ZeroQ – ANDROID MOBILE APPLICATION

A Project

Presented to the faculty of the Department of Computer Science

California State University, Sacramento

Submitted in partial satisfaction of
the requirements for the degree of

MASTER OF SCIENCE

in

Computer Science

by

Subashini Hariharan

SPRING
2017

ZeroQ – ANDROID MOBILE APPLICATION

A Project

by

Subashini Hariharan

Approved by:

______________________________________, Committee Chair
Dr .V. Scott Gordon

______________________________________, Second Reader
Ben White

____
Date

Student: Subashini Hariharan

I certify that this student has met the requirements for format contained in the University format manual, and that this project is suitable for shelving in the Library and credit is to be awarded for the project.

___________________________, Graduate Coordinator   ______________________
Dr. Jinsong Ouyang   Date

Department of Computer Science

Abstract

of

ZeroQ- ANDROID MOBILE APPLICATION

by

Subashini Hariharan

People shop on both weekdays and weekends, but most people find free time during the weekends to get groceries, household items. Studies show that around 40 million people shop on the weekends in the USA. The difficult part of the shopping after spending a lot of time in buying things is waiting in a long queue for billing the items.

With attractive deals, coupons and discount offers, people tend to go shopping very often and that is increasing day by day, especially on weekends. Though people do online shopping, the need for traditional shopping has not been reduced. Although there exist applications to help people know their wait time and where they stand in their long queue, there is no application to help people completely avoid standing in a queue to get their items billed and to help them in finding things quickly inside of a big store.

In this project, the ZeroQ Android application helps people avoid having to stand in a long queue by: (1) allowing them to scan (the barcode) on the items they want as they are putting them into their (physical) shopping cart, (2) confirm which scanned items they wish to purchase, (3) confirm the purchase and pay their bill using PayPal (or cancel purchase/payment), and (4) generate a receipt for the purchased items. Once the payment is made, a confirmation mail is sent to the customer with an e-bill. This e-bill is later verified by the person in the store as they leave. Later, the e-bill will be saved in the cloud for future reference and can be retrieved later. As it is quite easy to misplace paper bills, organizing them in this way has advantages. ZeroQ also adds the expenses we spend for shopping every time and tracks them in a visualized manner by generating charts. Moreover, a product search feature will be available to know quickly whether a particular item or product is in stock or not.

______________________________, Committee Chair
Dr. V. Scott Gordon

____
Date

ACKNOWLEDGEMENTS

It is with my immense gratitude that I acknowledge the support and help of my Professor, Dr. Scott Gordon, who has always encouraged me into this research. Without his continuous guidance and persistent help, this project would not have been a success for me. I want to thank Ben White for his contribution in completion of this project. I am grateful to the California State University, Sacramento and the department of Computer Science without which this project would have not been an achievement. I also thank my family and friends, for their endless love and support throughout my life.

TABLE OF CONTENTS

Page

List of Figures

Figures Page

1. INTRODUCTION

1.1 Motivation

People do shopping on both weekdays and weekends, but most people find free time during the weekends to get groceries, household items. Studies [1] show that around 40 million people are shopping on the weekends in the USA. The most inconvenient part of the shopping after spending a lot of time in buying things is waiting in a long line to checkout for billing the items. With attractive deals, coupons and discount offers, people tend to go shopping very often and that is increasing day by day, especially on weekends. Though people do shop online, the need for traditional shopping has not been reduced. During holiday seasons we can see a substantial number of people inside the store, searching for items and their availability, information and later waiting in a long line to check out after their shopping. Currently, there exists many applications like LINE HERE, atQ, QLess, to help people know their wait time and where they stand in their long queue, Walmart for finding reviews about the item, Wallet – Budget tracker for separately uploading their bills to cloud and checking whether the item is available online (in online shopping). Although, these applications help people in diverse ways, still we need to spend a lot of time for shopping, organizing bills and tracking the purchases.

1.2 Proposed System

The Proposed solution is an Android Mobile Application, “ZeroQ” which tries to avoid the above-mentioned limitations by avoiding people to stand in a long checkout line, sharing the generated invoice directly to cloud, helping them in finding things quickly inside of a big store, tracking purchases and limiting expenses. Why a Mobile Application? A Mobile Application is a set of software which is designed and developed in such a way that it can run on any mobile device, smart phone or tablet. ZeroQ is an Android mobile application which runs on an Android platform with Internet support. In this project, the ZeroQ Android application helps people avoid having to stand in a long queue by: (1) allowing them to scan (the barcode) on the items they want as they are putting them into their (physical) shopping cart, (2) confirm which scanned items they wish to purchase, (3) confirm the purchase and pay their bill using PayPal (or cancel purchase/payment), and (4) generate a receipt for the purchased items. Once the payment is made, a confirmation email is sent to the customer with an e-bill. This e-bill is later verified by the person in the store as they leave. Later, the e-bill will be saved in the cloud for future reference. As it is quite easy to misplace paper bills, organizing them in this way has advantages. ZeroQ also adds and tracks shopping expenses in a visualized manner by generating charts. Moreover, a product search feature will be available to know quickly whether an item/product is available in that store or if it is out of stock.

1.3 Related Work

Currently, there exist many applications which are similar to “ZeroQ” but not exactly what it does. Android applications like atQ, QLess, LINE HERE help us to know our waiting time in a checkout line after shopping. However, we should still wait for making payment. This feature is fulfilled by “ZeroQ” and helps people in saving their time. Also, there are applications like Barcode Scanner, QR & Barcode Scanner which are helpful in scanning the barcode that is present on any item. It simply scans the barcode number and the same is displayed on the screen but not any information or details about the item when we scan it. But, “ZeroQ” provides complete information or details about the items when we scan the barcode on it, which is one of the challenging features of the project. We also came across many applications like Walmart, Costco for online shopping which helps us to know whether any item is available i.e in stock or not, but there are no mobile applications which provides information about item availability when we do traditional shopping. “ZeroQ” helps us in achieving this functionality too. Moreover, invoices are generated by it after we shop as similar to online shopping applications. Although we have Wallet- Budget tracker, expenses tracker applications for tracking purchases, they are of a separate application and not integrated with a complete shopping application beginning from searching items, checking out, payment and tracking the expenses that are spent each time.

2. REQUIREMENTS ANALSYSIS AND SYSTEM SPECIFICATIONS

2.1 Software Requirements Specification Document

2.1.1 Data Requirements

The set of data that is involved in any project is defined using data requirements. For this project, the main data required is the login information to register the application and the item’s information. Without this information the application cannot process the transaction.

2.1.2 Functional Requirements

Functional requirements are properties that must exist in the final system. For any mobile application, we need to download the application from the play store. The application could be either free or paid depending upon the store or merchant. To use the application, the user needs to register and login to the application after installing by providing login information. Once, he or she logins into the application, they can use all the features.

2.1.3 Performance Requirements

Response time, scalability, platform dependencies, tolerance are the performance requirements that should be considered when developing any system. The application or system should be able to respond quickly when the user interacts with the application. The application should be developed in such a way that it should be scalable enough to accept new features when we want to expand the application complexity. The application should run in all the specified software and hardware requirements from the design phase of the project. Also, the tolerance rate (fault tolerance) of the application should be at a higher level in case of network issues, connectivity issues, and when the application crashes or stops. It should be able to deliver the information about any of those issues to the user when the system is no longer able to provide results when the user wants.

2.1.4 System Requirements

The application should be installed into a device, system or any machine in such a way that it should have basic requirements like supporting software and hardware of the device, accessing in-built software, say camera for mobile device, internet permissions, and potential security issues such as virus or any malware detection.

2.1.5 Testing and Maintainability Requirements

The application should be able to meet all the possible good and bad test cases under a test environment. Application should be developed in such a way that it does not have any issues or crashes when the user is using the application. It should be able to extend itself when we expand the code or implement any new functions to the existing application.

2.2 Validation

Validating any application is an important criterion before releasing the application to the users. If there is no validation, the information entered by users may be redundant, formatted inappropriately and cannot be maintained. For example, we can validate mobile number in a way that it should use only digits and letters. Suppose, if the validation is not done, there are chances for the user to enter a wrong phone number and save it. In case of any emergency issues, the authorized person cannot contact the respective person. Similarly, validations for all the fields that are used to save information in any application are highly necessary. In this application, I have done several validations in the Login Page and Home Page.

In the Login Page, I have validated all the login information that is required for the user to sign up for the first time. Fields like username, email, password, mobile number are validated appropriately by displaying error messages. The username should not contain any digits, password should be minimum of six letters, the email should be a valid address and if the email id is already registered, an error message is shown saying that email id exists. The mobile number should contain only digits, when scanning the item, scan operation should be done properly to fetch the barcode number, if not a message is shown. When searching the item using any name, it should not contain any digits in the search field

3. SYSTEM DESIGN

3.1. Design Approach

This project is based on the functional design approach, which helps in understanding the design of the project in a simpler way by explaining its flow, use cases, and implementation more like a modular approach. For example, there are different modules in this project which have separate functionality and, other sub functionalities/modules. All the modules are designed, implemented and integrated together to make a flawless working application.

3.2 Detailed Design

The detailed design including modules and sub modules of the application is as follows:

1. User Registration:

If the user wants to use the ZeroQ, they must download the application from the play store, install and register it by providing login information. Once, they registers the registered information is stored on the server and can be validated, checking the valid credentials for the next time he logins with the application.

2. Instant Search:

The instant search helps in finding whether the item is available in the store or not. This could help the users save time by searching for the item which is out of stock.

3. Scan the Item:

This feature helps people by allowing them to scan the barcode that is available on the item. Once the user scans the item, they can see all the information about the item like barcode number, name, quantity, price, net price, and serial number. Users can scan any number of items they wish and keep adding to their physical shopping cart. Later, they can purchase any item they want by checking the items in the cart. Suppose if the stock is not available for any items, “stock not available” message is also shown.

4. Checkout and Payment:

Checkout is made in an easy step to avoid hassle in this application. The user can just check in with the checkbox from the cart. Depending upon the quantity of the items, the payment is calculated along with the tax rate (Ex: 8%). If the user wishes to purchase, they can proceed by clicking the “PayNow” button or they can cancel the purchase at this stage and proceed and with shopping for other items or they can exit the application.

5. Invoice Generation and Uploading:

Once, the user confirms the purchase and pays the amount, the invoice is generated at the same time and we can share it instantly to cloud (share to drive, send email)

6. Tracking Purchase:

The application allows us to track purchases by adding the various expenses we spent and view them in the form of charts (pie chart) as for other charts like line graphs, bar graphs we need a large dataset. Viewing expenses is interactive with different options like view by category, by and between dates, or view all.

3.3 Application Design

The main aim of the system design is to explain the scenario using use case diagrams. Use case diagrams clarify the flow of the application by deriving the use cases for all the functionalities in form of diagrams for the users.

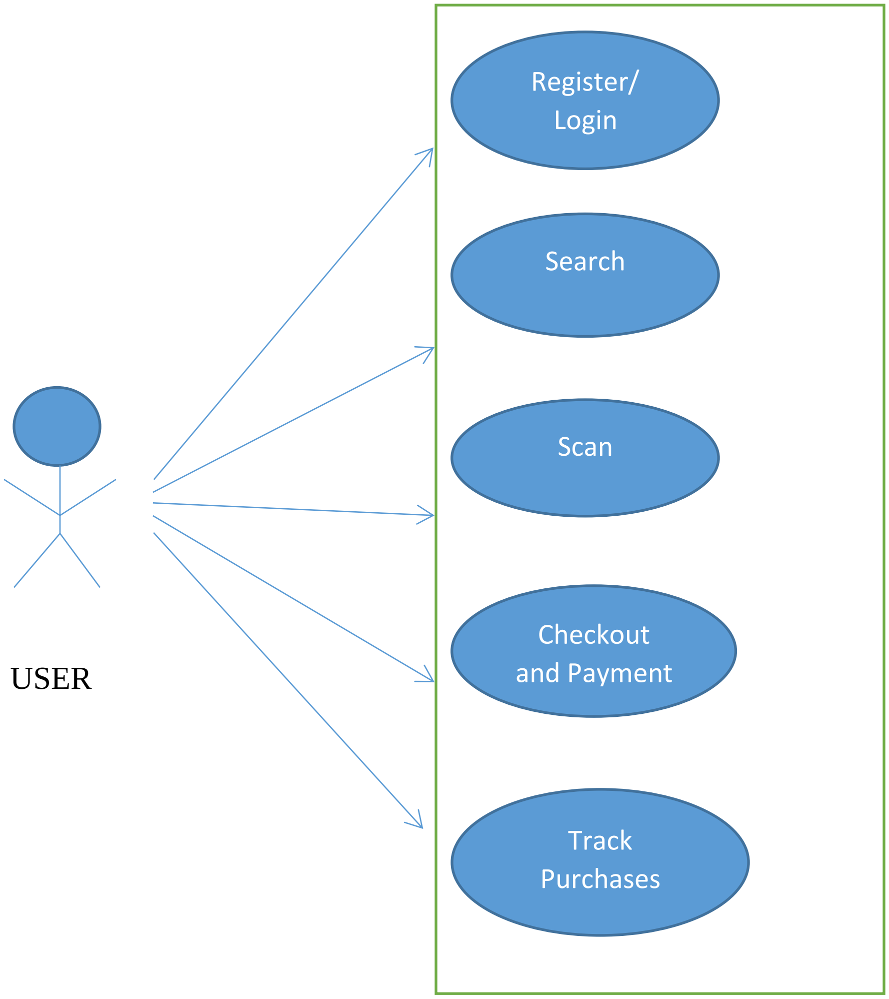


Figure 1: Use Case Diagram for User

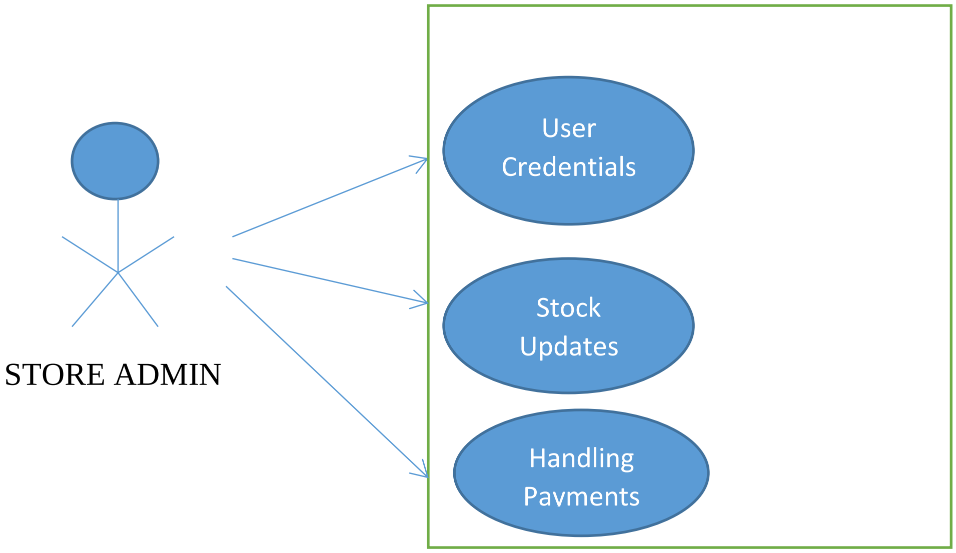


Figure 2: Use Case Diagram for Store Admin

3.4 User Interface Design

User Interface Design for any application should be very simple. We should have only a few clicks or navigation among the features when using the application to avoid hassle. In this application, there are two main screens, the Login and Home screens.

The login page is the first page which appears when the user uses the application. In that page, if he is a new user, he can sign up or if he is an existing user, he can login with the credentials.

The next screen is the homepage where the users can select features and use the app. The following image is the home screen. As seen from the image, this screen has all the key features.

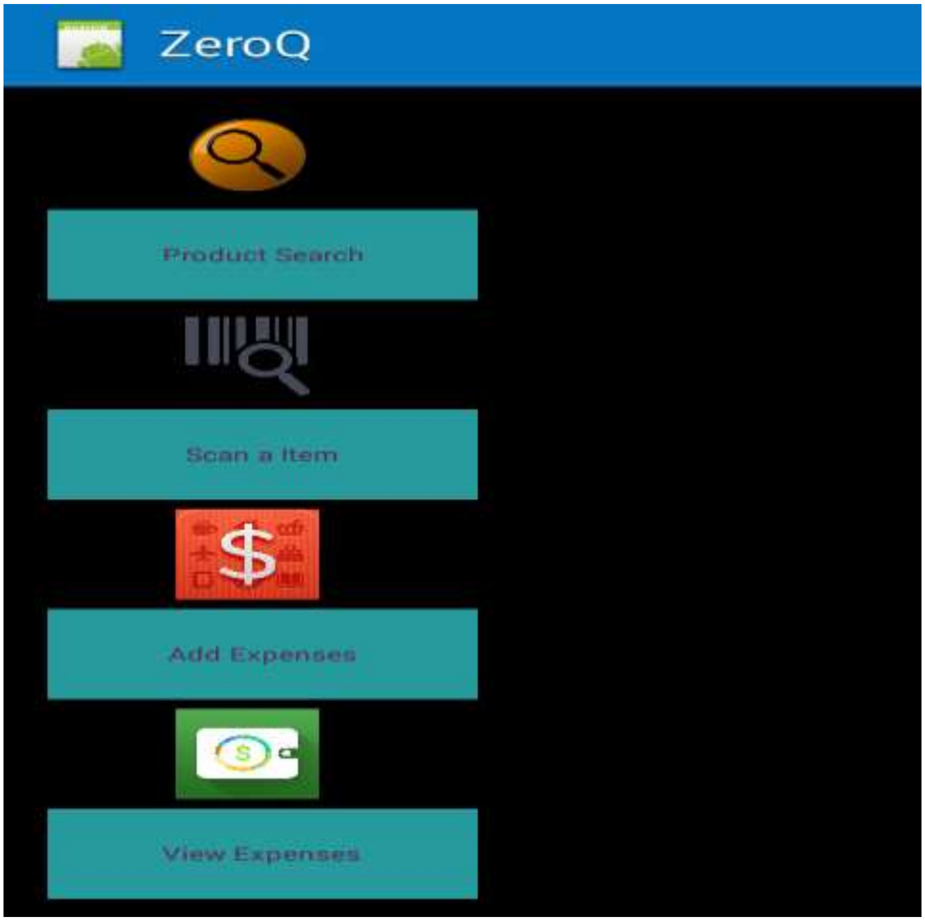


Figure 3: Home Screen

3.5 Database Design

The database should be designed in such a way that it should be easy to access and manipulate. Database definition and database manipulation operations should be performed accordingly to add, delete, and update values. In this project, I have used a MySQL database which is an open source database, easy to install and use. The database server could be installed by providing user credentials such as username and password. The created database is hosted remotely and I am accessing the database using SOAP web services. Web services act as the interface between the Android application and the database. It acts as a service between two electronic devices and they are communicated via HTTP protocol (using, World Wide Web). As the Android application cannot connect directly with the database server, we are using Web Services. The information to be transferred or fetched could be in XML or JSON formats. In this project, when we scan the item, the information is sent via web services in JSON format and displayed to users from MySQL database (products.db). The following figure, explains how the process takes place.

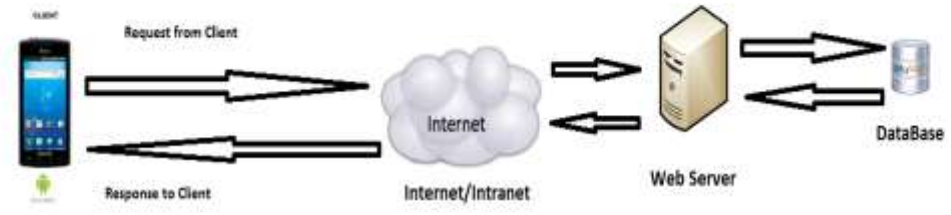


Figure 4: Application Work Flow

3.6 Entity Relationship Diagram

There are three tables involved for this project under one database (products) which was created in MySQL. One table includes information about the items. When the user scans the item, information from this table (item_details) is fetched through web services. There are seven attributes or fields for this table which includes item_code, item_name, price, quantity, net price, serial_no, stock. Secondly, another table (user) has login and registration information about the users who sign up/sign in. It has five attributes including, user_id, name, e mail, password, and phone number. The third table is for tracking purchases or expenses and it has five attributes namely, id, category, amount, date and notes. As the three tables do not depend upon each other as per the design, the entity relationship diagram is as follows:

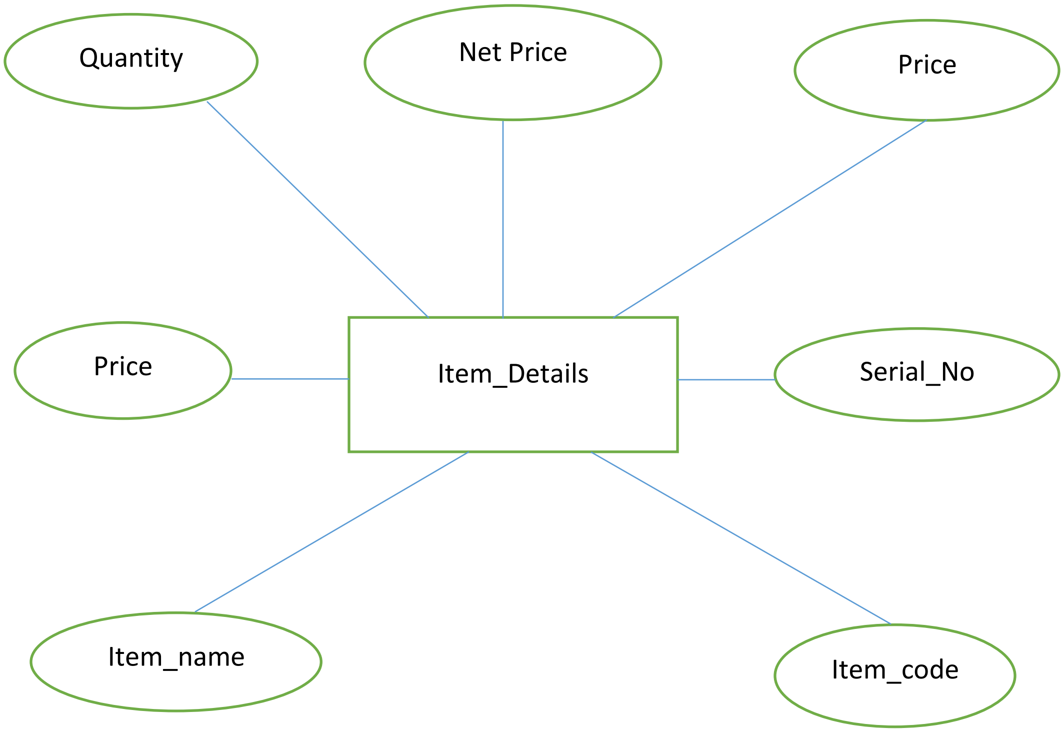


Figure 5: ER Diagram for table Item_Details

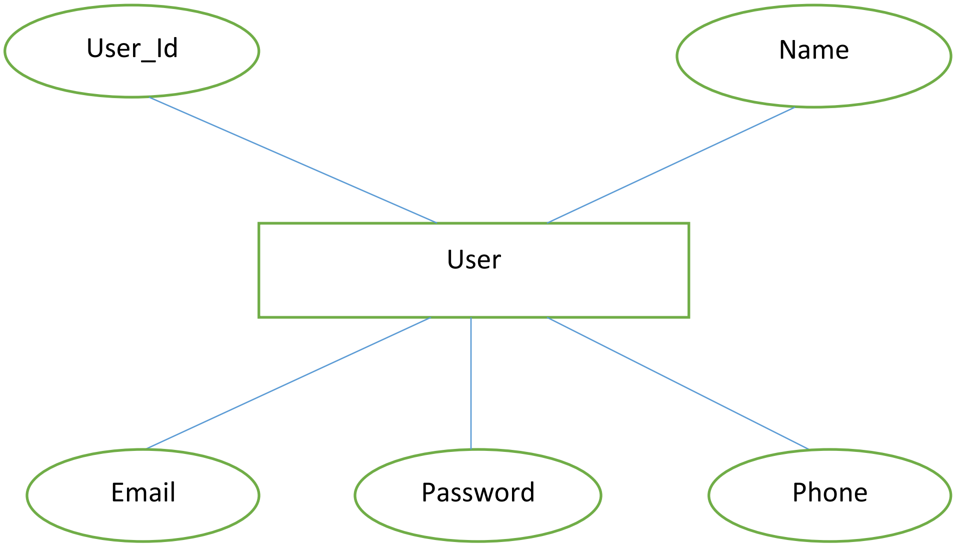


Figure 6: ER Diagram for table User

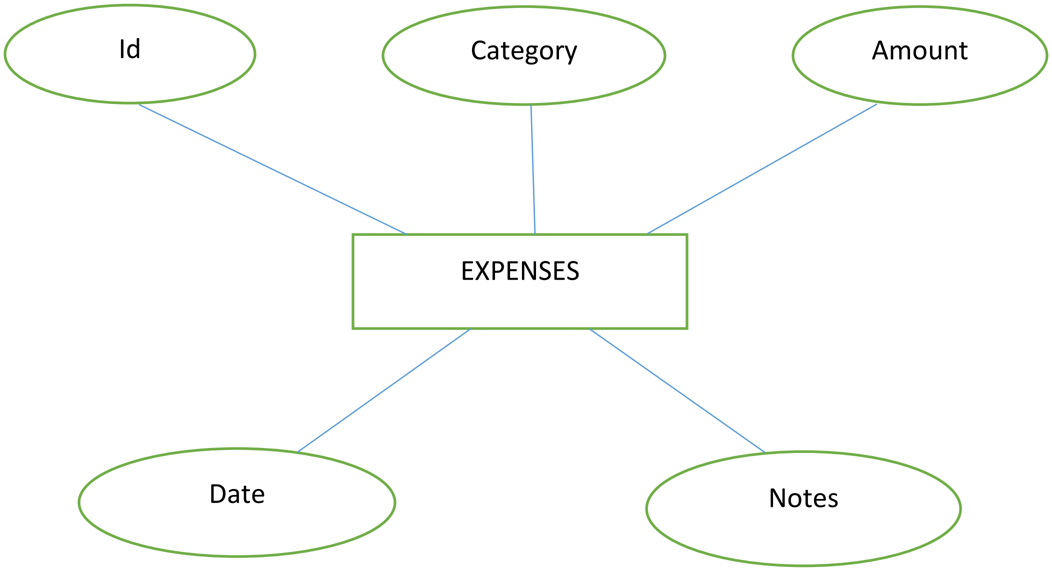


Figure 7: ER Diagram for table Expenses

3.7 Methodology

Agile Software Development Methodology [2] is widely used in many projects as it has many advantages. After gathering the project requirements, it is reviewed frequently in the form of small iterations and made into action by executing it. After completing tasks each iteration, it could be reviewed properly and moved to next iteration. The main advantage of this methodology is that we can change the requirements or design even in the middle of the project when the situation arises. Also, code maintenance is easier in contrast to Waterfall Approach. In Waterfall methodology, there is no flexibility in changing the requirements when we develop the project because we must understand the working flow of the project at least 80% even before the start of the project and work according to that. Only if the design process is done, we can move to construction, testing, and support. Though this is not a team project, I have approached this project using agile methodology by applying its principles. To illustrate, for example: In one of the modules, "Scan an Item", before start developing it, I have analyzed the design requirements initially and planned to execute them in the form of short iterations which are called "Sprints". I tried to execute the planned tasks of the iterations accordingly, but I could not able to finish some of the tasks on that particular iteration and they are carried to next sprint and are called "Product Backlogs". Also, as mentioned, I have changed some of the requirements for the convenience of the user interface while developing the project which is the main advantage of this methodology. Hence, for these reasons, I had decided to follow the principles of agile methodology.

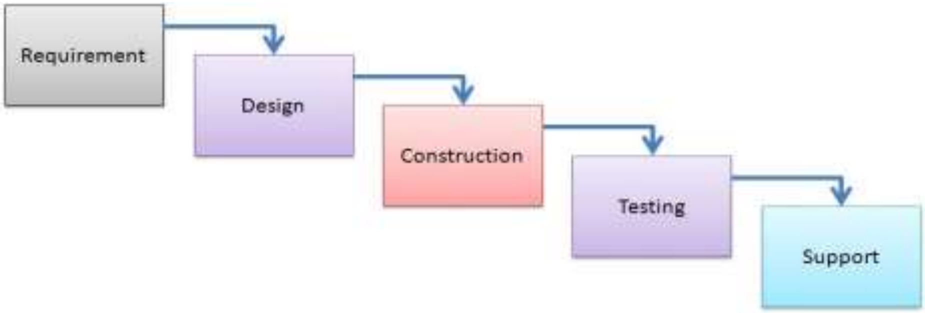


Figure 8: Waterfall Methodology

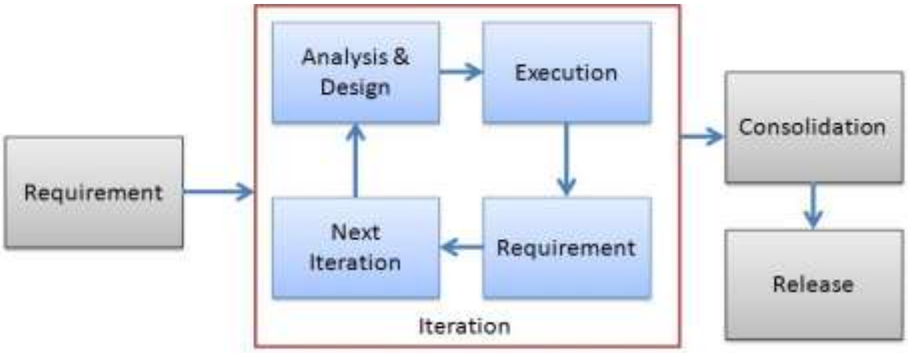


Figure 9: Agile Methodology

4. IMPLEMENTATION, TESTING AND MAINTENANCE

4.1 Introduction to Programming Languages, IDE'S, Tools and Technologies used for this Implementation

4.1.1 Java

As the project is developing an Android Application, the default programming language is Java. All Android applications are built using Java in Android Studio or Eclipse or both. Java is a popular and widely used language throughout the world. As mentioned in, [3] Java is one of the powerful programming languages like C, C++. developed by Sun Microsystems which has many powerful features as described below. After the development of C, C++, Java has come into evolution by addressing their drawbacks [4] . It is one of the open source projects [5] that could be easily installed in our machine. The language is also easy to learn, understand and implement. Java is used in various kinds of applications like Web, Desktop, Mobile, and Big Data. Many powerful features are supported by Java including various libraries, application services, graphics library for 2D/3D applications. The language is flexible enough to maintain code complexity, test, implementation, integration and support. Apart from these, there are other key features which make Java more special. It is object oriented programming language, one of the important hierarchies in the programming languages which is used to implement real time applications, it provides for code reusability, it has a platform independence feature including any virtual machines(Write Once Read Everywhere), as in no need to write the code for different OS as the Java Compliers convert the java source files to bytecode and this could be interpreted by any machine and the actual code is compiled irrespective of any machine, OS. It is more secured as the compilers are designed efficiently to figure out any kind of errors.

4.1.2 IDE's, Tools and Technologies:

4.1.2.1 Android Studio

Android Studio [6] is exclusively designed for developing Android applications. It consists of all Android SDK tools to design, develop, maintain, test, debug and publish our app. The IDE is designed very efficiently which makes the developer's job easy. It also supports the IntelliJ IDE, the main idea behind this IDE is that it automatically senses the variables, methods, classes, built-in functions or it could be anything else when we press the first letter of it. Say, suppose we declared few variables or methods that starts with an 'S', it automatically senses everything that starts with an 'S' and makes suggestions. It also supports Git as a version control system to maintain the app changes and push them into github. All java files, layout files (for design) are integrated into a single project easily. After the completion of project, the whole application could be put as an .APK (Android Package) file, in which we can run that APK file in any device and use the application.

Other main tools include Android SDK, ADB, and Gradle Build.

4.1.2.2 Android Software Development Kit (SDK):

One of the main tools used in developing android applications, as it packages many core features into one SDK and it can be used in the application easily. This helps us to avoid writing lot of code, and building applications faster.

4.1.2.3 Android Debug Bridge (ADB):

Android SDK uses ADB tool as a connection device which allows us to connect the Android Devices or Emulator with the machine via USB. After developing or while developing applications, we can connect with the device to check how the application runs. Later, we can debug and run the applications.

4.1.2.4 Gradle Build:

Gradle Scripts are the recent feature that is added to Android Studio. It is basically an automated build system which is used to automate the various phases involved in designing an application that includes design, development, test, debug, and publish. We need to configure the project and modules by mentioning all the supported jar files, SDK’s, version name, level, compiled SDK version, build tools version. to ensure that the developed app is compatible with the testing device/emulator. Gradle is also similar to Ant and Maven which helps in maintaining java projects (repositories).

4.1.2.5 Android Device Monitor:

If we want to access all the hidden files that are generated when we run the application, we can use the monitor. We can select any project and explore the files that are related to that project. But, as they are hidden files, we need root permissions to access them. Suppose, if we run the app in device, we need to root the device and run commands in adb shell to get permissions.

4.1.2.6 SDK Manager:

It is one of the main tools to maintain the updates of all the installed components required to run the project. It also notifies us when the project is not compatible with device or any other compatibility issues and to download any component that is required.

4.1.2.7 AVD Manager:

It is used to create virtual devices of any desired API level to support higher level SDK's incase our device does not support. Using emulators to test the application is difficult as it might be little slower when compared to real device.

4.1.2.8 SQLite Database:

Android also supports inbuilt database which is Android SQLite to develop any small applications and perform any CRUD (Create, Update, and Delete) operations. As it is not flexible enough to support substantial number of data, for complex applications we are using other external databases.

4.1.3 Eclipse IDE

Android applications could be developed in Eclipse IDE in which we can compile, run, debug and deploy using ADT (Android Development Tools). In this project, I have used Eclipse IDE to create Java SOAP Web Services for connecting the database and used Android Studio to develop the application.

4.1.4 MySQL Database

In this project, I have used MySQL database to store the data. This is one of the popular open source relational database management systems. We can perform all DDL, DML, DCL operations using this database. This also supports different programming language applications. The applications could connect the database using separate ways which includes PHP myadmin WAMP, LAMP, Web Services. To use this database, we should first download, install and configure the MySQL instance in our machine. While configuring, we should give access credentials which could be used further whenever you open the MySQL shell.

4.1.5 Apache Tomcat Server

It is one of the open source enterprise Java servlet containers which are widely used in many applications to maintain their database. The database is kept in this remote server and could be accessed whenever needed through MySQL shell.

4.1.6 Soap UI

I have used SOAP Web Services in this application to connect the application with database. SOAP Web Services are implemented by writing the service as a method and we should call this method in corresponding Android java files, the page where we want to call and display the details. The web service method could be tested in SOAP UI [7] by passing the parameters or fields in the database. The service method returns success if it returns the actual result and fails if it does not. In this way, we can test the service before calling it in Android Java source files. The UI is very interactive and easy to use.

4.2 Security and Permissions in Android

Security notions in Android are quite high. Whenever a new Android Application is created, a unique user and group ID. This makes the maintenance of the application in an easier way to avoid any security or privacy issues. As the application is created uniquely, it becomes private and no one can access other’s applications.

Permissions are another important concept which is included in AndroidManifest.XML configuration file. This is required if the application wants to access the external features. For ex, if the application wants to access the Internet, Camera or it could be any feature, it requires permissions. It is included within the tags as it is an XML file. Permissions are automatically created for the basic applications at the time when we create the application. If the app uses higher level API or SDK we must explicitly mention the permissions inside uses-permissions tag to access the features or components.

4.3 Test Plan and Test Activities

Test plan is necessary for any project to plan the testing phase and decide the scope of the project. Test plan involves collecting design specifications about the project, wiring test cases, executing them manually or automatically using automated testing tools. Testing any application is highly important. Test plan is a method of documenting the test cases, specification plans and other basic level details about how the application works.

Test Activities for this project includes various testing like:

- Black Box testing: In this project, sample test cases are written and manual testing is done to check the functionality of the application.
- White Box testing: Once the application meets the user requirements and functionalities according to the test cases, its internal logic are completely tested to ensure that the application does not have any logical errors or issues.
- Unit Testing: I have tested all the modules of the application individually by running as a test program.
- Integration testing: After testing the modules individually, tested them by integrating all the sub modules, modules into one application.
- System Testing: It refers to checking whether the system in which the application is built meets the necessary requirements like software support. For example: In this project, I have checked whether the device in which the application developed is compatible with the software (Android Studio)

- End to End Testing: Tested the complete environment of application by connecting the device with different machines, installing as an APK file, with the database and in local network.
- Usability Testing: Finally, usability testing is performed by testing the application's flow, UI design and how flexible and easy the application is easy to use.

4.4 Application Maintenance

Apart from designing and developing the application, maintaining the application is one of the important characteristics. The developer/owner of the application should be concerned about the maintenance of the application by fixing the issues. The solution for fixing such type of issues when the application crashes by using any of the "Crash Reporting Service" that are available for mobile applications. I have used "Crashlytics" [8] reporting service in this application to track the issues when the user faces. Crashlytics is a lightweight reporting service for mobile applications including Android, iOS by providing respective SDKs. It also enables by analyzing the data in the form of reports which includes all the details about the number of issues faced, affected users because of the issues and finally indicating about the root cause of the crash by including the exact line in the source code of the application (error code) due to which the application might get crashed. The report analytic engine that is used by Crashlytics to perform real time crash reporting analysis is using "Fabric". Crashlytics is a part of Fabric platform, together they are acquired by Google. Crashlytics and Fabric SDK are added to the project and imported to make it work. To understand the process, when using the application, suppose if the user faces any issues related to network connectivity, database connectivity. the application could not be used further by them. The application developer/owner receives an email related to the issue at the same time when the user faces the crash. The developer could keep track of all such issues, fix them and release it as a latest version in the google playstore and hence a new software update is available to the users. He can view the crashes in the form of reports by accessing his Fabric dashboard. Applications like Yelp, Groupon, Walmart, Waze uses Crashlytics to maintain their application and fix crash issues. The following figure illustrates about how the report look like:

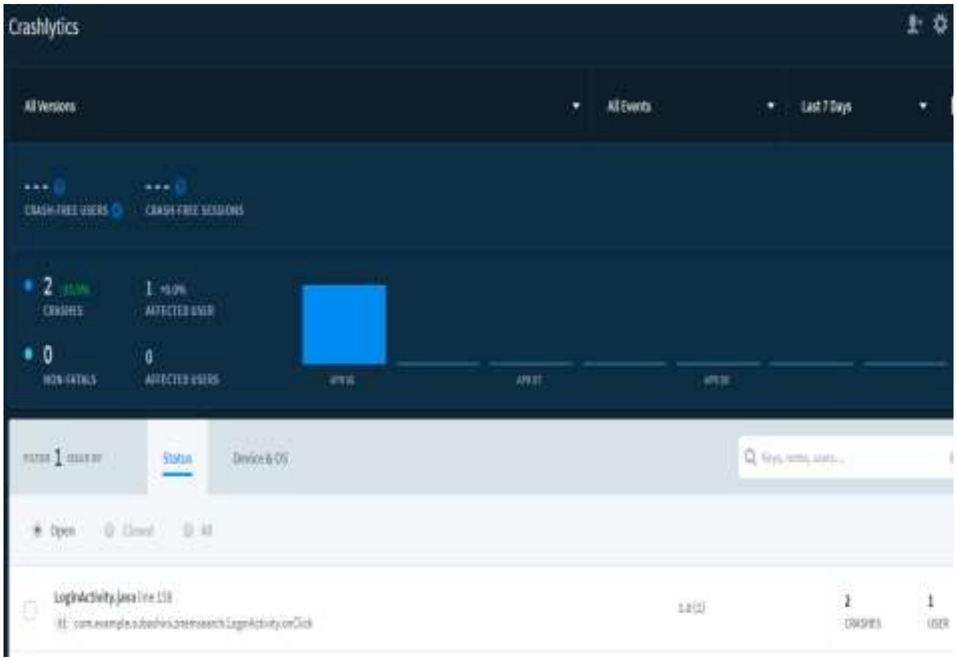


Figure 10: Crashlytics Report Engine

5. RESULTS AND DISCUSSIONS

5.1 User Interface Representation

To make the application interactive, different controls have been used and designed using the layout file. Following are the important controls that are designed and used in this application:

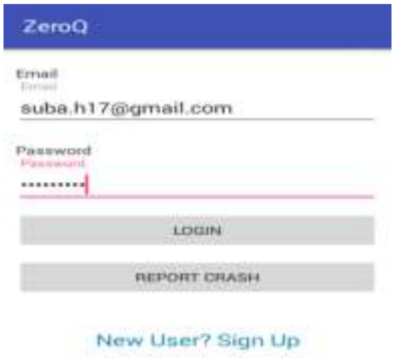
- Text View: The text view component belongs to the view group as a part of GUI. It displays the text or content view of any activity to the user and allows them to edit.
- Edit Text: This allows itself to be editable in the text box.
- Button: One of the important components in which the application needs. It is mainly associated with action when the user clicks it. We can represent the button using any text which holds the action class on it.
- Image Button: Suppose, if we want to have an image for the button which we have designed, we can include using this control by adding the source or path of the image file within the tags in the layout file.
- List View: This is a key component under the view group which helps in displaying the information about anything when we click the action button. It also allows us to scroll through the screen and have a look about the information displayed. Using the list adapter, the content is pulled from the database.

- Checkbox: It is the control component which allows us to use or make use of the function by just clicking on the check box button. When we include check box widget in the application, we can see a small box in the screen on which we can check it and it will be selected.

5.2 Description of Features and the Approach

1. User Login/Registration:

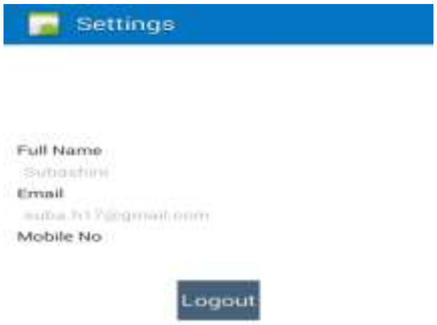
If the user wants to use the ZeroQ, we must download the application from the play store, install and register it by providing login information. As shown in the Figure 11, the login information includes user name and password. For the new user, the user must sign up by providing Full Name, Email Id and Password. Once, he registers, the registered information is stored in the server and can be validated, checking for the valid credentials for the next time he logins with the application. After using the application, he can logout as shown in Figure 12.



Email: suba.h17@gmail.com

Password: •••••••••

11: Login Screen



Full Name

Subashini

Email

suba.h17@gmail.com

Mobile No

Figure 12: Logout Screen

2. Instant Search:

The below Figure 13, is the instant search which helps in finding whether the item is available in the store or not. This could help the users in saving time by searching for the item which is not available in the store because of no stock. Search model is created using Web Services which displays the information to the user.

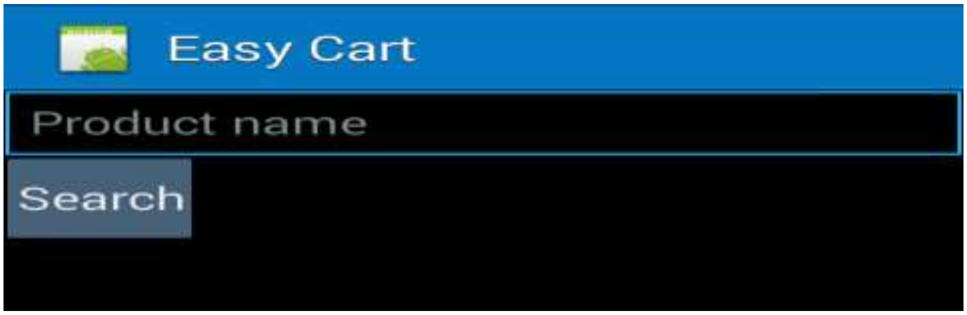


Figure 13: Search Screen

3. Scan the Item:

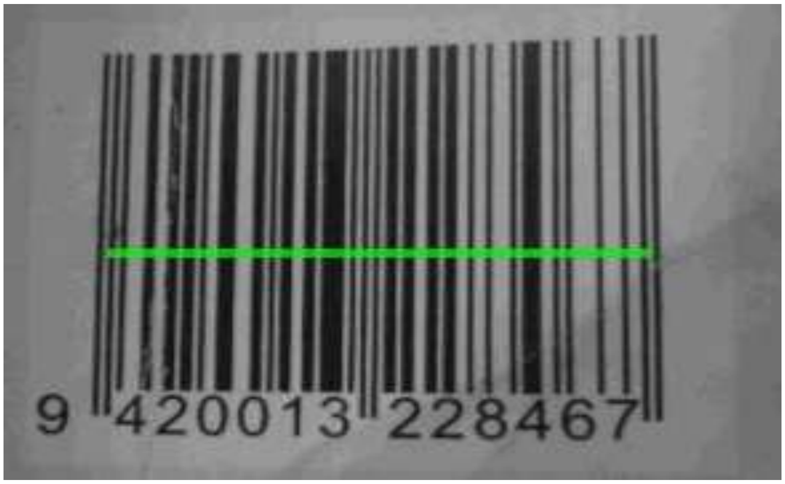
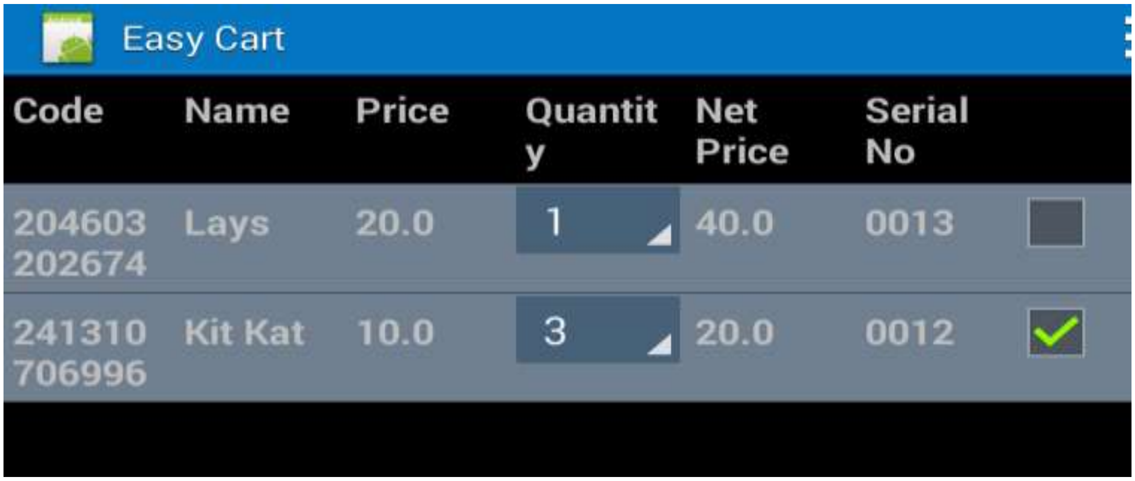


Figure 14: Scan Screen



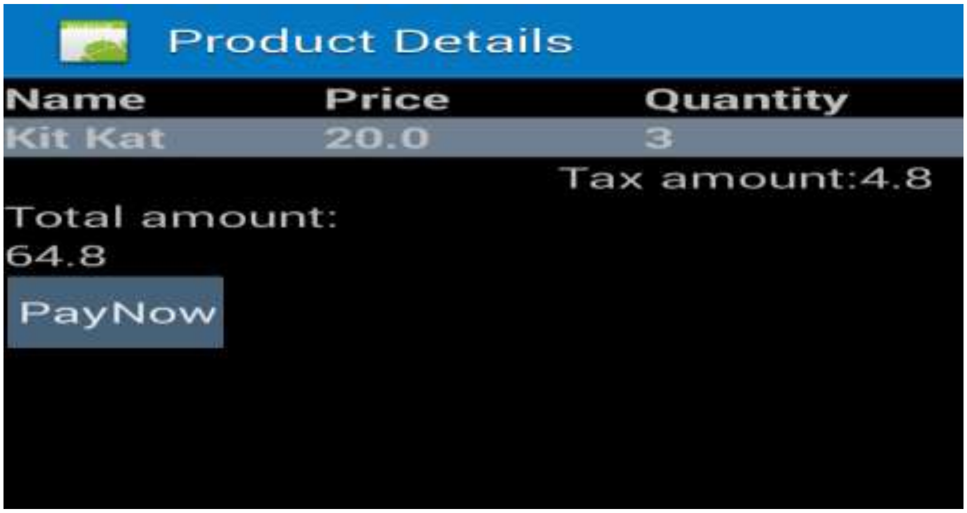
Code	Name	Price	Quantity	Net Price	Serial No	
204603202674	Lays	20.0	1	40.0	0013	□
241310706996	Kit Kat	10.0	3	20.0	0012	☑

Figure 15: Scanned Product Details Screen

In this project, I have used Zxing (Zebra Crossing) library to scan the barcode on items and it supports many kinds of barcode formats- UPC-A, UPC-E, EAN-8, EAN-13, Code 39, Code 93, Code 128, QR Code, Codabar. To use this scanning feature in our project, we should zxing jar files. This feature helps people by allowing them to scan the barcode that is available on the item [9]. Once the user scans the item, they can see all the information about the item like barcode number, name, quantity, price, net price, serial number. Users can scan any number of items they wish and keep adding to their physical shopping cart as shown in the Figure 15. Later, they can purchase any item they want by checking the items in the cart. It also has the option to select quantity of the items and suppose if the stock is not available for any items, "stock not available" message is also shown.

4. Checkout and Payment:

Checkout is made in an easy step to avoid hassle in this application. The user can just check in with the checkbox from the physical shopping cart. Depending upon the quantity of the items, the payment is calculated along with the tax rate (Ex: 8%). A separate payment screen is displayed which confirms the purchase. If the user wishes to purchase, they can proceed by clicking the "PayNow" button as shown in the Figure 16, else they can cancel the purchase at this stage also proceed and with shopping other items or they can exit the application. I have used PayPal [10] as a payment gateway in this project. It is widely used by many applications in recent days and easy to use. PayPal supports many applications in providing their gateway to use it. To implement this payment gateway in our application, we should add the PayPal Android SDK library into our project. There are two ways in which we can use and test this gateway. First, the “sandbox” mode in which it acts as a real environment that enables the users to use the gateway as shown in Figure 17. However, in sandbox mode the amount is not detected when we test the transaction process. It offers people the environment to know how the PayPal actually works. To use sandbox account, we need to create a developer account in PayPal and then sandbox accounts (test accounts). After creating the test account, we can create an application and generate a unique client id for that application and this is required to use the gateway by including in the project. Second, “live” mode, the real environment for processing payments. To use live mode also we should follow the same steps as sandbox mode. The reason for choosing sandbox mode over live mode for this project is, to avoid real time transactions as it detects money from my account. Also, the gateway provides two options for making the payment. We can either pay with our PayPal account or using our card as shown in Figures 18, 19, 20. We should just scan our credit/debit card, so that the details are fetched automatically and the payment is done. Once the payment is made, confirmation screen is displayed, which includes the Payment Id, status and amount as shown in the Figure 21.

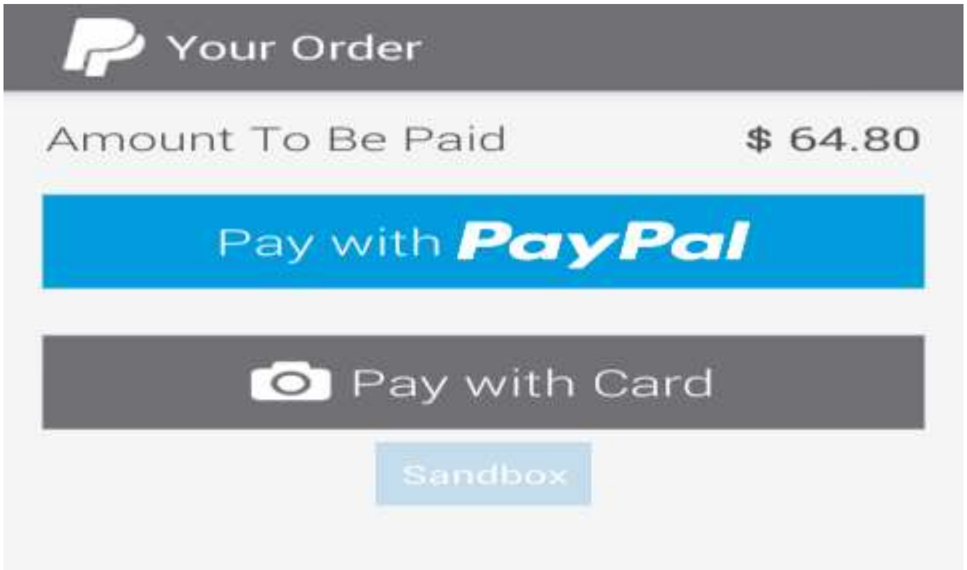


Name	Price	Quantity
Kit Kat	20.0	3

Tax amount:4.8

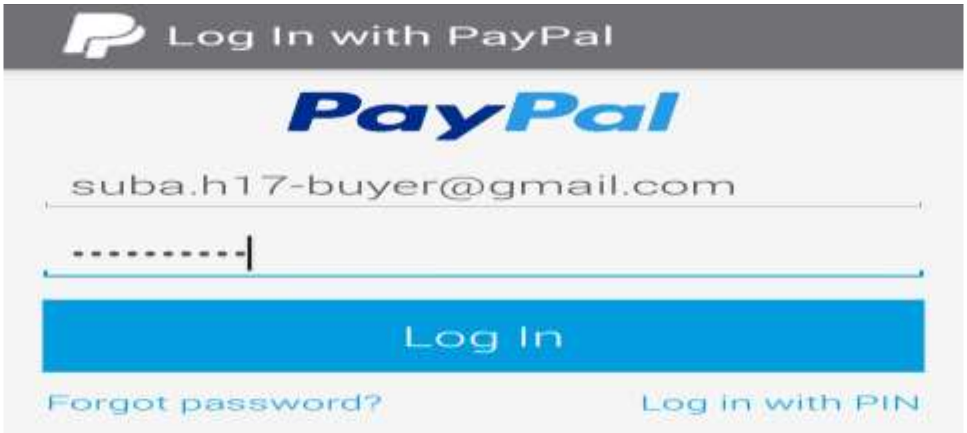
Total amount:
64.8

Figure 16: PayNow Screen



Amount To Be Paid $ 64.80

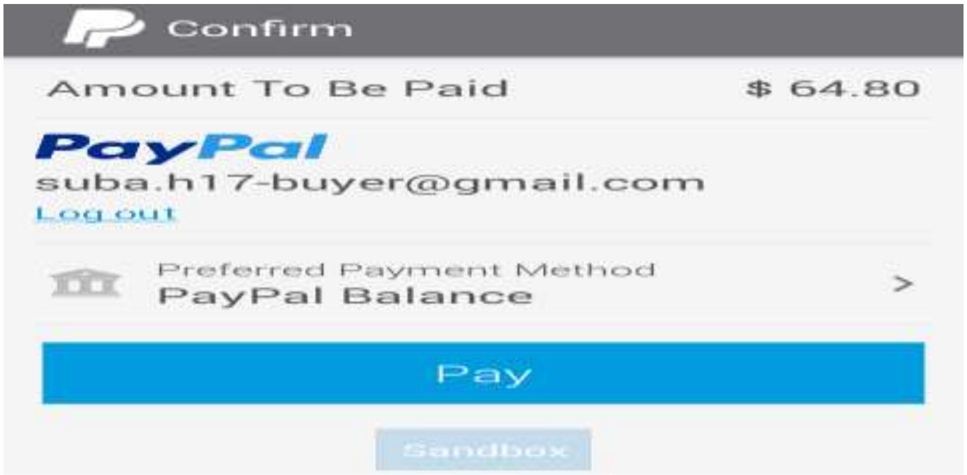
Figure 17: Payment via PayPal



suba.h17-buyer@gmail.com

••••••••••

Figure 18: Payment via PayPal Account Login

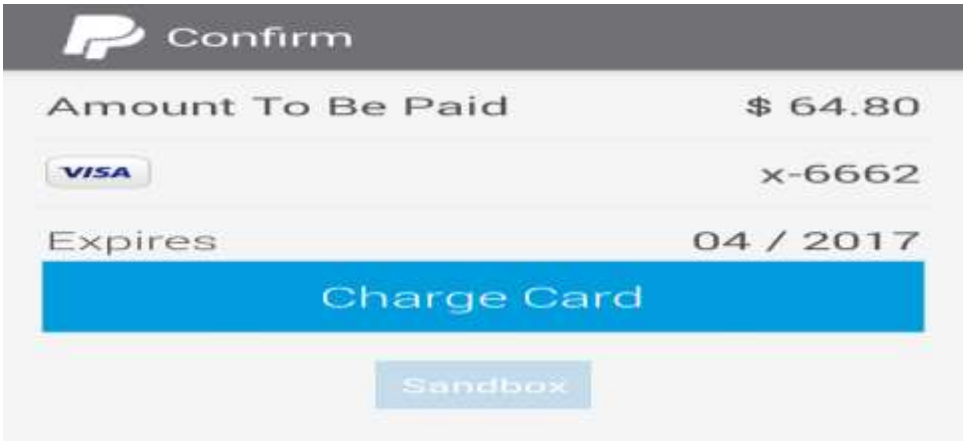


Amount To Be Paid $ 64.80

suba.h17-buyer@gmail.com

Preferred Payment Method
PayPal Balance

Figure 19: Payment via PayPal Account

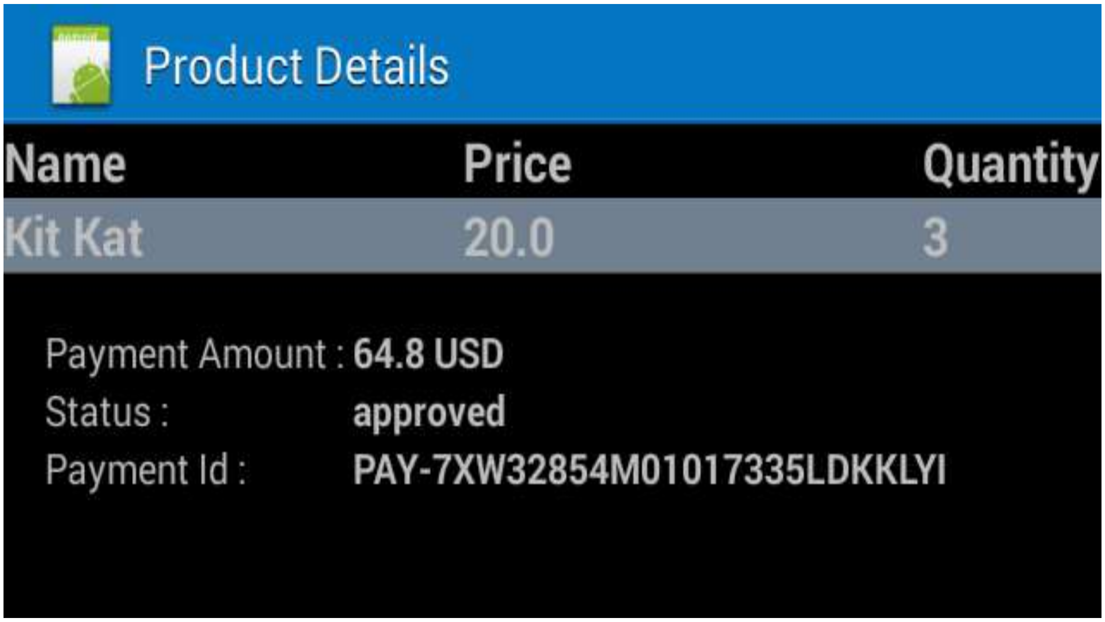


Amount To Be Paid $ 64.80

VISA x-6662

Expires 04 / 2017

Figure 20: Payment via PayPal Card



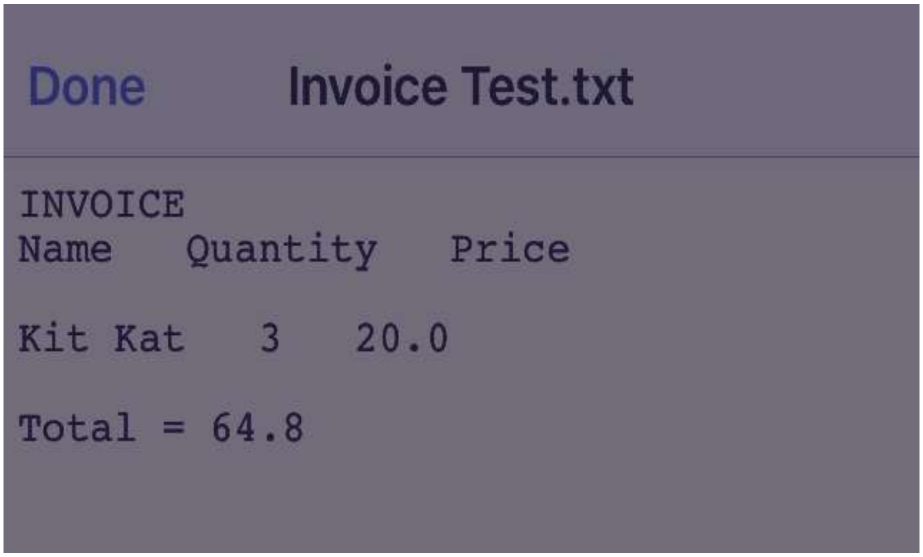
Name	Price	Quantity
Kit Kat	20.0	3

Payment Amount : 64.8 USD
Status : approved
Payment Id : PAY-7XW32854M01017335LDKKLYI

Figure 21: Payment Confirmation Screen

5. Invoice Generation and Uploading:

When the user confirms the purchase and pays amount, invoice is generated as shown in Figure 22 on time and we can share it instantly to cloud (share to drive, send email). The invoice is generated as a text file which includes all the items the user purchases with amount. The invoice is later could be shared in drive or we can send a mail copy of it.

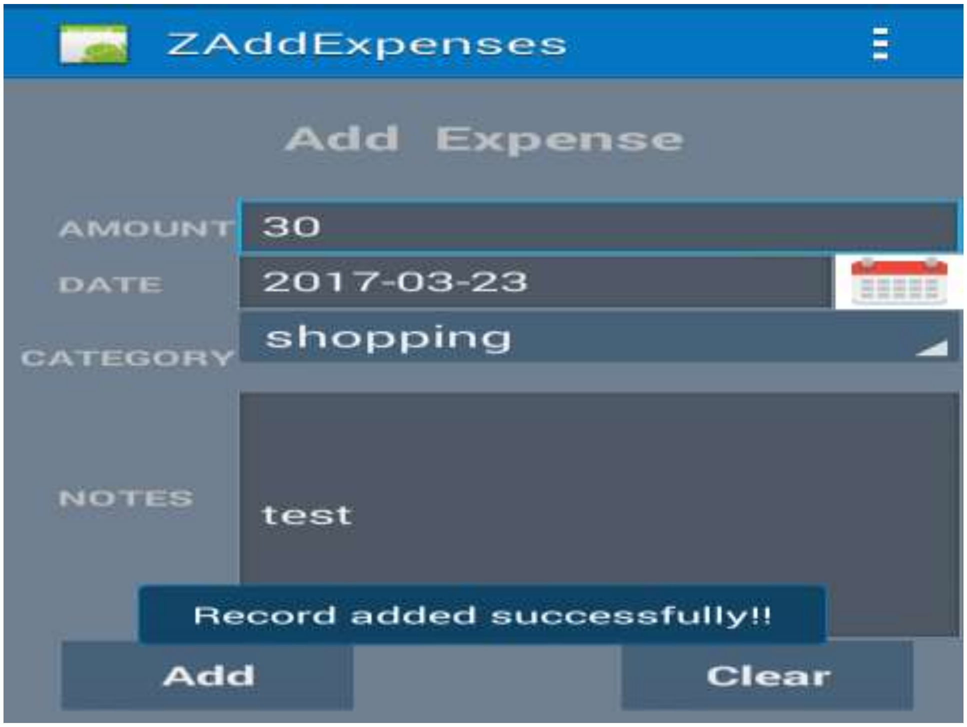


```
INVOICE
Name   Quantity   Price

Kit Kat   3   20.0

Total = 64.8
```

Figure 22: Invoice

6. Tracking Purchase:

ZeroQ also adds the expenses we spent for shopping and for any other categories every time and track purchases in a visualized manner by generating charts- pie chart [11] when we add the expense bills as shown in Figure 23 .This also helps people to plan their budget wisely and restrict them by spending on redundant things. Viewing expenses like Figures 24 and 25 are made interactive by choosing different options like view by category, by and between dates, or view all.



AMOUNT 30

DATE 2017-03-23

CATEGORY shopping

NOTES test

Record added successfully!!

Figure 23: Adding Expenses Screen

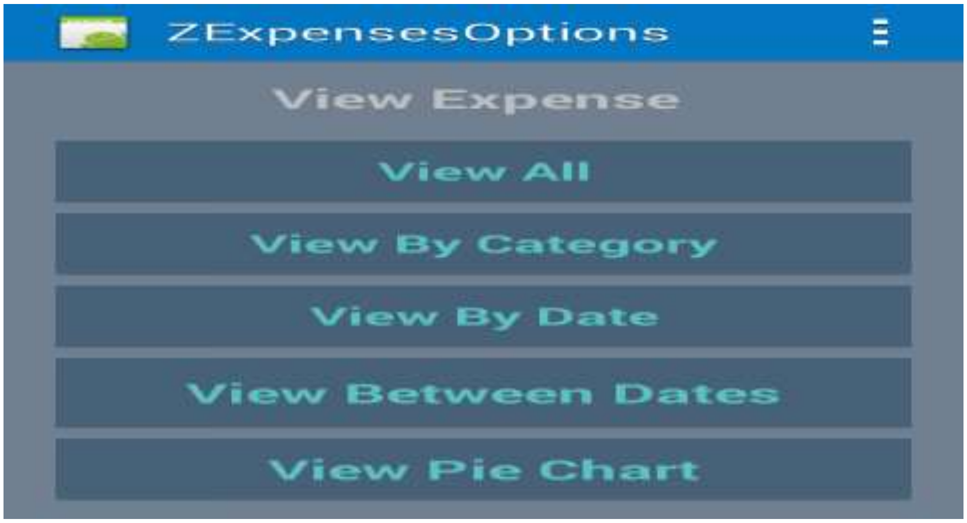


Figure 24: Viewing Expenses Screen

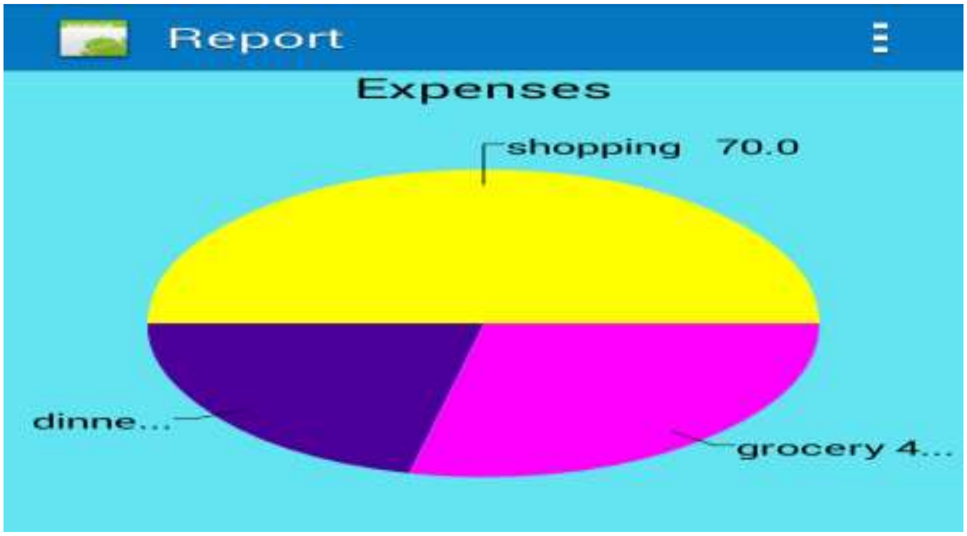


Figure 25: Viewing Expenses Charts

6. FUTURE WORK

There are several features of the project that would be quite challenging to be added. The desirable features that ZeroQ could have are,

- Including visual tour (location of items) inside the store to find items quickly using Google Map and Indoor Map API. It will help us in saving time and being tired too on searching for the items.
- Another feature is allowing the application to read RFID tags which are attached along with the barcode tags in many clothing stores. The RFID tags could be read using NFC (Near Field Communication) which allows the users to exchange data between two devices and read tags. Also, this could be used for making payments by turning on the NFC feature in our devices and make reading the credit/debit cards. To use this feature, our device should support NFC feature and the store owners should also agree to take the RFID tags along with the items after we scan and buy the item as many of the tags are quite expensive. For cheaper tags, if they agree, we can implement this feature.
- Moreover, implementing this application in iOS platform as an iPhone App is one of the important considerations as we have many iPhone users.

7. CONCLUSION

I have learned a lot from this project on how to develop Android Application and publishing it in real time, use Web Services using SOAP UI, reporting using chart libraries, other libraries for scanning, payment, SDKs, requirement gathering, barcode, QR, RFID formats.

As mentioned, other existing applications does not help in avoiding people to stand in a long checkout line, instantly searching about availability of products, tracking purchases (all together in a single application). If people use ZeroQ in the future, they have several advantages which includes, easy checkout, payment, sharing invoices instantly as it is quite easy to misplace paper bills and organizing them in easily, particularly helpful for elderly people by avoiding them to wait for a long time in the checkout line, providing quick information about items available, tracking purchases which helps users to budget their expenses wisely and finally stores can reduce scanning machinery and power consumption.

Bibliography

[1] J. Goodman, "Grocery Shopping: Who, Where and When," October 2008. [Online]. Available: http://timeuseinstitute.org/Grocery%20White%20Paper%202008.pdf. [Accessed 16 April 2017].

[2] "Agile Methodology," 4 July 2013. [Online]. Available: https://www.codeproject.com/Articles/616070/Agile-Methodology. [Accessed 16 April 2017].

[3] P. Jain, "What Makes Java a Powerful Programming Language," 11 February 2013. [Online]. Available: https://www.weblinkindia.net/blog/what-makes-java-a-powerful-programming-language. [Accessed 16 April 2017].

[4] A. Rongala, "Benefits of Java over Other Programming Languages," 7 May 2015. [Online]. Available: https://www.invensis.net/blog/it/benefits-of-java-over-other-programming-languages/. [Accessed 16 April 2017].

[5] "Java SE Downloads," [Online]. Available: http://www.oracle.com/technetwork/java/javase/downloads/index-jsp-138363.html. [Accessed 16 April 2017].

[6] "Android Developers," [Online]. Available: https://developer.android.com/index.html. [Accessed 7 April 2017].

[7] "SoapUI," [Online]. Available: https://www.soapui.org/. [Accessed 31 March 2017].

[8] "Crashlytics, Fabric," [Online]. Available: https://fabric.io/kits/android/crashlytics/features. [Accessed 10 April 2017].

[9] C. Aliferi, "Android Barcode and Qr Scanner Example," 2 December 2014. [Online]. Available: https://examples.javacodegeeks.com/android/android-barcode-and-qr-scanner-example/. [Accessed 12 April 2017].

[10] B. Khan, "Android Paypal Integration Tutorial," 1 May 2016. [Online]. Available: https://www.simplifiedcoding.net/android-paypal-integration-tutorial/. [Accessed 15 March 2017].

[11] "Android Pie Chart using MpAndroidChart Library Tutorial," Numetric Technologies, 29 January 2016. [Online]. Available: Android Pie Chart using AchartEngine Library Tutorial. [Accessed 16 April 2017].

[12] P. Kapoor, "E-NOTICE APPLICATION For Android Phones," 30 May 2014. [Online]. Available: https://www.slideshare.net/kapoorpriyanka/report-on-enotice-app-an-android-application. [Accessed 12 April 2017].