COMPARATIVE STUDY AND IMPLEMENTATION OF ALGORITHMS IN MACHINE LEARNING

This project report is submitted to

Rashtrasant Tukadoji Maharaj Nagpur University

in the partial fulfilment of the requirement for the award of the degree

of

Bachelor of Engineering in Computer Technology

by

Ms. Shreya Patle (CT18123)

Ms. Rutuja Kale (CT18037)

Mr. Shubham Chambhare (CT18125)

Ms. Prachi Patel(CT18031)

Mr. Raghav Shukla (CT18121)

Under the guidance of

Mr. V. P. Mahatme

Associate Professor

2020-2021

DEPARTMENT OF COMPUTER TECHNOLOGY
KAVIKULGURU INSTITUTE OF TECHNOLOGY AND SCIENCE

RAMTEK – 441 106
DEPARTMENT OF COMPUTER TECHNOLOGY

KAVIKULGURU INSTITUTE OF TECHNOLOGY AND SCIENCE

RAMTEK – 441106

CERTIFICATE

This is to certify that the project report entitled **‘Comparative Study and Implementation of Algorithms in Machine Learning’** carried out by Ms. Shreya Patle (CT18123), Ms. Rutuja Kale (CT18037), Mr. Shubham Chambhare (CT18125), Ms. Prachi Patel (CT18031), Mr. Raghav Shukla (CT18121) of the B.E. third year of Computer Technology, during the academic year 2020-2021, in the partial fulfilment of the requirement for the award of the degree of **Bachelor of Engineering (Computer Technology) offered by the Rashtrasant Tukadoji Maharaj Nagpur University, Nagpur**

Mr. V. P. Mahatme
Guide

Mr. V.P. Mahatme Dr. B. Ram Rathan Lal **Head of the Department Principal**

Date:

Place: Ramtek

ACKNOWLEDGEMENT

We are grateful to our respected guide **Mr. Rahul Bansod** for his kind, disciplined and invaluable guidance which inspired us to solve all the difficulties that came across during completion of project.

We express our special thanks to **Mr.V.P.Mahatme**, Head of the Department, for his kind support, valuable suggestion and allowing us to use all facilities that are available in the Department during this project.

Our sincere thanks are due to **Dr.B.Ram Rathan Lal**, Principal, for extending all the possible help and allowing us to use all resources that are available in the Institute. We are also thankful to our **Parents** and **Friends** for their valuable cooperation and standing with us in all difficult conditions.

Project-mates

ABSTRACT

Invoices are an inherent part of any business, be it manufacturing or services. In addition invoice processing is a huge task, especially manual invoicing. Android based application aims for an easy invoice generation and to overcome the major challenges in manual invoice processing. Therefore the development of invoice generation through android based application is essentially required. With the advancement in technology, mobile phones or smart phones are in a hurry, becoming the central computer and communication device in people's lives. Mobile phones have got influence due to user friendly operating system that is Android with its wide range applications. The availability of android based applications for invoice generation have advantages of instant information, quicker approval, reduces or eliminates errors and mainly increases efficiency. The application is developed using Android operating system environment. The application is standalone application, it does not require any internet connection for its operations. The application has a main focus on invoice generation and it will generate a GST added invoice. Due to the Constantly Evolving Technology, The Most major share of the business is the business of business. Every business connections on which processes are time-consuming and laborious. Smartphone technology is one such boon to these business owners.

KEYWORDS: Invoice, Android, Quicker approvals, Efficiency, GST invoices, Standalone.

CONTENTS

List of Figures

CHAPTER 1

INTRODUCTION

Invoices are an inherent part of any business, be it manufacturing or services. In addition, invoice processing is a huge task, especially manual invoicing. The surprising fact is that most businesses around the world still practice manual processing of invoices. PDF's are still used as a format to send and receive invoices. A large amount of time and resources are required to complete the scan and capture of physical invoices. This causes delays in the invoice process and hinders cash flow visibility.

An invoice can be defined as a sales invoice. Sales invoice is a document sent by a provider of a service or product to the purchaser. An invoice may contain many sections. The sections or format of an invoice may differ from business to business and also from businessman to businessman. The usual sections in an invoice include:

- The date that the invoice was created. Don't forget this. The date of the invoice starts the clock ticking on the customer. There are terms (a time limit for payment), that can be included like the date so that everyone knows when the payment is due.
- **Names and addresses** of customer and supplier. If the creation of the invoice in an accounting software is taking place, the only need is the email address of the customer, but it's still a good idea to collect and include the physical address, in case there is a need to send a real letter or document.

• **Contact names** of individuals at the two businesses (or business and individual). It's a good customer relations rule to make sure that spelling are correct. • **Description of items purchased**, either products or services, **including prices** and quantities. Often you will have standard item descriptions and inventory numbers. But be as specific and detailed as possible, when you create the invoice. This avoids confusion and not knowing about things issues. • **Terms of payment.** For example, the provider might specify "net 30 days," which means that the entire amount is due within 30 days.

Businesses must understand that paper invoices are time-intensive and often error prone. Cycle time for paper invoicing is usually very high. It is imperative to comprehend the steps involved in paper processing to know why paper invoicing is wasteful and digitization/automation makes sense.

1.1 Steps in Paper Invoicing

Even though the details might differ from firm to firm, most businesses go through the following steps when it comes to manual invoicing.

1. Capturing Invoices:

Suppliers would generate and print invoices and mail them to the buyer. Once you receive it, invoices need to be sorted and sent to the Account Payable (AP) department. Typically, AP staff manually key in the details into the system.

2. Quality Assurance:

In order that payments are accurate and payment terms are correct and complete, invoices need to be verified. An AP staff usually reviews each invoice for accuracy and completeness of information. For instance, invoices need to be checked for errors such as missing Purchase Order (PO) number or incorrect account number. Then, invoices need to be validated to confirm that the billing amount matches the order amount and the quantity of goods/services rendered. Finally, the cost center should look at the tax treatment for the invoice.

3. Getting Approvals:

Most often, invoices need to be approved by one or more higher authorities prior to payment. AP staff need to route invoices to the appropriate personnel whose approvals might be required. The whole process needs to be monitored so that delays are minimal. If there are any disputes or discrepancies, the AP staff should call the supplier to clear the same.

4. Accounting:

The AP staff need to record the expenses in the General Ledger of the firm. The staff would then file invoices or transport to archives for storage. This step has been used by the AP staff so that invoices can be easily retrieved when needed.

5. Payment:

When all the required approvals have been received, the payment is processed by the AP system. The payment might be made through paper check or Automated Clearing

House (ACH).

1.2 Major Challenges in Manual Invoice Processing

Paper processing or manual processing of invoices presents a number of challenges that result in unnecessary costs while reducing efficiency. Here are the reasons why paper invoicing is a wasteful process:

1. It Is Labor Intensive: Apart from the huge amount of time spent on paper handling and data entry, manual cross-reference for validity becomes necessary. This takes up quite a bit of time because AP staffs need to compare the invoice with a PO or contract. If that information is unavailable, then they might have to send it to some other authorities for cross verification. Depending on the size of the firm, there might be thousands or even millions of such invoices being sent around the firm. This is a waste of time for all personnel involved.

2. High Chances of Errors: Typically, invoices are received through various means such as mail, courier, fax, email and other formats. Multiple sources could mean confusion unless of these invoices are entered into a single system. This has to be done manually leading to wasteful time and possible errors. Some of the errors include paying invoices without delivery of goods/services, paying the same invoice twice, duplication of invoices, paying incorrect amounts and entering the wrong PO number.

3. Difficulty in Monitoring: Paper invoices get lost very easily. Unless paper invoices are files or stored in the right place, there are chances of them getting lost in the sea of other papers. Monitoring each and every invoice is a tough task and AP personnel usually end up spending too much time tackling queries from suppliers about payment status.

Unlike the past, AP departments are considered as a key to the firm’s bottom line. They are no longer cost center’s or processing center’s. Therefore, AP staff just cannot afford to waste time and resources that manual processing requires. Most often the aim of the AP department should be to drive savings through various means such as penalty avoidance, discount capture and lower labour costs. They should also look at improving the efficiency with which staff can access payment/invoice information.

1.3 About the Project

The society is more and more pervaded by the computer controlled devices.

Therefore the invoices which are an important part of any business be it manufacturing or services need an upgrade. The development of android based application for invoice generation is an upgrade for manual invoices and to help businessman in maintaining their records and to make the life less tiring and less on paperwork. The application mainly focuses on invoice generation. The application generates the GST added invoice .

1.4 Motivation

Businesses must understand that paper invoices are time-intensive and often error prone. Cycle time for paper invoicing is usually very high. According to a Hackett Group 2011 P2P Study and Ariba customer results, the cycle time for paper invoicing can take up to 22 days. A large amount of time and resources are required to complete the scan and capture of physical invoices. This causes delays in the invoice process and hinders cash flow visibility. Paper processing or manual processing of invoices presents a number of challenges that result in unnecessary costs while reducing efficiency.

It is imperative that you comprehend the steps involved in paper processing to know why paper invoicing is wasteful and digitization/automation makes sense. Electronic and automated invoice processes can result in savings of 60-80% compared to traditional paper-based processing. The study also reveals that companies who make the switch to e-invoicing can normally expect a payback period of 6-18 months. The advantages of automation are:

- **Instant information**: When invoices are automated, anybody can access the invoice at anytime from anywhere.
- **Quicker Approvals**: Since information on invoices is freely available, approvals for the same can be obtained quickly.
- **Save costs**: When payments are made on time, penalties are avoided and sometimes-even discounts are given by suppliers for those who pay in time regularly.
- **Reduces or Eliminates Errors:** When the automated invoices are integrated with the AP system, workflow gets streamlined and information becomes easily accessible. More importantly, when data is captured, the possibility of errors

becomes minimal. Also, the need for constant checking and monitoring of errors gets eliminated.

- **Easier Audit:** Auditing invoices in case of paper invoicing is very difficult. Sometimes invoices get misplaced or lost or even worse, they would have been destroyed by staffs who might not know their importance. In case of automation, all invoices are available at all points of time, making audit easier.
- **Increases Efficiency:** When less time is spent on data entry, cross referencing, getting approvals and filing papers, the efficiency of the AP staff goes up and they can spend time on other critical aspects of the AP process.

1.5 Organization of the Report

Chapter 1 includes the introduction of the project which comprises information regarding invoices, manual invoices and disadvantages of paper invoicing. Chapter 2 presents the literature review which involves the detailed study of the existing system which has some relevance with the proposed system. This chapter also specifies the Aims and Objectives of the project.

Chapter 3 presents overall ideas of the project in terms of the block diagram and also specifies the flow of working of the proposed system.

Chapter 4 contains the system description and all the functional and non functional requirements are specified in this chapters.

Chapter 5 contains the result of the proposed approach of the system. All the results are show with the help of snapshots and are discussed in details. Chapter 6 presents the conclusion and future scope of the proposed system.

CHAPTER 2

LITERATURE REVIEW

In the United States, the Council for Electronic Billing and Payment of the National Automated Clearing House Association **(NACHA)** is credited broadly promoting and communicating various forms of electronics billing in the early 2000’s. NACHA promoted activities and initiatives that facilitated the adoption of electronic payments in the areas of internet commerce, electronic bill payment and presentment financial electronic data interchange **(EDI).** International payments, electronic checks, electronic benefits transfer **(EBT)** and student lending. Certain electronic billing applications also provide the ability to electronic settle payment for good or service. When invoices are not designed on a specific template and instead on Word or Excel applications, their files are stored on local drives. In other words, they have to carry their laptops with them. The security risk is here. Online invoicing maker eliminates all these possibilities. In other words, the files and documents can be accessed from anywhere, anytime.

The businessman can keep track of invoices, expenses, reports, etc. through interactive dashboards and also offers other important information at one place. The best part of the business is the application of the business. E-invoices or online invoicing adds to improving accuracy, costs and speed of the invoicing process, a unified platform can actually bring a lot to the table of the invoicing world! The cloud-based invoicing software allows the user to manage and perform daily invoicing needs, and the payment of costs.

Since online invoicing is more pocket-friendly, it can help the accounting department to cut down to 80% by eliminating the use of paper, envelopes, stamps and other associated postage and mailing costs. Moreover, an additional factor is the payment method and other options for suppliers and vendors. Ensure that the online invoicing solution is more efficient and more efficient.

2.1 Billing System Design Based on Internet Environment

Muzhir Shaban Al-Ani (2014) has proposed that paper bills are now the primary channel of communication between companies and their customers. However, their potential for personalization is limited, and they are not interactive. If a customer wants to react to something in his paper bill – for example, to make a customer service inquiry or to order a new service – a telephone call can be an option. Internet Billing promises far more than a new and inexpensive way to deliver billing information. Industry experts predict that Internet Billing will fundamentally change the way companies interact with their customers. Eventually, the Internet Bill will be an interactive entry to a host of additional services including customer self-care, automated sales one-toone marketing. The Internet Bill will become the gateway through which customers and companies have electronic one to one dialogs. Businesses and consumers are banking on the Internet in more than one sense. Despite the early proliferation of electronic banking applications on private networks through dial-up services, most electronic banking applications have migrated to the Internet. Electronic billing is one of the fastest growing technologies for corporate law departments. Recent surveys indicate that roughly 15 percent of corporate legal 9ygdepartments require electronic bills from their law firms, and another 15 percent are considering it. If the person is a law firm with corporate clients, the person have probably seen acceleration in the number of requests from clients who want their bills submitted electronically. Choosing electronic billing and matter management systems are among the most important technology decisions that a law department can make, with significant potential consequences both positive and negative . The concept of electronic billing is not new. Since the advent of the Internet, a small number of consumers have been using this electronic medium to pay bills online after receiving standard paper invoices via regular Postal Service. What is new in the electronic billing arena is the concept of electronic bill presentment. With electronic bill presentment, companies that send bills (billers) post consumers' statements to the Internet, enabling consumers to view the statements and make e-payments . With ever increasing spread of Internet, Bill presentment and payment is becoming a new type of service area for periodic billers like Telephone Companies, Electricity etc. Internet based bill presentment and payment system converts billing centers from cost centers to revenue centers and for customers (payer) the system is a personalized service. Internet based bill presentment and payment system provides direct personalized communication channel between Billers and Payers, opens a new revenue channel by cross-selling advertisements. Drastic reduction of costs that are associated with paper based billing system.

2.2 Experimental Billing Schemes in Next Generation Network (I Changed)

P.S. Barreto, et al. has presented a discussion concerning the performance of network scenarios for billing purposes. Using the results of packet losses in an experimental platform simulating a NGN (Next Generation Network) environment. The authors has evaluated on each scenario the impact in the billing process with different traffic flows comparing the total revenue calculus for two billing schemes: (1)charging per packet and (2)reducing the value corresponding to undelivered packets. Our results show that the environments that use Differentiated Services are both convenient for costumers and service providers. Most of the billing systems are flat rate-based, which means that the customers are charged indistinctly, without considering the real network resources utilization and the type of services which is provided. This method is abundantly unused due to its simplicity, as it does not require any charging, accounting and metering policies. Despite of that, this method is not appropriate to implement congestion control through charging. The increasing demand for QoS (Quality of Service) induces the development of better adjusted billing mechanisms which discriminate each kind of traffic stimulating users to choose, according to their personal needs, the most appropriate type of service, avoiding excessive resources use and allocation. As the traffic is differentiated based on performance requirements, a QoS discriminative billing mechanism is expected to charge differently for different type of services. In order to implement congestion control through charging, there must be a billing method that relates the bill to the real use of resources. Based on this concept one simple method is proposed: charge per sent packet.

2.3 Billing System: Introduction

Hatem Mostafa (2005) has proposed an effective and accurate billing system for Telecommunication companies, need to be able to assure their revenue. Billing systems process the usage of network equipment that is used during the service usage into a single Call Detail Record (CDR). The billing process involves receiving billing records from various networks, determining the billing rates associated with the billing records, calculating the cost for each billing record, aggregating these records periodically to generate invoices, sending invoices to the customer, and collecting payments received from the customer. Billing system is very complex starting from network elements that generate usage to the billing system to usage collection, mediation, rating, and invoicing. The system user navigates through the company site and views company services, and decides to order one of the available services. If the user has no account, signs up for a new account, else user can signs in. Then the user asks to conduct an order with the selected service. The service may be prepaid where he has to pay to have credits to use the service, or it may be postpaid, where the user has to pay if the service has installation or setup fees, and later on he will pay for his usage of each billing cycle. The billing system should provide service to the user, collect user usage records, and generate invoices of each credit expire, each billing cycle depends on the billing type, collect payments and adjust customers' balances. Invoice is the entity generated by the billing system to inform the customer that he must pay for his service usage or ordering. Each invoice includes the customer account, date of payment, line items of the invoice, and invoice sales taxes. Invoices affect the account debits in the billing system.

2.4 Android OS

As wireless communication has been rapidly developed, mobile applications, services are growing much and more popular, like instant messaging, downloading of a variety of contents, mobile commerce, mobile banking, Internet access, etc. Technology development has simplified business, enriched the entertainment and made personal communication more convenient for mobile device customers. Mobile devices such as cellular phone, PDA and smart phone are opened the door to a lot of security threats like malicious code, vulnerabilities of mobile phone, attacks on communication, data robbery and injure, intruders, hackers, virus, spam, eves drooping, tracing, Jung mails, etc. information security will become a critical issue to mobile devices and be of great concern to mobile devices users, just like what computer users do today. This survey discusses the various features of mobile communication and mobile computing. This introduction part provides an extensive outline of mobile malware, mobile threats, virus, spam's, the problems what user faces in the current scenario. Moreover, it explains the differences among security solutions towards Smartphone and personnel computers. Android is a new mobile platform which is purely open source. Android applications can use trendy level of hardware and software, as well as local and server data. Android should have security mechanism to guarantee security of user contacts, data, information, application and network. Open source platform needs strong and high security architecture to secure all information. It was designed with multifaceted security that provides flexibleness needed for an open source, whereas provided that protection for all users of the platform premeditated to a software 15 stack, android includes an operating system, middleware and core application as a Complete. Android architecture is designed for ease of development for developers. Security modules have designed to minimize the load for developers. Developers have to merely work on additional security improvement aspect level controls. Developers are not aware with securities that apply by defaults on application. It is also designed with focused on user's perspective. Developers can view how applications work and control the built-in applications. Researchers are working to create a novel security application to prevent these threats and issues. Android is one of the popular mobile OS in smart phone sectors developed by Google. It is based on a modified version of the Linux kernel and other open source software, and is designed primarily for touch screen mobile devices such as smart phones and tablets. In addition, Google has further developed Android TV for televisions, Android Auto for cars, and

Wear OS for wrist watches, each with a specialized user interface. Variants of Android are also used on game consoles, digital cameras, PCs and other electronics.

Initially developed by Android Inc., which Google bought in 2005, Android was unveiled in 2007, with the first commercial Android device launched in September 2008. The operating system has since gone through multiple major releases, with the current version being 9 "Pie", released in August 2018. The core Android source code is known as Android Open Source Project (AOSP), and is primarily licensed under the Apache License.

Android is also associated with a suite of proprietary software developed by Google, called Google Mobile Services(GMS) that very frequently comes pre-installed in devices, which usually includes the Google Chrome web browser and Google Search and always includes core apps for services such as Gmail, as well as the application store and digital distribution platform Google Play, and associated development platform. These apps are licensed by manufacturers of Android devices certified under standards

imposed by Google, but AOSP has been used as the basis of competing Android ecosystems, such as Amazon.com's Fire OS, which use their own equivalents to GMS. Android has been the best-selling OS worldwide on smartphones since 2011 and on tablets since 2013. As of May 2017, it has over two billion monthly active users, the largest installed base of any operating system, and as of December 2018, the Google Play store features over 2.6 million apps.

2.5 History of Android

The name Andrew and the noun Android sharethe Greekroot andros, which means man. Andy Rubin picked android.com as his personal website, and his colleagues used Android as his nickname at work. That eventually became the name of the company he founded, and the name of the operating system they developed. Android Inc. was founded in Palo Alto, California, in October 2003 by Andy Rubin, Rich Miner, Nick Sears, and Chris White. Rubin described the Android project as "tremendous potential in developing smarter mobile devices that are more aware of its owner's location and preferences".The early intentions of the company were to develop an advanced operating system for digital cameras, and this was the basis of its pitch to investors in April 2004. The company then decided that the market for cameras was not large enough for its goals, and by five months later it had diverted its efforts and was pitching Android as a handset operating system that would rival Symbian and Microsoft Windows Mobile.

Rubin had difficulty attracting investors early on, and Android was facing eviction from its office space. Steve Perlman, a close friend of Rubin, brought him $10,000 in cash in an envelope, and shortly thereafter wired an undisclosed amount as seed funding. Perlman refused a stake in the company, and has stated "I did it because I believed in the thing, and I wanted to help Andy."In July 2005, Google acquired Android Inc. for at least $50 million. Its key employees, including Rubin, Miner and White, joined Google as part of the acquisition. Not much was known about the secretive Android at the time, with the company having provided few details other than that it was making software for mobile phones.At Google, the team led by Rubin developed a mobile device platform powered by the Linux kernel. Google marketed the platform to handset makers and carriers on the promise of providing a flexible, upgradeable system. Google had "lined up a series of hardware components and software partners and signalled to carriers that it was open to

various degrees of cooperation". Speculation about Google's intention to enter the mobile communications market continued to build through December 2006. An early prototype had a close resemblance to a BlackBerry phone, with no touchscreen and a physical QWERTY keyboard, but the arrival of 2007's Apple iPhone meant that Android "had to go back to the drawing board".Google later changed its Android specification documents to state that "Touch screens will be supported", although "the Product was designed with the presence of discrete physical buttons as an assumption, therefore a touchscreen cannot completely replace physical buttons". By 2008, both Nokia and BlackBerry announced touch-based smartphones to rival the iPhone 3G, and Android's focus eventually switched to just touchscreens. The first commercially available smartphone running Android was the HTC Dream, also known as T-Mobile G1, announced on September 23, 2008. On November 5, 2007, the Open Handset Alliance, a consortium of technology companies including Google, device manufacturers such as HTC, Motorola and Samsung, wireless carriers such as Sprint and T-Mobile, and chipset makers such as Qualcomm and Texas Instruments, unveiled itself, with a goal to develop "the first truly open and comprehensive platform for mobile devices". Within a year, the Open Handset Alliance faced two other open source competitors, the Symbian Foundation and the LiMo Foundation, the latter also developing a Linux-based mobile operating system like Google. In September 2007, InformationWeek covered an Evalueserve study reporting that Google had filed several patent applications in the area of mobile telephony.Since 2008, Android has seen numerous updates which have incrementally improved the operating system, adding new features and fixing bugs in previous releases. Each major release is named in alphabetical order after a dessert or sugary treat, with the first few Android versions being called "Cupcake", "Donut", "Eclair", and "Froyo", in that order. During its announcement of Android KitKat in 2013, Google explained that "Since these devices make lives so sweet, each Android version is named after a dessert", although a Google spokesperson told CNN in an interview that "It's kind of like an internal team thing, and preferred to be a little bit. In 2010, Google launched its Nexus series of devices, a lineup in which Google partnered with different device manufacturers to produce new devices and introduce new Android versions. The series was described as having "played a pivotal role in Android's history by introducing

new software iterations and hardware standards across the board", and became known for its "bloat-free" software with "timely ... updates". At its developer conference in May 2013, Google announced a special version of the Samsung Galaxy S4, where, instead of using Samsung's own Android customization, the phone ran "stock Android" and was promised to receive new system updates fast. The device would become the start of the Google Play edition program, and was followed by other devices, including the HTC One Google Play edition, and Moto G Google Play edition. In June 2014, Google announced Android One, a set of "hardware reference models" that would "allow to easily create high-quality phones at low costs", designed for consumers in developing countries. In September, Google announced the first set of Android One phones for release in India. However, Recode reported in June 2015 that the project was "a disappointment", citing "reluctant consumers and manufacturing partners" and "misfires from the search company that has never quite cracked hardware". Plans to relaunch Android One surfaced in August 2015, with Africa announced as the next location for the program a week later.

A report from The Information in January 2017 stated that Google is expanding its low-cost Android One program into the United States, although The Verge notes that the company will presumably not produce the actual devices itself. Google introduced the Pixel and Pixel XL smartphones in October 2016, marketed as being the first phones made by Google, and exclusively featured certain software features, such as the Google Assistant , before wider rollout. The Pixel phones replaced the Nexus series, with a new generation of Pixel phones launched in October 2017.

2.6 Features of Android

Android's default user interface is mainly based on direct manipulation, using touch inputs that loosely correspond to real-world actions, like swiping, tapping, pinching, and reverse pinching to manipulate on-screen objects, along with a virtual keyboard. Game controllers and full-size physical keyboards are supported via Bluetooth or USB. The response to user input is designed to be immediate and provides a fluid touch interface, often using the vibration capabilities of the device to provide haptic feedback to the user. Internal hardware, suchas accelerometers, gyroscopes and proximity sensors are used by some applications to respond to additional user actions, for example adjusting the screen from portrait to landscape depending on how the device is oriented, or allowing the user to steer a vehicle in a racing game by rotating the device, simulating control of a steering wheel. Android devices boot to the home screen, the primary navigation and information "hub" on Android devices, analogous to the desktop found on personal computers. Android homescreens are typically made up of app icons and widgets; app icons launch the associated app, whereas widgets display live, auto-updating content, such as a weather forecast, the user's email inbox, or a news ticker directly on the homescreen. A homescreen may be made up of several pages, between which the user can swipe back and forth. Third-party apps available on Google Play and other app stores can extensively re-theme the homescreen, and even mimic the look of other operating systems, such as Windows Phone. Most manufacturers customize the look and features of their Android devices to differentiate themselves from their competitors. Along the top of the screen is a status bar, showing information about the device and its connectivity. This status bar can be "pulled" down to reveal a notification screen where apps display important information or updates. Notifications are "short, timely, and relevant information about your app when it's not in use", and when tapped, users are directed to a screen inside the app relating to the notification. Beginning with Android 4.1 "Jelly Bean", "expandable notifications" allow the user to tap an icon on the notification in order for it to expand and display more information and possible app actions right from the notification. An All Apps screen lists all installed applications, with the ability for users to drag an app from the list onto the home screen. A resents screen lets users switch between recently used apps.

2.7 Linux Kernel

Android's kernel is based on the Linux kernel's long-term support (LTS) branches. As of 2018, Android targets versions 4.4, 4.9 or 4.14 of the Linux kernel. The actual kernel depends on the individual device. Android's variant of the Linux kernel has further architectural changes that are implemented by Google outside the typical Linux kernel development cycle, such as the inclusion of components like device trees, ashmem, ION, and different out of memory (OOM) handling. Certain features that Google contributed back to the Linux kernel, notably a power management feature called "wakelocks", were initially rejected by mainline kernel developers partly because they felt that Google did

not show any intent to maintain its own code. Google announced in April 2010 that they would hire two employees to work with the Linux kernel community, but Greg Kroah Hartman, the current Linux kernel maintainer for the stable branch, said in December 2010 that he was concerned that Google was no longer trying to get their code changes included in mainstream Linux. Google engineer Patrick Brady once stated in the company's developer conference that "Android is not Linux", with Computerworld adding that "Let me make it simple for you, without Linux, there is no Android". ArsTechnica wrote that "Although Android is built on top of the Linux kernel, the platform has very little in common with the conventional desktop Linux stack". In August 2011, Linus Torvalds said that "eventually Android and Linux would come back to a common kernel, but it will probably not be for four to five years". In December 2011, Greg Kroah-Hartman announced the start of Android Mainlining Project, which aims to put some Android drivers, patches and features back into the Linux kernel, starting in Linux 3.3. Linux included the autosleep and wakelocks capabilities in the 3.5 kernel, after many previous attempts at merger. The interfaces are the same but the upstream Linux implementation allows for two different suspend modes: to memory (the traditional suspend that Android uses), and to disk (hibernate, as it is known on the desktop).Google maintains a public code repository that contains their experimental work to re-base Android off the latest stable Linux versions. The flash storage on Android devices is split into several partitions, such as system for the operating system itself, and foruser data and application installations. In contrast to desktop Linux distributions, Android device owners are not given root access to the operating system and sensitive partitions such as system are read-only. However, root access can be obtained by exploiting security flaws in Android, which is used frequently by the open-source community to enhance the capabilities of their devices, but also by malicious parties to install viruses and malware. Android is a Linux distribution according to the Linux Foundation, Google's open-source chief Chris DiBona. and several journalists. Others, such as Google engineer Patrick Brady, say that Android is not Linux in the traditional Unix-like Linux distribution sense; Android does not include the GNU C Library (it uses Bionic as an alternative C library) and some of other components typically found in Linux distributions. With the release of Android

Oreo in 2017, Google began to require that devices Existing devices upgraded to Oreo, and new products launched with older SoCs, were exempt from this rule.

2.8 Development Kit

Android software development is the process by which new applications are created for devices running the Android operating system. Google states that "Android apps can be written using Kotlin, Java,and C++ languages" using the Android software development kit(SDK), while using other languages is also possible. All non-JVM languages, such as Go, JavaScript, C, C++ or assembly, need the help of JVM language code, that may be supplied by tools, likely with restricted API support. Some languages/programming tools allow cross-platform app support, i.e. for both Android and iOS. Third party tools, development environments and language support have also continued to evolve and expand since the initial SDK was released in 2008. In addition, with major business entities like Walmart, Amazon, Bank of America etc. eyeing to engage and sell through mobiles, mobile application development is witnessing a transformation.

2.8.1 Android JDK

The Java Development Kit (JDK) is an implementation of either one of the Java Platform, Standard Edition, Java Platform, Enterprise Edition, or Java Platform, Micro Edition platforms released by Oracle Corporation in the form of a binary product aimed at Java developers on Solaris, Linux, macOS or Windows. The JDK includes a private JVM and a few other resources to finish the development of a Java Application. Since the introduction of the Java platform, it has been by far the most widely used Software Development Kit (SDK). On 17 November 2006, Sun announced that they would release it under the GNU General Public License (GPL), thus making it free software. This happened in large part on 8 May 2007, when Sun contributed the source code to the OpenJDK.

2.8.2 Android SDK

The Android software development kit (SDK) includes a comprehensive set of development tools. These include a debugger, libraries, a handset emulator based on QEMU, documentation, sample code and tutorials. Currently supported development platforms include computers running Linux (any modern desktop , Mac OS X 10.5.8 or

later. As of March 2015, the SDK is not available on Android itself, but software development is possible by using specialized Android applications.Until around the end of 2014, the officially-supported integrated development environment (IDE) was Eclipse using the Android Development Tools (ADT) plug-in, though IntelliJ IDEA IDE (all editions) fully supports Android development out of the box, and NetBeans IDE also supports Android development via a plug-in. As of 2015, Android Studio, made by Google and powered by IntelliJ, is the official IDE; however, developers are free to use others, but Google made it clear that ADT was officially deprecated since the end of 2015 to focus on Android Studio as the official Android IDE. Additionally, developers may use any text editor to edit Java and XML files, then use command line tools (Java

Development Kit and Apache Ant are required) to create, build and debug Android applications as well as control attached Android devices (e.g., triggering a reboot, installing software package(s) remotely). Enhancements to Android's SDK go hand-in-hand with the overall Android platform development. The SDK also supports older versions of the Android platform in case developers wish to target their applications at older devices. Development tools are downloadable components, so after one has downloaded the latest version and platform, older platforms and tools can also be downloaded for compatibility testing.Android applications are packaged in .apk format and stored under /data/app folder on the Android OS (the folder is accessible only to the root user for security reasons). APK package contains .dex files (compiled byte code files called Dalvik executables), resource files, etc.

2.9 XML

Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine readable. The W3C's XML 1.0 Specification and several other related specifications—all of them free open standards—define XML.The design goals of XML emphasize simplicity, generality, and usability across the Internet. It is a textual data format with strong support via Unicode for different human languages. Although the design of XML focuses on documents the language is widely used for the representation of arbitrary data structures such as those used in web services. Several schema systems exist to aid in the

definition of XML-based languages, while programmers have developed many application programming interfaces (APIs) to aid the processing of XML data.

2.10 JAVA

Java is a general-purpose computer-programming language that is Concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to bytecode that can run on any Java Virtual Machine (JVM) regardless of computer architecture. As of 2016, Java is one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers. Java was originally developed by James Gosling, a Canadian, at Sun Microsystems (which has since been acquired by Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its original features from SmallTalk, with a syntax similar to C and C++, but it has fewer low-level facilities than either of them.

The Java programming language adheres the following qualities:

- **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.

2.11 Aim and Objectives

The aim of the proposed system is to implement an Android System for Invoice Generation with the following objectives:

- To create print and share GST Invoices
- To digitalize the billing system
- To develop a system that will lessen process and delay in terms of releasing receipts and customer billing
- To Manage Inventory and Account book for small business

- To Maintain the Sales and Purchases with the Sales Record and Manage the Expenses

CHAPTER 3

PROPOSED APPROACH

The system is completely standalone. Once the application has been installed by the user it will work without any internet connectivity. In the proposed system the user will first start the application. The user will get their two options. The first option is to login and the second option is register. By selecting the register option, the user will be directed to a registration page where the user will register on the application after that the user will be directed towards the login page where the user can log into the application. After the user is logged in into the application the user will be able to perform different functions. The data is stored locally onto the user's device after successfully logged in. Through this application the businessman can maintain their invoices and sales efficiently.

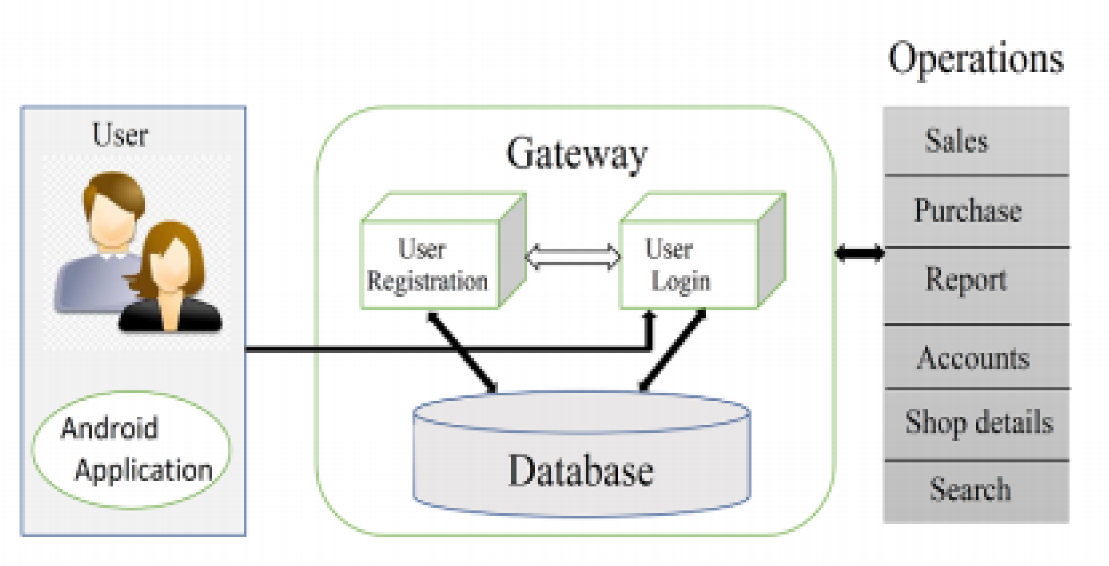


Fig.3.1 Architecture of the implemented system

The proposed system mainly focuses on term invoice and invoice generation. It also has different functions they are sales , purchase

3.1 Invoices

It is a recycle view menu which shows a list of invoices generated by the shopkeeper. In this menu the basics of a generated invoice are shown like the name of the customer, product, quantity, cost. These invoices can be viewed directly with the name of the customer, all the generated invoices are in a list view which is easy for the shopkeeper to seek through the list of invoices.

3.1.1 Generated Invoice

In recycle view the name of the customer, the name of the product which is brought by the customer, the quantity of the product and the total cost of the product which is calculated while generating an invoice are shown in the invoice's. The list of the generated invoice is shown in First-Come-First-Served algorithm. The stacking of invoices helps in maintaining the sales done by the shopkeeper. All the generated invoices are locally stored in a database on the user system. For storing of data SQLite is used and Database Helper is used to display the data which is stored.

3.2 Sales and Purchases

The shop must have some products to sell, for selling those products shop also need to buy those products from the wholesalers this is all handled under the sales and purchase section which can be found under the side menu on the homepage. The sales and purchase have a specific quantity for a product. All the products which are purchased can be sell by the shopkeeper. The shopkeeper can buy as much as products from the wholesaler and also different quantities.

3.2.1 Sales

The sales must be managed differently by the shopkeeper so the sales menu is under the sales and purchase option in side menu. The sales option calculates the no of sales and the total cost of the sales done within a certain period. It also shows the total number of products and their quantities which are sold by the shopkeeper.

3.2.2 Purchase

The purchase must be managed differently by the shopkeeper so the purchase menu is under the sales and purchase option in the side menu. The purchase option calculates and adds a product to the shops product list this also shows the number of purchases made within a certain period of time and also shows the number of products and their quantities.

3.3 Account

The account is another function that the system provides. The account is included to maintain all the monetary transaction of the businessman. The account section is maintain all the monetary information of the sales and purchase of the user. The account section calculates the profit and loss on the basis of sales and purchases within a certain period of time. The profit is calculated on the basis of simple calculation if the gross sales value is greater than the gross purchase value then the shopkeeper is in profit or the gross sales value is less than the gross purchase value then the shopkeeper is in loss. The account section can be found under the side menu on the homepage.

3.4 Report

The report is another function in the proposed system. The report plays an important part in the business. This function can be used by the user to generate the monthly report of the sales and purchase. The report can be generated with the help of account function's stored data in the database. The report can be generated of two types the first one is sales report , which will include the sales of the products. And the second one is purchase report which will include all the purchase record of the month.

CHAPTER 4

SYSTEM DESCRIPTION

4.1 Methods in Android

The various methods of Android platform are used for the implementation. The description about it is as follows:

1. The onCreate() method

- When an Activity first call or launched then onCreate(Bundle savedInstanceState) method is responsible to create the activity.
- Whenever orientation(i.e. from horizontal to vertical or vertical to horizontal) of activity gets changed or when an Activity gets forcefully terminated by any Operating System then savedInstanceState i.e. object of Bundle Class will save the state of an Activity.
- After Orientation changed then onCreate(Bundle savedInstanceState) will call and recreate the activity and load all data from savedInstanceState.
- Basically Bundle class is used to stored the data of activity whenever above condition occur in app.
- The onCreate() is not required for apps. But the reason it is used in app is because that method is the best place to put initialization code.
- The initialization code can also be put in onStart() or onResume() and while loading the app first time, it will work same asonCreate().

2. The setContentView (View view)

- The setContentView() is a very important function when it comes to programming with Android.
- Basically what this function does is display the Layout created thorugh XML or the dynamically created layout view in the Screen.
- It sets the activity content to an explicit view. This view is placed directly into the activity's view hierarchy. It can itself be a complex view hierarchy. When calling this method, the layout parameters of the specified view areignored.Both the width and the height of the view are set by defaults to MATCH_PARENT.

3. Intent

- For linking the button intent is used. Intent is an abstract description of an operation to be performed.

- It can be used with start activity to lunch an activity,broadcastIntent to send it to any interested broad cast Receiver components, and startService(Intents) or bind Service(Intent, serviceConnection, int) to communicate with a background service.
- Intent provides a facility for performing late runtime binding between the codes in different applications. Its most significant use is in the launching of activities, where it can be thought of as the glue between activities. It is basically a passive data structure data structure holding an abstract description of an action to be performed.
- Android application components can connect to other Android applications. This connection is based on a task description represented by an Intent object. • Intents are asynchronous messages which allow application components to request functionality from other Android components.
- Intents allow you to interact with components from the same applications as well as with components contributed by other applications. For example, an activity can start an external activity for taking a picture.
- Intents are objects of the android.content.Intent type. Your code can send them to the Android system defining the components you are targeting. For example, via the startActivity() method you can define that the intent should be used to start an activity.
- An intent can contain data via a Bundle. This data can be used by the receiving component.

4.2 Layouts

A layout defines the visual structure for a user interface, such as the UI for an activity or app widget. Layout can be declared by declare UI elements in XML. Android provides a straightforward XML vocabulary that corresponds to the View classes and subclasses, such as those for widgets and layouts. The Android framework gives the flexibility to use either or both of these methods to build your app's UI. For example, one

can declare the app's default layouts in XML, and then modify the layout at runtime. In the Layout Editor, one can quickly build layouts by dragging UI elements into a visual design editor instead of writing the layout XML by hand. The design editor can preview the layout on different Android devices and versions, and can dynamically resize the layout to be sure it works well on different screen sizes.

The Layout Editor is especially powerful when building a new layout with Constraint Layout. A layout manager provided in a support library that's compatible with Android 2.3 (API level 9) and higher. The advantage to declaring UI in XML is at it enables one to better separate the presentation of the application from the code that control its behaviour. The UI description are external to application code , which means that, it can modify or adapt without having to modify the source code and recompile. For example, one can create XML layouts for different screen orientation, different device screen sizes, and different languages. Additionally, declaring the layout in XML makes it easier to visualize the structure of your UI, its easier to debug problems.

4.3 Manifest

Manifest file for an android application is a resource file which contains all the details needed by the android system about the application. It is a key file that works as a bridge between the android developer and the android platform. It helps the developer to pass on functionality and requirements of our application to Android. This is an xml file which must be named as AndroidManifest.xml and placed at application root. Every Android app must have AndroidManifest.xml file. Android's manifest.xml allows to define the packages, API's, libraries needed for the application.

• Basic building blocks of application like activities, services and etc. •

Details about permissions.

• Set of classes needed before launch.

AndroidManifest.xml must be the file named and it should be placed in root folder of application. And elements must be present in the file and is allowed only once. Elements that are present inside, has no specific order forced. All values for any elements should be supplied as attributes and not as CDATA.As such, the name in the manifest's package attribute should always match project's base package name where user can keep activities and other app code. Of course, user can have other sub-packages in the project,

but those files must import the R.java class using the namespace from the package attribute.

However, beware that, once the APK is compiled, the attribute also represents the app's universally unique application ID. After the build tools perform the above tasks based on the package name, replace the package value with the value given to the application Id property in the project's build gradle file (used in Android Studio projects). This final value for the package attribute must be universally unique because it is the only guaranteed way to identify the app on the system and in Google Play.App activities, services, and broadcast receivers are activated by intents.Anintent is a message defined by an intent object that describes an action to perform, including the data to be acted upon, the category of component that should perform the action, and other instructions.

4.4 Android Studio

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as the primary IDE for native Android application development.

Android Studio was announced on May 16, 2013 at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0. The current stable version is 3.3, which was released in January 2019.

4.4.1 Features of Android Studio

The following features are provided in the current stable version:

- Gradle-based build support
- Android-specific refactoring and quick fixes
- Lint tools to catch performance, usability, version compatibility and other problems • Pro Guard integration and app-signing capabilities
- Template-based wizards to create common Android designs and components

- A rich layout editor that allows users to drag-and-drop UI components, option to preview layouts on multiple screen configurations
- Support for building Android Wear apps
- Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine • Android Virtual Device (Emulator) to run and debug apps in the Android studio. **4.5 Use Case Diagram**

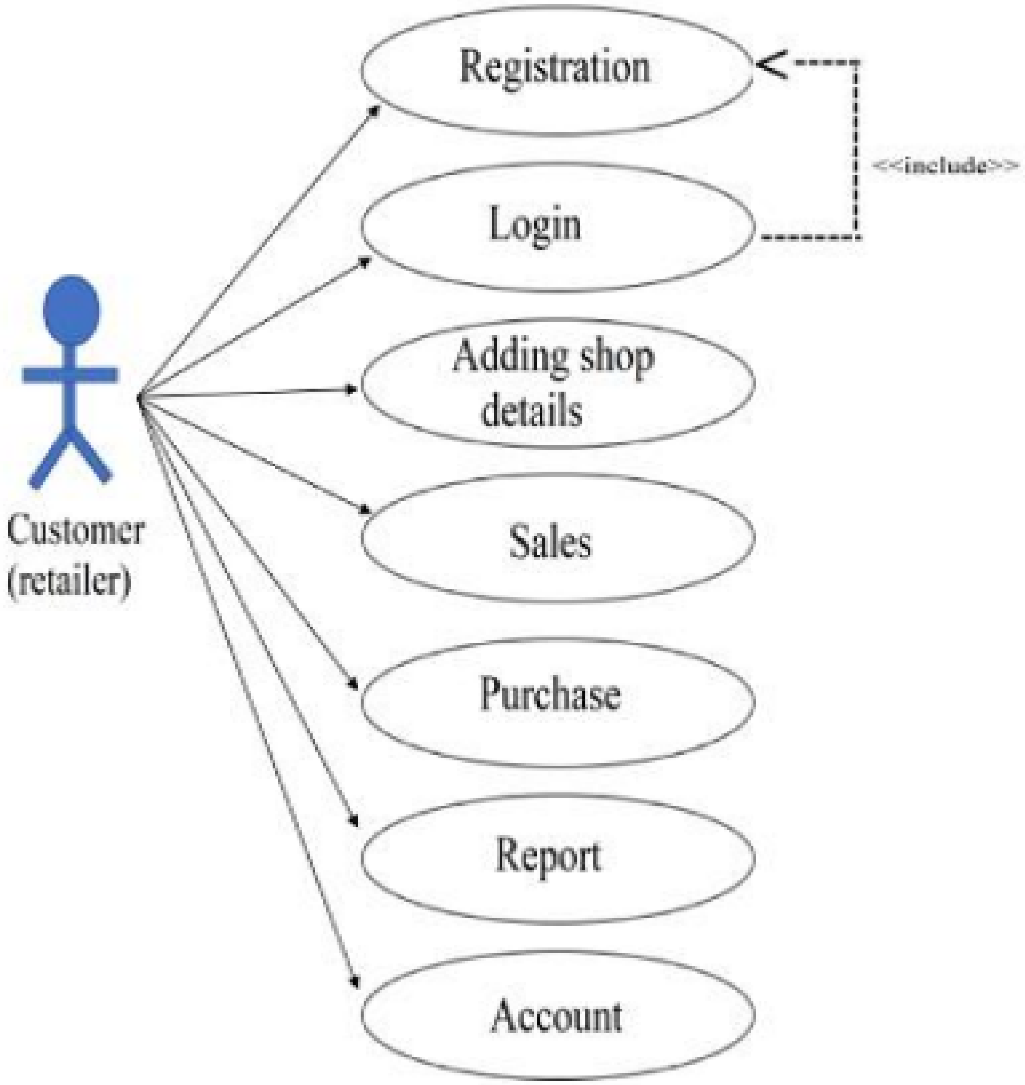


Fig 4.1 Use Case Diagram

4.6 Invoice Generation

For generating an invoice in the system there is a floating action button on the homepage of the application. On clicking the floating action button it will redirect the shopkeeper to an invoice generation page, here the shopkeeper will enter the name of the customer, the product which the customer is buying, the no of quantity the customer is asking for and the Goods and services tax which is applicable for that product. After entering all these details, the shopkeeper will click on the generate button which will generate the total cost of the product which the customer has to pay. After that the shopkeeper can decide whether to store the generated invoice into the database or discard it. If the shopkeeper decides to save the generated invoice then he simply has to click on the Add button which will directly store the information to the database and it can also be seen in the Generated Invoice List in the Invoice's homepage.

4.6.1 Calculate

After the shopkeeper has entered the name of the customer, product, the no. of quantity, and tax percentage (GST%) the user will click on generate to calculate the total cost of the product. Even if a single data is not entered and the user clicks on Generate the application won't calculate the total cost of the product and will highlight the texts which is missing.

4.6.2 Add

After the shopkeeper has successfully calculated the total cost of the product and received the payment from the customer then the shopkeeper can enter the invoice which is generated or the shopkeeper can discard the invoice if the payment is not completed or the customers refuses to but the goods. If the payment is successfully completed then shopkeeper can add the generated invoice to the database which will shop-up in the Generated Invoice's.

CHAPTER 5

RESULT AND DISCUSSION

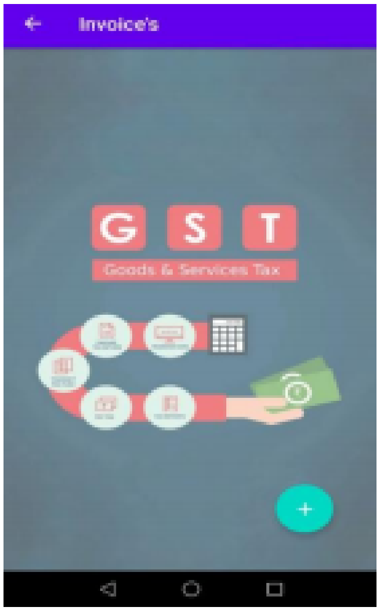
5.1 Formation of Home Page

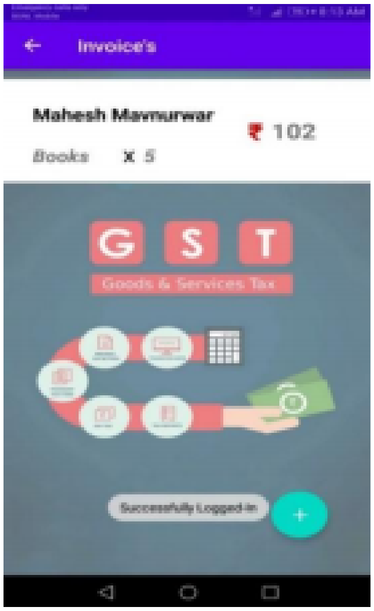
The Home page of proposed application will include one modules, the first one is invoice generation. which is an addition sign button which intent you to next page when clicked. And there is an empty space in home page which shows the mini report of every customer purchased from the shop. The home page is also can be called as mini report page. There are one button and a mini report on the home page of the application.

- Add invoice.
- Mini report.

The mini report on the home page is displayed after every invoice is generated. The mini report includes as follows:

- Customer Name.
- Product.
- Unit of Product.
- Price (Sum of product which includes GST).





Mahesh Mavnurwar ₹ 102

Books x 5

Successfully Logged-In

Fig 5.1
Home Page Fig 5.2 Invoice Record

5.2 Formation of Invoice Generation Module

In the invoice generation module, the user will provide the Information needed to generate the invoice. The information to be provide is the customer name, product, Quantity, Price (per unit), and includes GST. And then the last total Amount of the invoice by Rupee. This information of customer reflects in the Invoice as well as in the Home Page. Which is Shown in the Previous Figures. The page includes two buttons. One Calculate button and the other one is Add button. From this page the information is saved. This will also check if user does not provide any information about customer it will not save any information and shows the Error.

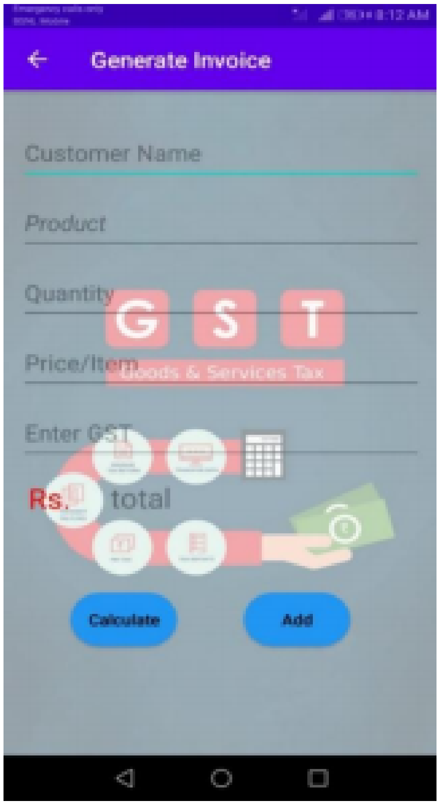
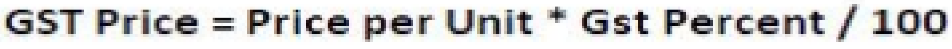


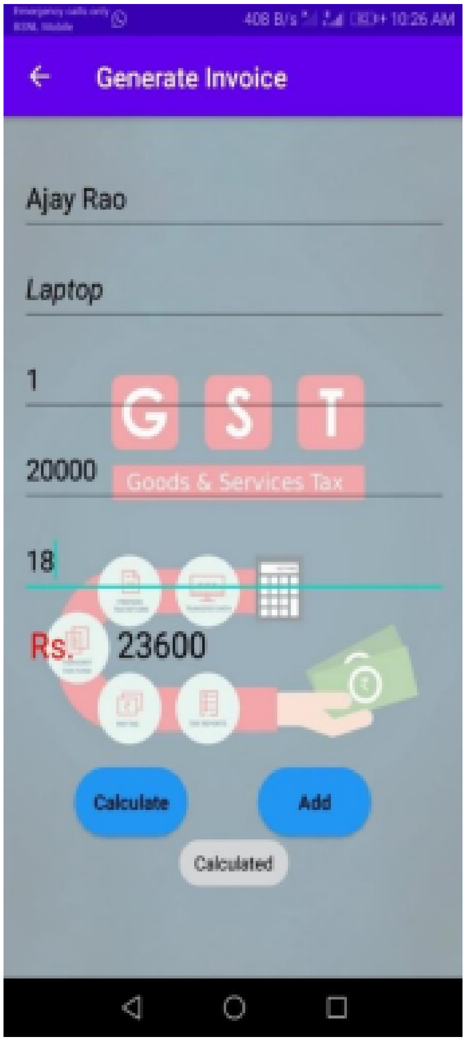
Fig 5.3 Invoice Template

5.3 Calculate Button

The calculate button is used to calculate the GST used in the invoice. The calculate button calculates the GST by the percentage of the goods decided by the government. Then the price per unit and the GST price is total in the bottom which shows

the total Amount of the invoice. By clicking the calculate button then it will calculate All the way the total Amount which is needed in invoice. After the calculation of the GST in the Amount the popup text will be shown as calculated. Which means the GST is calculated and then the sum of the amount including GST is printed in the total amount blank.


GST Price = Price per Unit * Gst Percent / 100



Generate Invoice

Ajay Rao

Laptop

1

20000

18

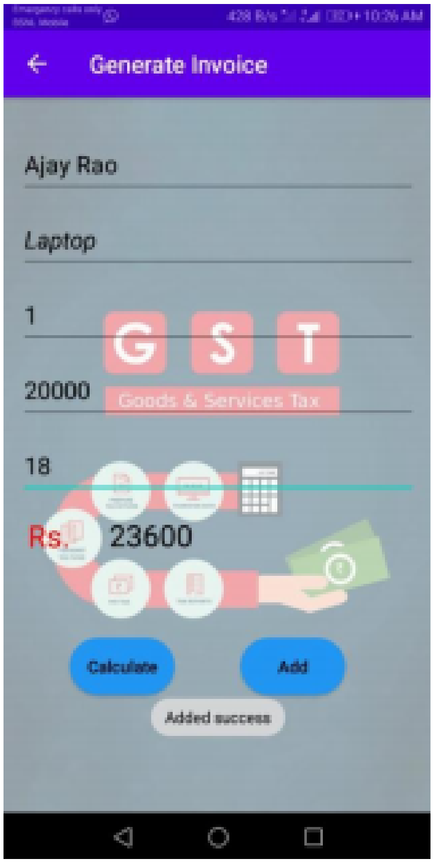
Rs. 23600

Fig 5.4 Invoice with GST Calculation

5.4 Add Button

In the invoice generation module the information provided by the user is saved by the add button. This button saves the customer name, product, units purchased, Price per

unit, GST percent and the total amount is saved by the user by one click. The button saves the information and will then proceed for the generation of invoice. After clicking on the Add button the popup text will be shown as Added successfully. And then the information is reflected in the home page as well. This module intent to the next page after saving the information. The saved content will be reflects on the invoice which is been generated after the information is saved.



Ajay Rao

Laptop

1

20000

18

Rs. 23600

Fig 5.5 Saving Invoice

CHAPTER 6

CONCLUSION

Working on this project has been an enriching experience. This project has given the opportunity to work on a new platform and learn a completely new language. This application has been designed for mobile users and it is very useful for businessman as it generates the invoice with all the taxes. The human calculation is not required to calculate the total amount as this will benefit the businessman to focus more on their customers and increase their businesses. It costs nothing to download this application in android devices; just one thing to do is to install an APK and keep smart phone in the pocket. As generation is improvising gradually development of applications like this can save a lot of money and time, instead of maintaining the record manually the businessman can store their records of customers only on one touch. Many applications like these have been developed in Android for computing the invoices. This application will not only benefit many existing relevant applications, but also accelerate the emergence of more applications relying on accurate invoice generation.

The digitization of invoices not only help the businessman but it is also time and money saving for customer. Also the digital invoice can easily reach to end users and one can make payments accordingly.

6.1 Limitations of the Study

• The user has to enter the data manually for estimating the invoices. • The calculations used to estimate the total calculation requires addition of taxes that may vary from one product to other product or from one service to the other service.

• There is a delay of few seconds showing the previous invoices.

6.2 Future Scope of the Work

• The complete digitization of business records can be done.

• Annual Report can be easily made with the help of invoices.

REFERENCE

1. Assimakopoulos Nikitas A . , Anastasis N. Riggas & Giorgos K.Kotsimpos, “A Systemic Approach for an Open Internet Billing System”, 2003 *International Conference of IEEE.*

2. Muzhir Shaban Al-Ani, Rabah Noory, Dua’a Yaseen Al-Ani , “Billing System Design

Based on Internet Environment “, 2012 *International Journal of Advanced Computer Science And Applications, Vol 3 No 9.*

3. Barreto P. S, G Amvame-Nze , C.V. Silva, J.S.S. Oliveria, H.P. de Carvalho, H.Abdalla Jr, A.M Soares, and R. Puttini, “ A Study of Billing Schemes in an Experimental Next Generation Network” , 2005 *SpringerVerlag Berlin Heidelberg.*
4. NN Murthy, BM Mehtre, KPR Rao, GSR Raman, PKB Harigopal and K.S. Babu, “Technologies for E-Commerce : An Overview”, 2000 *CMC Center -R&D, CMC Limited Old Mumbai Highway, Gachibowli Hyderabad – 500 019,Andhra Pradesh.*
5. EWB Team, “Electronics Extra Work Billing System : Online Step -byStep Instructions”,2001 *Revision 2, ISSC EWB Release 1.1 Instructions.*
6. Shiquin Li . GuilinWang. Jianying Zhou . Kefei Chen, “ Fair and Secure Mobile Billing Systems”, 2008 *Springer Science+ Business Media ,LLC.*
7. Crookes J. “ Multiservice Billing System – a platform for the future “, 1996 *BT Technol JVol 14 N0 3.*
8. Mostafa Hatem, “Billing System Introduction “, 2005 *codeproject.*

I have corrected the report somewhere and the rest u have to correct. Wherever correction is required I have marked with red color. There are very few corrections now.

Common mistakes:

1. Avoid I, we You , he , they
2. Check the left margin (as per the guidlines)