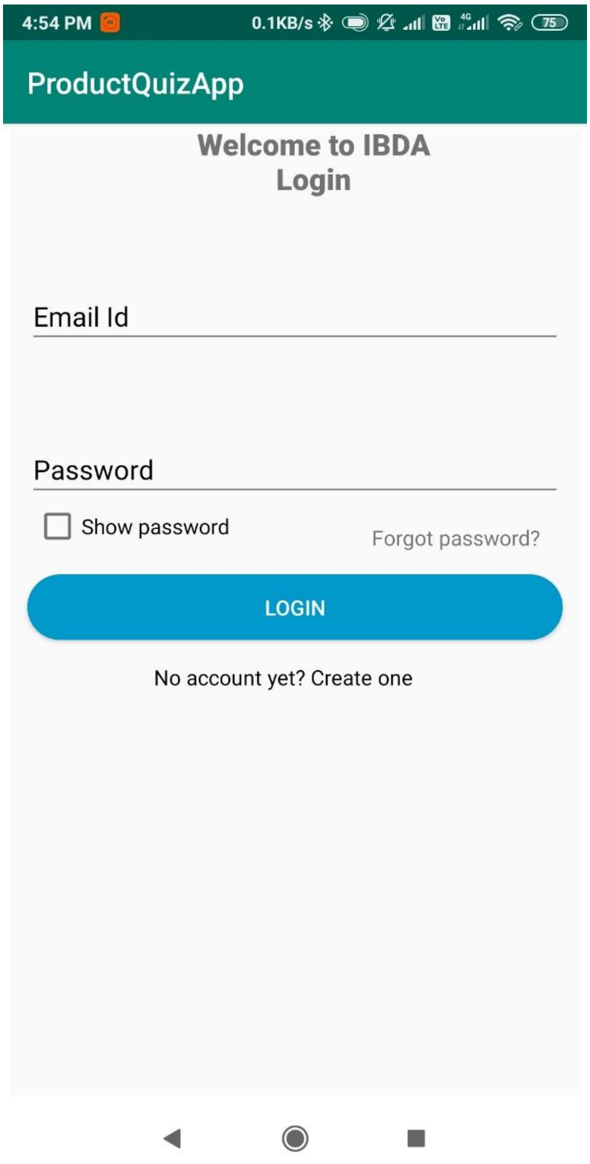
9) The admin can also view the list of products according to the shopkeeper.

10) The admin can also check the number of active users.

11) The admin can update product price and also, he can update the product description.

3.13 User Interface Design:





ProductQuizApp

Welcome to IBDA
Login

Email Id

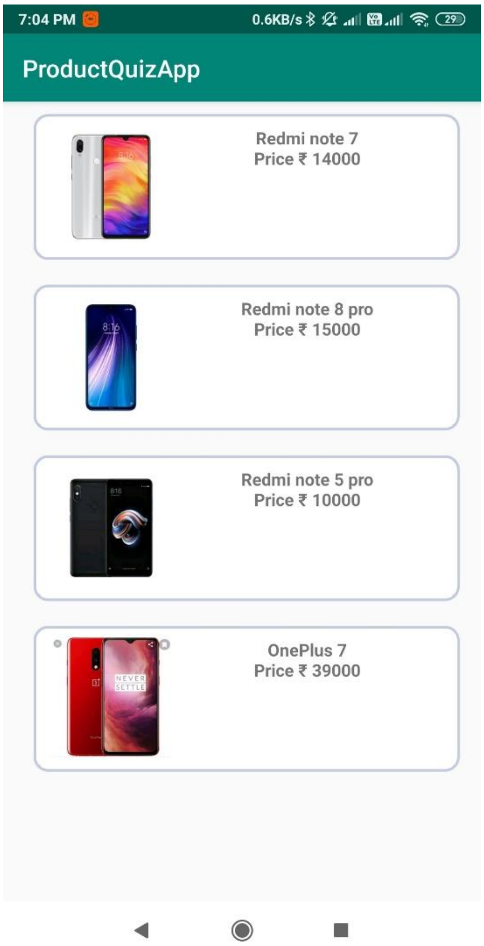
Password

□ Show password

Forgot password?

LOGIN

No account yet? Create one



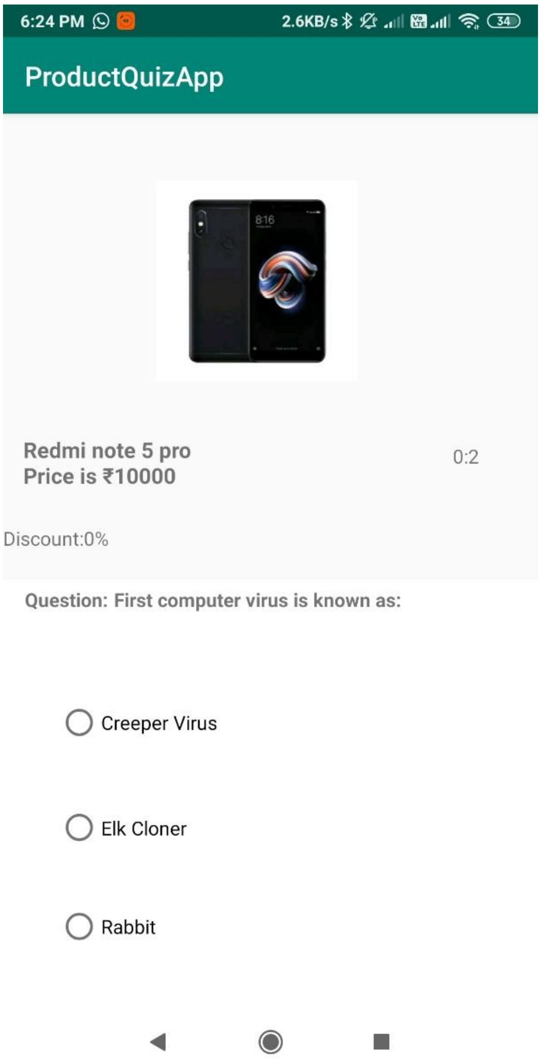
ProductQuizApp

Redmi note 7
Price ₹ 14000

Redmi note 8 pro
Price ₹ 15000

Redmi note 5 pro
Price ₹ 10000

OnePlus 7
Price ₹ 39000



ProductQuizApp

Redmi note 5 pro
Price is ₹10000

0:2

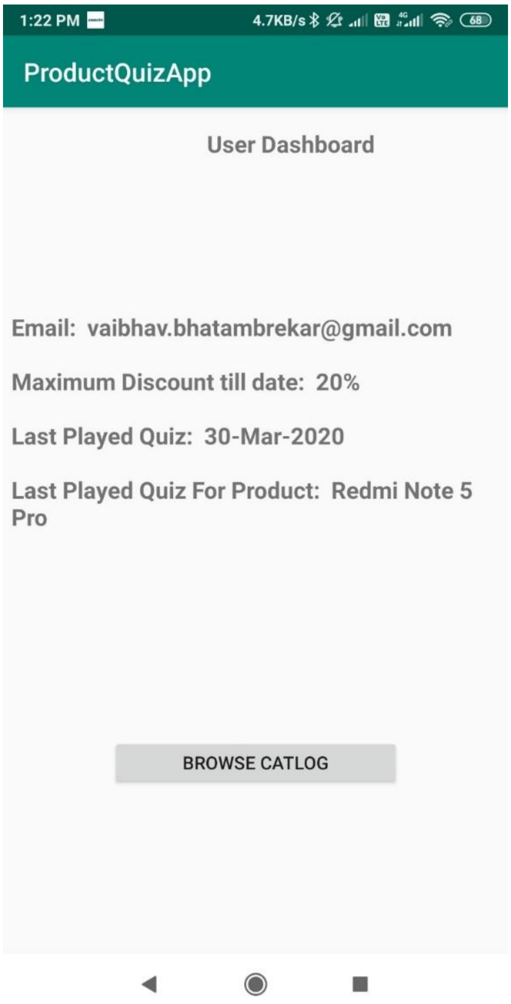
Discount:0%

Question: First computer virus is known as:

○ Creeper Virus

○ Elk Cloner

○ Rabbit



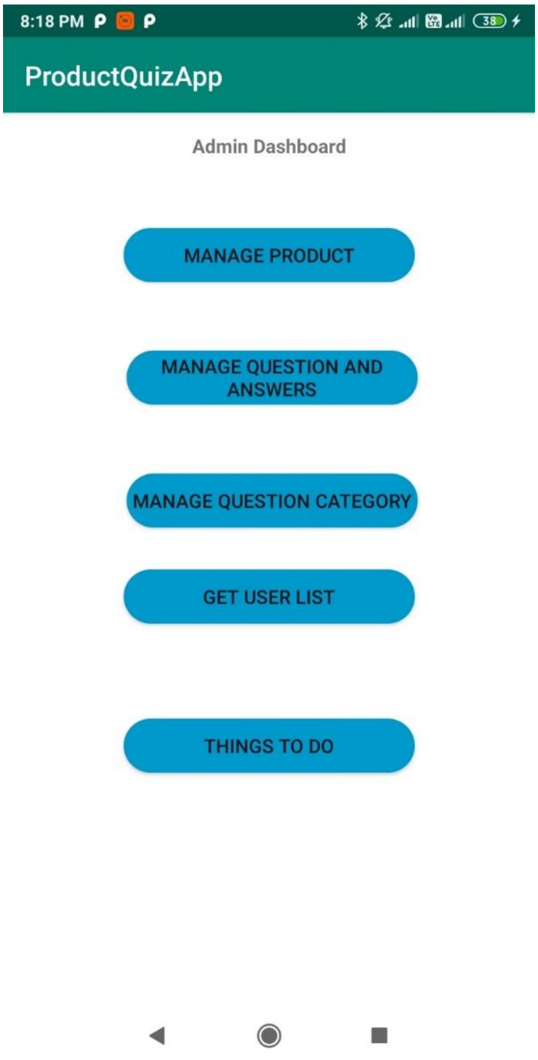
User Dashboard

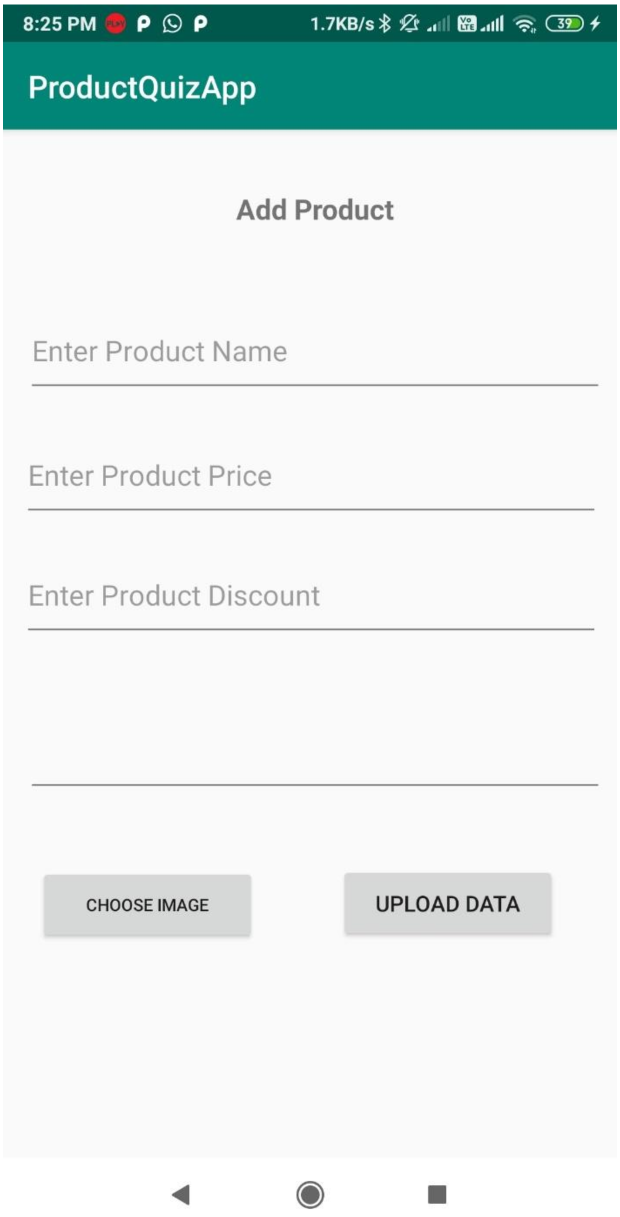
Email: vaibhav.bhatambrekar@gmail.com

Maximum Discount till date: 20%

Last Played Quiz: 30-Mar-2020

Last Played Quiz For Product: Redmi Note 5 Pro


8:18 PM
ProductQuizApp
Admin Dashboard
MANAGE PRODUCT
MANAGE QUESTION AND ANSWERS
MANAGE QUESTION CATEGORY
GET USER LIST
THINGS TO DO

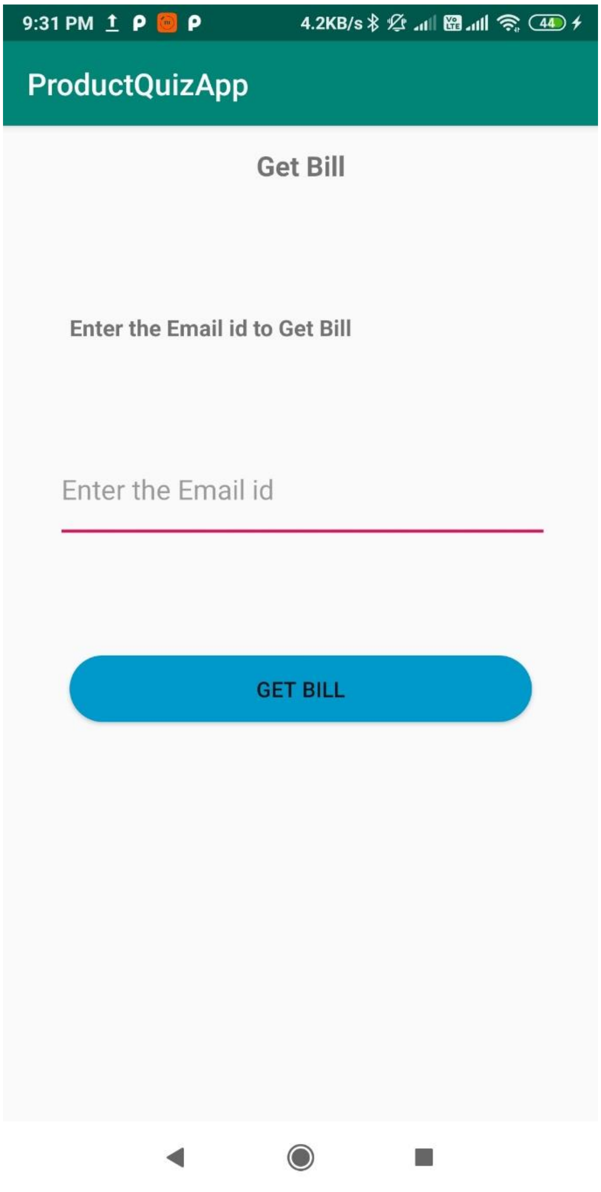


Add Product

Enter Product Name

Enter Product Price

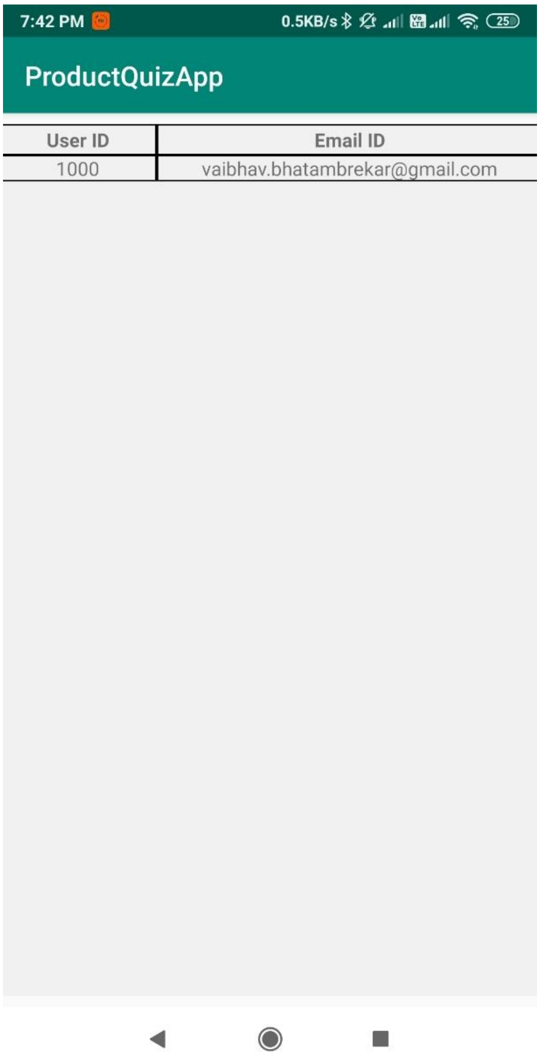
Enter Product Discount



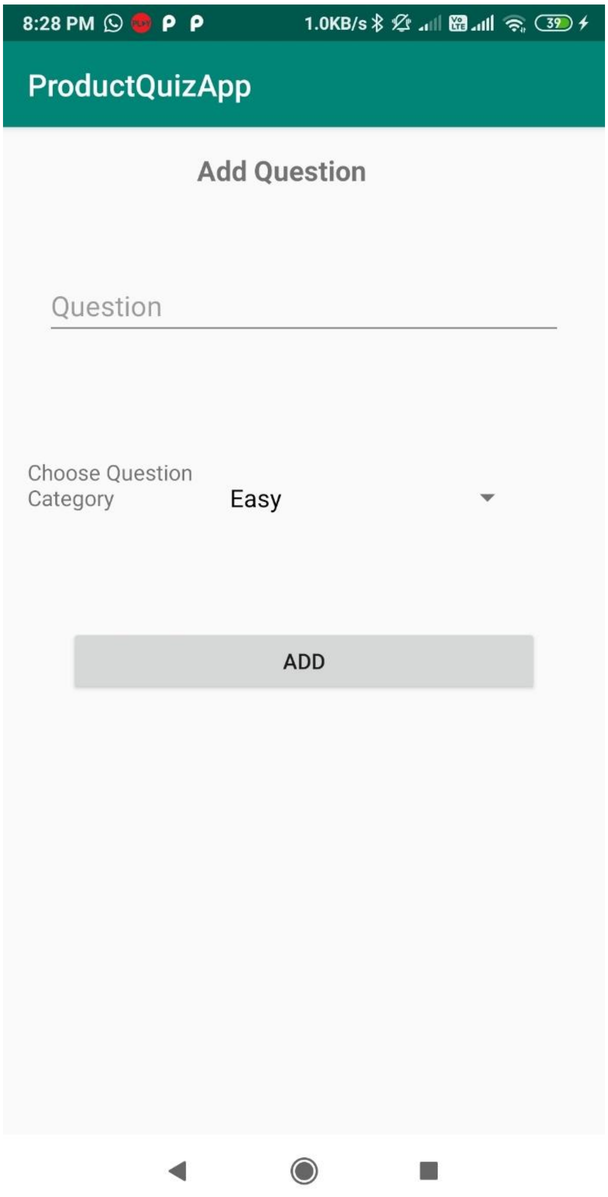
Get Bill

Enter the Email id to Get Bill

Enter the Email id



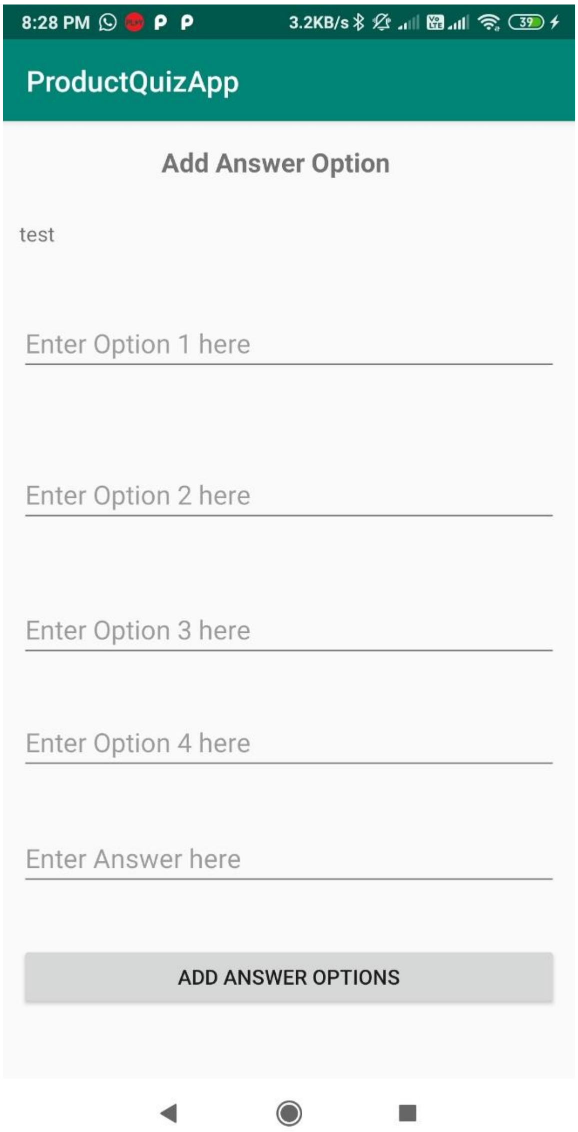
User ID	Email ID
1000	vaibhav.bhatambrekar@gmail.com



Add Question

Question

Choose Question Category Easy



8:28 PM

3.2KB/s

ProductQuizApp

Add Answer Option

test

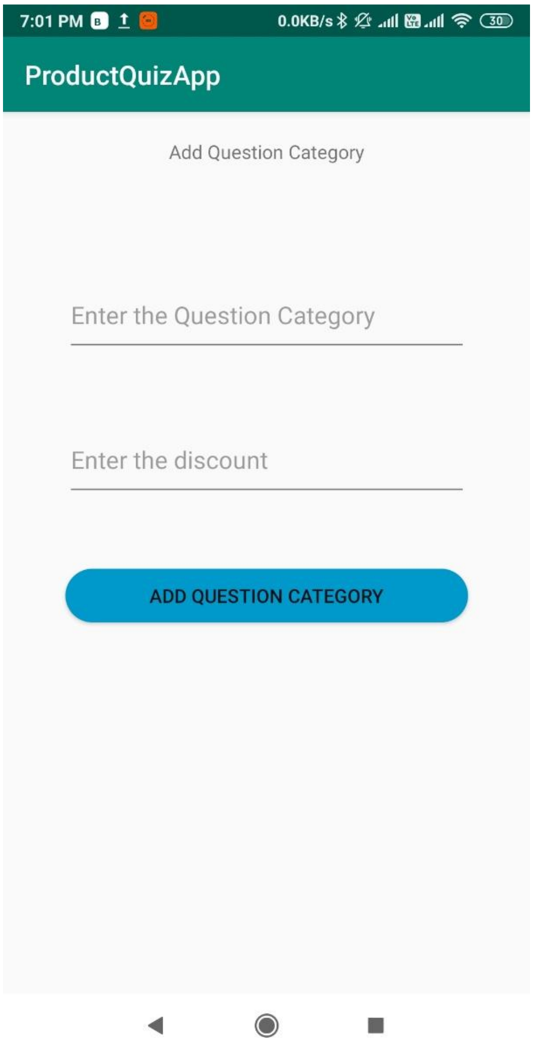
Enter Option 1 here

Enter Option 2 here

Enter Option 3 here

Enter Option 4 here

Enter Answer here


7:01 PM
0.0KB/s
ProductQuizApp
Add Question Category
Enter the Question Category
Enter the discount
ADD QUESTION CATEGORY

3.14 Data Dictionary:

Sr no	Field Name	Data Type	Width	Description
1	user_id	Integer	10	User ID
2	Email	Varchar	50	Email Address(id)
3	Password	Varchar	50	Password Address(id)
4	Product_id	Integer	10	Product ID
5	Product_name	Varchar	50	Product Name
6	product_img_url	Varchar	255	Product Image Path
7	Price	Integer	10	Price of Product
8	question_id	Integer	10	Question ID
9	question_text	Varchar	255	Question Text
10	answer_id	Integer	10	Answer ID
11	answer_text	Varchar	255	Answer Text
12	is_right_choice	Integer	10	flag Yes/No
13	question_category_id	Integer	10	Question Category ID
14	question_category_text	Varchar	30	Question Category

Sr no	Field Name	Data Type	Width	Description
15	shop_id	Integer	10	Shop ID
16	shop_name	Varchar	50	Name of Shop
17	shop_address_id	Integer	10	Shop Address ID
18	shop_address_text	Varchar	255	Details of Shop Address
19	shopkeeper_id	Integer	10	Shopkeeper ID
20	shopkeeper_name	Varchar	50	Name of Shopkeeper
21	final price	Integer	10	Final Price after discount
22	customer_discount	Integer	10	Maximum discount of customer
23	answered_right	Varchar	30	Yes/No
24	quiz_play_date	Date		Date of Quiz played

3.15 Table Specifications:

User:

Sr no	**Field Name**	**Data Type**	**Width**	**Constraint**
1	user_id	Integer	10	Primary Key
2	email	Varchar	50	Not Null
3	password	Varchar	50	Not Null

Product Master:

Sr no	**Field Name**	**Data Type**	**Width**	**Constraint**
1	Product_id	Integer	10	Primary Key
2	Product_name	Varchar	50	Not Null
3	product_img_url	Varchar	255	Not Null
4	Price	Integer	10	Not Null

Question Master:

Sr no	Field Name	Data Type	Width	Constraint
1	question_id	Integer	10	Primary key
2	question_text	Varchar	255	Not Null
3	question_category_id	Integer	10	Foreign Key

Answer Master:

Sr no	Field Name	Data Type	Width	Constraint
1	answer_id	Integer	10	Primary Key
2	question_id	Integer	10	Foreign Key
3	answer_text	Varchar	255	Not null
4	is_right_choice	Integer	10	Not null

Question Category:

Sr no	**Field Name**	**Data Type**	**Width**	**Constraint**
1	question_category_id	Integer	10	primary key
2	question_category_text	Varchar	30	Not null

Customer_Product Discount:

Sr no	**Field Name**	**Data Type**	**Width**	**Constraint**
1	user_id	Integer	10	Foreign Key
2	user_discount	Integer	10	Not null
3	discount_date	Date		Not null

Customer_Quiz:

Sr no	Field Name	Data Type	Width	Constraint
1	user_id	Integer	10	Foreign key
2	question_id	Integer	10	Foreign key
3	Product_id	Integer	10	Foreign key
4	answered_right	Varchar	30	Not null
	quiz_play_date	Date		Not null

Shop_Master:

Sr no	Field Name	Data Type	Width	Constraint
1	shop_id	Integer	10	Primary key
2	shop_name	Varchar	255	Not null

Shopkeeper_Master:

Sr no	Field Name	Data Type	Width	Constraint
1	shopkeeper_id	Integer	10	Primary Key
2	shopkeeper_name	Varchar	255	Not null
3	shop_id	Integer	10	Foreign Key

ShopAddress:

Sr no	Field Name	Data Type	Width	Constraint
1	shop_address_id	Integer	10	Primary Key
2	shop_address_text	Varchar	255	Not null
3	shop_id	Integer	10	Foreign Key

Customer Bill:

Sr no	Field Name	Data Type	Width	Constraint
1	Bill_no	Integer	10	Foreign Key
2	user_id	Integer	10	Foreign Key
3	product_id	Integer	10	Not null
4	final_price	Float	10	Not null
5	bill_date	Date		Not null
6	discount	Integer	10	Not null

3.16 Test Procedures:

Unit Testing

In unit testing, initially the focus is on each component individually, ensuring that it functions properly. Various unit-testing strategies include testing of module interfaces, local data structures, boundary conditions and all independent paths.

Integration Testing

The Integration Testing is on design and the construction of the software architecture. The individually tested components are integrated as a complete software package. Black Box test case design techniques are the most prevalent during integration. Top-down integration testing is an incremental approach to construction of program structure, thereby verifying major control or decision points early in the test process. Bottom up integration testing begins construction and testing with atomic modules i.e. components at the lowest levels in the program structure Regression testing is re-execution of some subset of tests that have been already been conducted to ensure that changes have not propagated unintended side effects.

- **Validation Testing**

As the software has been integrated a set off higher order tests are conducted. Validation criteria, established during requirement analysis must be tested. Validation testing provides final assurance that the software meets all the requirements and the specifications, functional, behavioural and performance requirements. Black-box testing techniques are exclusively used during validation.

Performance Testing

Performance Testing is designed to test the run-time performance of the software within the context of an integrated system. It occurs throughout all the steps in the testing process. These are often coupled with stress testing and usually require both hardware and software implementation. That is often necessary to measure resources utilization.

Security Testing:

Security testing attempts to verify that protection mechanisms built into a system will protect from improper penetration.

- **Stress Testing:**

Stress Testing executes a system in a manner that demands resources in abnormal quantity, frequency or volume.

Alpha/Beta Testing:

Customer conducts the alpha test at the developer's site. The software is used in a natural setting with the developer recording errors and usage problem. Alpha tests are conducted in a controlled environment Beta Testing is conducted at one or more customer sites by the end user of the software. The 'beta test' is a live application of the software in an environment that cannot be controlled by developer.

The Customer records all the problems that are encountered during beta testing and result of the problem report is given to software engineer to make modifications and then prepare the release of the software product to entire customer base.

Test Cases:

Test Case ID	Scenario to test	Steps to perform	Expected result	Actual result	Pass/Fail
TCLogin1	Login into the application as customer or administrator	1.Open login page of application 2. Enter the valid username 3. Enter the valid password. 4.Click on the login button.	Application should accept the valid username and valid password entered by the user should direct user to dashboard page of the application	Login to application is successful	Pass

Test Case ID	**Scenario to test**	**Steps to perform**	**Expected result**	**Actual result**	**Pass/Fail**
TCLogin1.1	Login into application	1.Open login page of application 2.Enter the invalid username 3.Enter the valid password. 4.Click on the login button.	Application should not accept the invalid username and application should throw message "Invalid username or password".	Login denied with appropriate message.	Pass

Test Case ID	**Scenario to test**	**Steps to perform**	**Expected result**	**Actual result**	**Pass/Fail**
TCLogin 1.2	Login into application	1.Open login page of application 2. Enter the valid username 3. Enter the invalid password. 4.Click on the login button.	Application should not accept the invalid username and application should throw message "Invalid username or password". Please enter the valid username or password.	Login denied with appropriate message.	Pass

Test Case ID	**Scenario to test**	**Steps to perform**	**Expected result**	**Actual result**	**Pass/Fail**
TC Login 1.3	Login into application	1.Open login page of application 2. Enter the valid invalid username 3. Enter the invalid password. 4.Click on the login button.	Application should not accept the invalid username and application should throw message "Invalid username or password".	Login denied with appropriate message.	Pass

Test Case ID	**Scenario to test**	**Steps to perform**	**Expected result**	**Actual result**	**Pass/Fail**
TC Login 1.4	Login into application	1.Open login page of application 2. Enter the valid invalid username 3. Enter the invalid password. 4.Click Back button.	Application should display the alert box to exit the application or continue	Application should display the alert box to exit the application or continue	Pass

Test Case ID	**Scenario to test**	**Steps to perform**	**Expected result**	**Actual result**	**Pass/Fail**
TCREGCUST 2	Registration for customer	1. click on no account yet create one on the login form 2. Enter the valid details and click on the create account button	Customer must create in the system and should able to login	Customer gets created in the system and customer is able to login	Pass

Test Case ID	**Scenario to test**	**Steps to perform**	**Expected result**	**Actual result**	**Pass/Fail**
TCREGCUST 2.1	Registration for customer	1. click on no account yet create one on the login form 2. Enter the invalid details like (used email id) and click on the create account button	validation error must occur like email id is already used or email id is not in proper format	Validation error has been occurred. And also, in second time the email id is already used error has been shown	Pass

Test Case ID	**Scenario to test**	**Steps to perform**	**Expected result**	**Actual result**	**Pass/Fail**
TCADDITEM 3	Add Product into the system	1.click on add item on the admin dashboard 2. Fill all the product details and click on add	must show the toast message that product added successfully	shown the message product added successfully	Pass

Test Case ID	**Scenario to test**	**Steps to perform**	**Expected result**	**Actual result**	**Pass/Fail**
TCSHOWUSER 4	Show user list to admin	1.click on the show user list button	Must show the user list in the tabular format	Showing the user list in tabular format	Pass

Test Case ID	**Scenario to test**	**Steps to perform**	**Expected result**	**Actual result**	**Pass/Fail**
TCGENBILL 5	Generate bill	1. click on things to do button in the admin dashboard 2. Enter the email id of registered customer so that system can generate bill	Must show the generated bill to admin which contains Bill No, product name, discount, final price to user.	System is able to show all the details	Pass

CHAPTER 4

USER MANUAL

4.1 User Manual

The user manual is prepared reflexively because it is an item that must accompany every system.

This manual will help user to navigate through the application and help user out whenever there is any trouble while using the application.

The main goal of the application is to give the discount to customer the discount based on his/her intellect.

Because if we go to shop and purchase the product then either we get flat discount or we get 5 to 10 % discount.

For that also we have to demand to shopkeeper then shopkeeper will either give the discount or maybe not.

Using this application, we are eliminating the concept of flat discount or any other discount.

In this application user has to play quiz to get the discount on the product. As user will give the more correct answers, he/she will get the more discount.

In this application user can see list of products, as well as user can view the dashboard in which user able to see last played quiz date, maximum discount gets, last quiz played for the product name.

As the user play quiz for one product. User will not allow to play the quiz again for 6 months.

In the quiz there are different types of questions and having the different complexity level and based on that complexity level user will get the discount.

So, this is the short description about application.

Now we see in detail.

Information about system:

For Customer:

Intellect Based Discount Application has following features:

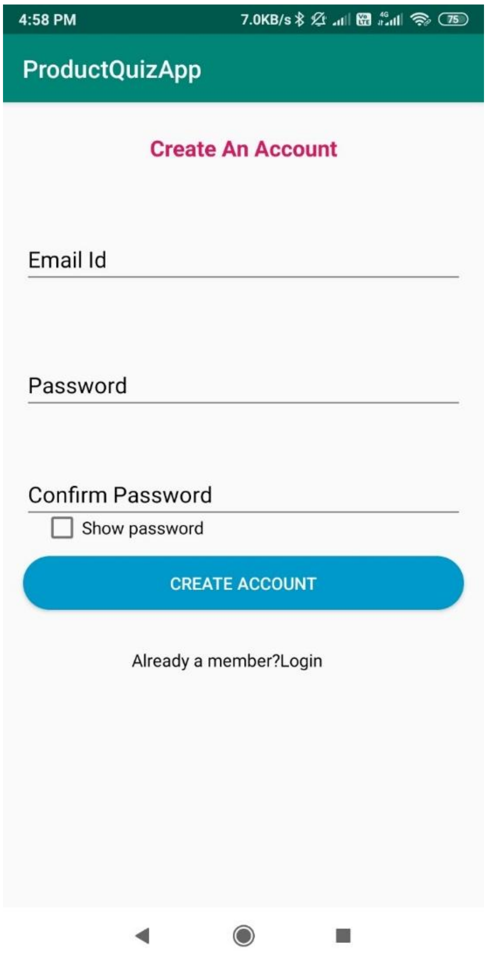
- Registration
- Dashboard
- Browse for Product (Show list of products)
- Select the Product
- Play Quiz
- Get Discount
- Shop and Pay

For Admin:

- Add Product/update product/delete product
- Add Questions/update question/delete question.
- Add Answer option/update option/delete option
- Add Question Category/update question category.
- Get user list
- Things to do.

For Customer

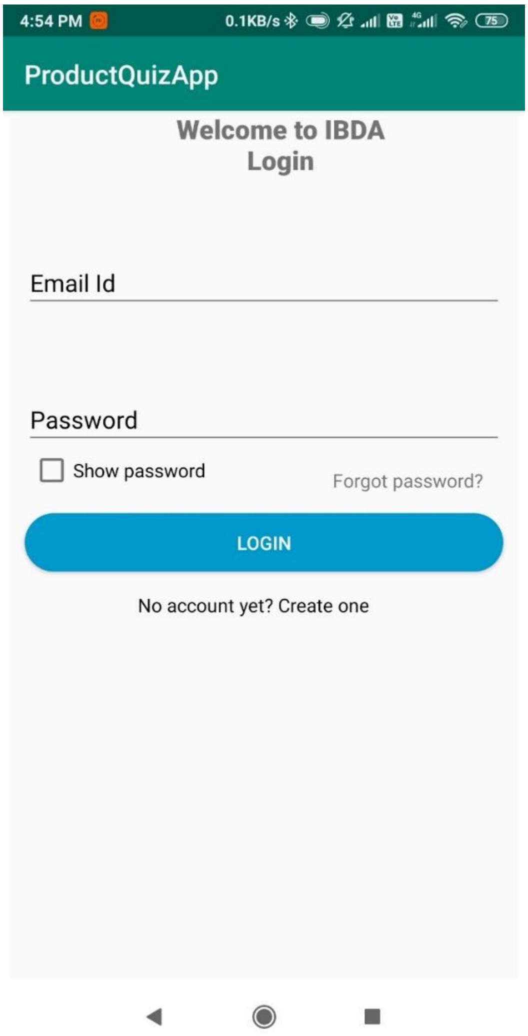
Registration:



For the registration user need to enter the Email Id and password.

Once the user is registered. User can login into system.

Login:



4:54 PM

0.1KB/s

ProductQuizApp

Welcome to IBDA
Login

Email Id

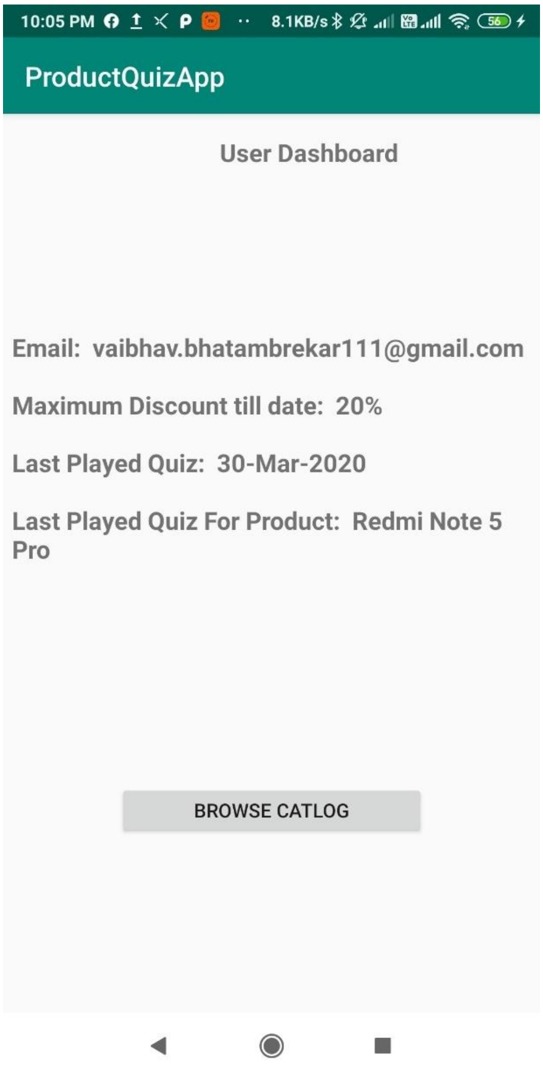
Password

□ Show password

No account yet? Create one

For login into the system user has enter the valid Email and password. After the successful login user will redirect to user dashboard page.

Dashboard:



User Dashboard

Email: vaibhav.bhatambrekar111@gmail.com

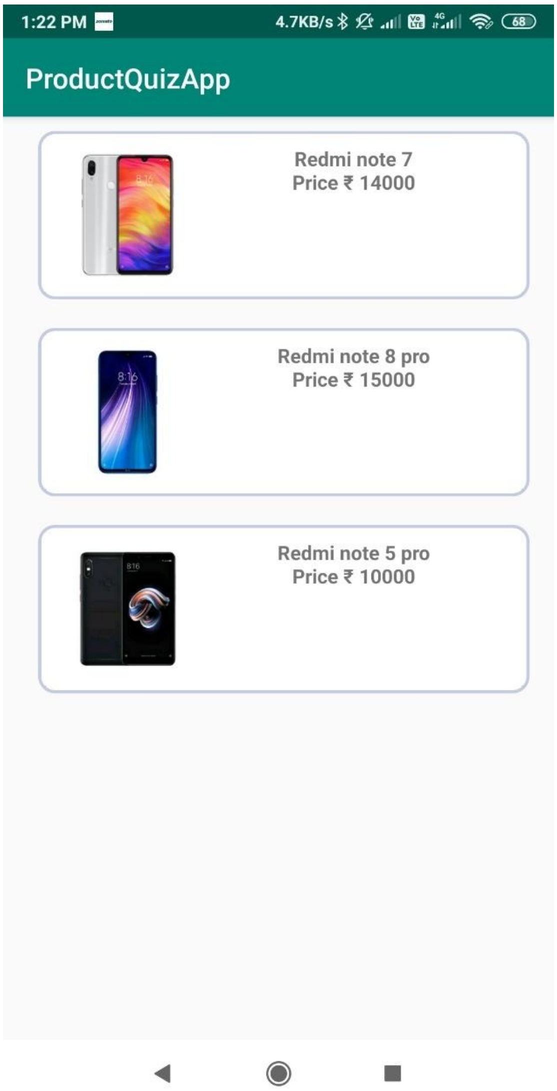
Maximum Discount till date: 20%

Last Played Quiz: 30-Mar-2020

Last Played Quiz For Product: Redmi Note 5 Pro

In the user dashboard user can able to see maximum discount, last quiz played date, the product name for which last quiz played.

Browse for product:



1:22 PM 4.7KB/s 68

ProductQuizApp

Redmi note 7
Price ₹ 14000

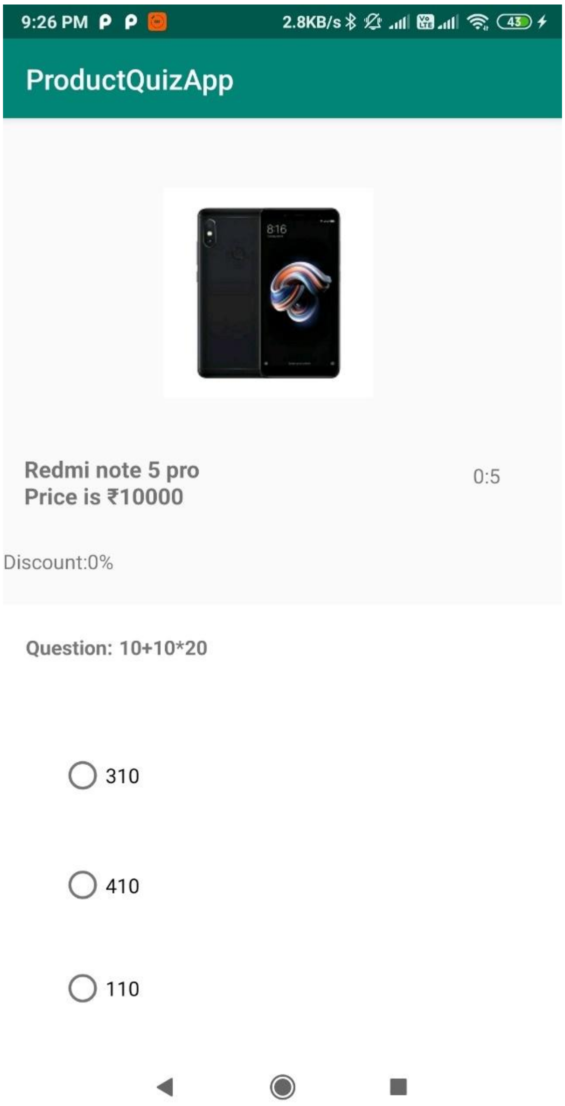
Redmi note 8 pro
Price ₹ 15000

Redmi note 5 pro
Price ₹ 10000

Above page shows list of products in the system.

After selecting the product, we can start the quiz.

Play Quiz:



9:26 PM

ProductQuizApp

Redmi note 5 pro
Price is ₹10000

0:5

Discount:0%

Question: 10+10*20

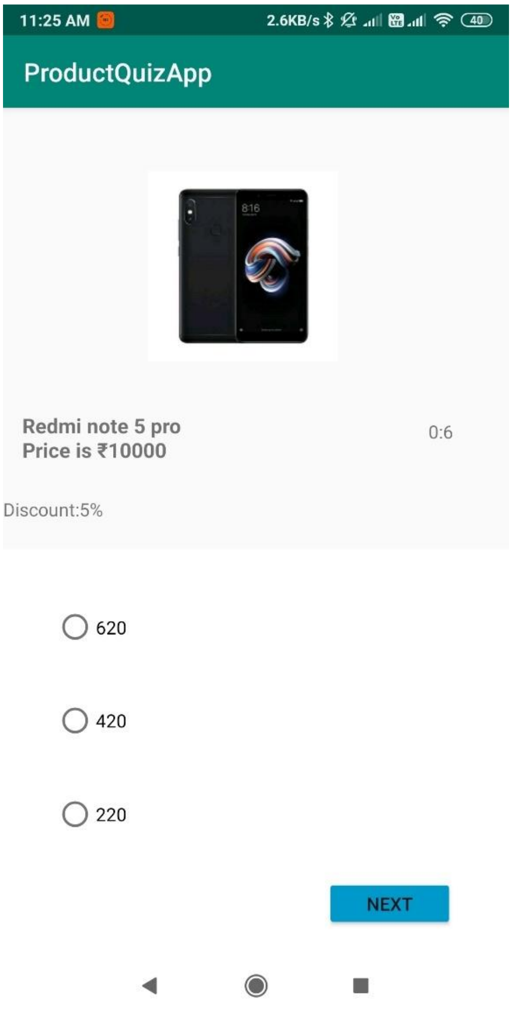
○ 310

○ 410

○ 110

Quiz page has question and for that question having the 4 answer options. For each question having a particular timer depending upon the complexity of question. And also, for each question having the different discount depending upon the complexity of the question.

User has to select one option and then click to next to load the next question. As many as user will give the correct answer user will get more discount.



ProductQuizApp

Redmi note 5 pro
Price is ₹10000

0:6

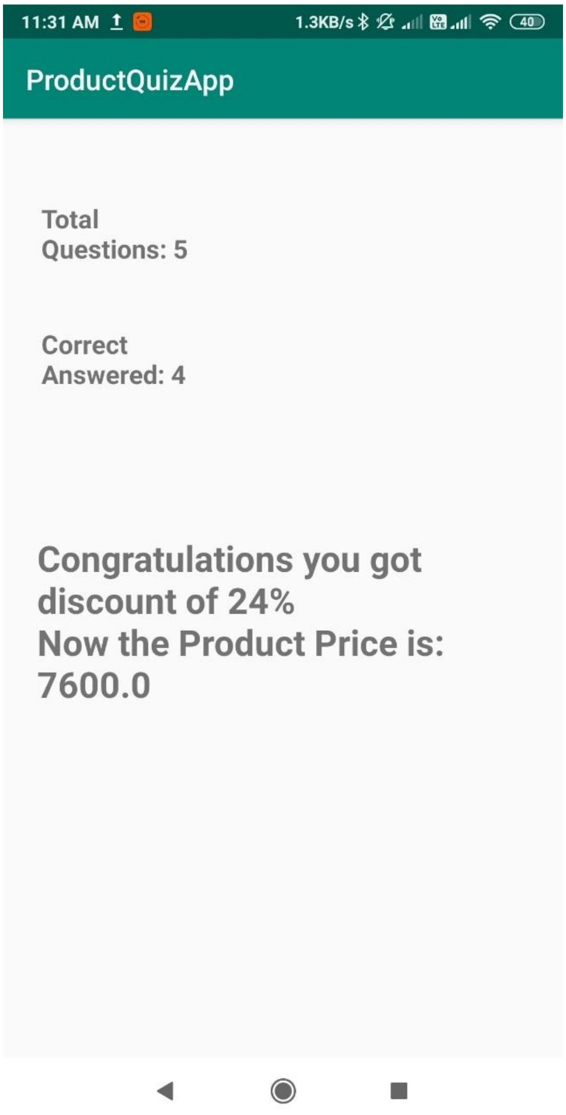
Discount:5%

○ 620

○ 420

○ 220

As the user will give the wrong answer or else the questions are finished then user will redirect to the discount page.



Total
Questions: 5

Correct
Answered: 4

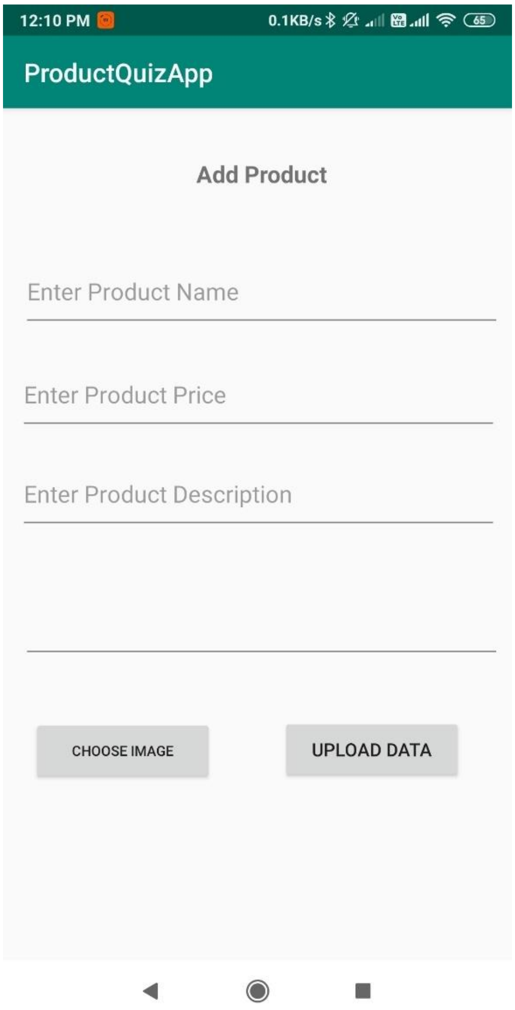
Congratulations you got discount of 24%
Now the Product Price is: 7600.0

For the Admin user:

Here in the system the admin is the user who has all the rights for insert, update, delete the data into the system.

Let’s see the admin features one by one

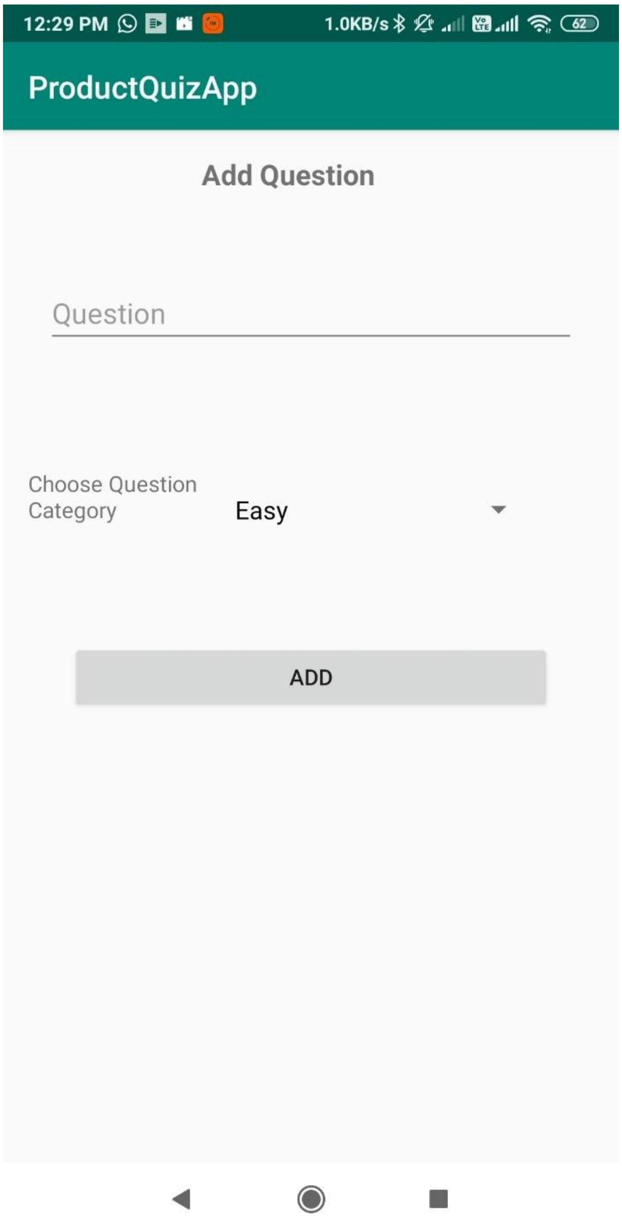
Add Product:



For Adding the Product into the system admin has to enter the product name, product price, product description and also admin has to select the product image which we can upload into the firebase database.

For the successful insertion of the product admin will get toast message that product added successfully and admin will redirect to the product list page.

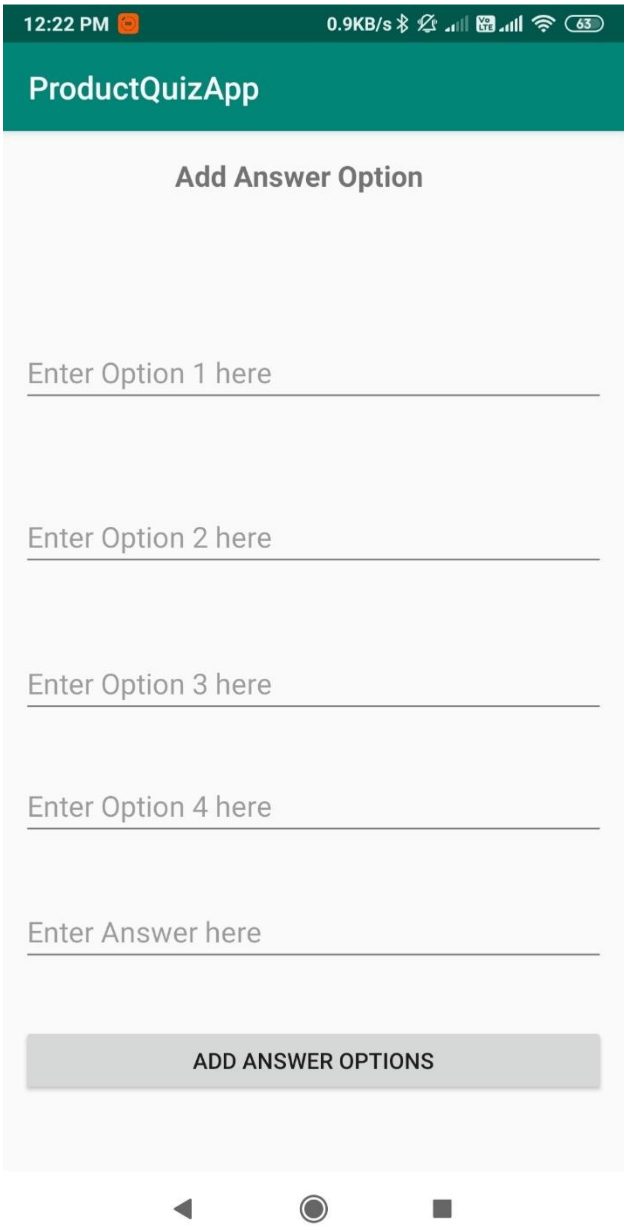
Add Question:



Question

Choose Question Category Easy

For adding the question into the system admin go to the add question page and then admin enter the question into the question textbox and then admin select the question category and then click on add then admin will redirect to add answer options page.



ProductQuizApp

Add Answer Option

Enter Option 1 here

Enter Option 2 here

Enter Option 3 here

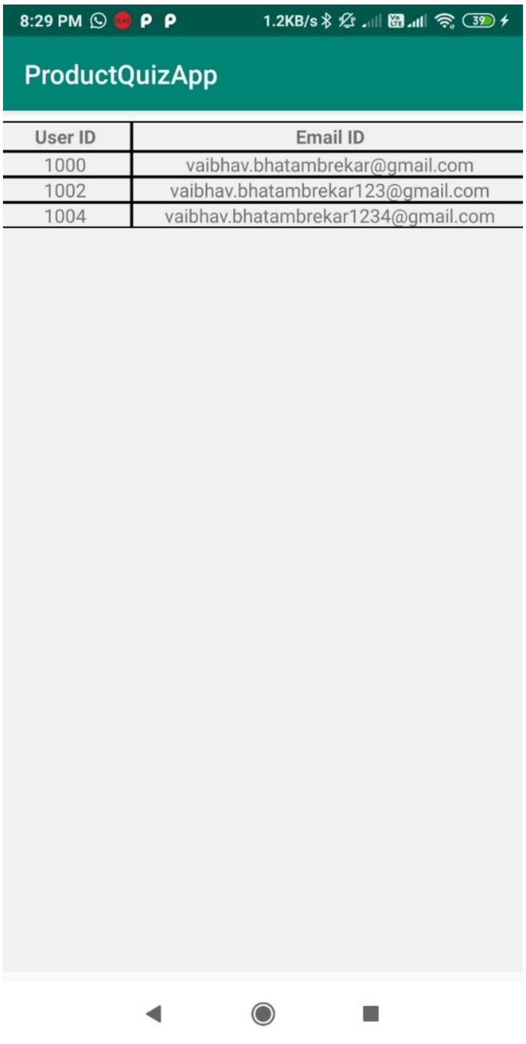
Enter Option 4 here

Enter Answer here

In this page admin has to enter the answer options and correct answer so that system will verify the correct answer.

After adding the answer option, the admin will get toast message that Question and answer option added successfully.

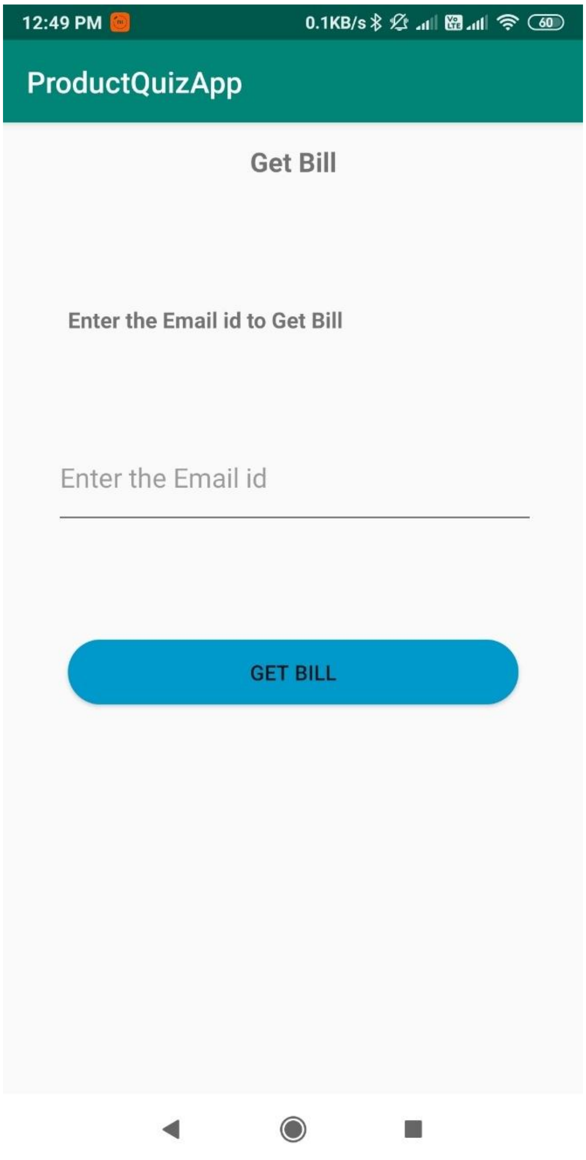
Get User List:



User ID	Email ID
1000	vaibhav.bhatambrekar@gmail.com
1002	vaibhav.bhatambrekar123@gmail.com
1004	vaibhav.bhatambrekar1234@gmail.com

The Get user list page shows the list of Registered users in the system.

Things to Do (Generate Bill):



Get Bill

Enter the Email id to Get Bill

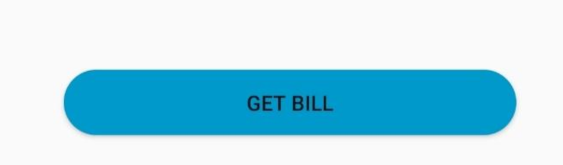
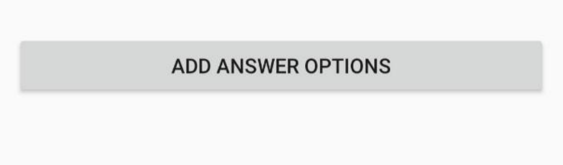
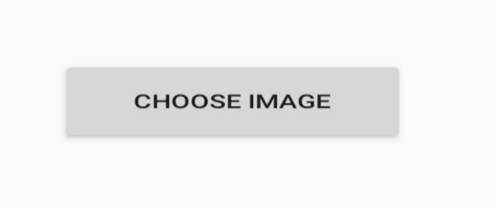
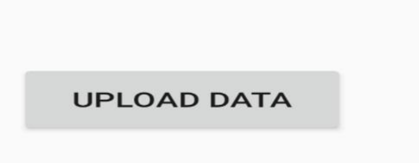
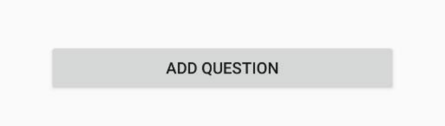
Enter the Email id

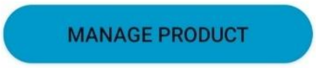
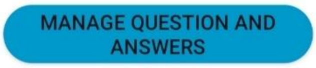
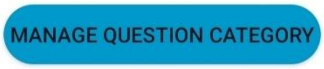
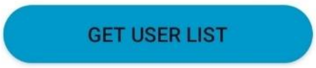
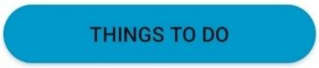
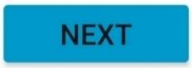
For generating the bill user has to click on Things to do button

After opening the Get Bill page admin has to enter the Email id, then click on Get bill button, then system will generate bill and then shows to the admin.

After generating the bill user can get that bill from admin. So, in this way the user manual is explained.

4.2 Operational Manual

Buttons	**Use**
	To generate Bill
	To Add Answer Options
	Choose image from gallery
	To add the product into system
	To add question into system

	To go to Add Product Page.
	To go to add question page
	To go to add question category page.
	To get user registered user list.
	To go to get bill page.
	To load the next question in the quiz

4.3 Program Specifications:

Module: Dashboard

Purpose: To show Details of user

Input: User ID

Output: Member details are show on page.

Module: Browse Catlog (Browse Product)

Purpose: Shows the list of Products (Mobile phones)

Input: Click on button

Output: Product List will Display on page.

Module: Select the Product

Purpose: To start the quiz

Input: Click on button

Output: Page will redirect to quiz

Module: Play Quiz

Purpose: Display Multiple choice questions

Input: Select any one answer

Output: Verify the answer if correct get discount or else

redirect to quiz details page.

Module: Get Discount

Purpose: To show final price and get discount code

Input: Mobile number

Output: Get the discount code and verify from admin.

Module: Shop and Pay

Purpose: To confirm product shopping

Input: Discount code

Output: Product will add into cart.

Module: Add Product

Purpose: To Add product into system

Input: product name, product price, product image

Output: Show Message product added successfully.

Module: Add Questions

Purpose: Add quiz questions into database

Input: question, question category

Output: Show Message Question added successfully.

Module: Add Answer option

Purpose: Add quiz questions answers into database

Input: answer, options

Output: Show Message Answers added successfully.

Module: Add Question Category

Purpose: Add quiz questions category into database

Input: category

Output: Show Message Question category added successfully.

Module: Generate Bill

Purpose: Generate bill for customer

Input: discount code and user id

Output: Show Message bill generated successfully.

Module: Shop and pay

Purpose: To confirm purchase of product.

Input: click on button

Output: Show Message product added to users' cart successfully.

Drawbacks and limitations:

As the database is large. It requires multiple administrators to keep it up to date.

The applications need to be more secure as it has large data.

The online payment is not there in the application.

Enhancements:

Primarily this application is for limited shops, further it will be available for more number of shops.

The GUI improvements are needed to make application more responsive and better.

The search for product option can be provided.

The email alert functionality will be created to get alert to users that discounts are available for user.

Feedback system will be provided in the application.

Conclusion:

All the requirements are gathered and implemented according to the user or shopkeepers.

The system includes:

- Add product
- Customer user registration.
- User Dashboard
- Admin Dashboard
- Play quiz
- Get discount
- Get the user list.

The application is tested successfully.

As we know that any project even on completion requires constant improvements and changes which gives way for release of new version. We made this application user friendly.

The application is user friendly. And easy to use for any kind of user.

For developing this application, we used the technologies such as android studio, SQL Server database which is widely used now a days in android application development. And its use increasing day by day.

Bibliography

Sites:

http://www.stackoverflow.com

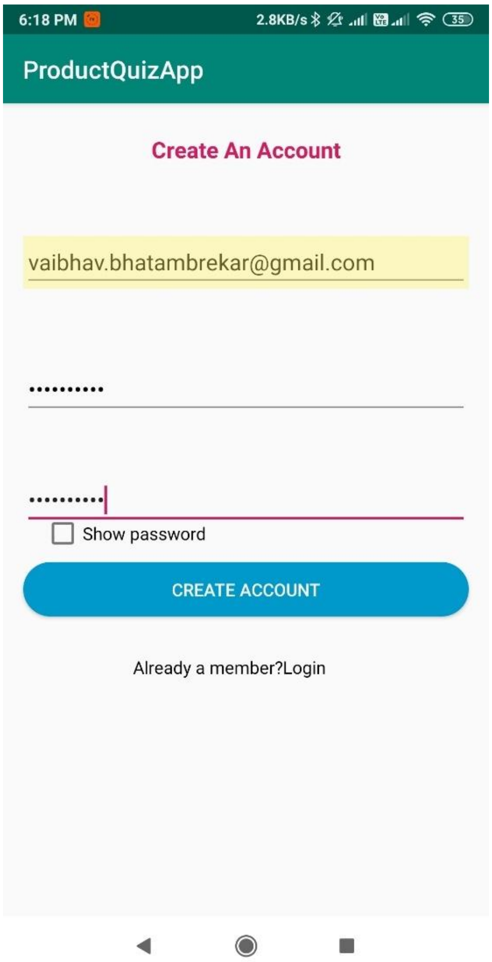
http://www.google.com

http://www.youtube.com

https://developer.android.com/studio.

ANNEXURE 1

USER INTERFACE SCREEN



6:18 PM

2.8KB/s

ProductQuizApp

Create An Account

vaibhav.bhatambrekar@gmail.com

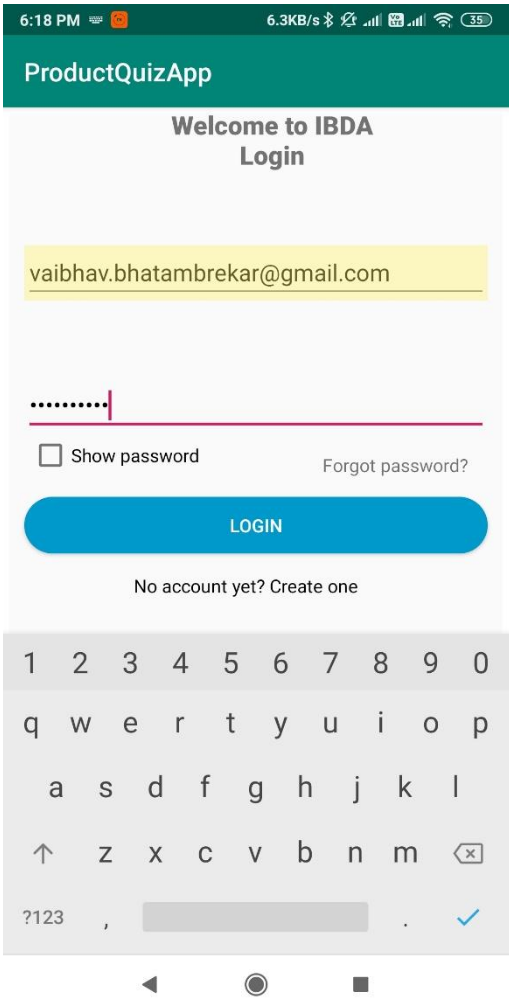
••••••••••

••••••••••

□ Show password

CREATE ACCOUNT

Already a member?Login



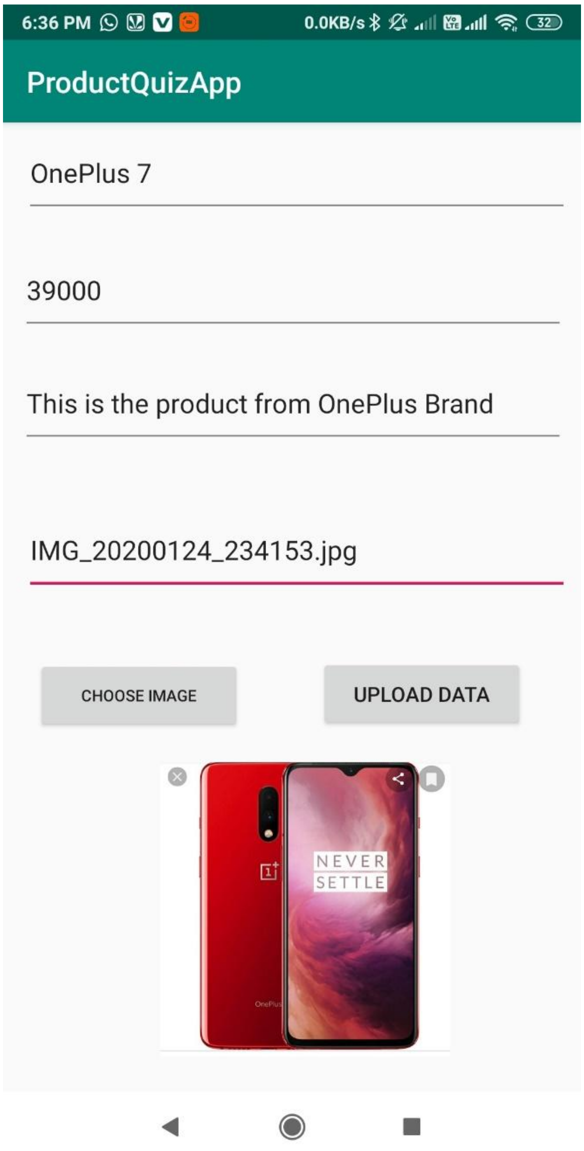
Welcome to IBDA
Login

vaibhav.bhatambrekar@gmail.com

••••••••••

□ Show password

No account yet? Create one



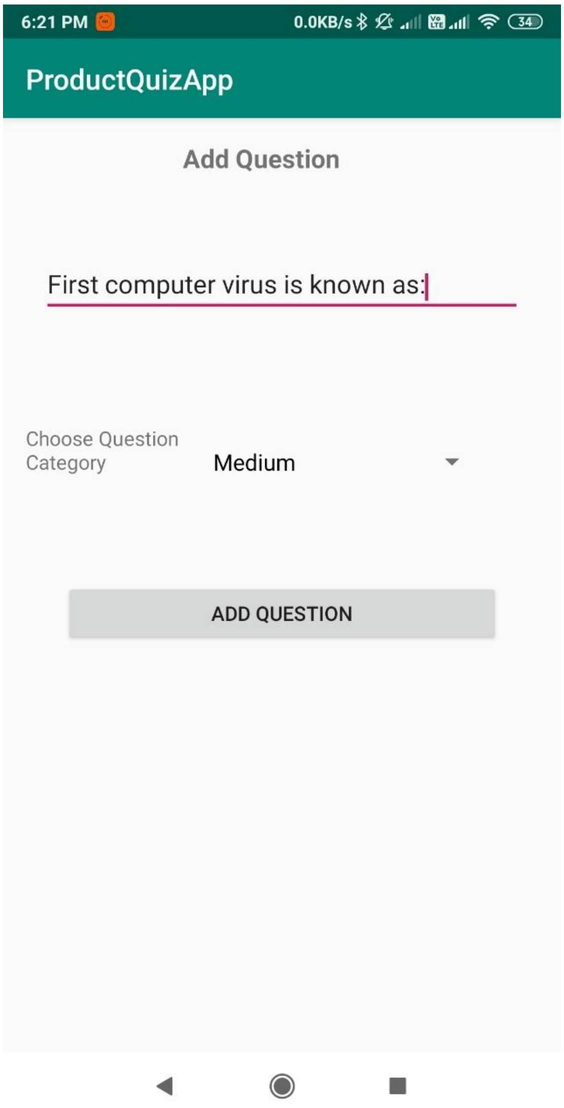
ProductQuizApp

OnePlus 7

39000

This is the product from OnePlus Brand

IMG_20200124_234153.jpg

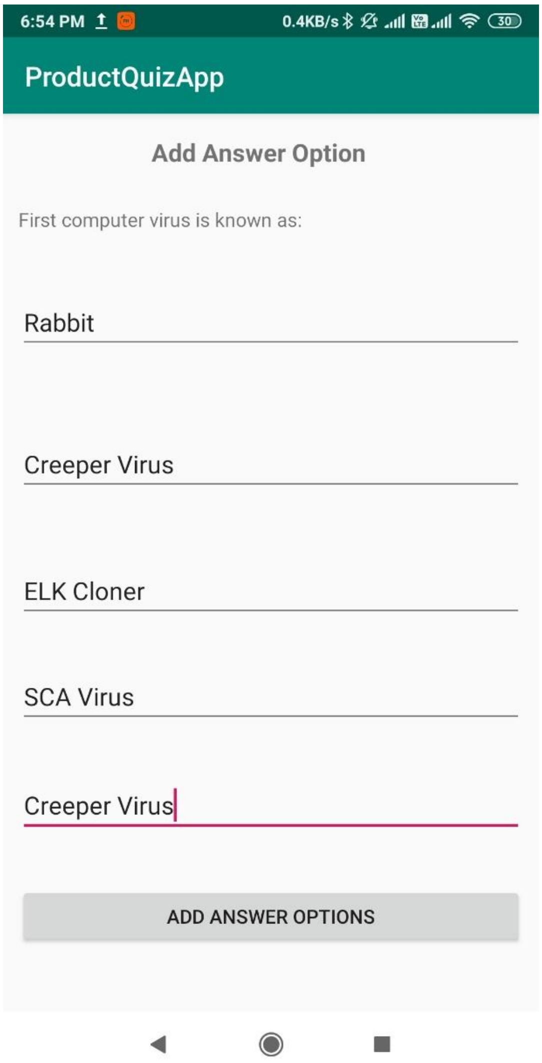


ProductQuizApp

Add Question

First computer virus is known as:

Choose Question Category: Medium



ProductQuizApp

Add Answer Option

First computer virus is known as:

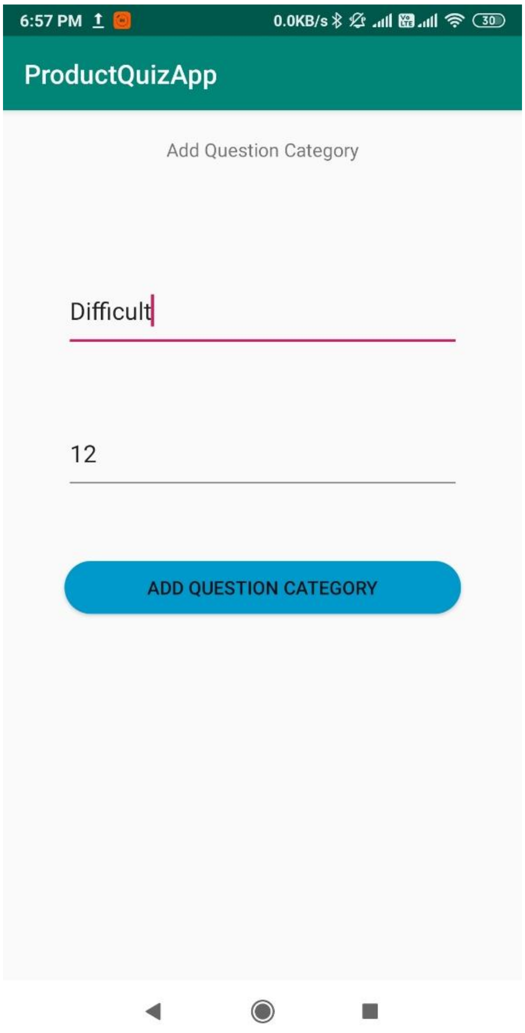
Rabbit

Creeper Virus

ELK Cloner

SCA Virus

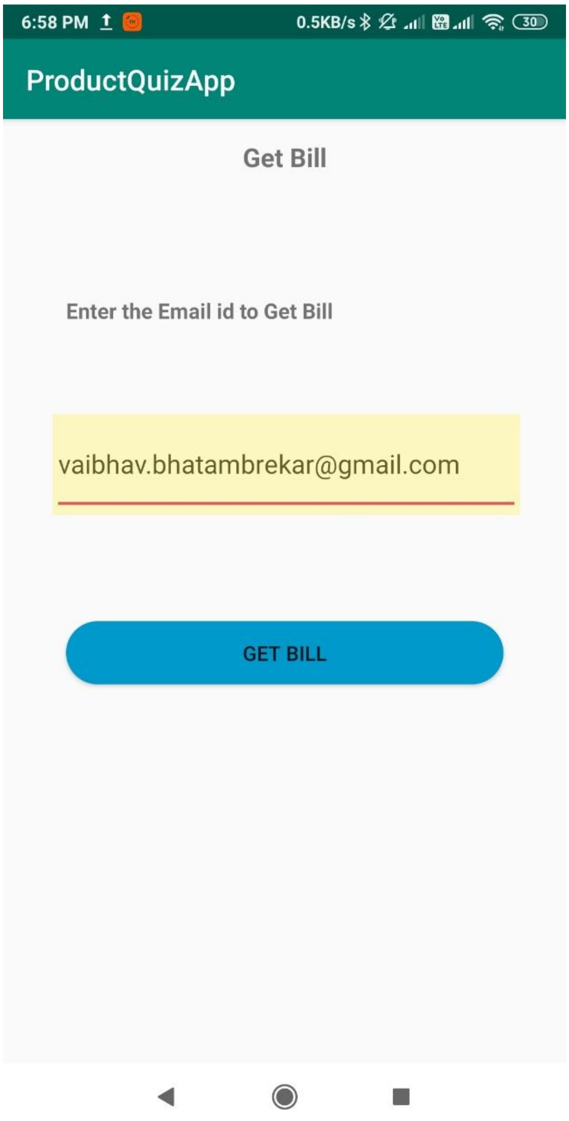
Creeper Virus



Add Question Category

Difficult

12

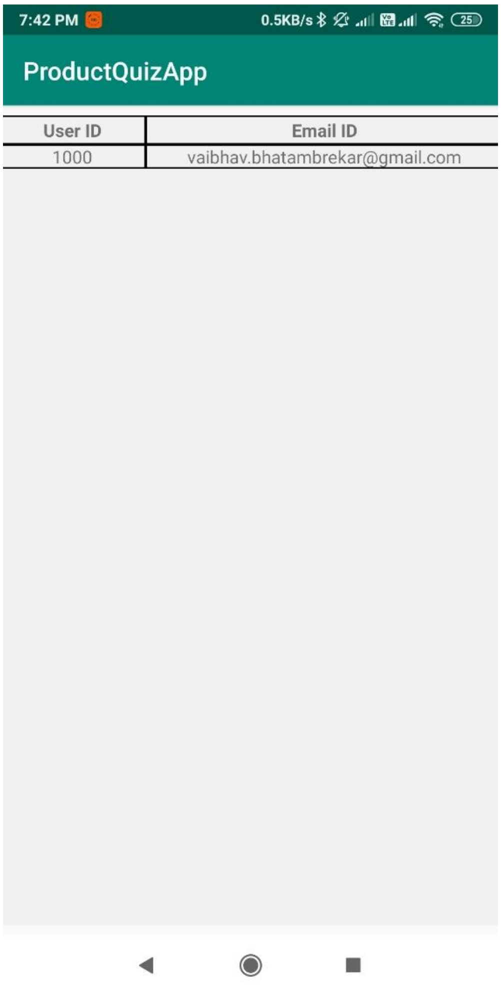


Enter the Email id to Get Bill

vaibhav.bhatambrekar@gmail.com

ANNEXURE 2

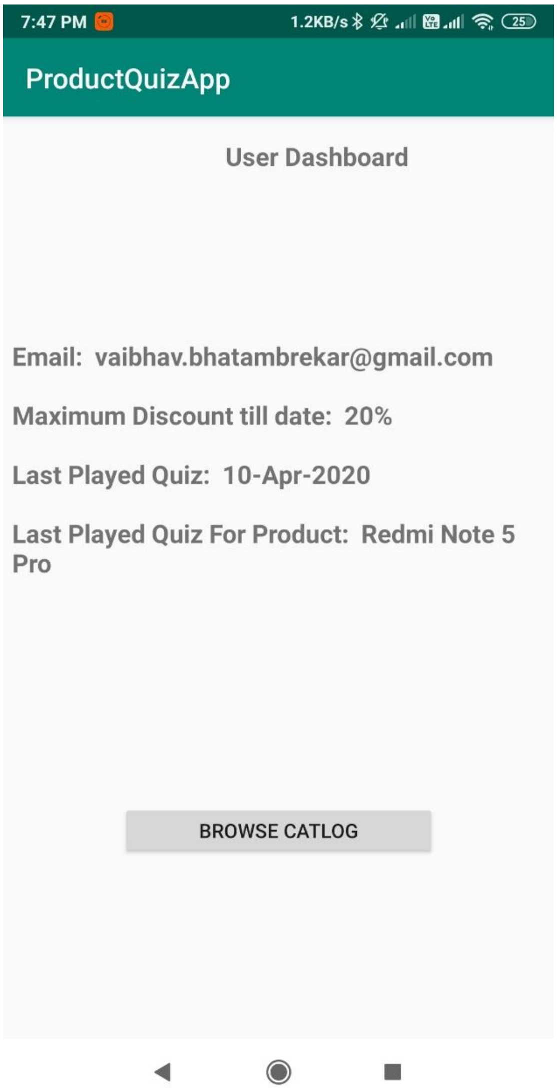
OUTPUT REPORTS WITH DATA



7:42 PM 0.5KB/s 25

ProductQuizApp

User ID	Email ID
1000	vaibhav.bhatambrekar@gmail.com



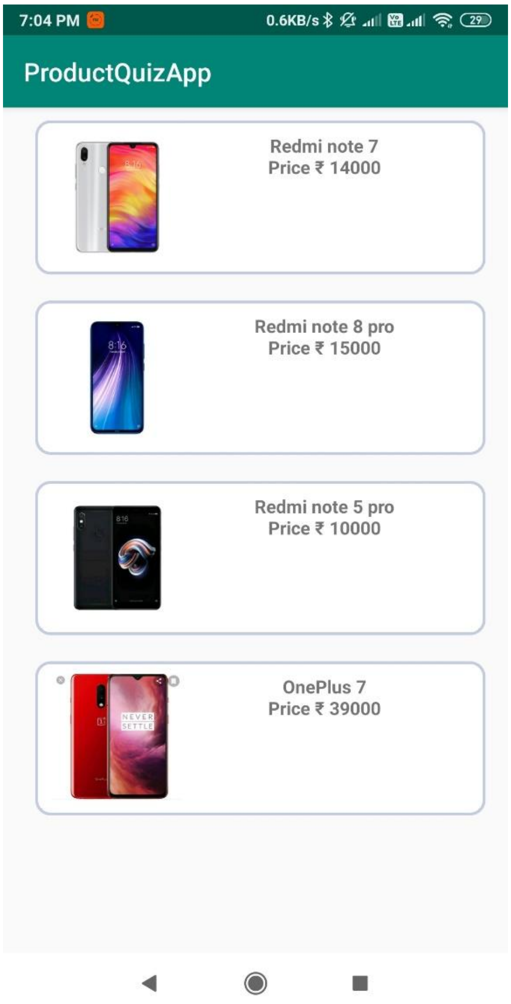
User Dashboard

Email: vaibhav.bhatambrekar@gmail.com

Maximum Discount till date: 20%

Last Played Quiz: 10-Apr-2020

Last Played Quiz For Product: Redmi Note 5 Pro



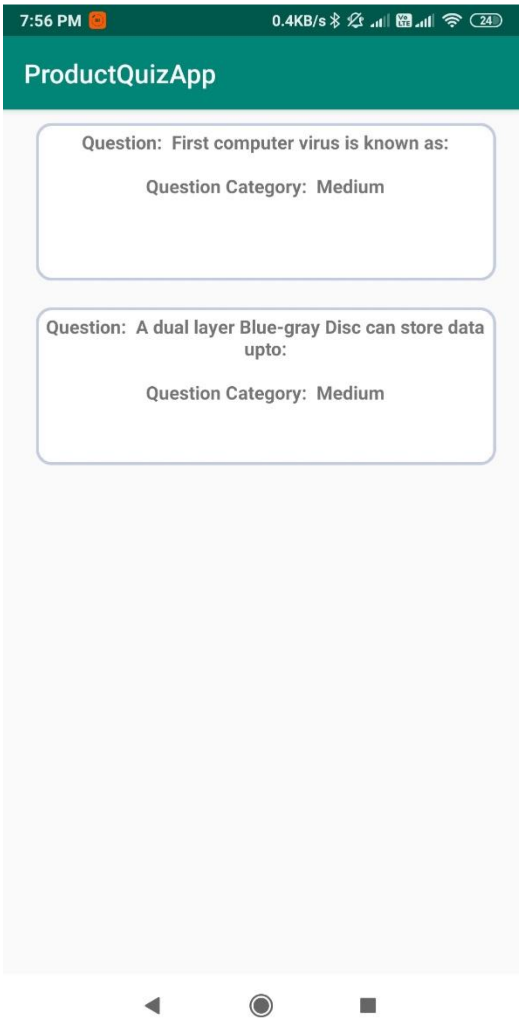
ProductQuizApp

Redmi note 7
Price ₹ 14000

Redmi note 8 pro
Price ₹ 15000

Redmi note 5 pro
Price ₹ 10000

OnePlus 7
Price ₹ 39000



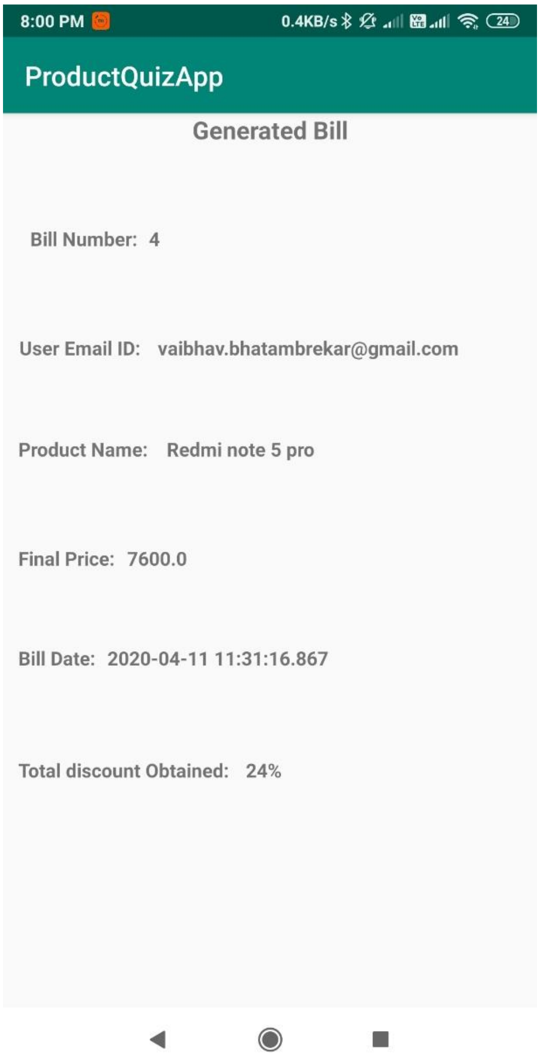
ProductQuizApp

Question: First computer virus is known as:

Question Category: Medium

Question: A dual layer Blue-gray Disc can store data upto:

Question Category: Medium



Generated Bill

Bill Number: 4

User Email ID: vaibhav.bhatambrekar@gmail.com

Product Name: Redmi note 5 pro

Final Price: 7600.0

Bill Date: 2020-04-11 11:31:16.867

Total discount Obtained: 24%

ANNEXURE 3

SAMPLE PROGRAM CODE

Login page Design code:

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android
"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/border1"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/loginemail"
        android:layout_width="360dp"
        android:layout_height="44dp"
        android:layout_marginStart="25dp"
        android:layout_marginEnd="25dp"
        android:layout_marginBottom="69dp"

        android:ems="10"

        android:hint="Email Id"

        android:inputType="textPersonName"

        android:textColorHint="#000000"

        app:layout_constraintBottom_toTopOf="@+id/login
pass"

        app:layout_constraintEnd_toEndOf="parent"

        app:layout_constraintStart_toStartOf="parent"


app:layout_constraintTop_toBottomOf="@+id/appname" />


    <EditText

        android:id="@+id/login pass"

        android:layout_width="360dp"

        android:layout_height="44dp"

        android:layout_marginStart="25dp"

        android:layout_marginEnd="25dp"

        android:layout_marginBottom="12dp"

        android:ems="10"
        android:hint="Password"

        android:inputType="textPassword"

        android:textColor="#000000"

        android:textColorHighlight="#000000"

        android:textColorHint="#000000"


app:layout_constraintBottom_toTopOf="@+id/showpass"

        app:layout_constraintEnd_toEndOf="parent"

        app:layout_constraintStart_toStartOf="parent"


app:layout_constraintTop_toBottomOf="@+id/loginemail"

/>


    <Button

        android:id="@+id/btnlogin"

        android:layout_width="360dp"

        android:layout_height="44dp"

        android:layout_alignParentTop="true"

        android:layout_alignParentRight="true"

        android:layout_marginStart="25dp"
        android:layout_marginEnd="25dp"
        android:layout_marginBottom="25dp"
        android:background="@drawable/buttoneffect"
        android:text="Login"
        android:textColor="#FFFFFF"

app:layout_constraintBottom_toTopOf="@+id/btngotoregist
er"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@+id/showpass" />

    <TextView
        android:id="@+id/forgotpass"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignEnd="@+id/btngotoregister"
        android:layout_marginTop="14dp"
        android:layout_marginEnd="37dp"
        android:clickable="true"
        android:text="Forgot password?"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/showpass"
        app:layout_constraintTop_toBottomOf="@+id/login
pass" />


    <TextView
        android:id="@+id/appname"
        android:layout_width="167dp"
        android:layout_height="49dp"
        android:layout_alignStart="@+id/btngotoregister"
        android:layout_marginStart="109dp"
        android:layout_marginTop="10dp"
        android:layout_marginEnd="105dp"
        android:layout_marginBottom="68dp"
        android:fontFamily="sans-serif-medium"
        android:text="Welcome to IBDA Login"
        android:textAlignment="center"
        android:textSize="20sp"
```

```
android:textAppearance="@style/TextAppearance.AppCom
pat.Display1"
    android:textStyle="bold"
    android:typeface="normal"

app:layout_constraintBottom_toTopOf="@+id/loginemail"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"

app:layout_constraintStart_toStartOf="@+id/loginemail"
    app:layout_constraintTop_toTopOf="parent" />

  <TextView
    android:id="@+id/btngotoregister"
    android:layout_width="190dp"
    android:layout_height="40dp"
    android:layout_alignParentStart="true"
    android:layout_alignParentBottom="true"
    android:layout_marginBottom="266dp"
```

```
        android:clickable="true"
        android:text="No account yet? Create one"
        android:textColor="#000000"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@+id/btnlogin" />

    <CheckBox
        android:id="@+id/showpass"
        android:layout_width="137dp"
        android:layout_height="29dp"
        android:layout_marginStart="26dp"
        android:layout_marginEnd="96dp"
        android:layout_marginBottom="28dp"
        android:text="Show password"
        android:textSize="14sp"

app:layout_constraintBottom_toTopOf="@+id/btnlogin"
```

```
app:layout_constraintEnd_toStartOf="@+id/forgotpass"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/loginpass" />
</android.support.constraint.ConstraintLayout>
```

Login page java code

```
package com.example.productquizapp;

import android.app.AlertDialog;

import android.app.ProgressDialog;

import android.content.DialogInterface;

import android.content.Intent;

import android.content.SharedPreferences;

import android.support.annotation.NonNull;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.text.TextUtils;

import android.text.method.HideReturnsTransformationMethod;

import android.text.method.PasswordTransformationMethod;

import android.view.View;

import android.widget.Button;
import android.widget.CheckBox;
import android.widget.CompoundButton;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.Random;

import static
com.example.productquizapp.ConnectionClass.password;

public class MainActivity extends AppCompatActivity {
    ConnectionClass conn;
    private FirebaseAuth mAuth1;
    EditText email1, pass1;
    Button btn1login;
    CheckBox showpass;
    TextView forgotpass,appname,btngoregister;
    ProgressDialog;
    AlertDialog.Builder builder;
    SharedPreferences pref;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        email1 = (EditText) findViewById(R.id.loginemail);
        pass1 = (EditText) findViewById(R.id.loginpass);
        btngoregister = (TextView)
findViewById(R.id.btngotoregister);
        btn1login = (Button) findViewById(R.id.btnlogin);
        forgotpass = (TextView)findViewById(R.id.forgotpass);
        appname = (TextView)findViewById(R.id.appname);
        showpass = (CheckBox) findViewById(R.id.showpass);

        pref =
getSharedPreferences("user_details",MODE_PRIVATE);


        mAuth1 = FirebaseAuth.getInstance();
        progressDialog = new ProgressDialog(this);

        btngoregister.setOnClickListener(new
View.OnClickListener() {
      @Override
      public void onClick(View view) {
        startActivity(new Intent(MainActivity.this,
Register.class));
      }
        });

        btn1login.setOnClickListener(new View.OnClickListener()
{
```

```
            @Override
            public void onClick(View view) {
                progressDialog.setMessage("Authenticating");
                progressDialog.show();
                userlogin();



            }
        });

        forgotpass.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new
Intent(MainActivity.this,ForgotPass.class));
            }
        });

        showpass.setOnCheckedChangeListener(new
    };



CompoundButton.OnCheckedChangeListener() {
      @Override
      public void onCheckedChanged(CompoundButton
buttonView, boolean isChecked) {
        if(isChecked)
        {

pass1.setTransformationMethod(HideReturnsTransformationMet
hod.getInstance());
        }
        else
        {

pass1.setTransformationMethod(PasswordTransformationMetho
d.getInstance());
        }


      }
    });
    }

    public void onBackPressed() {


        builder = new AlertDialog.Builder(this);

        builder.setMessage("Do you want to close this
application ?")
                .setCancelable(false)
                .setPositiveButton("Yes", new
DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int id) {
                        mAuth1.signOut();
                        finishAffinity();

                    }
                })
```

```
            .setNegativeButton("No", new
DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int id) {
                    //  Action for 'NO' Button
                    dialog.cancel();


                }
            });
        //Creating dialog box
        AlertDialog alert = builder.create();
        //Setting the title manually
        alert.setTitle("AlertDialog Exit");
        alert.show();


    }


    private void userlogin () {


        String email, pass;
        email = email1.getText().toString();
        pass = pass1.getText().toString();


 if (TextUtils.isEmpty(email) || TextUtils.isEmpty(pass)) {
            Toast.makeText(MainActivity.this, "Some Fields are
empty", Toast.LENGTH_SHORT).show();
            return;
        }
        progressDialog.setMessage("Authenticating");
        progressDialog.show();


        if(email.equals("admin") && pass.equals("admin"))
        {
            startActivity(new
Intent(MainActivity.this,AdminDashboard.class));
            progressDialog.dismiss();
        }
        else {


                try
                {
```

```
            Connection con = (Connection)

ConnectionClass.CONN();

            if (con == null) {

              Toast.makeText(MainActivity.this, "Error in

connection with SQL server",

Toast.LENGTH_SHORT).show();

            }

            else

            {

              String query = "select * from Registration

where Email='" + email+ "' and Password='" +pass + "'";

             // String query = "select * from user where

User_Name  ='" + email+ "' and Password='" +pass + "'";

              Statement stmt = con.createStatement();

              ResultSet rs = stmt.executeQuery(query);


              if(rs.next())

              {


                   SharedPreferences.Editor editor = pref.edit();
```

```
                        editor.putString("userOID",rs.getString(1));
                        editor.commit();
                        progressDialog.dismiss();
                        startActivity(new
Intent(MainActivity.this,User_Dashboard.class));


                    }
                    else
                    {
                        progressDialog.dismiss();
                        Toast.makeText(MainActivity.this, "Invalid
Credentials", Toast.LENGTH_SHORT).show();
                    }
                }
            }
            catch (Exception ex)
            {
                ex.printStackTrace();
            }
```

```
        }

    }


}
```

Main Quiz Page

Design code:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android
"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
```

```
  android:id="@+id/l1"

  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical"
  tools:context=".scroll">

  <ImageView
    android:id="@+id/imageView2"
    android:layout_width="227dp"
    android:layout_height="148dp"
    android:layout_marginStart="73dp"
    android:layout_marginTop="49dp"
    android:layout_marginEnd="73dp"
    android:layout_marginBottom="19dp"
    android:scrollbars="vertical"

app:layout_constraintBottom_toTopOf="@+id/radioGroup"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
```

```
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@android:color/background_light" />

    <TextView
        android:id="@+id/product_nametxtview"
        android:layout_width="252dp"
        android:layout_height="46dp"
        android:layout_marginStart="15dp"
        android:layout_marginTop="20dp"
        android:layout_marginEnd="40dp"
        android:layout_marginBottom="-10dp"
        android:text="hello"
        android:textSize="16sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toTopOf="@+id/button"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@+id/imageView2"
/>
```

```
<TextView
    android:id="@+id/timertxtview"
    android:layout_width="50dp"
    android:layout_height="50dp"
    android:layout_marginStart="330dp"
    android:layout_marginTop="-30dp"
    android:layout_marginBottom="10dp"
    android:text="TextView" />

  <TextView
    android:id="@+id/txtdiscount"
    android:layout_width="170dp"
    android:layout_height="30dp"
    android:layout_marginTop="0dp" />

  <ScrollView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
```

```
android:id="@+id/scrollview1"
android:layout_marginTop="10dp">

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="0dp"
    android:background="@android:color/white"
    android:orientation="vertical">

    <TextView
        android:id="@+id/question"
        android:layout_width="283dp"
        android:layout_height="36dp"
        android:layout_marginStart="16dp"
        android:layout_marginTop="20dp"
        android:layout_marginBottom="23dp"
        android:text="Hello world"
        android:textStyle="bold"
```

```
app:layout_constraintBottom_toBottomOf="parent"

app:layout_constraintStart_toStartOf="@+id/textView"

app:layout_constraintTop_toBottomOf="@+id/textView"
            app:layout_constraintVertical_bias="0.579" />


        <RadioGroup
            android:id="@+id/radioGroup"
            android:layout_width="358dp"
            android:layout_height="300dp"
            android:layout_marginStart="20dp"
            android:layout_marginEnd="20dp"
            android:layout_marginBottom="13dp"
            android:orientation="vertical"

app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
```

```
app:layout_constraintTop_toBottomOf="@+id/question">

        <RadioButton
          android:id="@+id/option3"
          android:layout_width="350dp"
          android:layout_height="20dp"
          android:layout_margin="20dp"
          android:layout_weight="1"
          android:text="demo" />

        <RadioButton
          android:id="@+id/option1"
          android:layout_width="350dp"
          android:layout_height="10dp"
          android:layout_margin="20dp"
          android:layout_weight="1"
          android:text="demo" />

        <RadioButton
          android:id="@+id/option2"
          android:layout_width="350dp"
          android:layout_height="10dp"
          android:layout_margin="20dp"
          android:layout_weight="1"
          android:text="demo" />

        <RadioButton
          android:id="@+id/option4"
          android:layout_width="350dp"
          android:layout_height="10dp"
          android:layout_margin="20dp"
          android:layout_weight="1"
          android:text="demo" />

  </RadioGroup>

  <Button
        android:id="@+id/button20"
        android:layout_width="100dp"
```

```
        android:layout_height="40dp"
        android:layout_marginStart="250dp"
        android:layout_marginBottom="20dp"

android:backgroundTint="@android:color/holo_blue_dark"
        android:text="Next" />

    </LinearLayout>

  </ScrollView>

</LinearLayout>
```