**SANDIP INSTITUTE OF TECHNOLOGY AND RESEARCH CENTER**

**(Department of Computer Engineering)**

****

**Mini Project Report**

**On**

**RESTAURANT BILLING SYSTEM AND ANALYSIS**

**Subject-:**

***SKILL* DEVELOPMENT LAB**

**Guided By Presented By**

Prof. Harish Patil Minaz Inamdar

Yash Garudkar

Mohit Sonawane

Ratna Phalak

**Signature of External Examiner Signature of Internal Examiner**

**ABSTRACT**

The following report details the development of restaurant billing system and analysis in python language. To keep the order and calculations easy. This system provides cohesion between the main elements of restaurant business- the restaurant floor and back office. Waiters have an interface they can use to take orders, the order then shows up on an interface for the kitchen staff to complete. The manager can use another interface to edit the menu, generate reports/analyze statistics, and perform stock control.

**Table of Content**

**1. Introduction…………………………………………………………………………04**

1.1 Purpose/ Chapter Overview………………………………………………....04

1.2 Objective…………………………………………………………………….

1.3 system overview…………………………………………………………….04

1.4 Scope………………………………………………………………………..

**2. Problem Statement…………………………………………………………..............07**

**3. Design………………………………………………………………………………...05**

3.1 Chapter Overview…………………………………………………………...05

3.2 Programming Methodology…………………………………………………05

2.2.1 Waterfall…………………………………………………………..06

2.2.2 Spiral………………………………………………………………07

2.2.3 Agile………………………………………………………………08

2.3 Chosen Methodology……………………………………………………….09

2.4 Program Architecture…………………………………………………….…10

2.5 Development Language………………………………………………….…11

**3. Implementation…………………………………………………………………….13**

3.1 Chapter Overview………………………………………………………….13

3.2 Software and tools…………………………………………………………13

3.3 GUI design for front end…………………………………………………..18

**4. Result……………………………………………………………………………….19**

4.1 Chapter Overview…………………………………………………………19

4.2 Snapshots of output………………………………………………………..19

**5. Conclusion…………………………………………………………………………22**

5.1 Chapter Overview………………………………………………………...22

5.2 Future Improvements……………………………………………………..22

**6. References…………………………………………………………………………29**

1. **INTRODUCTION**
   1. **PURPOSE**

The purpose of this chapter is to give the reader context, and enough information about existing restaurant systems ensure full understanding of the issues incurred in the industry.

* 1. **OBJECTIVE**

The main objective of this 4 week project is to develop a server model, which deals with “Restaurant billing system and analysis”. The system has part for the management side.

The management side the staff is allowed to edit information regarding menu list, price, maintain bill regarding the orders, etc.

* 1. **SYSTEM OVERVIEW**

We propose to build a software project that can efficiently handle and manage various activities of a restaurant and all these activities will be happening under the supervision of the administrator. The businesses in restaurants are now growing constantly. At the same time, the need for managing its operations and tasks arises. The best way to optimize these activities is growing the business online as well. Today's generation encourages high-tech services especially over the Internet. Hence the project is developed proficiently to help restaurant owners automate their business operations. This project serves the best way of maintaining customer's information and caters their needs.

* 1. **SCOPE**

We are greatly constrained by “Time limitations” as only 4 weeks are given to complete the project. So we limited our scope of project to “billing system and analysis” only.

The sole purpose of the project is to learn the application development in python, how python application works to produce more commercial applications for daily business.

We will try our best to make better usage of Python and surely all in a better team work.

* 1. **PROBLEM STATEMENT**

The businesses in restaurants are now growing constantly. At the same time, the need for managing its operations and tasks arises. The best way to optimize these activities is growing the business online as well. Today’s generation encourages high-tech services especially over the Internet.

RESTAURANT BILLING SYSTEM AND is Desktop Application to restaurant management. Hence the project is developed proficiently to help restaurant. This system wake to provide service facility to restaurant and owners automate their BILLING OPERATIONS AND ANALYSIS.

As per the new rule Goods and Service Tax (GST) levy on both AC and non-AC restaurants to 5%. Every restaurant charges 5% GST breakup as 2.5% State GST (SGST) and 2.5 % Central GST (CGST) without any other service tax and any other VAT charges.. We have developed a RESTAUNRANT BILLING SYSTEM AND ANALYSIS with inbuilt GST calculation in the bill, as well as it shows the amount in rupees how much GST is applied.

RESTAUNRANT BILLING SYSTEM AND ANALYSIS project fully developed in python language which is currently demanded in market using python GUI Tkinter. Python is a programming language that lets you work quickly and integrate systems more efficiently.

RESTAUNRANT BILLING SYSTEM AND ANALYSIS project include weekly analysis of sales of dishes in restaurant.

1. **DESIGN**
   1. **CHAPTER OVERVIEW**

This chapter follows the design of the system following on from the requirements identified. The chapter will explain and justify the chosen software development methodology, and detail the architecture of the system. Following this, the design phase of each system layer will be described in detail.

* 1. **PROGRAMMING METHODOLOGIES**

It is important that the correct programming methodology is chosen for a given project. The methodology chosen should reflect the type of system being developed. For example, developing a program to be used by aircraft would require a methodology heavily focused on testing, to reduce risk. Alternatively, a single developer creating software with no risk and with no other stakeholders may not use a methodology at all. There is no ‘one size fits all’ method, and as a result there are a large range of programming methodologies used throughout the industry. In this section several widely used techniques will be detailed, before justification of the chosen methodology for the project.

**2.2.1 WATERFALL**

This is the first published software engineering process described by Royce in 1970, who described a model consisting of seven phases to be competed in consecutive order. The waterfall methodology requires the entire scope of the project to be known at the start of the project. All planning is completed in one section, and therefore requires big upfront requirements gathering. The methodology is best suited for projects where the requirements are static, and unlikely to change. The waterfall methodology has been widely used throughout history, notably by the US Department of Defense (DOD) who used it for several projects throughout the 1980’s and 1990’s However since this time the use of this methodology has declined due to it’s inflexibility.

**Advantages:**

1. Work on a stage has to have been completed and signed off before moving onto the next. This means the developers are forced to focus on one stage a time.

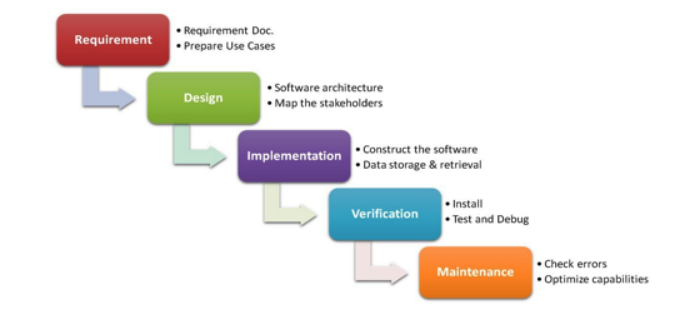
2. Big upfront design means developers don’t need to waste time gathering more information or changing specifications, which costs time and money.

**Problems:**

1. Most criticism aimed at the Waterfall method is based on several key assumptions made in development.

2. Firstly, the assumption that all requirements are already well defined, and that changes to requirements can be managed without rethinking plans.

3. Secondly, that the time taken to research new the techniques and innovations required in making the project can be estimated at the beginning.



**2.2.2 SPIRAL**

First detailed in a paper entitled ‘A Spiral Model of Software Development and Enhancement’ by Barry Boehm. Like most methodologies, the process passes through four main phases: Planning, Risk Analysis, Development and Evaluation. The model uses iterations to cycle through these phases multiple times, each iteration represents a ‘spiral’. A “prototype’ system is produced at the end of each phase, as the Spiral Methodology is risk driven, each spiral addresses one or more major risks.Each spiral builds on it’s precursor until all the risks have been Addressed . Each iteration is typically between 6 months and 2 years long.

**Advantages**:

1. Flexible, iterative nature means the development process can scope with changes in requirements

2. Software can be produced early in the development lifecycle

3. Risk driven nature of development results in risk avoidance

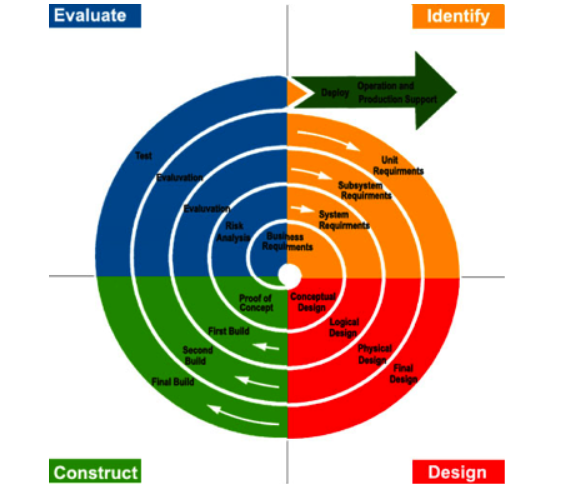
1. Advantageous for mission critical software, due to thorough documentation.

**Problems:**

1. Long iteration times means projects are often expensive

2. Not advantageous for small projects

3. Requires skilled managers and experience in risk analysis



**2.2.3 AGILE METHODOLOGY**

Agile Methodologies were introduced throughout the 1990’s to combat the rigidness of the waterfall based methods used throughout the industry. Many versions of the process were implemented over the decade, before a set of key principles known as ‘The Agile Manifesto’ was published in 2001. The manifesto based on 12 key principles, focuses on ‘lightweight’ development methods. Agile development focuses on breaking down the development process into time­‐boxes (fixed length time intervals) known as iterations. A subset of the identified user stories are chosen, developed and tested every iteration. Once an iteration is completed, stakeholders are able to view and evaluate the work before giving feedback. The length of iterations can vary from 1 to 4 weeks, and usually multiple iterations are completed

to release a product of feature. Agile methodologies prioritize producing working software, customer collaboration, code ownership and flexible specifications.

**Advantages:**

1. Systems can be developed very quickly

2. Low cost

3. Fast adaptation to change

4. Increased customer satisfaction caused by feedback

**Problems:**

1. Lack of development documentation results from frequent changes

2. Often requires experienced project manager

3. Flexible nature of project may cause scope creep

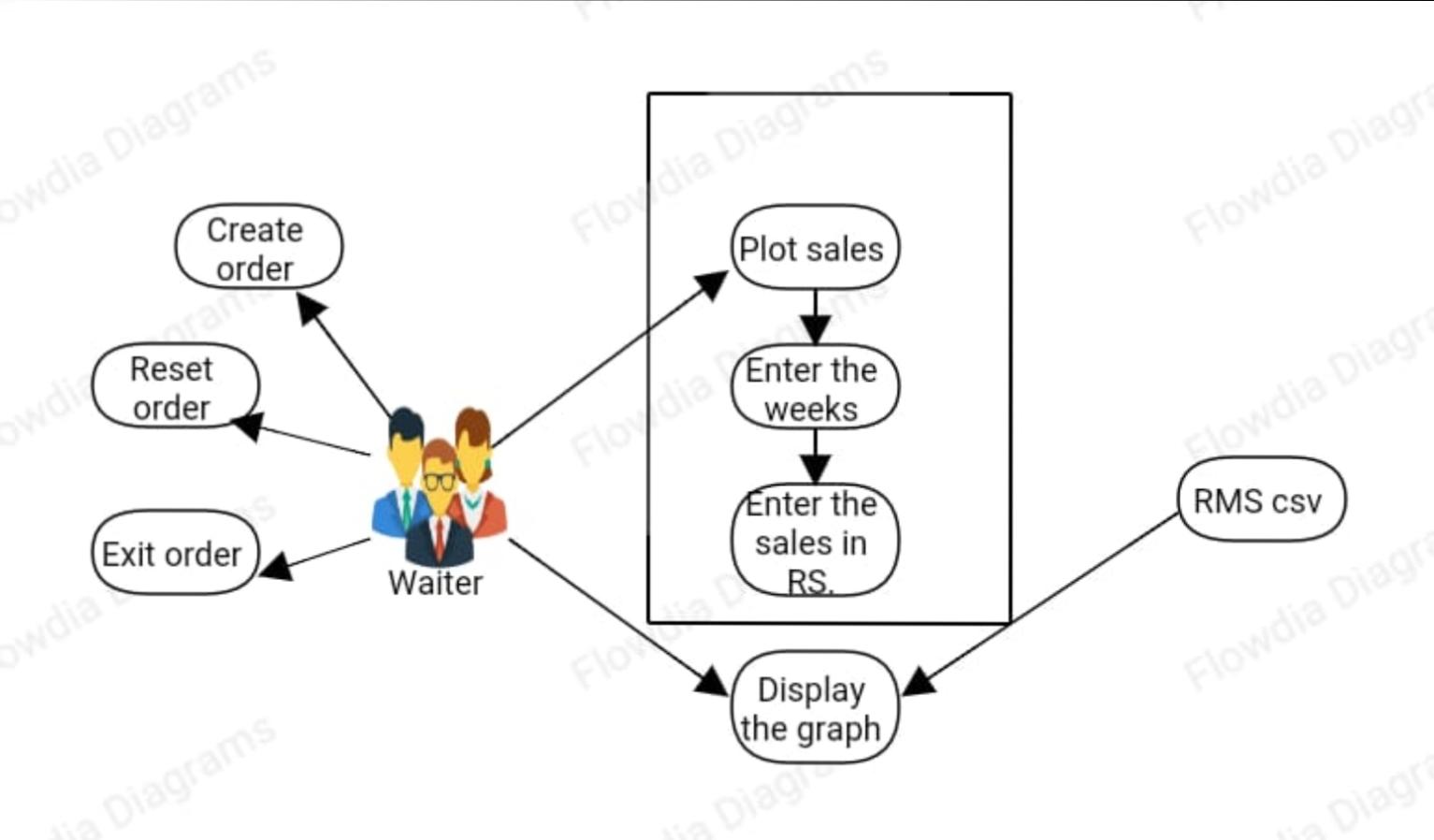
4. Intensive work hours needed to complete requirements for iteration end There are several variations of agile methodologies, each sharing similar principles and characteristics. However, each variation is implemented with a different style.

* 1. **CHOSEN METHODOLOGY**

When choosing the methodology to be used throughout the project, the volatility of the requirements specification was assessed, along with the time constraints placed on the project lifecycle. From this information, Spiral Methodology was ruled out, due to its long iteration time and small nature of the project. Thanks to familiarity with waterfall methodologies from previous projects, it is clear that there is a lot of evidence to suggest that its big up front requirements gathering and its rigid structure hinders the progress of projects. It is estimated that up to 60% of a companies resources can be wasted using the methodology, by writing unused and incorrect code.

Therefore, an agile methodology was the best option available. Due to the short time frame of the project, an agile methodology would help a system to be developed with the most useable features in a short space of time. A feature driven development was chosen throughout the project, allowing a focus on designing and building a set of requirements or ‘features’ in each iteration. This allowed value to be delivered to the stakeholders and for the project to remain responsive to changing requirements.

* 1. **PROGRAM ARCHITECTURE**

****

* 1. **DEVELOPMENT LANGUAGES**

Decisions made at this stage of the design process have a large affect on the outcome of a project. Therefore it was important to choose development languages and tools that suited the architecture that had been chosen.

**PYTHON**

[Python](https://www.geeksforgeeks.org/python-programming-language/) is a widely used general-purpose, high level programming language. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code. Python is a programming language that lets you work quickly and integrate systems more efficiently. Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

**Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

**Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

**Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

**Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

**PYTHON FEATURES**

Python's features include −

**Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.

**Easy-to-read** − Python code is more clearly defined and visible to the eyes.

**Easy-to-maintain** − Python's source code is fairly easy-to-maintain.

**A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.

**Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.

**Portable**− Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

**Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.

**Databases**− Python provides interfaces to all major commercial databases.

**GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

**Scalable** − Python provides a better structure and support for large programs than shell scripting. Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

1. It supports functional and structured programming methods as well as OOP.
2. It can be used as a scripting language or can be compiled to byte-code for building large applications.
3. It provides very high-level dynamic data types and supports dynamic type checking.
4. IT supports automatic garbage collection.
5. It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

**Python graphical user interfaces (GUIs)**

Tkinter − Tkinter is the Python interface to the Tk GUI toolkit shipped with Python. We would look this option in this chapter.

wxPython − This is an open-source Python interface for wxWindows [http://wxpython.org](http://wxpython.org/).

JPython − JPython is a Python port for Java which gives Python scripts seamless access to Java class libraries on the local machine [http://www.jython.org](http://www.jython.org/).

There are many other interfaces available, which you can find them on the net.

1. **IMPLEMENTATION**

**3.1 CHAPTER OVERVIEW**

In this chapter I will justify the tools and techniques used to implement the project, including development environment and programming languages. I will then highlight

**3.2 SOFTWARE AND TOOLS**

In this section I will justify the software development tools, such as IDE’s and repositories, used when implementing the system.

Technologies used - Python 3.6.

Python Tkinter GUI

Language used - Python

3.2.1 Tkinter

Tkinter Programming



Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps −

Import the Tkinter module.

Create the GUI application main window.

Add one or more of the above-mentioned widgets to the GUI application.

Enter the main event loop to take action against each event triggered by the user.

Example

#!/usr/bin/python

import tkinter

top = tkinter.Tk()

# Code to add widgets will go here...

top.mainloop()

This would create a following window −



Tkinter Widgets

Tkinter provides various controls, such as buttons, labels and text boxes used in a GUI application. These controls are commonly called widgets.

There are currently 15 types of widgets in Tkinter. We present these widgets as well as a brief description in the following table −

|  |  |
| --- | --- |
| Sr.No. | Operator & Description |
| 1 | [Button](https://www.tutorialspoint.com/python/tk_button.htm)  The Button widget is used to display buttons in your application. |
| 2 | [Canvas](https://www.tutorialspoint.com/python/tk_canvas.htm)  The Canvas widget is used to draw shapes, such as lines, ovals, polygons and rectangles, in your application. |
| 3 | [Checkbutton](https://www.tutorialspoint.com/python/tk_checkbutton.htm)  The Checkbutton widget is used to display a number of options as checkboxes. The user can select multiple options at a time. |
| 4 | [Entry](https://www.tutorialspoint.com/python/tk_entry.htm)  The Entry widget is used to display a single-line text field for accepting values from a user. |
| 5 | [Frame](https://www.tutorialspoint.com/python/tk_frame.htm)  The Frame widget is used as a container widget to organize other widgets. |
| 6 | [Label](https://www.tutorialspoint.com/python/tk_label.htm)  The Label widget is used to provide a single-line caption for other widgets. It can also contain images. |
| 7 | [Listbox](https://www.tutorialspoint.com/python/tk_listbox.htm)  The Listbox widget is used to provide a list of options to a user. |
| 8 | [Menubutton](https://www.tutorialspoint.com/python/tk_menubutton.htm)  The Menubutton widget is used to display menus in your application. |
| 9 | [Menu](https://www.tutorialspoint.com/python/tk_menu.htm)  The Menu widget is used to provide various commands to a user. These commands are contained inside Menubutton. |
| 10 | [Message](https://www.tutorialspoint.com/python/tk_message.htm)  The Message widget is used to display multiline text fields for accepting values from a user. |
| 11 | [Radiobutton](https://www.tutorialspoint.com/python/tk_radiobutton.htm)  The Radiobutton widget is used to display a number of options as radio buttons. The user can select only one option at a time. |
| 12 | [Scale](https://www.tutorialspoint.com/python/tk_scale.htm)  The Scale widget is used to provide a slider widget. |
| 13 | [Scrollbar](https://www.tutorialspoint.com/python/tk_scrollbar.htm)  The Scrollbar widget is used to add scrolling capability to various widgets, such as list boxes. |
| 14 | [Text](https://www.tutorialspoint.com/python/tk_text.htm)  The Text widget is used to display text in multiple lines. |
| 15 | [Toplevel](https://www.tutorialspoint.com/python/tk_toplevel.htm)  The Toplevel widget is used to provide a separate window container. |
| 16 | [Spinbox](https://www.tutorialspoint.com/python/tk_spinbox.htm)  The Spinbox widget is a variant of the standard Tkinter Entry widget, which can be used to select from a fixed number of values. |
| 17 | [PanedWindow](https://www.tutorialspoint.com/python/tk_panedwindow.htm)  A PanedWindow is a container widget that may contain any number of panes, arranged horizontally or vertically. |
| 18 | [LabelFrame](https://www.tutorialspoint.com/python/tk_labelframe.htm)  A labelframe is a simple container widget. Its primary purpose is to act as a spacer or container for complex window layouts. |
| 19 | [tkMessageBox](https://www.tutorialspoint.com/python/tk_messagebox.htm)  This module is used to display message boxes in your applications. |

Geometry Management

All Tkinter widgets have access to specific geometry management methods, which have the purpose of organizing widgets throughout the parent widget area. Tkinter exposes the following geometry manager classes: pack, grid, and place.

[The pack() Method](https://www.tutorialspoint.com/python/tk_pack.htm) − This geometry manager organizes widgets in blocks before placing them in the parent widget.

[The grid() Method](https://www.tutorialspoint.com/python/tk_grid.htm) − This geometry manager organizes widgets in a table-like structure in the parent widget.

[The place() Method](https://www.tutorialspoint.com/python/tk_place.htm) − This geometry manager organizes widgets by placing them in a specific position in the parent widget.

3.2.2 Matplotlib

Matplotlib is a plotting library for Python. It is used along with NumPy to provide an environment that is an effective open source alternative for MatLab.

matplotlib.pyplot is a plotting library used for 2D graphics in python programming language. It can be used in python scripts, shell, web application servers and other graphical user interface toolkits.

There are several toolkits which are available that extend python matplotlib functionality. Some of them are separate downloads, others can be shipped with the matplotlib source code but have external dependencies.

* Basemap: It is a map plotting toolkit with various map projections, coastlines and political boundaries.
* Cartopy: It is a mapping library featuring object-oriented map projection definitions, and arbitrary point, line, polygon and image transformation capabilities.
* Excel tools: Matplotlib provides utilities for exchanging data with Microsoft Excel.
* Mplot3d: It is used for 3-D plots.
* Natgrid: It is an interface to the natgrid library for irregular gridding of the spaced data.

3.2.3 Numpy

NumPy is a Python package. It stands for 'Numerical Python'. It is a library consisting of multidimensional array objects and a collection of routines for processing of array.

Operations using NumPy

Using NumPy, a developer can perform the following operations −

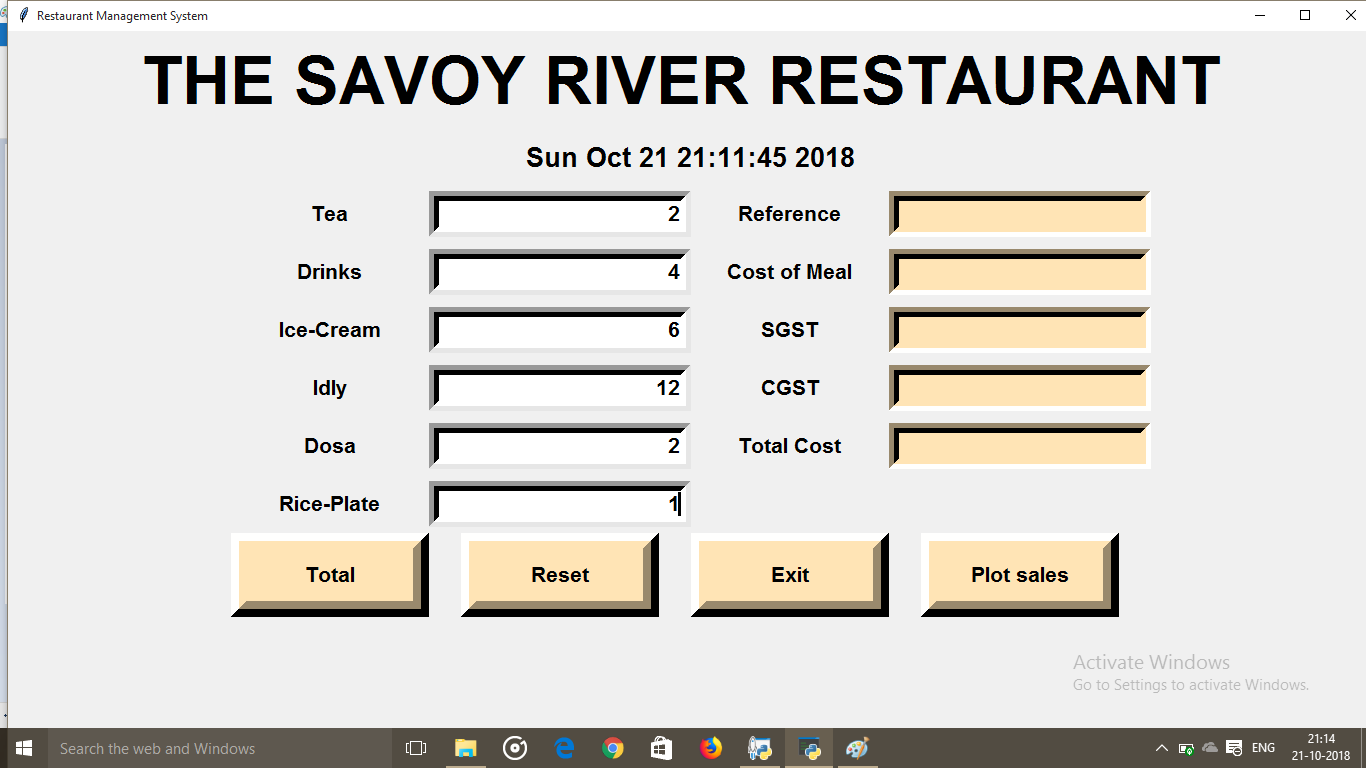
* Mathematical and logical operations on arrays.
* Fourier transforms and routines for shape manipulation.
* Operations related to linear algebra. NumPy has in-built functions for linear algebra and random number generation.

**NumPy – A Replacement for MatLab**

NumPy is often used along with packages like SciPy (Scientific Python) and Mat−plotlib (plotting library). This combination is widely used as a replacement for MatLab, a popular platform for technical computing. However, Python alternative to MatLab is now seen as a more modern and complete programming language.

It is open source, which is an added advantage of NumPy.

**3.3 GUI DESIGN FOR FRONT END**



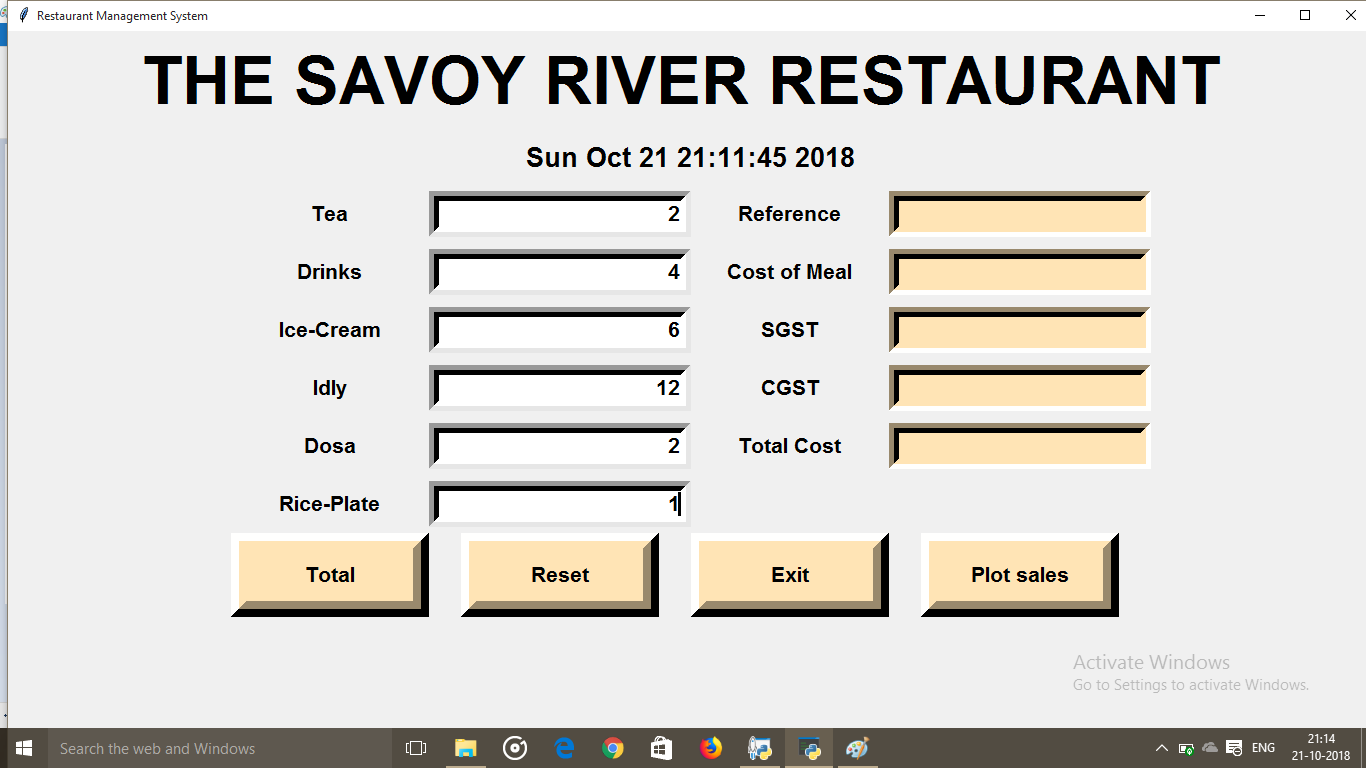
1. **RESULT**

**4.1 CHAPTER OVERVIEW**

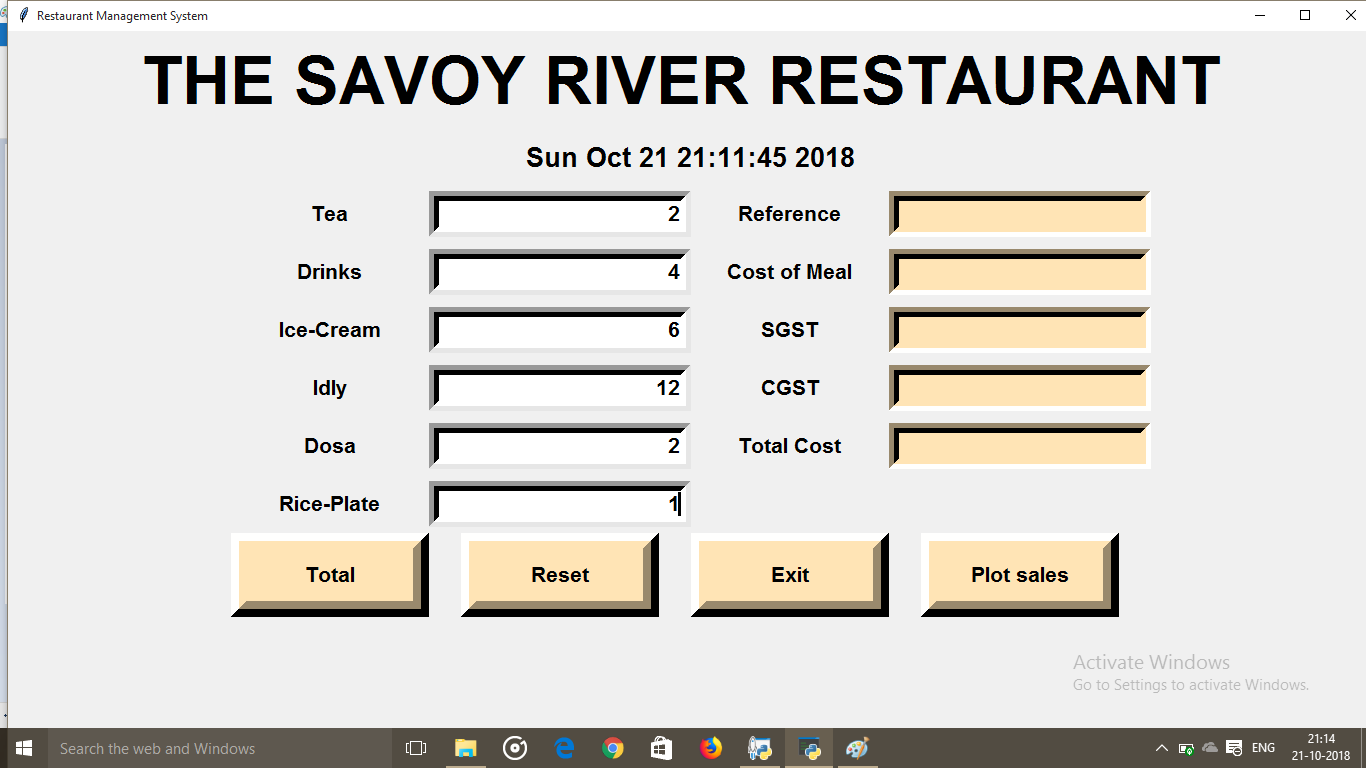
This chapter demonstrates the operation of the system, showcasing a run-through of key features and data validation.

**4.2 SNAPSHOTS OF OUTPUT**

GUI – Main display window with name of restaurant name current time and date with calculator



Buttons –



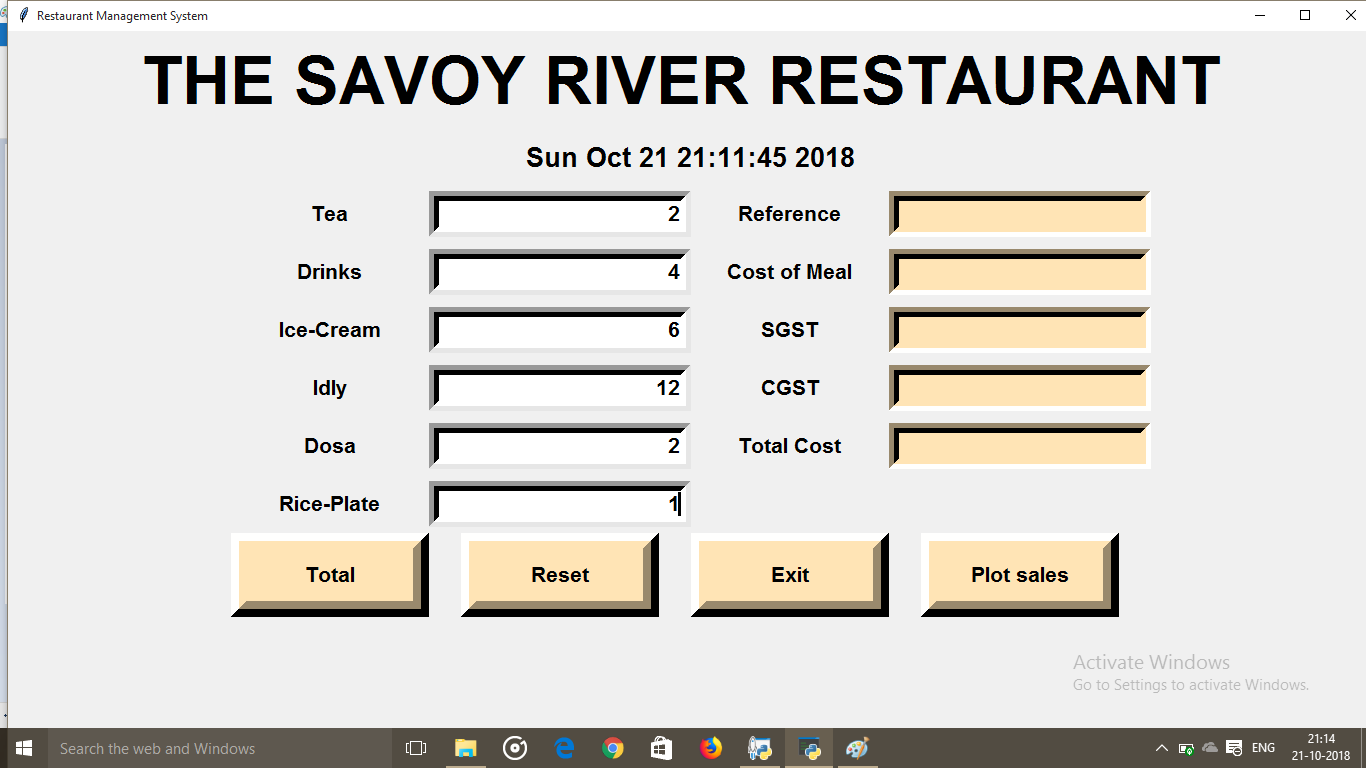
Exit button closes the window

Reset button resets all the values

Total button gives the calculation of Bill

Plot sales plot the graph as per weekly sale

We need to insert number of menu items taken by customer ,after inserting no. of items



Generated Bill

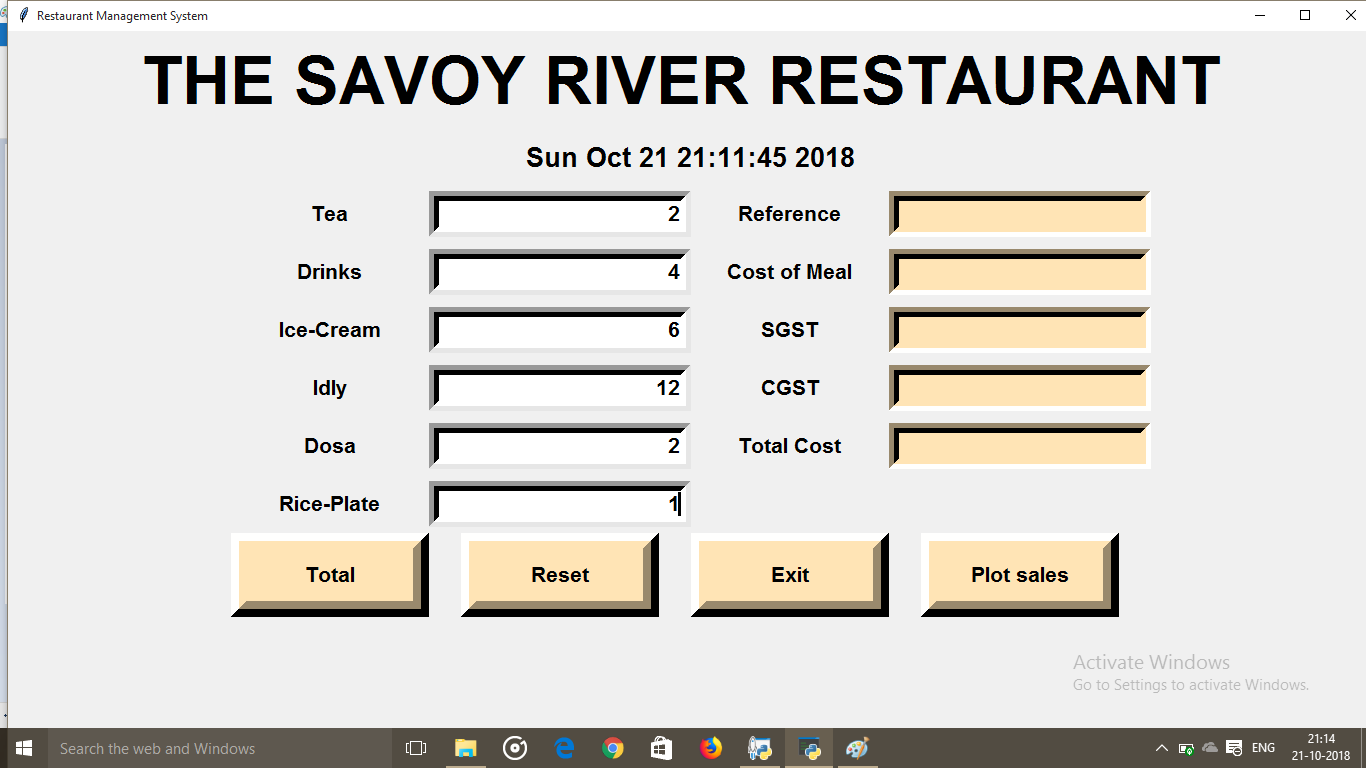
**/===== BILL NO**

**/===== TOTAL COST OF MEAL**

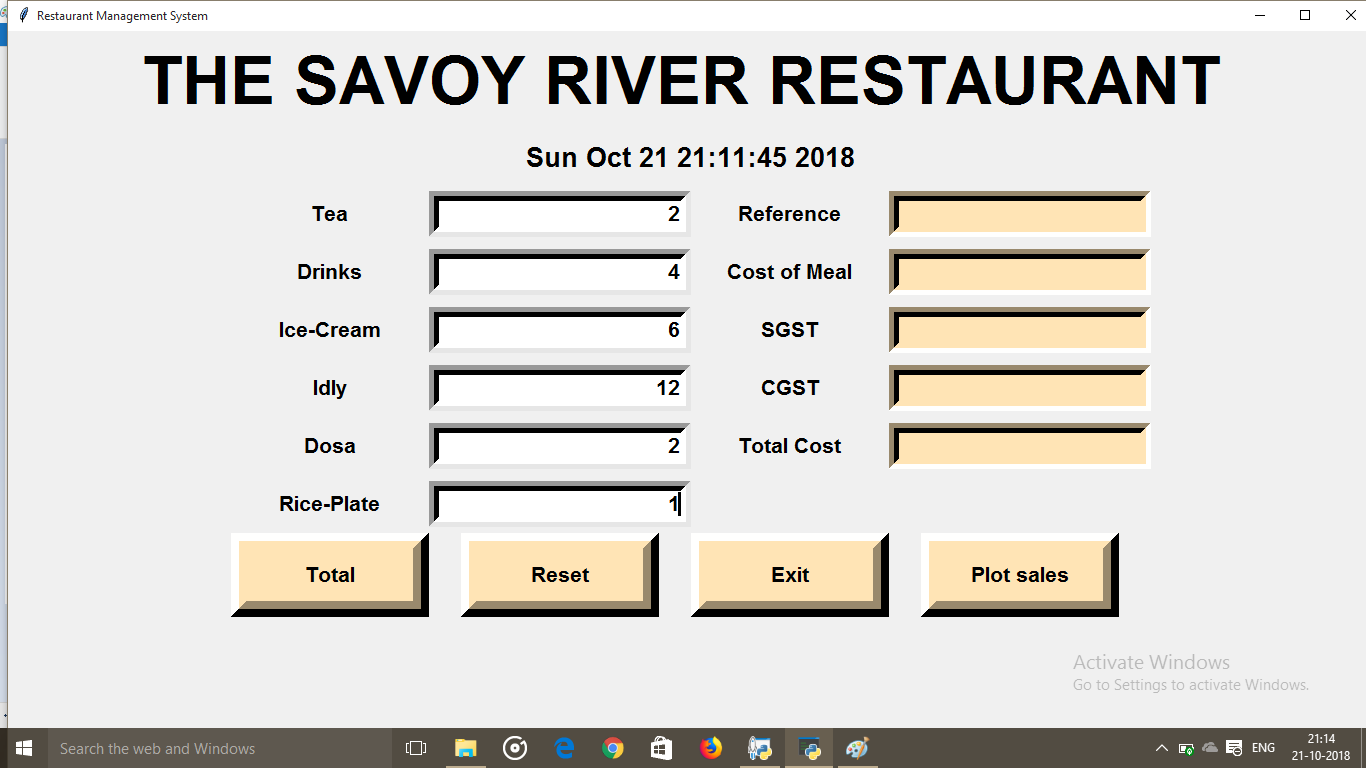
**/===== TOTAL SGST**

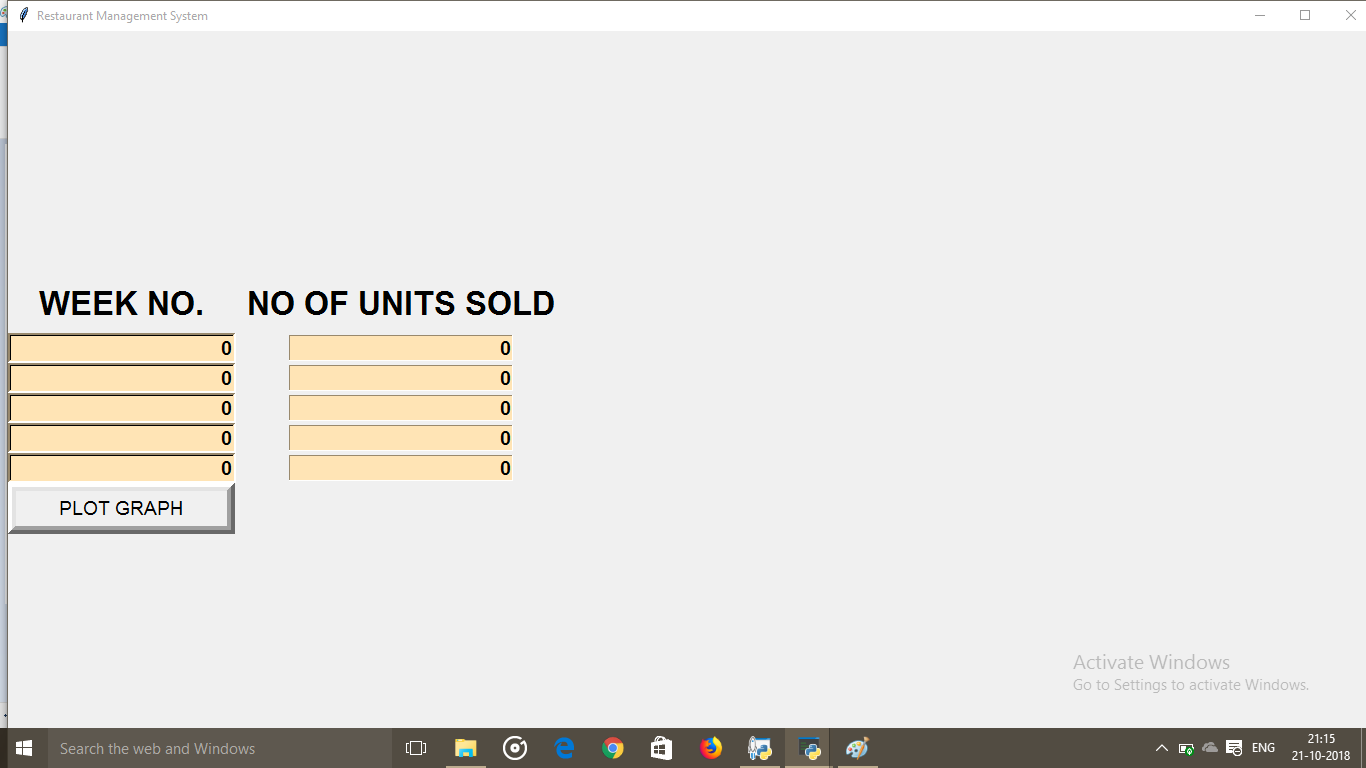
**/===== TOTAL CGST**

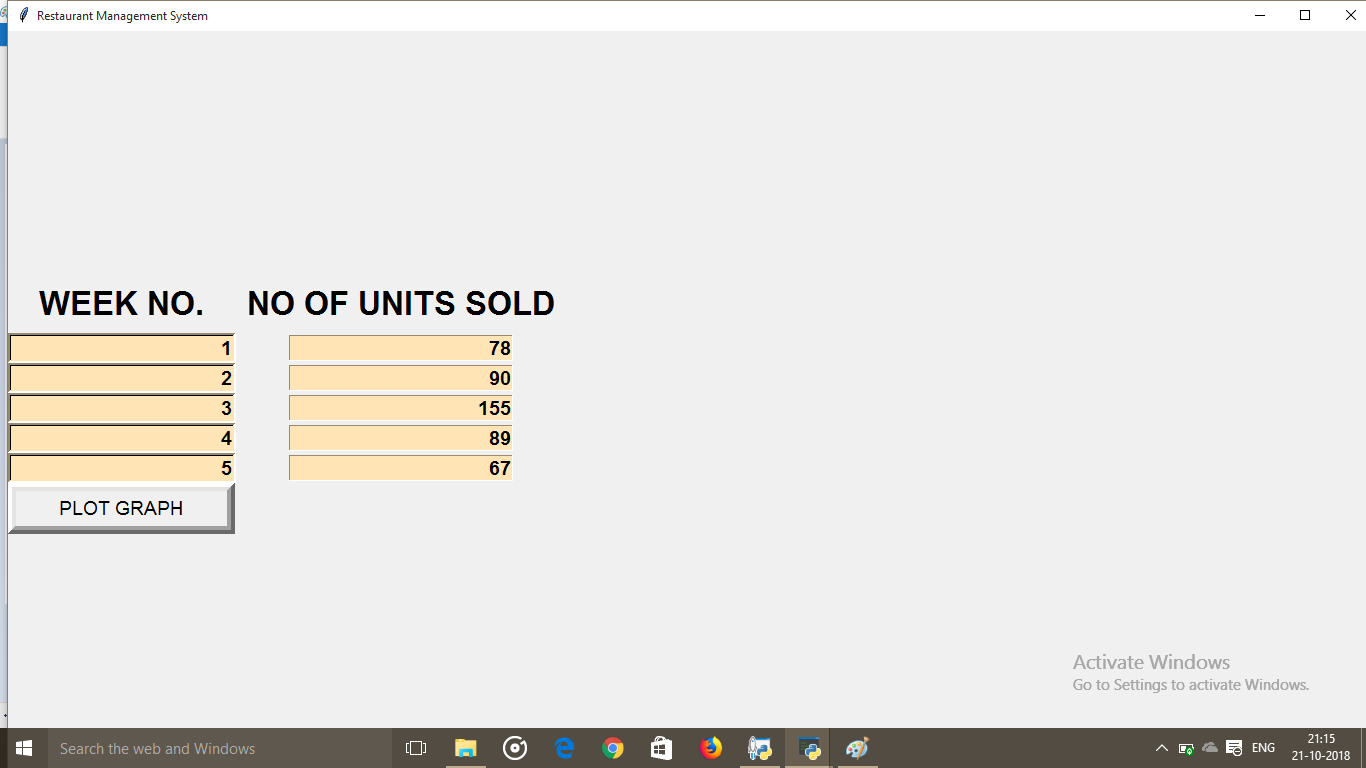
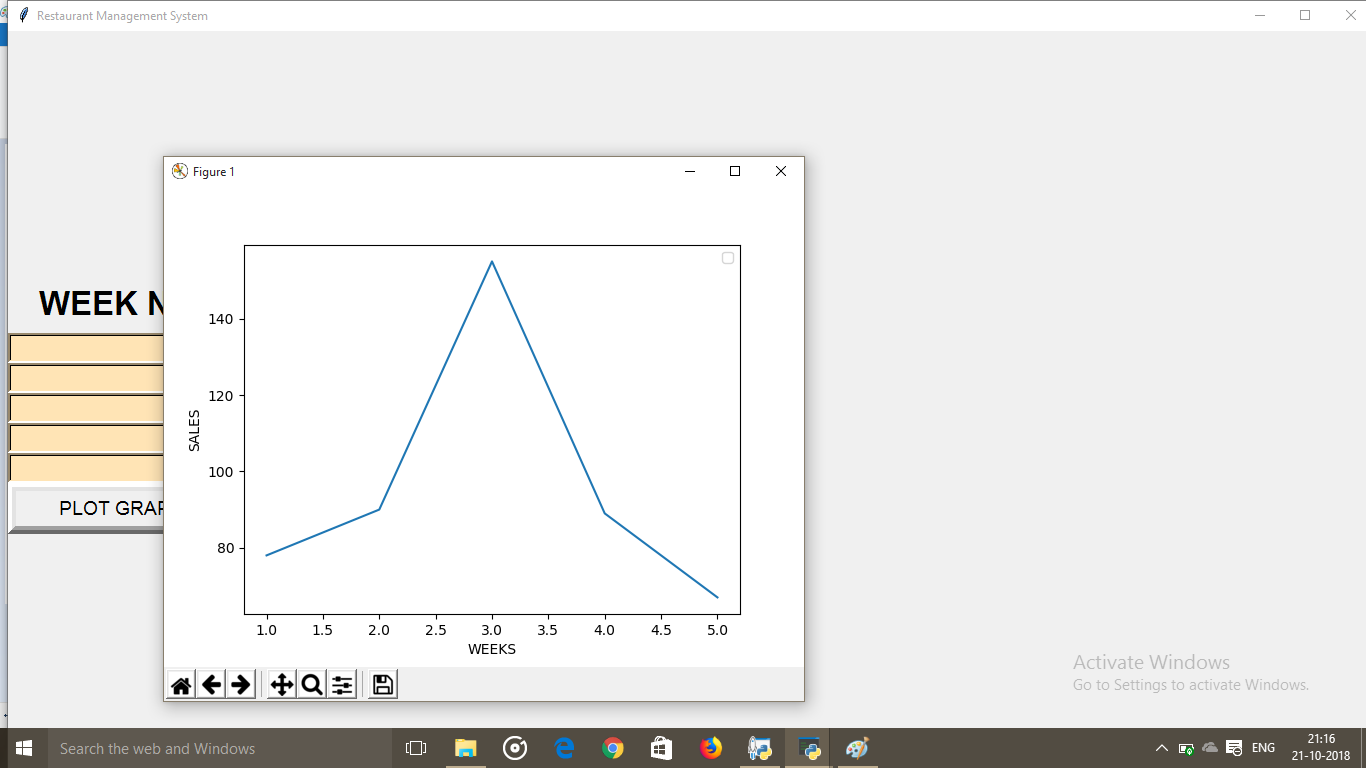
**/===== TOTAL COST INCLUDING GST**



After clicking TOTAL button it will automatically generate the bill

****

After clicking Plot sales this window will generate.



1. **CONCLUSION**

**5.1 CHAPTER OVERVIEW**

This chapter reflects on the work completed throughout the project, and evaluates the results. Highlighting the challenges presented by the project before discussing future extensions.

This project has really been faithful and informative. It has made us learn and understand the many trivial concepts of Python Language. As we have used python Tkinter as a GUI and Matplotlib and numPy it provides various controls, such as buttons, labels and text boxes to build a user friendly application.

The fast growing use of internet confirms the good future and scope of the proposed project.

Finally it has taught us a valuable lifelong lesson about the improvements and working and interacting in a group.

**5.2 FUTURE IMPROVEMENTS**

Improvements in the future could include an integrated iOS application, which would allow customers to order food directly from their own phones. The system could let customers create

reservations on their phones On the fly payments can be implemented in iOS and Mobile devices to allow staff to take payments at the table, similar to the system used in Apple Stores.

A feature which enables chef’s to inform waiters about specials and food items that are out of stock can be implemented.

1. **REFERENCES**

[1] Inamo Restaurant using E-­TABLE System [Online] Available at:

http://www.inamo-­restaurant.com/index.php

[2] E-­‐TABLE [Online] Available at:

http://www.e-­table-­interactive.com/

[3]

Eating out in modern American society: Why do people make the choice to eat

outside the home? [Online] Available at:

https://library.uvm.edu/jspui/bitstream/123456789/233/1/Epter%20Thesis.pdf

[4]

Does a Restaurant Need a Point of Sale System? [Online] [Video] Available at:

https://www.youtube.com/watch?v=c7ELx2BLeWo