**A PROJECT REPORT ON**

**GUI BASED MARKSHEET**



Submitted in the partial fulfilment of award of

**MASTER OF COMPUTER APPLICATIONS**

Degree In

Computer Science and Engineering

Submitted To:

Dr Rajendra Kumar Dwivedi

Submitted By:

ABHAY Kr. SHUKLA

## DECLARATION

This is to certify that the project report entitled “**GUI Based Marksheet**” is done by me is an authentic work carried out for the partial fulfilment of the requirements for the award of the degree of MCA in Information Technology under the guidance of **Dr.Rajendra Kumar Dwivedi**. The matter embodied in this project work has not been submitted earlier for award of any degree or diploma to the best of my knowledge and belief.

Signature of the student

ABHAY KUMAR SHUKLA

2020104003 MCA

Madan Mohan Malaviya University of Technology, Gorakhpur

## CERTIFICATE

This is to certify that project entitled “**GUI Based Marksheet**” submitted for partial fulfillment of the degree of Master of Computer Application to through Madan Mohan Malaviya University of Technology, Gorakhpur, done by **Mr. ABHAY KUMAR SHUKLA**, Roll No. 2020104003 is an authentic work carried out by her under my guidance. The matter embodied in this project work has not been submitted earlier for award of any degree or diploma to the best of my knowledge and belief.

Internal Examiner/Guide External Examiner

Head of Department

## ACKNOWLEGEMENT

We had a great experience working on this project and we got to learn a plethora of new skills through this project. However, it would not have been possible without the kind support and help of many individuals. We would like to extend our sincere thanks to all of them. We are highly indebted to the teachers and especially Dr Rajendra Kumar Dwivedi for their guidance and constant supervision as well as providing necessary information regarding the project and also for their support in completing the project. We would like to express our gratitude towards our parents and friends for their kind cooperation and encouragement which help us in the completion of the project.

## ABSTRACT

A GUI Based Marksheet can be very useful within a administrative environment. Instead of marksheet manually or to sum up the total manually, it is very much time consuming and also may have some human errors like adding up the wrong total or adding wrong items into the marksheet. When making a hand-written marksheet the owner and customer both have to repeatedly check the total, subjects added, etc. It also sometimes results into a Bad Impression towards the school administration from a Customer. Ideally, user should be able to generate marksheet without any mistakes and quickly, enabling them to fasten or improve their process. To overcome this problem, we have come up with this project, that is, Gui Based Marksheetta Using Python.

## INDEX

1. Cover Page as per format
2. Declaration
3. Certificate
4. Acknowledgement
5. Abstract
6. Index/List of Contents consists following Chapters:
   * Chapter-I: Introduction 1
     + Introduction To Project 2
     + Objective 3
     + Project Scope 4
   * Chapter-II: System Analysis 5
     + SDLC Model 6-9
   * Chapter-III: Feasibility Study 10
     + Technical Feasibility 11
     + Operational Feasibility 11
     + Economic Feasibility 11
   * Chapter-IV: Environment 12
     + Hardware Specification 13
     + Language Used 13
   * Chapter-V: System Design 14
     + Data Flow Diagram 15-17
     + ER Diagram 18-21
   * Chapter-VI: Technology and Description 22-26
   * Chapter-VII: Output Screens 27-29
   * Conclusion 30
   * Bibliography 31

# CHAPTER-I INTRODUCTION

## INTRODUCTION

GUI Based Marksheet Using Python can be very useful within a business environment. Instead of doing manual work for making up a marksheet at School Administration, which gets tiring and time consuming, you can generate a marksheet including marks and subjects in just few clicks. When making up a marksheet manually at a School Administration may contain some human errors like adding wrong items into the marksheet or summing up their total also may end up wrong, it also sometimes results into a Bad Impression towards the School Administration from a Customer. Ideally, user should be able to generate marksheet without any mistakes and quickly, enabling them to fasten or improve their process. To overcome this problem, we have come up with this project, that is, GUI Based Marksheet Using Python.

The GUI Based Marksheet Using Python is very useful to small business or School Administration or cafe or food truck owners. This helps the owner to fasten the process which is bug free and easy to use. It also has a calculator to ease the use of the user. This project firstly has the menu and then adds up the selected items by customer and sums up the total of all items adds tax and service charges and displays total. To perform any other operation like division, multiplication, etc calculator is also

## OBJECTIVE

A GUI Based Marksheet can be very useful within a business environment. Instead of making marksheets manually or to sum up the total manually, it is very much time consuming and also may have some human errors like adding up the wrong total or adding wrong items into the marksheet. When making a hand written marksheet the owner and customer both have to repeatedly check the total, items added, etc. It also sometimes results into a Bad Impression towards the School Administration from a Customer. Ideally, user should be able to generate marksheet without any mistakes and quickly, enabling them to fasten or improve their process. To overcome this problem, we have come up with this project, that is, GUI Based Marksheet Using Python.

## SCOPE OF PROJECT

* This project will help the user in fast marksheeting.
* It will generate receipt on every transaction inputted to the system.
* The software will display view of calculations of every transaction.

# CHAPTER -II SYSTEM ANALYSIS

## SDLC MODEL

### What is SDLC?

Software development is the bread and butter of software engineers and developers all around. Every day, software engineers and professionals alike have to immerse themselves into the dynamics of the best Software Development Lifecycle (SDLC) methodology and approach to develop and deliver software in optimum conditions. But what is SDLC?

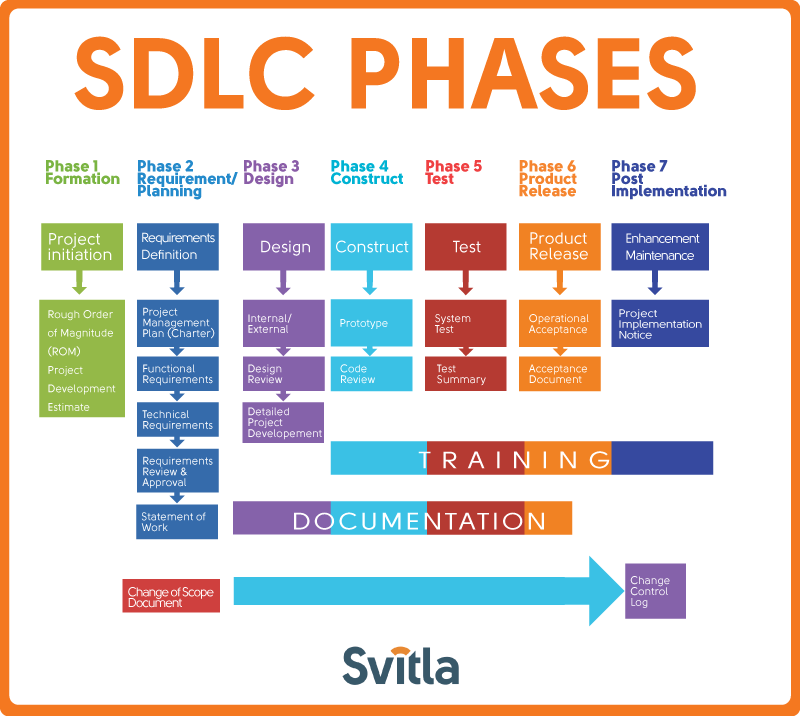
In the simplest terms, SDLC methodologies provide a systematic framework to design, develop and deliver software applications, from beginning to end. It is a series of steps that offer a foundation for the software development process. Having a structure to develop software is fundamental, which is why there are multiple software development methodologies available to choose from. It is increasingly important for software engineers to select the right SDLC model that meets specific requirements and concerns of the project to drive success. In this article, we go into the details of SDLC methodologies, their relevance, their advantages, disadvantages, and everything in between.

To a certain extent, SDLC methodologies can be thought of like a checklist of the different stages that must be performed to develop and deliver successful software applications. All SDLC methodologies share a common ground of distinct phases that include planning, analysis, design, building, testing, deploying, and maintenance. These SDLC phases provide the outline of what a software application project entail.

In the following section, we are going to explore how software development lifecycles impact the software development process.

* + The Software Development Process
  + The software development process, as with all great projects, starts with an idea. It takes planning, preparation, and management of phases and team members to reach a goal. SDLC is a mapped-out, regulated framework that typically follows the following universal phases to deliver high-quality software application.

## SDLC phases



### Formation phase

This basic, initial phase is the inception of an idea for a solution that improves an existing solution or develops an entirely new one. It helps define the magnitude of the project to plan resources.

### Requirement/Planning Phase

In this phase, requirements are gathered to formulate a design plan for the software application solution. This phase entails a thorough analysis to assess user needs, feasibility, development, improvements, and more. It is very important to include documentation to refine requirements and keep a record of the solution’s development. This phase involves the creation of a project charter which defines technical and functional requirements.

### Design Phase

This phase is focused on the design aspect of the software application solution in terms of the selected technical and functional requirements and the results of the thorough analysis of the software’s viability.

### Development Phase

This phase is the “meat” of the software development process. In this phase, software engineers are solely focused on building a prototype of the solution to perform a code review and ultimately create the solution itself. The team works on transforming software specifications into a working and reliable solution.

### Testing Phase

This crucial phase tests the software to ensure that everything works as it intended. In the testing phase, software engineers are able to detect defects, bugs, and errors in the software solution and ultimately have a quality product that meets business expectations. Quality Assurance (QA) specialists perform a series of tests to evaluate the status of the solution.

### Release Phase

Once the software application is fully developed and tested, it moves to the release phase.

In this phase, the software goes live and is released to the end user for

actual use of the product. In essence, the software is fully operational in a live environment where end users utilize it.

### Maintenance Phase

This post-release phase is tasked with keeping the software completely operational, updating it to meet quality standards, and enhancing it throughout its life to ensure it continues to attract and retain users.

The software development process sets the tone and defines a goal from which developers kick-start a project. Ultimately, following a software development process is intended to develop software faster and with a few hiccups as possible.

Now that we’ve covered the universal SDLC phases, let’s assess how important it is to follow software development methodologies in an IT environment.

# CHAPTER -III FEASIBILITY STUDY

#### Technical Feasibility:

The proposed system is developed using Active Server Page, VB Script and HTML as front-end tool and Oracle 8 as the back end. The proposed system needs a Personal Web Server to serve the requests submitted by the users. The Web browser is used to view the web page that is available within the Windows operating system itself. The proposed system will run under Win9x, NT, and win2000 environment. As Windows is very user friendly and GUI OS it is very easy to use. All the required hardware and software are readily available in the market. Hence the system is technically feasible.

#### Operational Feasibility

Operational feasibility is a measure of how well a proposed system solves the problems and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

Operational feasibility reviews the willingness of the organization to support the proposed system. This is probably the most difficult of the feasibilities to gauge. In order to determine this feasibility, it is important to understand the management commitment to the proposed project. If the request was initiated by management, it is likely that there is management support, and the system will be accepted and used.

However, it is also important that the employee base will be accepting of the change. The proposed system is operationally feasible because of the following reasons.

1. The customer is benefited more as most of his time is saved. The customer is serviced at his place of work.
2. The cost of the proposed system is almost negligible when compared to the benefits gained.

#### Economical Feasibility

As the necessary hardware and software are available in the market at a low cost, the initial investment is the only

cost incurred and does not need any further enhancements. Hence it is economically feasible.

The system is feasible in all respects and hence it encourages taking up the system design.

# CHAPTER -IV ENVIRONMENT

#### DEVELOPER END:

**HARDWARE REQUIREMENTS:**

Processor : Intel i3 Processor or above.

RAM : 4GB

Hard disk : 40GB or above.

Monitor : 15’’ LCD or CRT Monitor or above.

Keyboard : Standard windows keyboard

#### SOFTWARE REQUIREMENTS:

Development Kit : Python 3 or above. IDE : Vs Code

Platform : Window 7/XP And Above.

#### LANGUAGE AND TECHNOLOGY:

Language : Python.

Platform : Vs Code

# CHAPTER -V SYSTEM DESIGN

## Data Flow Diagram:

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyse an existing system or model a new one. Like all the best diagrams and charts, a DFD can often visually “say” things that would be hard to explain in words, and they work for both technical and nontechnical audiences, from developer to CEO. That’s why DFDs remain so popular after all these years. While they work well for data flow software and systems, they are less applicable nowadays to visualizing interactive, real-time or database-oriented software or systems.

## Data Flow Diagram

Data flow diagrams were popularized in the late 1970s, arising from the book *Structured Design*, by computing pioneers Ed Yourdon and Larry Constantine. They based it on the “data flow graph” computation models by David Martin and Gerald Ezrin. The structured design concept took off in the software engineering field, and the DFD method took off with it. It became more popular in business circles, as it was applied to business analysis, than in academic circles.

Also contributing were two related concepts:

* + Object Oriented Analysis and Design (OOAD), put forth by Yourdon and Peter Coad to analyse and design an application or system.
  + Structured Systems Analysis and Design Method (SSADM), a waterfall method to analyse and design information systems. This rigorous documentation approach contrasts with modern agile approaches such as Scrum and Dynamic Systems Development Method (DSDM.)

## DFD rules and tips:

* + Each process should have at least one input and an output.
  + Each data store should have at least one data flow in and one data flow out.
  + Data stored in a system must go through a process.
  + All processes in a DFD go to another process or a data store.
  + **DFD Level 1**: Provides a more detailed breakout of pieces of the Context Level Diagram. You will highlight the main functions carried out by the system, as you break down the high-level process of the Context Diagram into its subprocesses.
  + **DFD Level 2**: Then goes one step deeper into parts of Level 1. It may require more text to reach the necessary level of detail about the system’s functioning.

## Examples of how DFDs can be used:

Data flow diagrams are well suited for analysis or modelling of various types of systems in different fields.

**DFD in software engineering:** This is where data flow diagrams got their main start in the 1970s. DFDs can provide a focused approached to technical development, in which more research is done up front to get to coding.

**DFD in business analysis:** Business analysts use DFDs to analyse existing systems and find inefficiencies. Diagramming the process can uncover steps that might otherwise be missed or not fully understood.

**DFD in business process re-engineering:** DFDs can be used to model a better, more efficient flow of data through a business process. BPR was pioneered in the 1990s to help organizations cut operational costs, improve customer service and better compete in the market.

**DFD in agile development:** DFDs can be used to visualize and understand business and technical requirements and plan the next steps. They can be a simple yet powerful tool for communication and collaboration to focus rapid development.

**DFD in system structures:** Any system or process can be analysed in progressive detail to improve it, on both a technical and non-technical basis

Diagram

Description automatically generated

**Fig -2**: GUI Based Marksheet Using Python

Diagram

Description automatically generated

**Fig -2:** User Case Diagram

## ER DIAGRAM:

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes. They mirror grammatical structure, with entities as nouns and relationships as verbs.

ER diagrams are related to data structure diagrams (DSDs), which focus on the relationships of elements within entities instead of relationships between entities themselves. ER diagrams also are often used in conjunction with data flow diagrams (DFDs), which map out the flow of information for processes or systems.

Uses of entity relationship diagrams

#### Database design:

ER diagrams are used to model and design relational databases, in terms of logic and business rules (in a logical data model) and in terms of the specific technology to be implemented (in a physical data model.) In software engineering, an ER diagram is often an initial step in determining requirements for an information systems project. It’s also later used to model a particular database or databases. A relational database has an equivalent relational table and can potentially be expressed that way as needed.

#### Database troubleshooting:

ER diagrams are used to analyse existing databases to find and resolve problems in logic or deployment. Drawing the diagram should reveal where it’s going wrong.

#### Business information systems:

The diagrams are used to design or analyse relational databases used in business processes. Any business process that uses fielded data involving entities, actions and interplay can potentially benefit from a relational database. It can streamline processes, uncover information more easily and improve results.

#### Business process re-engineering (BPR):

ER diagrams help in analysing databases used in business process re-engineering and in modelling a new database setup.

#### Education:

Databases are today’s method of storing relational information for educational purposes and later retrieval, so ER Diagrams can be valuable in planning those data structures.

#### Research:

Since so much research focuses on structured data, ER diagrams can play a key role in setting up useful databases to analyse the data.

The components and features of an ER diagram

ER Diagrams are composed of entities, relationships and attributes. They also depict cardinality, which defines relationships in terms of numbers. Here’s a glossary:

#### Entity:

A definable thing—such as a person, object, concept or event—that can have data stored about it. Think of entities as nouns. Examples: a customer, student, car or product. Typically shown as a rectangle.

#### Entity type:

A group of definable things, such as students or athletes, whereas the entity would be the specific student or athlete. Other examples: customers, cars or products.

#### Entity set:

Same as an entity type, but defined at a particular point in time, such as students enrolled in a class on the first day. Other examples: Customers who purchased last month, cars currently registered in Florida. A related term is instance, in which the specific person or car would be an instance of the entity set.

#### Entity categories:

Entities are categorized as strong, weak or associative. A **strong entity** can be defined solely by its own attributes, while a **weak entity** cannot. An associative entity associates entities (or elements) within an entity set.

## SYMBOLS USED IN E-R DIGRAM

**Entity**

**Relationship**

**Attribute Weak Entity**

**Weak Entity Relationship**

**Multivalued Attribute**

**Key Attribute**

**E-R DIAGRAM**

**Diagram

Description automatically generated**

**Fig -3**: GUI Based Marksheet Using Python Flowchart

# CHAPTER -VI

**TECHNOLOGY AND DESCRIPTION**

## TECHNOLOGY:

What is Python?

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

It is used for:

* + web development (server-side),
  + software development,
  + mathematics,
  + system scripting. What can Python do?
  + Python can be used on a server to create web applications.
  + Python can be used alongside software to create workflows.
  + Python can connect to database systems. It can also read and modify files.
  + Python can be used to handle big data and perform complex mathematics.
  + Python can be used for rapid prototyping, or for production-ready software development.

Why Python?

* + Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
  + Python has a simple syntax similar to the English language.
  + Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
  + Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
  + Python can be treated in a procedural way, an object-oriented way or a functional way.

Good to know

* + The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular.
  + In this tutorial Python will be written in a text editor. It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse which are particularly useful when managing larger collections of Python files.

Python Syntax compared to other programming languages

* + Python was designed for readability, and has some similarities to the English language with influence from mathematics.
  + Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
  + Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

## Source Code and Commands:

Text

Description automatically generated**Text

Description automatically generated**

(1) (2)

Text

Description automatically generated

(3)

Text

Description automatically generated

(4)

Text

Description automatically generated

(5)

Text

Description automatically generated

**CHAPTER - VI OUTPUT SCREEN**

* + GUI – Main display window with name of School Administration name current time and date with calculator

Graphical user interface, text, application

Description automatically generated

* + Buttons

Shape

Description automatically generated with medium confidence



* + Generated Marksheet

Graphical user interface, application

Description automatically generated

## Conclusion

The urge for the digital GUI Based Marksheets is increasing day by date. GUI Based Marksheet Using Python is a perfect solution for this. Through this the ease of access and flexibility of the day to day works in the School Administration is made simpler. The features such as marksheet number, CGPA and SGPA make this software user friendly. Both the management side and worker site can manage the data easily using such a system. It is very good and reliable system which can be in corporate to the chain of schools so can easily maintained and addressed.

## Bibliography

1. Mayur Jain,” Smart Home System Using Android Mobile Devices”, Journal of Computing Technologies Vol 2, Issue 3 ISSN 2278 – 3814.
2. IEEE std. 802.15.4 - 2003: “Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low-Rate Wireless Personal Area Networks (LRWPANs)”
3. KiumiAkingbehin, Akinsola Akingbehin., “Alternativ es for Short Range Low Power Wireless Communications,” IEEE.2005: 94 - 95.
4. QIN Tinghao, DOU Xiaoqian, “Application of ZigBee Technology in Wireless Sensor Network,” Instrumentation Technology, 2007, pp.57-59.
5. A.R. Al-Ali and M. AL-Rousan. “Java-Based Home Automation System”. IEEE Transaction on Consumer Electronics, Vol.50, No. 2, May 2004.
6. N. Sriskanthan and Tan Karand. “Bluetooth Based Home Automation System”. Journal of Microprocessors and Microsystems, Vol. 26, pp.281-289, 2002