Design and Implementation of the Server CashOut Mobile Application

Tonbara Atte, *Student, University of Regina, attestephanie@gmail.com, 200396187*

***Abstract*—According to research by ctvnews.ca in 2012 [1]. Almost 200,000 people in Canada are servers. We will assume that the number would have increased over seven years. Servers are tasked with handling payments for Cash, Debit/Credit and often it is harder to reconcile tabs at the end of the shift and also keep track of tips for tax filing. This document details the design and implementation of a mobile application for servers and waitresses around the country, the application is called the Server CashOut.**

This application helps the server to cash out at the end of their shifts easily without making mistakes or stressing over calculations. The application uses the Firebase API database to store, read and write data. The Server enters the required data and the application shows cash out report where the user can view Cash Out Values. The development phase was successful as we were able to implement most of the features laid out in the proposal. However, we had some setback, which is discussed further in the report.

I. Application Description

THE Server cashout was built as a solution to aid waitress/servers at restaurants accurately and efficiently cash out at the end of their shift. The Server CashOut is built on the current cash out system of the East Side Marios restaurant, but it can be adapted to any system used by any restaurant. The application works by taking inputs like the gross receipts (which is the total of all orders taken by the server throughout their shifts excluding tax), and also cash due. Cash due is the amount of cash the restaurant expects from the server, If the amount is positive, then the server owes the restaurant, if negative the restaurant owes the server. At East Side Marios, the server is required to tip out to the Host staff, Kitchen staff and also the bar.

The Hosts and kitchen tip is included in the cashout to the manager while the bar tip is given directly to the bar. Atmost the server is only given the information on the gross receipts and the cash due, they are required to calculate the cash out and tip out by themselves. It may be an issue for someone who is not good with maths or feels like its is a time consuming task. That being the need for the proposal to develop this application, so cash out can be very easy and swift.
Application feature summary

- CashOut feature
- TipOut feature
- Tip take home summary

II. Design

This application will be developed using Kotlin language on the Android Studio Integrated Development Environment. The database/ backend will be managed using the Firebase API. Firebase allows users to read, write and support data. The database is a RealTime database and the data is stored as soon as it entered the application. For this application, the database rules are public and can be accessed by anybody. The application will be tested on the Google Pixel 2 Android Virtual Device.

Once the user launches the application they are met with a landing page, which is a form that takes Gross Receipts, Cash Due and Cash on Hand as input. This form was implemented using EditText, TextView and a button(Cash Out). An onClickListener() is placed on the Cash Out button, once the button is clicked, the program checks if any of the three input fields is empty using function isEmpty(), if any is empty it will throw an error message otherwise the text is extracted from the EditText and Converted to a String and then a Double.

In the program, we then calculate values for :
CashOut =(3% of Gross Receipts) + 0.50 + cashDue
TipOut= (1% of Gross Receipts)
TakeHome= Cash_On_hand -/+ cash_compute
TakeHome Percentage= (TakeHome/Grossreceipts) *100

After entering the gross receipts, Cash Due and the cash on hand into the Server CashOut app, the server instantly gets the cash-out value, tip out and the take home amount. After the values are entered and onclick() of the Cash Out Button. The values from each event is given a unique ID using the push.key(). Take Home, Take Home Percentage, Tip Out, Cash Out together with the keys are stored on the Firebase database. This launches another activity, where the report of the calculation is displayed.

To demonstrate how the application works, take for example the scenario below: A Server heard about this new server cash out application and would like to try it out.

- Step 1:Enter the following data.
  - Gross Receipts= $51
  - Cash Due= $16.95
  - Cash on Hand =$25
  - takeHome=5.51
- Step 2:User Clicks CashOut
- Step 3:Displays the following data

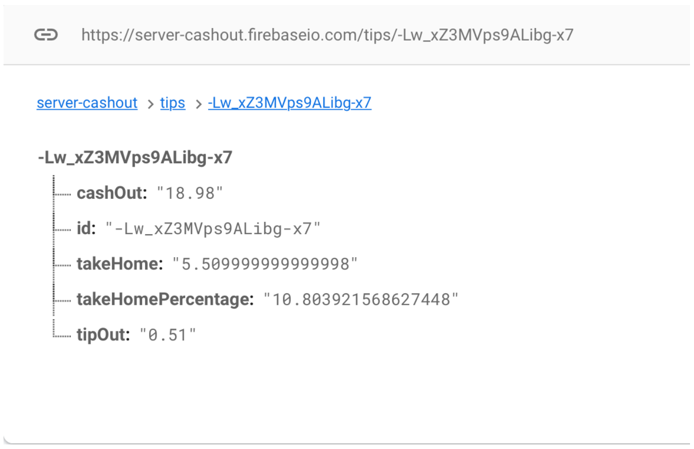
III. Limitations

- In the proposal, one of the key features, we tried to implement was the history feature, where users can keep

track of their take home tips. Although, we are able to store data of the firebase database . We could not retrieve data as time was a factor.
- Also in the application, anybody can access the data because the database rules are set to public. This is a security concern especially if the application will be developed on the enterprise level.

IV. Conclusion

In this report, we have been able to document the design and implementation of the Server CashOut application. Our main objective for the development of the application was to ease the cash out process for server and we have been able to fulfill that objective. For the future implementation, the designer should consider the incremental development of the history feature.



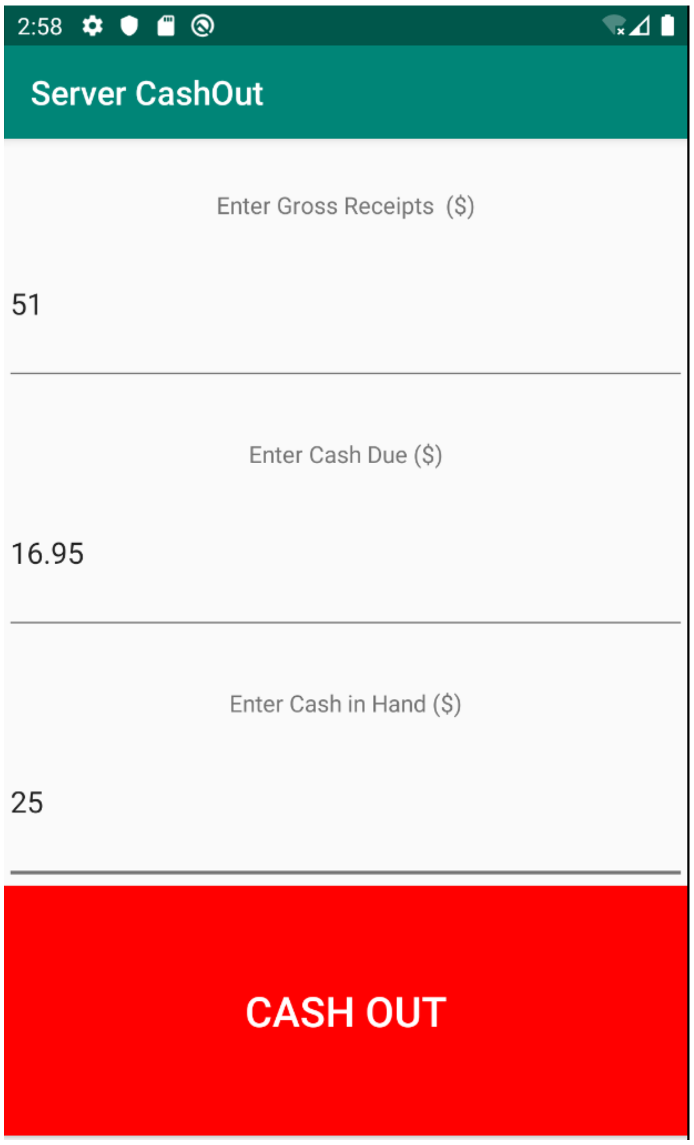
https://server-cashout.firebaseio.com/tips/-Lw_xZ3MVps9ALibg-x7

server-cashout > tips > -Lw_xZ3MVps9ALibg-x7

```
-Lw_xZ3MVps9ALibg-x7
├ cashOut: "18.98"
├ id: "-Lw_xZ3MVps9ALibg-x7"
├ takeHome: "5.509999999999998"
├ takeHomePercentage: "10.803921568627448"
└ tipOut: "0.51"
```

Fig. 2. Screenshot of data on database

Appendix



2:58

Server CashOut

Enter Gross Receipts ($)

51

Enter Cash Due ($)

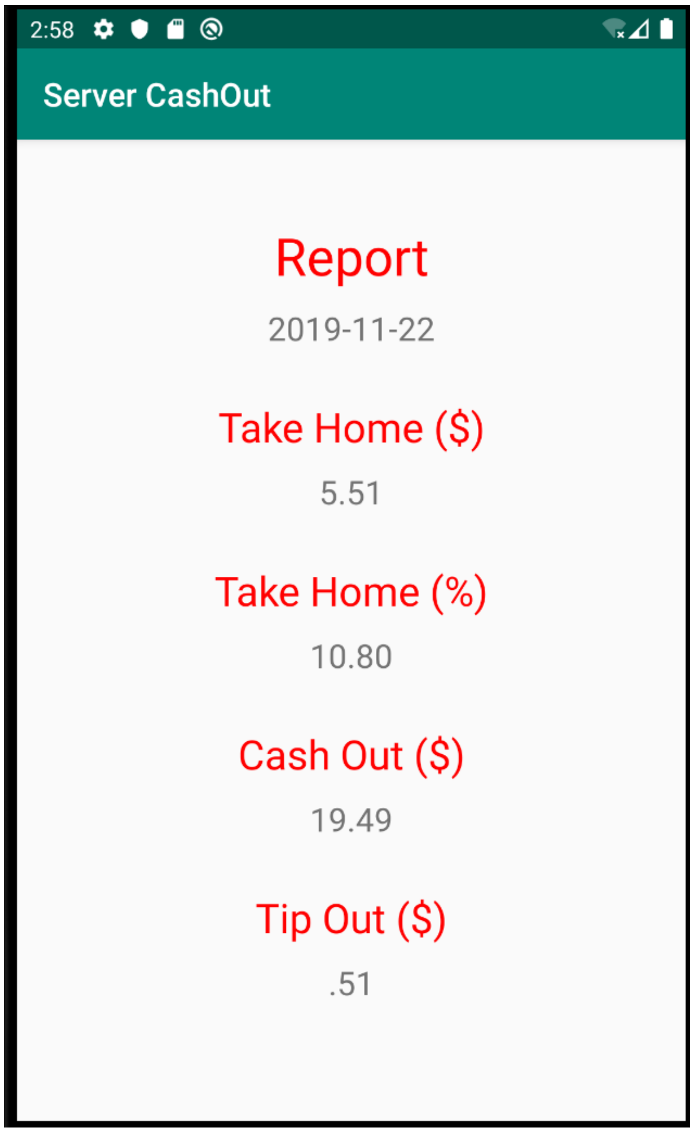
16.95

Enter Cash in Hand ($)

25

CASH OUT

Fig. 1. Screenshot of the Main Activity



2:58

Server CashOut

Report

2019-11-22

Take Home ($)

5.51

Take Home (%)

10.80

Cash Out ($)

19.49

Tip Out ($)

.51

Fig. 3. Screenshot of the Report

REFERENCES

[1] CTVNews. (2012). Tax crackdown finds many servers not declaring tips. [online] Available at: https://www.ctvnews.ca/canada/tax-crackdown-finds- many-servers-not-declaring-tips-1.870411 [Accessed 11 Nov. 2019].

[2] GitHub. (2019). tonbara-stephanie/ServerCash. [online] Available at: https://github.com/tonbara-stephanie/ServerCash [Accessed 19 Dec. 2019].