Scientific Voyage
Vol. 1 No. 2
Page 39-48 (2020)

SCIENTIFIC VOYAGE
ISSN: 2395-5546

GCECT
Publication
Apr.-Jun.

Real-time invoice generation using Raspberry Pi on android platform

Aniruddha Sarkar[1] and Arkadip Ray[2]

[1]*Department of Computer Science and Engineering, Government College of Engineering & Ceramic Technology, Kolkata 700010, WB, India*
Email: sarkar4540@gmail.com

[2]*Department of Information Technology, Government College of Engineering & Ceramic Technology, Kolkata 700010, WB, India*
Email: arka1dip2ray3@gmail.com

***Abstract*:** Human operator based retail invoicing systems have put data accumulation, bill printing and real-time data processing into practice in retail outlets and supermarkets. These systems provide real-time centralised billing and provide invoicing facility; however can seldom be implemented in small and medium scale shops, due to size and implementation cost. The primary objective is to design a system, suitable for small and medium scale shop owners, which will allow the basic invoicing process – invoice generation, invoice printing and enlisting of invoices for further usage. In addition, the application will provide a fast and simple user experience, removing the requirement of a skilled operator.

Keywords: Android application; Socket protocol, MySQL database, Middleware server, Raspberry Pi

1. Introduction

Digital invoice generation, printing and storage systems are one of reliable and consistent systems used to keep track of the sales of a retail outlet or a store. It is relatively hassle-free and easy to use than maintaining pen-and-paper book of records for billing and invoices.

However, the available digital invoicing systems are designed for large-scale retail outlets and usually requires dedicated skilled operators at the terminals for data entry. Also, these systems come with significantly large price tag, due to hardware. Thus, for addressing these problems, the solution must be able to perform the following:

(i) Create invoices and provide real-time calculations, (ii) Provide user experience for faster checkouts, such that the storekeeper himself can operate the system in small and medium scale stores, (iii) Store invoices and enlist them later, (iv) Allow to export the collected data, (vi) Provide invoices in a digital printable document format, such as Portable Document Format (PDF) and (vii) Uses cost-effective hardware solutions.

In this paper, we have created one such digital invoicing system, which is usable in small and medium scale retail outlets and stores, and is able to perform the aforementioned. We have leveraged the Android platform in the system, as it is one of the major smartphone platforms around the world and has already reached millions, which include shopkeepers and retail outlet cashiers. This eliminates the possibility of buying costly hardware, thus reduces the cost of implementation. An Android Application is developed to provide the interface to use the system. The application is straightforward and can be used to create, store, enlist and print invoices. For faster checkouts, the application allows entry of items using barcode scanning and retrieval of information of previously added items. It communicates to a Middleware Server, to store, fetch and print invoices. The Server is integrated with MySQL database for storing and retrieving data. MySQL database provides additional functionalities for exporting data and connecting to other systems, which enables further analysis of the data. The Server, along with MySQL database and other components, is deployed on Raspberry Pi running Rasbian Operating System, which is connected to an Android smartphone using a Wi-Fi network. The server uses CUPS Printing Server to connect a printer with Raspberry Pi. The server stores the printed invoices in PDF format for digitally sending the documents to customers.

2. Related research works

This section provides the literature review of existing systems related to this research topic. Sung[1] presents the major players and analysis of global smartphone market. The paper also describes that smartphone market growth is overall constant and has already reached billions. Husni et al.[2] proposes Shopping Application System that uses NFC and RFID technology to allow the customers to have cashless billing using their Android smartphones. Samal et al. [4]

recommends a Shopping Application for customers to generate improved billing and cart management. Sainath et al. [3], suggests 1D-barcode scanning in a shopping cart and Sancheti et al. [5] uses 2D-barcode scanning in a shopping cart, to automate the billing process of a supermarket. Krishnamoorthy et al. [6] purports IoT technology to automate the shopping experience. These systems, due to their heavy customer interaction, are error-prone.

Sachdeva et al. [7] presents a review paper on Raspberry Pi, which discusses its benefits and usage. Dhage et al. [8], proposes Raspberry Pi as a Bluetooth enabled printer adapter. Krishnamoorthy et al. [9] introduced automated shopping experience based on embedded sensors using Raspberry Pi with ultrasonic sensor. Pangasa et al. [10] analysed the main components of automation in billing systemlike scanning system, microcontroller and transmission medium. Megalingam et al. [11] introduced automatic billing of products using an RFID reader.

3. Proposed system

This section describes the peripherals used, overall architecture of the system and the components of the proposed system, namely – Android Application, Socket Protocol, MySQL Database, Middleware Server and CUPS Print Server. Configuration of Raspberry Pi is also discussed here.

3.1 Materials

The peripherals required to build the proposed system are – Raspberry Pi, Android smartphone and Printer.

3.1.1 Rasberry Pi 3 B+

Every embedded system requires a Computing Unit, which is a Central Server or a specially designed Computer, for storing data, running various algorithms on the collected data and printing the data. The computing unit used in this system is Raspberry Pi3 B+. It is a fully-fledged computer with HDMI, USB, Ethernet, Wi-Fi and many other features packed into one business card-sized board. It is powered through a micro-USB port, using 5V 2A adapter. It contains an ARM-based SOC whichstores data in an 8GB microSD card and runs on Raspbian OS which is a Linux distribution containing X11 server and package management.

3.1.2 Printer

The proposed system requires a printer, for printing invoices or bills for customer. The printer used in this system is HP LaserJet Pro P1108. It is a monochrome Laser printer and is connected using USB technology.

3.1.3 Android smartphone

As of 2019, over 24,000 various types of Android smartphones are available in the market. These phones appear in various sizes as well as shapes, and comes with variety of hardware. These are usually equipped with touch screen. The benefit of an Android smartphone comes with the Android platform, which allows applications targeted for the platform to run on almost all of these devices. The Android smartphone we have used is Asus Max Pro M1.

3.2 Overall architecture

The android application has been used as an interface to control the system. It provides options to the user, based on the selection of which, the application sends data to the Middleware Server using the Socket Protocol. The Middleware Server communicates with the Android Application to receive data from the user, which is stored in the MySQL database and is sent to the printer. Fig 1 describes the overall architecture of the proposed system.A MicroSD card, with capacity of 8 GB has been used to store the Raspbian Operating System, Middleware Server, MySQL database, the Shell Scripts and CUPS server. The MicroSD card is inserted into the relevant slot in Raspberry Pi and powered on. An android phone is connected, using Wi-Fi, to the Access Point created by configuring the Raspberry Pi, thus enabling Socket Connection between the Android Application and the Middleware Server. The Printer is connected to the Raspberry Pi using a USB cable.

3.3 Android application

Android Platform supports a multitude of devices, and targeting the platform to build application allows us to target these devices. Applications targeted for the platform can be created using the Android Software Development Kit or SDK. The SDK contains all toolchains required to create an Android Application. Android Studio is an Integrated Development Environment which provides various GUI features to control the Android SDK and allows rapid development. Fig 2 describes the flowchart of the Application. The application allows the shopkeeper or, the salesperson to create, save, view and print invoices. The application also allows fast retrieval of information regarding previously entered products using serial numbers. The serial numbers can be encoded in 1D as well as 2D barcodes. The application uses Zxing library to decode barcodes captured using the device's camera.

3.4 Socket protocol

Socket Protocol is based on the Transmission Control Protocol, and used to communicate between different applications. Socket allows communication between applications using IP addresses and port numbers. Socket follows the Server-Client model of communication, where one server can respond to multiple clients. The clients initiate the requests and the server responds.In this system, we have used the Java Standard Library's Socket Server and Socket classes to allow communication between Android Application and Middleware Server.

3.5 MySQL database

In this system, MySQL Database server has been used to store the data of customers, invoices and products. MySQL database is an open-source Relational Database which allows operations using Structured Query Language. A database is created in the MySQL server and two tables are created in it – Bill and Product. MySQL server exposes its API in multiple languages, thus one can use the APIs to perform further data analysis and operations on the collected data. The table schema used for creating Bill and Product table are shown in Figs. 3 and 4, respectively.

3.6 Middleware server

The Middleware server, which connects to the Android Application, MySQL server and Print Server, controls the flow of data among these components. The server has been written in Java language. Java provides a huge Standard Library, of which we have primarily used the Java Socket API and the Java Printing API. The Server creates a Socket Server instance and waits for the Android Application to make a connection. On every new connection, its input stream is read for data and command. If the command is known, the Middleware Server performs the respective operation and writes to the output stream of the connection. The operations allowed by the server are – saving and printing bill, enlisting previous bills and getting product description by serial. In Fig. 5, the flowchart describes the operations performed by the Middleware Server.

To save a bill, the server reads the customer and product details from the Socket. It then connects to MySQL server using Java Database Connection (JDBC) library and stores data into tables. To print a bill, the server reads the customer and product details from the Socket and translates it to Hyper Text Markup Language (HTML). It then uses the Flying Saucer Library's PDF printing to convert the translated data to a Portable Document Format (PDF) file. Finally, using the Java Printing API, the PDF file is sent to the CUPS print server.To enlist previous bills, the server queries the MySQL database using JDBC library and fetches the previous bills. It then writes the data queried, to the output stream of the Socket.To get product description by serial, the server reads the serial from the Socket's input stream and queries the MySQL database using JDBC library, whether the serial exists or not. If the serial exists, the product description is written to the Socket's output stream.After every operation, an acknowledgement containing the success or failure of the operation is prompted to the Socket's output stream.

3.7 Common Unix printing system (CUPS)

CUPS or Common UNIX Printing System is an Internet Printing Protocol based server, which receives print requests from other processes and sends them to the respective printer. CUPS server is required for the Java Printing API to work. Along with CUPS, the respective driver, which can control the printer, must also be supplied. The CUPS is configured for HP LaserJet Pro P1108, by editing the configuration file for the CUPS server.

3.8 Configuration of RASPBIAN operating system

Raspbian OS is the official Operating System for Raspberry Pi and is shipped as a disk image file with an .img extension. It is supplied with X11 Server, which enables Graphical User Interface by connecting a monitor. One can write this image to a microSD card and use the same to run the Raspbian OS in Raspberry Pi.Java Runtime Environment 8, MySQL Community Server 8.0 and CUPS Server, all compiled for ARM 64-bit architecture, is installed within the OS. The compiled Middleware Server is then copied into the system. MySQL server is configured and the configuration to use the internal Wi-Fi module as a Access Point is written. The Access Point is configured to be password-protected using WPA2-PSK.

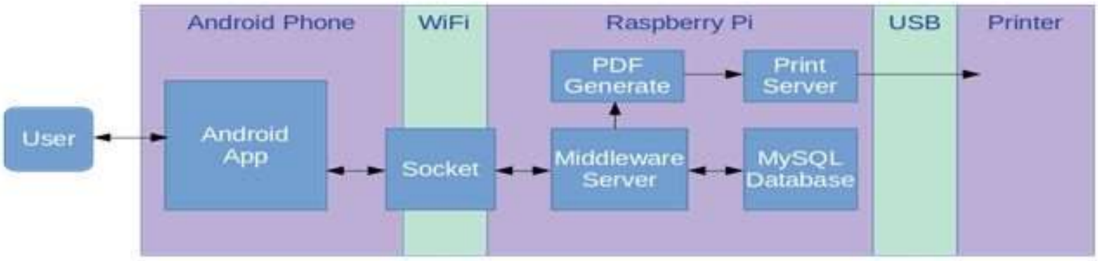


Fig. 1: Overall architecture of proposed system

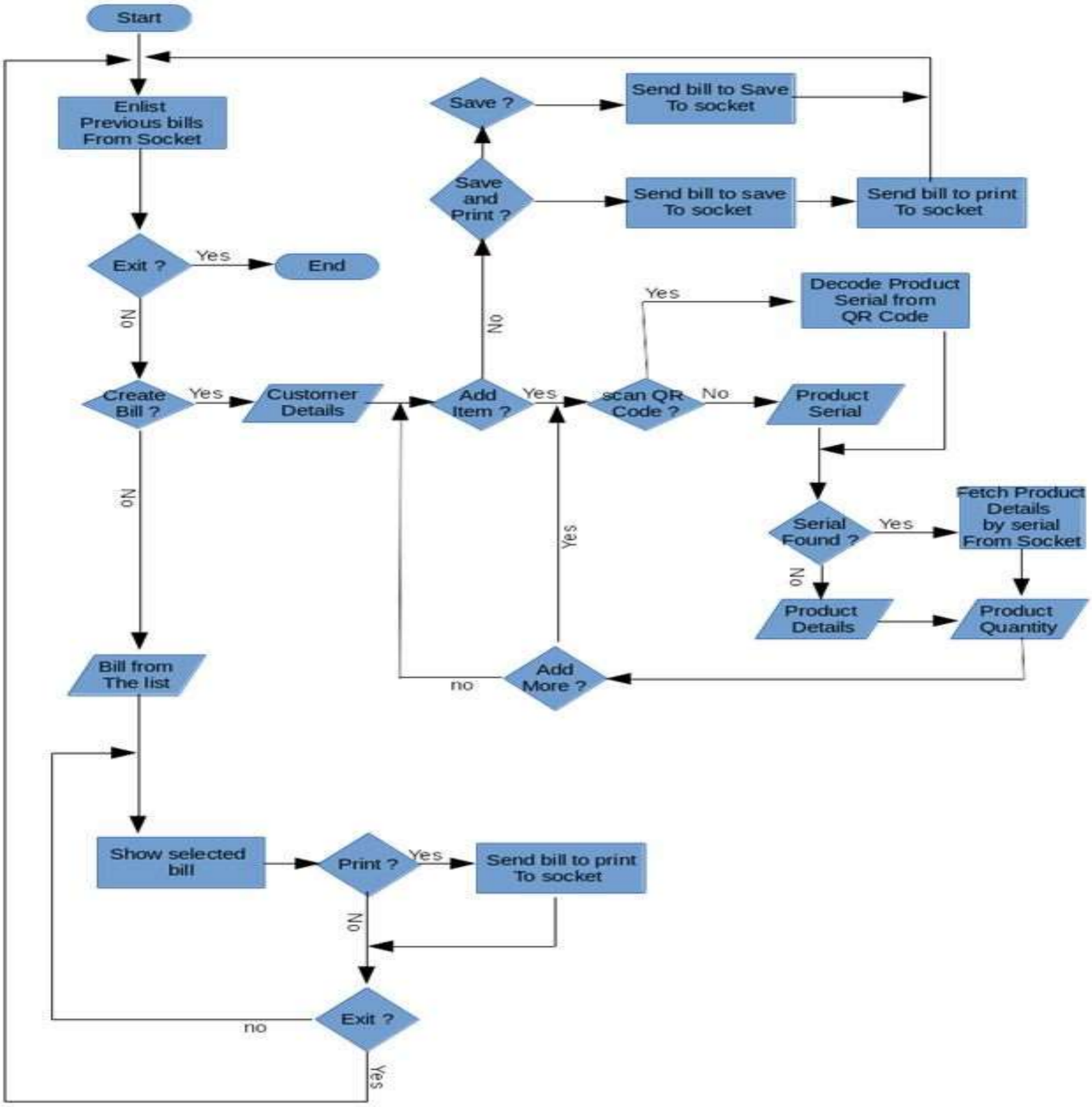
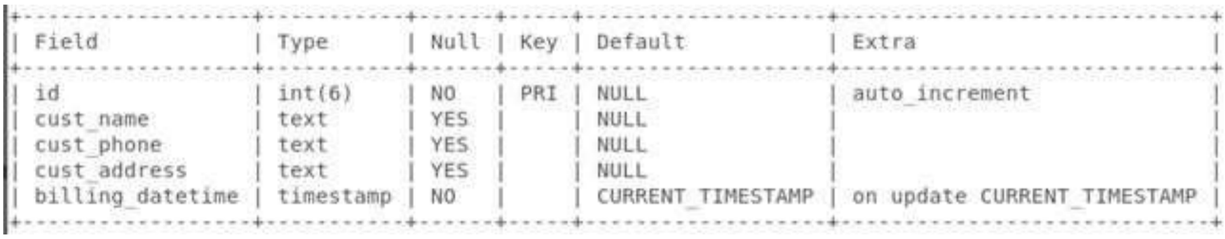
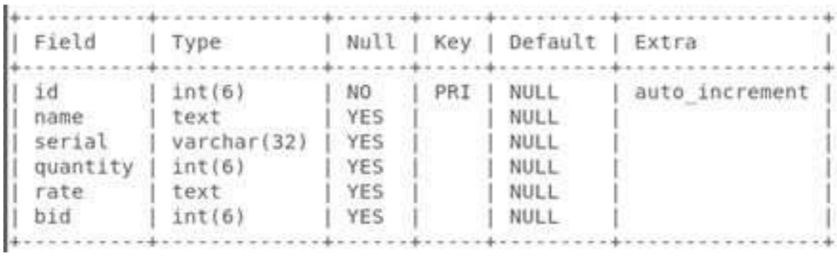


Fig. 2: Flowchart of android application



```
+------------------+-----------+------+-----+-------------------+-----------------------------+
| Field            | Type      | Null | Key | Default           | Extra                       |
+------------------+-----------+------+-----+-------------------+-----------------------------+
| id               | int(6)    | NO   | PRI | NULL              | auto_increment              |
| cust_name        | text      | YES  |     | NULL              |                             |
| cust_phone       | text      | YES  |     | NULL              |                             |
| cust_address     | text      | YES  |     | NULL              |                             |
| billing_datetime | timestamp | NO   |     | CURRENT_TIMESTAMP | on update CURRENT_TIMESTAMP |
+------------------+-----------+------+-----+-------------------+-----------------------------+
```

Fig. 3: Table schema for bill table



```
+----------+-------------+------+-----+---------+----------------+
| Field    | Type        | Null | Key | Default | Extra          |
+----------+-------------+------+-----+---------+----------------+
| id       | int(6)      | NO   | PRI | NULL    | auto_increment |
| name     | text        | YES  |     | NULL    |                |
| serial   | varchar(32) | YES  |     | NULL    |                |
| quantity | int(6)      | YES  |     | NULL    |                |
| rate     | text        | YES  |     | NULL    |                |
| bid      | int(6)      | YES  |     | NULL    |                |
+----------+-------------+------+-----+---------+----------------+
```

Fig. 4: Table schema for product table

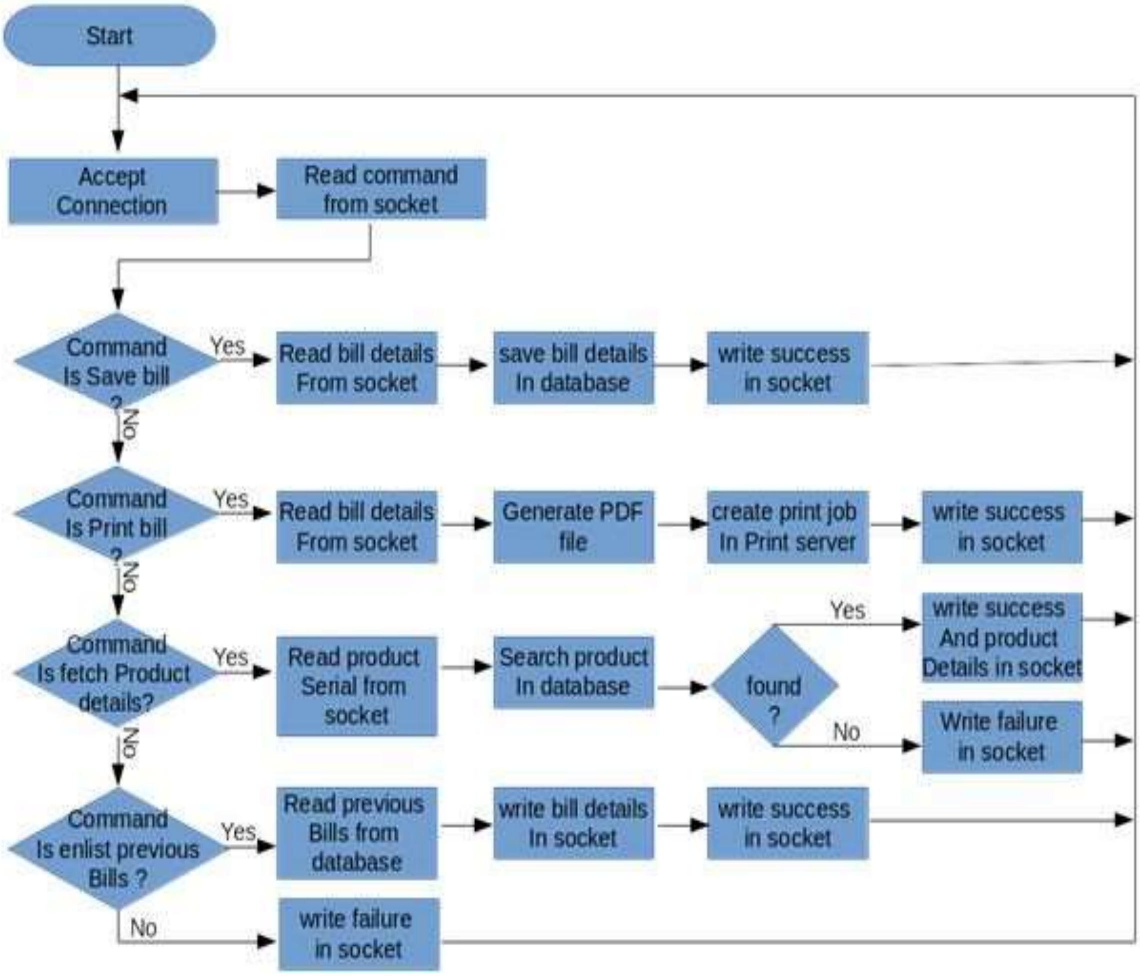

Start
Accept Connection
Read command from socket
Command Is Save bill ?
Yes
No
Read bill details From socket
save bill details In database
write success in socket
Command Is Print bill ?
Yes
No
Read bill details From socket
Generate PDF file
create print job In Print server
write success in socket
Command Is fetch Product details?
Yes
No
Read product Serial from socket
Search product In database
found ?
Yes
No
write success And product Details in socket
Write failure in socket
Command Is enlist previous Bills ?
Yes
No
Read previous Bills from database
write bill details In socket
write success in socket
write failure in socket

Fig. 5: Flowchart of middleware server

4. Analysis of testing results

This section describes the analysis of testing performed on the proposed system. The system is tested using a single smartphone, however, multiple smartphones can be simultaneously connected to the system.

4.1 Expense calculation and invoice creation

For creating invoices, the application asks for the customer details. Then for adding item, the item details can be either fetched by using serial number or can be provided directly in the input fields. After adding items, the quantity of an item can be adjusted accordingly, or can be made zero to remove the item. The total price of items in the cart is automatically calculated upon selection of items. An invoiceis generated by clicking the “Create Bill” button. Fig. 6(a) shows the application home window without any previously added invoices. Fig. 6(b) shows entry of data regarding invoice and cart expense calculation.

4.2 Storing invoices

The application sends the collected data to the Middleware server, which then updates the relevant table values in the MySQL database. One can store the created bill by clicking on the “Save” button. One can view the existing invoices by clicking on a previous bill entry in the home window of the application. Fig. 6(c) shows the response after clicking on “Save” button. Fig. 7 shows the MySQL database entries created after addition of an invoice. Fig. 8(a) shows the application home window after addition of an invoice. Fig. 8(b) shows a previously added invoice.

4.3 Printing invoices

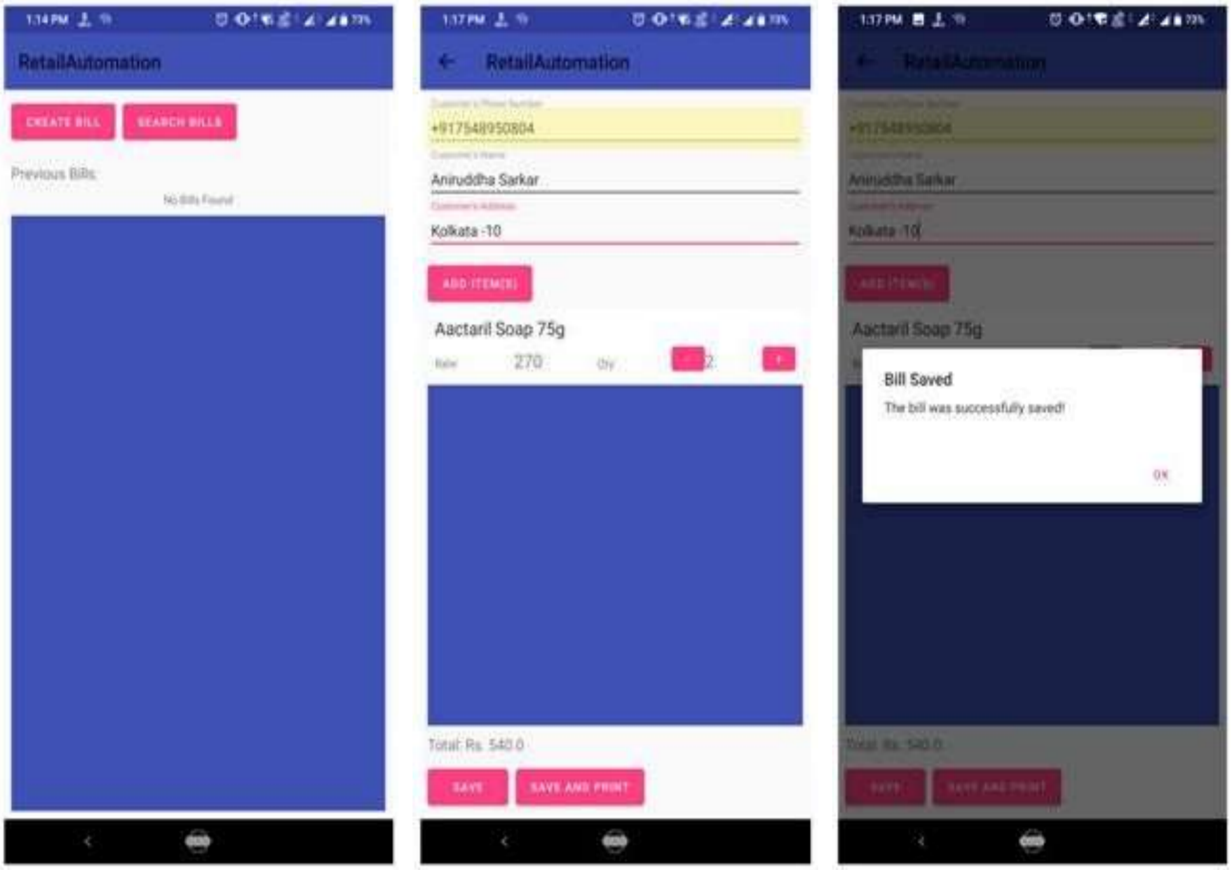
The application sends the collected data to the Middleware server, which then generates a PDF document and sends the file to the CUPS server to initiate a Print Job. One can print the created bill by clicking on “Save and Print” button, or can print an existing bill by opening the bill and clicking on “Print” button as shown in Fig. 8(c). Fig. 9 shows the document sent for printing.

4.4 Faster checkout using barcode scanning

Barcode scanning has been implemented in the application using Zxing Library, which support 1D as well as 2D barcodes. The barcode is scanned to get a serial number. If an item with the same serial number is previously entered, then the latest details are fetched from the Database and shown to the user. This helps to make the checkout faster by preventing the user to enter the name and price of the item every time. Fig. 10 shows the fetching of a previously entered product detail using QR code scanning.

5. Implementation of proposed system

The designed system provides facilities for creating, storing and printing bills anddeliversa simple and faster user experience. In contrast to the pen-and-paper method of invoicing, this system is reliable and relatively error-free to many human errors. As minimum as one operator can implement the system who usually is the storekeeper himself. The employment of barcode scanning enhances the efficiency of the system by decreasing the time required to add an item in an invoice. Thus, the system is suitable for implementation in small and medium scale stores.



+917548950804

Aniruddha Sarkar

Kolkata -10

Aactaril Soap 75g 270 2

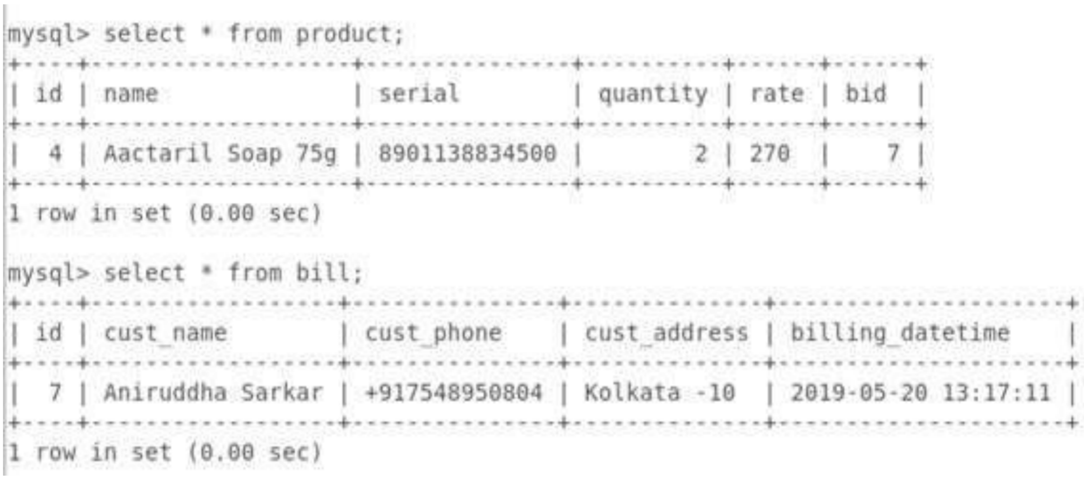
Total: Rs. 540.0

Bill Saved

The bill was successfully saved!

Fig. 6(a) **Fig. 6(b)** **Fig. 6(c)**

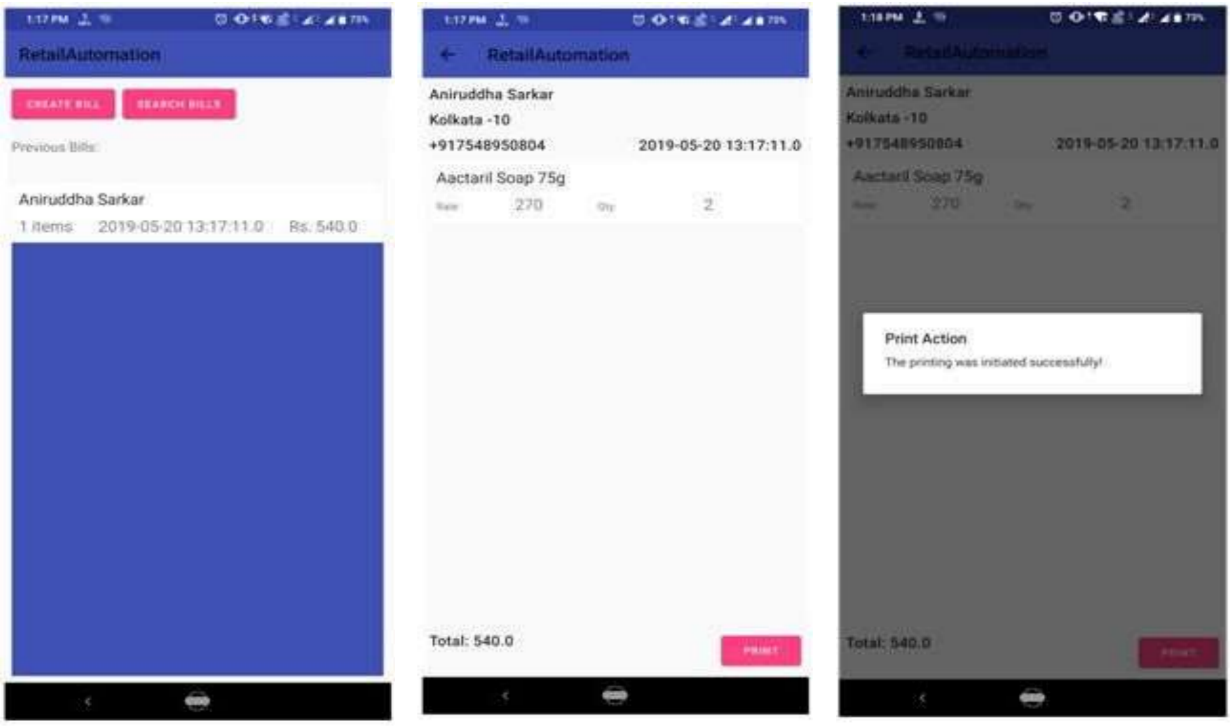
Fig. 6(a) Application home window without previous bills, (b) Creation of bill, (c) Response after saving bill



```
mysql> select * from product;
+----+-------------------+---------------+----------+------+-----+
| id | name              | serial        | quantity | rate | bid |
+----+-------------------+---------------+----------+------+-----+
|  4 | Aactaril Soap 75g | 8901138834500 |        2 | 270  |   7 |
+----+-------------------+---------------+----------+------+-----+
1 row in set (0.00 sec)

mysql> select * from bill;
+----+------------------+---------------+--------------+---------------------+
| id | cust_name        | cust_phone    | cust_address | billing_datetime    |
+----+------------------+---------------+--------------+---------------------+
|  7 | Aniruddha Sarkar | +917548950804 | Kolkata -10  | 2019-05-20 13:17:11 |
+----+------------------+---------------+--------------+---------------------+
1 row in set (0.00 sec)
```

Fig. 7: Entries in MySQL database after saving bill



Previous Bills:

Aniruddha Sarkar
1 items 2019-05-20 13:17:11.0 Rs. 540.0

Aniruddha Sarkar
Kolkata -10
+917548950804 2019-05-20 13:17:11.0

Aactaril Soap 75g
Rate 270 Qty 2

Total: 540.0

Aniruddha Sarkar
Kolkata -10
+917548950804 2019-05-20 13:17:11.0

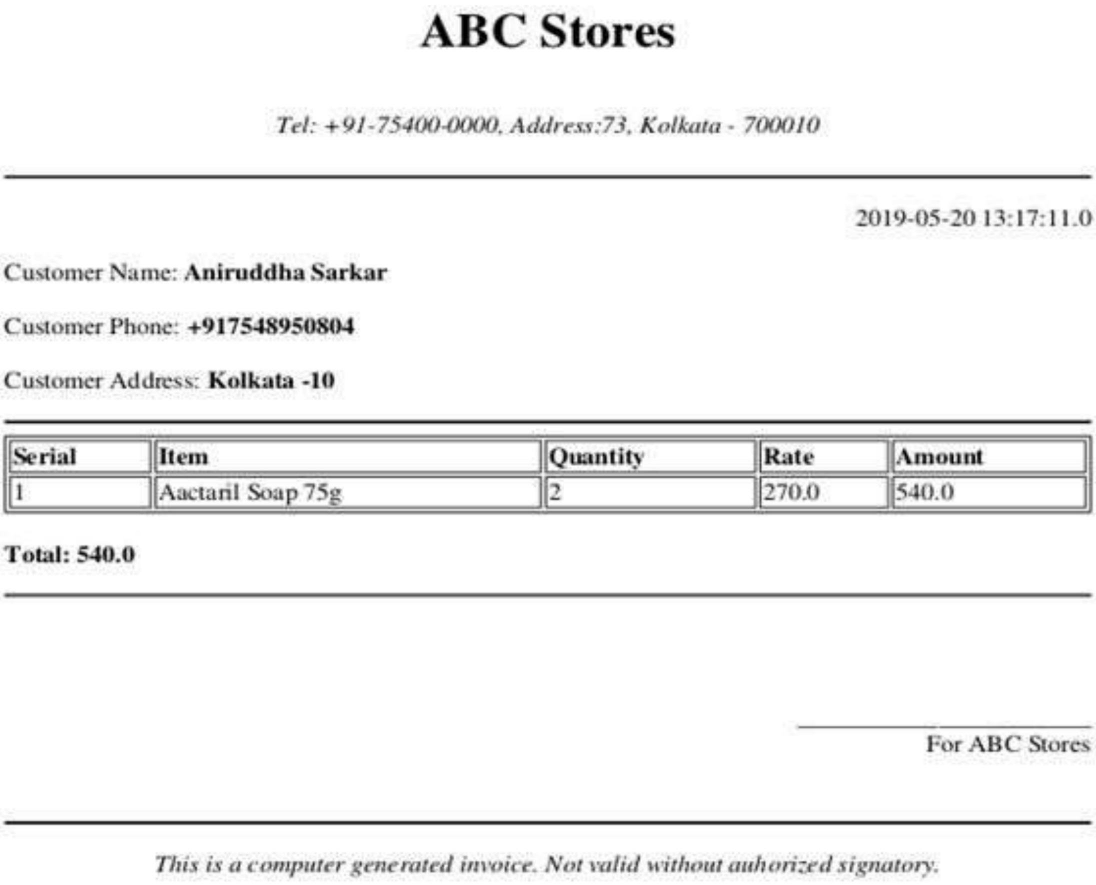
Aactaril Soap 75g
Rate 270 Qty 2

Print Action
The printing was initiated successfully!

Total: 540.0

Fig. 8(a) **Fig. 8(b)** **Fig. 8(c)**

Fig. 8: (a) Application home window with saved bill, (b) Previous bills, (c) Response after printing



ABC Stores

Tel: +91-75400-0000, Address:73, Kolkata - 700010

2019-05-20 13:17:11.0

Customer Name: **Aniruddha Sarkar**

Customer Phone: **+917548950804**

Customer Address: **Kolkata -10**

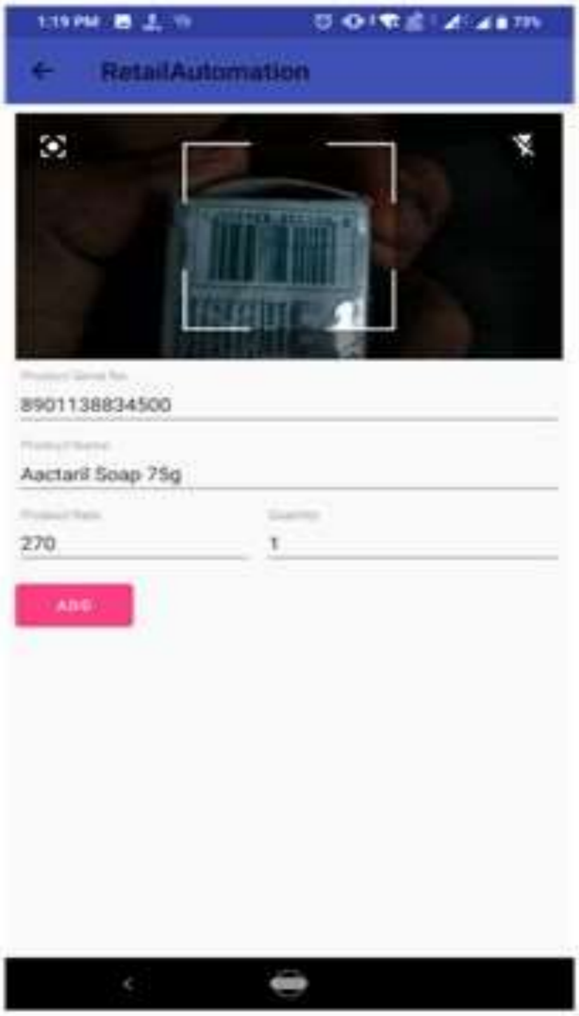
Serial	Item	Quantity	Rate	Amount
1	Aactaril Soap 75g	2	270.0	540.0

Total: 540.0

For ABC Stores

This is a computer generated invoice. Not valid without auhorized signatory.

Fig. 9: Invoice for printing



Product Serial No.
8901138834500

Product Name
Aactaril Soap 75g

Product Rate
270

Quantity
1

Fig. 10: QR code scanning for previously added products

6. Future works and conclusion

This work shows a successful demonstration of an invoicing system for small and medium scale stores, using which the basic billing process – creating, storing and printing bills can be done. The system is suitable for small and medium scale shops, but not suitable for large scale stores or supermarkets due to the lack of user management.

We can introduce more features into the application, including but not limited to – customer profile consisting of the products he/she buys, selling recommendations to help the shopkeeper, giving insights of the shop, recommendations for restocking of products, advertisement of products using SMS and sending bills via SMS service.Also, inventory management system, sending emails to customers containing the bills, and credit management for customer can be devised on the proposed system asadditional attributes. We can also include a payment gateway to allow cashless transaction.User management can be implemented in the system for keeping track of salespersons and improving security.

Acknowledgement:

We are indebted to Dr. Saibal Ray, Associate Professor (Physics), Government College of Engineering and Ceramic Technology, Kolkata for his guidance and constant encouragement in completing our research work.

References:

[1] H.C. Song, H.C., *Analysis of the global smartphone market and the strategies of its major players*, Hanyang University, University of Texas at Dallas (2010).

[2] E. Husni and S.P. Purwantoro, *Shopping application system with Near Field Communication (NFC) based on Android*, IEEE International Conference on System Engineering and Technology (ICSET), Bandung, Indonesia, Date of Conference: 11-12 September 2012.

[3] S. Sainath, K. Surender and V.V. Arvind, *Int. J. Comp. Appl.* **0975** (2014) 8887.

[4] S. Samal and S.P. Jena, 2014, *Int. J. Adv. Comp. Res.* **4** (2014) 236.

[5] K.K. Sancheti, P.G. Bhor and V.P. Patil, Int. J. Sci. Tech. Manag. Res. **3** (2018) 13.

[6] A. Krishnamoorthy, V. Vijayaranjan and R. Sapthagiri, Advances in Intelligent Systems and Computing **862** (2019), DOI https://doi.org/10.1007/978-981-13-3329-3_20.

[7] N.N. Dhage and S.D. Markande, *Bluetooth enabled printer adapter using raspberry pi,* IEEE International Conference on Pervasive Computing (ICPC), 8-10 Jan. 2015, Pune, India.

[8] P. Sachdeva and S. Katchii, *Int. J. Cur. Engg. Tech.* **4** (2014) 3818.

[9] A. Krishnamoorthy, V. Vijayarajan and R. Sapthagiri, R. (2019), *Automated Shopping Experience Using Real-Time IoT*, In: Satapathy S., Bhateja V., Somanah R., Yang XS., Senkerik R. (eds) Information Systems Design and Intelligent Applications. Advances in Intelligent Systems and Computing, Vol. 862, Springer, Singapore (2019).

[10] H. Pangasa and S. Aggarwal, *An Analysis of Li-Fi based Prevalent Automated Billing Systems in Shopping Malls*, 3rd International Conference on Computing Methodologies and Communication (ICCMC), IEEE, August 2019.

[11] R.K. Megalingam, S. Vishnu, S. Sekhar, V. Sasikumar, S. Sreekumar and T.R. Nair, *Design and Implementation of an Android Application for Smart Shopping,* International Conference on Communication and Signal Processing (ICCSP), pp. 0470-0474, IEEE (2019).