GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

UNIVERSITY OF POLYTECHNIC (GREATER NOIDA,UTTARPRADESH)

PROJECT-I On Customer Billing System

By

Bittu Kumar Thakur

&

Aamir Javed

Admission no.:19GPTC4060056

/19GPTC4060115

In partial fulfilment of requirements for the award of the degree

DIPLOMAINCOMPUTERSCIENCE&ENGINEERING

(Under the guidance of Sonam Bhati)

Galgotias University Department of Computer Science

ABSTRACT

The man objective of customer billing system is You can use this application to keep the records such as name, address, mobile number, paid amount, due amount, payment date etc. of your regular costumer. Moreover, if you have a new customer, you can add and edit the account at any time.

INDEX

CHAPTER1:INTRODUCTION

1.1INTRODUCTION

Customer Billing System Project is a simple console application designed to demonstrate the practical use of C programming language and its features as wells as to generate an application which can be used in any departmental store, shops, cafes etc. for billing to the customer.

1.5.2SOFTWAREREQUIREMENTSSPECIFICATION

This section includes the Software and hardware requirements for the smooth running of the application.

CHAPTER2: SOFTWARE REQUIREMENTS SPECIFICATION

2.1HardwareRequirements

Number	Description
1	PC with 250GB or more Hard disk.
2	PC with 2GBRAM.
3	PC with Pentium 1 and Above.

2.2SoftwareRequirements

Number	Description	Type
1	Operating System	Windows XP/Windows
2	Language	Python, local host server
3	Database	MySQL
4	IDE	Visual Code
5	Browser	Google Chrome

CHAPTER3: IMPLEMENTATION DETAILS

InthisSectionwewilldoAnalysisofTechnologiestouseforimplementingthepr

oject. 4.1: FRONTEND

4.1. PYTHON

Python is a widely used general-purpose, high level programming language.

It was created by Guido van Rossum in 1991 and further developed by the Python Software Foundation. It was designed with an emphasis on code readability, and its syntax allows programmers to express their concepts in

Local host server

Many people know from their own experience that it's not easy to install an Apache web server and it gets harder if you want to add MariaDB, PHP and Perl. The goal of XAMPP is to build an easy to install distribution for developers to get into the world of Apache. To make it convenient for developers, XAMPP is configured with all features turned on. In the case of commercial use please take a look at the product licenses, from the XAMPP point of view commercial use is also free. There are currently distributions for Windows, Linux, and OS X.

Our Team

Founders
Kai 'Oswald' Seidler

Oswald was one of the original co-founders of Apache Friends. He graduated 1999 from Technical University of Berlin with a Diplom-Informatiker degree (equivalent to a Master in Computer Science). In the 90's he created and managed Germany's biggest IRCnet server irc.fu-berlin.de, and co-managed one of the world's largest anonymous FTP server ftp.cs.tu-berlin.de. From 1993 until 1998 he was member of the internationally renowned "Projektgruppe Kulturraum Internet", a Berlin-based research project on net culture and network organization. In 2006, his third book, Das XAMPP-Handbuch, was published by Addison Wesley. From 2009 to 2011, he served as Technology Evangelist for Web Tier Products at Sun Microsystems/Oracle.

Together with Oswald, Kay co-founded the Apache Friends project in 2002. He currently works as freelance System Engineer and wrote several books about web technologies like Apache, MySQL and XAMPP itself.

Kay Vogelgesang

Beltran Rueda

Beltran is a project manager at Bitnami. He fell in love with open source and Linux since an early age and enjoys tinkering with web technologies and speaking at developer events. Beltran graduated from the Seville Engineering School with a Master's degree in Telecommunication.

Daniel Lopez Ridruejo

Daniel is a co-founder of Bitnami. Previously, he contributed to various web infrastructure open source projects, published multiple technical books on web technologies and became a member of the Apache Software Foundation. Daniel graduated from the Seville Engineering School with a Master's degree in Telecommunication and holds a Ms. Sc. on Optical Networking from DTU in Denmark.

4.1.1 MySQL

MySQL is an open-source relational database management system (RDBMS) based on Structured Query Language (SQL).Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language. A relational database organizes data into one or more data tables in which data types may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

MySQL is pretty easy to master in comparison with other database software like Oracle Database, or Microsoft SQL Server. MySQL can run on various platforms UNIX, Linux, Windows, etc. You can install it on a server or even in a desktop. Besides, MySQL is reliable, scalable, and fast. The official way to pronounce MySQL is My Ess Que Ell, not My Sequel. However, you can pronounce it whatever you like, who cares?

MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation).In 2010, when Oracle acquired Sun, Widenius forked the open-source MySQL project to create MariaDB.

MySQL is a component of the LAMP web application software stack (and others), which is an acronym for Linux, Apache, MySQL, Perl/PHP/Python. MySQL is used by many database
-driven web applications, including Drupal, Joomla, phpBB, and WordPress. MySQL is also used by many popular websites, including Facebook, Youtube, Twitter and so on.

fewer lines of code. Python is a programming language that lets you work quickly and integrate systems more efficiently.

4.2: BACKEND

4.2.1Local Host Server

This term is generally used in the context of networks. Localhost is not just the name for the virtual server but it is also its domain name. Just like .example, .test, or .invalid, ., .localhost is a top-level domain reserved for documentation and testing purposes. While accessing the domain, a loopback is triggered. If you access
“http://localhost” in the browser, the request will not be forwarded to the internet through the router. It will instead remain in your own system. Localhost has the IP address 127.0.0.1. This refers back to your own server.

Chapter 4:Features of a Billing Software

Every business uses different billing software for refining their accounting solutions. Your software should manage and contain the following features

1. Instant New Invoices- A good software automatically analysis your credit/debit tables in a professional manner. It should easily cite and
combine various schemes, timesheets, and consumer reports.

2. Individual Customer Tracking- Creates a single database that will easily segregate a clients purchases, relevant files along with a clean filter
search option for effortless accessibility.

3. Quick Receiving & Information Transferring- A central server facility that will help you send all the latest updates and information to
customers. All previous records shall be readily made available inside its database.

Screen Shot:

18/11/21
Customer Billing system
19:35:26
Customer Ref:
Name:
Mobile:
Gender:
Email:
Address:
PinCode:
Nationality:
No of peoples:
Tea
Coffee
Ice Cream
Choclate shake
Oreo shake
Banana shake
Cold coffee
Ice Tea
Tax and Total sum
Paid Tax:
Sub Total:
Total Cost:
Total
Reset
Exit
Save
Print

CHAPTER 5 : TESTING

5.1 : UNIT

TESTING

5.1.1Introduction

In computer programming, unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming, a unit could be an entire module, but it is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method. Unit tests are short code fragments created by programmers or occasionally by white box testers during the development process. It forms the basis for component testing. Ideally, each test case is independent from the others. Substitutes such as method stubs, mock objects, fakes, and test harnesses can be used to assist testing a module in isolation. Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended.

5.1.2

The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. A unit test provides a strict, written contract that the piece of code must satisfy. As a result, it affords several benefits.

1) Find problems early : Unit testing finds problems early in the development cycle. In test-driven development (TDD), which is frequently used in both extreme programming and scrum, unit tests are created before the code itself is written. When the tests pass, that code is considered complete. The same unit tests are run against that function frequently as the larger code base is developed either as the code is changed or via an automated process with the build. If the unit tests fail, it is considered to be a bug either in the changed code or the tests themselves. The unit tests then allow the location of the fault or failure to be easily traced. Since the unit tests alert the development team of the problem before handing the code off to testers or clients, it is still early in the development process.

2) Facilitates Change : *Unit testing allows the programmer to refactor code or upgrade system libraries at a later date, and make sure the module still works correctly (e.g., in regression testing). The procedure is to write test cases for all functions and methods so that whenever a change causes a fault, it can be quickly identified. Unit tests detect changes which may break a design contract.*

3) Simplifies Integration : *Unit testing may reduce uncertainty in the units themselves and can be used in a bottom-up testing style approach. By testing the parts of a program first and then testing the sum of its parts, integration testing becomes much easier.*

4) Documentation : *Unit testing provides a sort of living documentation of the system. Developers looking to learn what functionality is provided by a unit, and how to use it, can look at the unit tests to gain a basic understanding of the unit's interface (API).Unit test cases embody characteristics that are critical to the success of the unit. These characteristics can indicate appropriate/inappropriate use of a unit as well as negative behaviors that are to be trapped by the unit.*

5.1 : INTEGRATION TESTING

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

5.1.1 Purpose

The purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items. These "design items", i.e., assemblages (or groups of units), are exercised through their interfaces using black-box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter-process communication is tested and individual subsystems are exercised through their input interface. Test cases are constructed to test whether all the components within assemblages interact correctly, for example across procedure calls or process activations, and this is done after testing individual modules, i.e., unit testing. The overall idea is a "building block" approach, in which verified assemblages are added to a verified base which is then used to support the integration testing of further assemblages.Software integration testing is performed according to the software development life cycle (SDLC) after module and functional tests. The cross- dependencies for software integration testing are: schedule for integration testing, strategy and selection of the tools used for integration, define the cyclomatical complexity of the software and software architecture, reusability of modules and life-cycle and versioning management.Some different types of integration testing are big-bang, top-down, and bottom-up, mixed (sandwich) and risky-hardest. Other Integration Patterns[2] are: collaboration integration, backbone integration, layer integration, client-server integration, distributed services integration and high-frequency integration.

5.1.1.1 Big

In the big-bang approach, most of the developed modules are coupled together to form a complete software system or major part of the system and then used for integration testing. This method is very effective for saving time in the integration testing process. However, if the test cases and their results are not recorded properly, the entire integration process will be more complicated and may prevent the testing team from achieving the goal of integration testing.A type of big-bang integration testing is called "usage model testing" which can be used in both software and hardware integration testing. The basis behind this type of integration testing is to run user-like workloads in integrated user-like environments. In doing the testing in this manner, the environment is proofed, while the individual components are proofed indirectly through their use. Usage Model testing takes an optimistic approach to testing, because it expects to have few problems with the individual components. The strategy relies heavily on the component developers to do the isolated unit testing for their product. The goal of the strategy is to avoid redoing the testing done by the developers, and instead flesh-out problems caused by the interaction of the components in theenvironment.

5.1.1.2 Top-down And Bottom-up

Bottom-up testing is an approach to integrated testing where the lowest level components are tested first, then used to facilitate the testing of higher level components. The process is repeated until the component at the top of the hierarchy is tested.All the bottom or low- level modules, procedures or functions are integrated and then tested. After the integration testing of lower level integrated modules, the next level of modules will be formed and can be used for integration testing. This approach is helpful only when all or most of the modules of the same development level are ready. This method also helps to determine the levels of software developed and makes it easier to report testing progress in the form of a percentage.Top-down testing is an approach to integrated testing where the top integrated modules are tested and the branch of the module is tested step by step until the end of the related module.Sandwich testing is an approach to combine top down testing with bottom up testing.

CHAPTER7:
ADVANTAGES

- It’s fast, easy and comfortable.
- Less hassle for you.
- An online billing is simpler to manage.
- Helps in GST Tax calculation.

CHAPTER8:
CONCLUSION

You have the flexibility to do pricing your way & automates your billing and gives customers what they want, how they want it.

Our customer billing system enables to send accurate and clean invoices so that customers understand exactly how they're being billed and for what.