

STAFF MANAGEMENT SYSTEM

A Project Report

Submitted in partial fulfillment of the
Requirements for the award of the Degree of

BACHELOR OF SCIENCE (COMPUTER SCIENCE)

By

Panchal Yash Panchabhai

Seat Number – 533

Under the esteemed guidance of

Ms. Niramaye Deshpande

Assistant Professor



DEPARTMENT OF COMPUTER SCIENCE

NAGINDAS KHANDWALA COLLEGE

(Autonomous College)

(Affiliated to University of Mumbai)

MUMBAI 400064

MAHARASHTRA

2020-2021

(Note: All entries of the proforma of approval should be filled up with appropriate and complete information. Incomplete proforma of approval in any respect will be summarily rejected.)

PNR No.:

Roll no: 533

1. Name of the Student: Panchal Yash Panchabhai
2. Title of the Project: Staff Management System
3. Name of the Guide: Ms. Niramaye Deshpande
4. Teaching/Industry experience of the Guide: 04
5. Is this your first submission? Yes



Signature of the Student:

Signature of the Guide:

Date:

Date:

Signature of the Coordinator:

Date:

NAGINDAS KHANDWALA COLLEGE

(Affiliated to University of Mumbai)

MUMBAI-MAHARASHTRA-400064

DEPARTMENT OF COMPUTER SCIENCE



CERTIFICATE

This is to certify that the project titled, "**Staff Management System**", is bonafied work of **PANCHAL YASH PANCHABHAI** bearing Seat. No: 533 submitted in partial fulfillment of the requirements for the award of degree of BACHELOR OF SCIENCE in COMPUTER SCIENCE from University of Mumbai.

Internal Guide

Coordinator

(Don't write names of lecturers or HOD)

External Examiner

Date: 12-04-2021

College Seal

COMPANY CERTIFICATE

॥ जय चामुंडा माता नमः ॥

॥ श्री गणेशाय नमः ॥

Mob.: 8286344062
9757105494

॥ जय जापडी मां नमः ॥

CHAMUNDA FABRICATION

Room No-9, Giraja Niwas, Shivaji Nagar Road, Kurar Village, Malad (East), Mumbai - 400097.

D= 1/4/2021

To, whom so ever it may concern

This is to inform you that the student yash Panchal of Nagindas Khandwala college has successfully developed 'Staff Management System' for the Chamunda Fabrication. He was working on this project from 28-11-2020 to 17-3-2021.

For security reasons student is not allowed to disclose code in documentation.

Thanks & Regards



P. Panchal.

Abstract

Python is a high-level, interpreted and general-purpose dynamic programming language that focuses on code readability. The syntax in python helps the programmers to do coding in fewer steps as compared to Java or C++.

Python has topped the charts in the recent years over other programming languages like C, C++ and Java is widely used by the programmers. The language has undergone a drastic change since its release 25 years ago as many add-on features are introduced. The Python 1.0 had the module system of modula-3 and interacted with Amoeba Operating System with varied functioning tools.

The programmers of big companies use Python as it has created a mark for itself in the software development with characteristic features like –

- Interactive
- Interpreted
- Modular
- Dynamic
- Object – oriented
- Portable
- High level

The Python language has diversified application in the software development companies such as gaming, web frameworks and applications, language development, prototyping, graphic design application, etc.

It provides large standard libraries that include the areas like string operations, Internet, Web service tools, operating system interfaces and protocols. Most of the highly used programming tasks are already scripted into it that limits the length of the codes to be written in Python.

ACKNOWLEDGEMENT

It is my pleasure to be indebted to various people, who directly or indirectly contributed in the development of this work and who influenced our thinking, behavior and acts during the course of study.

I express my sincere gratitude to **Principal Dr. (Mrs.) Ancy Jose** for providing me an opportunity to undergo this Project as the part of the curriculum.

I am thankful to **Coordinator Dr. Sindhu P.M** for her support, cooperation, and motivation provided to me during the training for constant inspiration, presence and blessings.

I would also like to thank **Project guide Ms. Niramaye Deshpande** for his valuable suggestions which helps us lot in completion of this project.

Lastly, we would like to thank the almighty and our parents for their moral support and friends with whom we shared our day-to-day experience and received lots of suggestions that improved our quality of work.

Thanks for all your encouragement!

DECLARATION

I hereby declare that the project entitled, “**Staff Management System**” done at **Nagindas Khandwala College, Malad (West)**, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of degree of **BACHELOR OF SCIENCE (COMPUTER SCIENCE)** to be submitted as fifth semester project as part of our curriculum.

Panchal Yash Panchabhai



Name and Signature of the Student

TABLE OF CONTENTS

Chapter 1: Introduction

.....	
1.1 Background.....	10
1.2 Objective.....	11
1.3 Purpose and Scope	
1.2.1 Purpose.....	12
1.2.2 Scope.....	13
1.2.3 Applicability.....	13

Chapter 2: System Analysis

.....	
2.1 Existing System.....	14
2.2 Proposed System.....	14
2.3 Requirement Analysis.....	14
2.4 Hardware Requirements.....	14
2.5 Software Requirements.....	15
2.6 Justification of selection of Technology.....	16

Chapter 3: System Design

.....	
3.1 Module Division.....	17
3.2 Data Dictionary.....	18
3.3 ER Diagrams.....	20
3.4 UML Diagrams.....	21

Chapter 4: Implementation and Testing

.....	
4.1 Code (Place Core segments).....	28
4.2 Testing Approach.....	29
4.2.1 Unit Testing (Test cases and Test Results).....	32
4.2.2 Integration System (Test cases and Test Results).....	33

Chapter 5: Results and Discussions (Output Screens).....	37
Chapter 6: Conclusion and Future Work.....	38
Chapter 7: References.....	39
Chapter 8: Glossary.....	40

List of Tables

Table 1: Hardware Requirements.....	15
Table 2: Software Requirements.....	15
Table 3: Add Staff.....	18
Table 4: Attendance.....	18
Table 5: Salary.....	19
Table 6: Software Project System Type.....	32
Table 7: Login Test case.....	35

List of Figures

Fig 1: ER Diagram.....	20
Fig 2: Class Diagram.....	22
Fig 3: Use Case Diagram.....	24
Fig 4: Activity Diagram.....	26
Fig 5: Sequence Diagram.....	28
Fig 6: Gantt Chart.....	39

Chapter 1

Introduction

1.1 Background

Staff Worker's are the backbone of any company therefore their management plays a major role in deciding the success of an organization. Human Resource Management Software makes it easy for the employer to keep track of all records. This software allows the administrator to ed employees, add new employees as well as evaluate an employee's performance. Worker can be managed efficiently without having to retype back their information in the database. You can check to see if there are duplicate positions/employees in the database.

A flexible and easy to use Staff Management software solution for small and medium sized companies provides modules for personnel information management thereby organization and companies are able to manage the crucial organization asset people The combination of these modules into one application assures the perfect platform for re-engineering and aligning. Human Resource processes along with the organizational goals. This system brings about an easy way of maintaining the details of workers working in any organization.

It is simple to understand and can be used by anyone who is not even familiar with simple staff management system. It is user friendly and just asks the user to follow step by step operations by giving easy to follow options. It is fast and can perform many operations for a company.

The staff record system is very simple and for very beginner mini project. It is based on the menu-driven program for elementary database management. This project is a learning milestone for beginners who want to step into the database management project in Python.

The aim of this project is the work done or achieved and to give an assessment of the completed system, discuss the problem faced, limitation of the system and give future recommendations on how the system can be improved. The software product produced was fairly good, it achieved most of the user requirements, the user interface is good. The admin side validation is excellent.

1.2 Objectives

In this world of growing technologies, everything has been computerized. With a large number of work opportunities, the human workforce has increased. Thus there is a need for a system that can handle the data of such a large number of workers in an organization. This project simplifies the task of maintaining records because of its user-friendly nature.

The “STAFF MANAGEMENT SYSTEM” has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and in some cases reduce the hardships faced by this existing system. Moreover, this system is designed for the particular need of the company to carry out operations in a smooth and effective manner.

The main objective of the Project on Staff Management System is to manage the details of staff, salary, leaves, attendance, registration. It manages all the information about staff, registration, attendance. This project is totally built at the administrative end and thus only the administrator is guaranteed access. The purpose of the project is to build an application program to reduce the manual work for managing the worker, salary, work details. It manages all the details about the staff's attendance, registration.

The application is reduced as much as possible to avoid errors while entering the data. It also provides error messages while entering invalid data. No formal knowledge is needed for the user to use this system. This system. Thus by this, all it provides is user-friendly.

This project will allow the admin to add new workers after proper authentication. Admin can also mark the attendance and add the payment. It can also allocate worker's different work. The database should store all personal details of workers such as date of birth full name, email, dob, phone no, etc.

Without a staff management system, it's a tedious job for the human resource department to keep track of each and every worker and even harder for big contractor managers to assign tasks to the project team. An organization or company with a very large number of workers manages a greater volume of data. This activity can be daunting without a more sophisticated tool to store and retrieve data. The various levels of sophistication can be examined by looking at the evolutionary aspects of characterized into four stages of development: Personal Computer technology, Electronics database, Personalized database, and Graphic user interface.

User role check – after logging in, the user role will be checked from the database and the user interface will be displayed according to their role.

The objectives for this project are:

- The main objective of the “Staff Management System” is to manage worker for small business.
- A user-friendly front-end for the user to interact with the system.
- This project aims to simplify the task of maintaining a record of the workers of company.
- To develop a well-Designed Database to store worker’s information.
- To provide a comprehensive approach towards the management of workers information.
- An easy way to automate all functionalities of the employees of the Company.
- Provides full functional reports to management of the company.
- Committed to bringing the best way of management in the various forms of Staff Management.
- Manage the information of workers.
- To increase the efficiency of managing the attendance, salary.
- Tools to manage the inner operation of company-related worker’s Details, leave Salary and Attendance etc.

1.3 Purpose, Scope and Applicability

1.3.1 Purpose

- Manage the employees in a better way.
- Easy to find the various information related to employees.
- Maintains the salary, attendance , events, leaves and requests.
- Maintains the date of Hiring and Retirement.
- Less time consuming and easy to use.
- Provide computerized system for maintaining records.
- Huge data storage with less computer memory.
- Avoid human errors & efforts for maintaining daily data.

1.3.2 Scope

The scope of this project will be limited to the following:

- Admin will have access to their profiles and will be able to edit their details
- Complete elimination of paperwork in leave management by enabling an employee apply for leave as well as check their leave status through the system.
- Electronic leave application
- Work's in multiple pc's
- Administrator can use it with ease.
- More efficient & reliable
- Avoid data manipulations.
- Have a good user interface.
- It satisfy the user requirement.

1.3.3 Applicability

This system is expected to be user friendly and will offer easy access to data as well as services such as leave management, e-recruitment, and timely report generation, monitoring worker trainings and task management.

Without an Staff management system, it's a tedious job for the human resource department to keep track of each and every worker and even harder for a project manager to assign tasks to the project team. The management system will be developed to provide information of worker and many other facilities at the click of a button.

Chapter 2

System Analysis

2.1 Existing System

- The records are maintained in registerers.
- Time consuming process.
- Complicated Attendance marking and calculate salary.
- Need of extra manual effort.
- Not very much accurate.
- It used to take lot of time.

2.2 Proposed System

- Helps in maintaining the computerized employees details.
- Calculate the salary.
- Easy attendance marking.
- Add a new worker's profiles in system accordingly.
- Very fast and accurate.
- Doesn't require any extra hardware device.

2.3 Requirement Analysis

The necessities were gathered from approved sources. The necessities for improvement of a site is considered at each progression of the venture. The away from of data has led to an extraordinary help with the general prosperity of the task.

2.4 Hardware Requirements

The selection of hardware is very important in the existence and proper working of any software. When selecting hardware, the size and requirements are also important.

Table 2.1: Hardware Requirements

Processor	Intel CORE i3
RAM	4.0 GB
Hard Disk Drive	512 GB
Graphics Card	Nvidia, AMD

2.5 Software Requirements

Programming Requirements manage characterizing programming asset necessities and essentials that should be introduced on a PC to give ideal working of an application. These necessities or pre-essentials are commonly excluded from the product establishment bundle and should be introduced independently before the product is introduced.

Table 2.2: Software Requirements

Number	Description
1.	Windows 7,8,10
2.	Python Idle 3.7.3
3.	MySQL
4.	Tkinter - GUI

2.6 Justification of Platform

So, I opted Python base management system for the project because of various advantages.

Python integrates the Enterprise Application Integration that makes it easy to develop web services by invoking COM or COBRA components. It has powerful control capabilities as it calls directly through C, C++ or Java via Jython. Python also process XML and other markup languages as it can run on all modern operating systems through same byte code. With its strong process integration features, unit testing framework and enhanced control capabilities contribute towards the increased speed for most applications and productivity of application.

It is a great option for building scalable multi-protocol network applications. Python has varied advantageous feature and programmers prefer this language to other programming languages because it is easy to learn and code too.

The python language is dynamically typed so it has many design restrictions that are reported by some Python developer. It is even seen that it requires more testing time, and the errors show up when the applications are finally run.

Python is a robust programming language and provide an easy usage of the code lines, maintenance can be handled in a great way and debugging can be done easily too. It has gained importance across the globe as computer giant Google has made it one of its official programming languages.

What advantage Python provides to its users. But before that, I hope you all are aware of what features Python provides to us :-

- Extensive Libraries
- Extensible
- Embeddable
- Improved Productivity
- Simple and Easy
- Object – Oriented
- Portable
- Interpreted

Chapter 3

System Design (20 bold, centered)

3.1 Module Division

Admin:

- Staff details – Admin has access to view the staff details.
- View – Admin can view the newly added staff from the database.
- Add – Admin has access to add the staff.
- Register – Admin has access to register the worker
- Salary Type – Admin has access to select the Salary Type.
- Salary Cycle – Admin can also select the salary cycle.

Attendance:

- Attendance - Admin can mark the attendance
- Timing – Admin has access to time manage
- Overtime – Admin can also mark the overtime of the worker.

Salary:

- Total Salary – Admin has access to mark the salary paid or not.
- Salary paid or not – Admin can mark the salary paid or not.
- PF – Admin has access to deduct the PF amount from his Salary.
- Loan – Admin has access to deduct the loan amount from worker's salary if he taken.

3.2 Data Dictionary

So basically, data dictionary is a kind of inventory where all the data elements and also a brief description about database relationship, entities, source, format and usage.

1. Add Staff

Field Name	Data Type
Name	StringVar
Email	StringVar
Mobile	IntVar
Address	StringVar
Dob	IntVar
Age	IntVar
Gender	StringVar
Salary Type	StringVar
Salary Cycle	StringVar

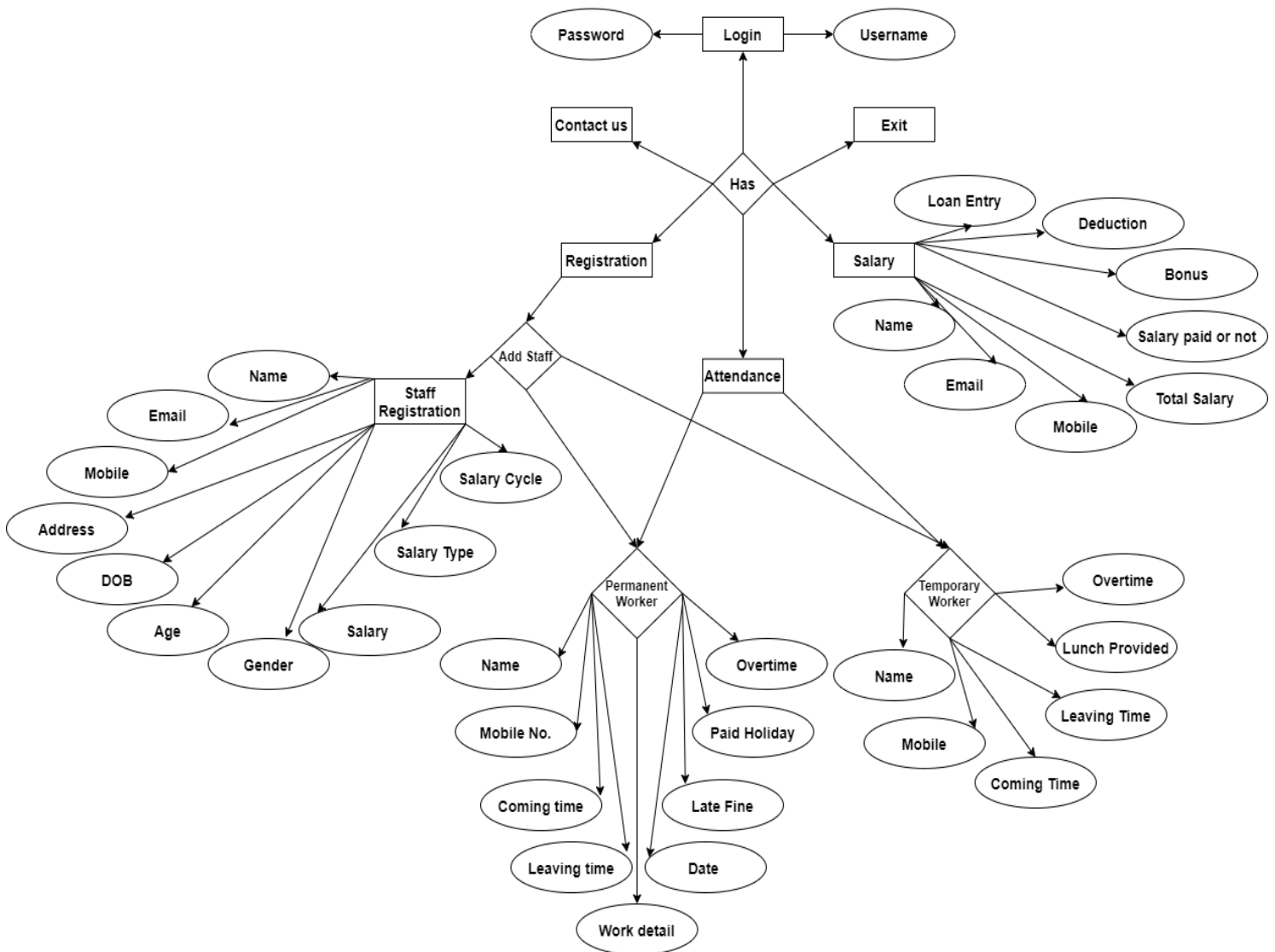
2. Attendance

Permanent Worker	Temporary Worker	Datatype
Name	Name	StringVar
Mobile No	Mobile No	IntVar
Coming Time	Coming Time	IntVar
Leaving Time	Leaving Time	IntVar
Work detail	Lunch Provided	StringVar
Date	Overtime	IntVar
Late fine		IntVar
Paid holiday		IntVar
Overtime		StringVar

3. Salary

Salary	Datatype
Name	StringVar
Mobile	IntVar
Email	StringVar
Total Salary	IntVar
Salary Paid or not	StringVar
Bonus	IntVar
PF,ESI - Deduction	IntVar
Loan Entry	IntVar

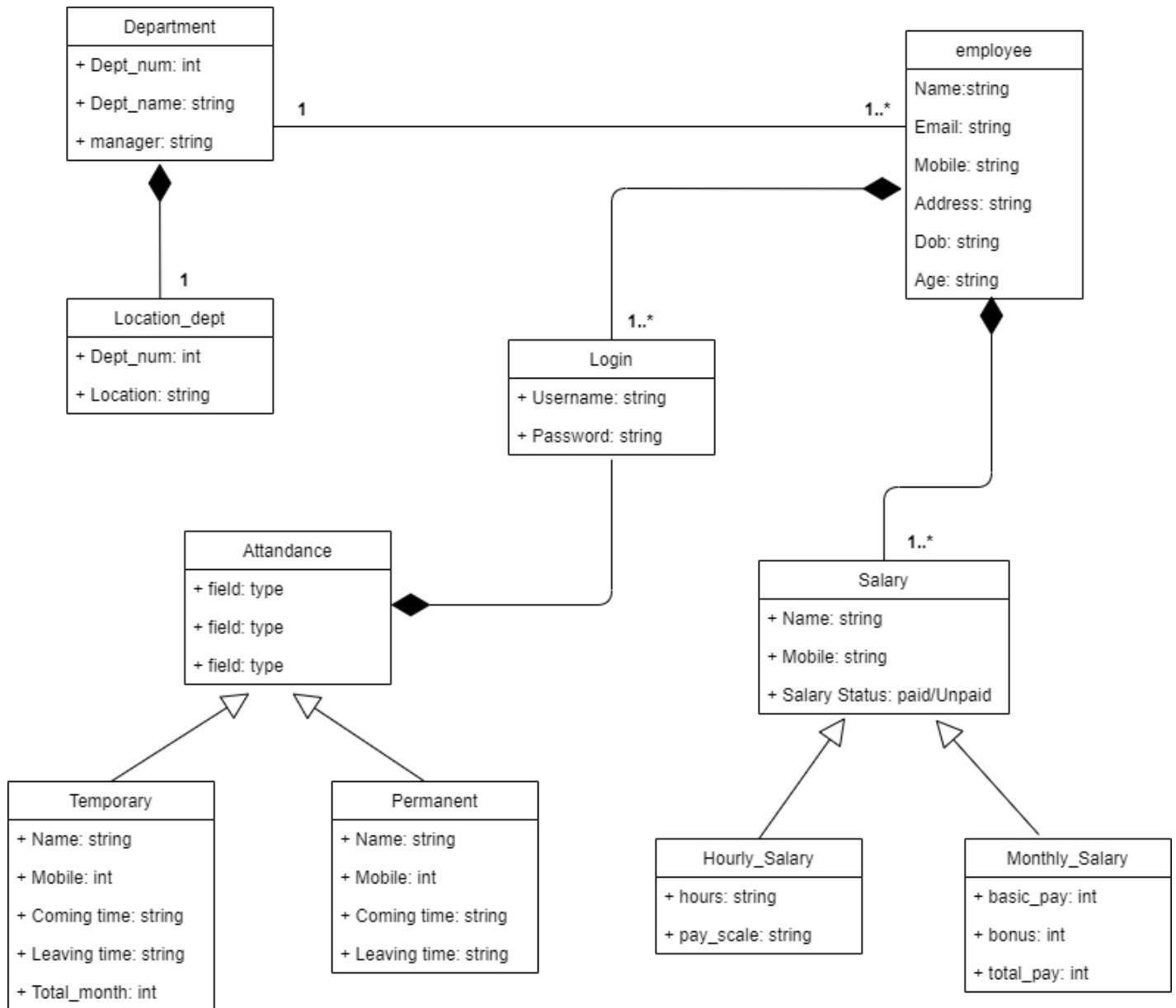
3.3 E-R Diagrams



An Entity-Relationship (ER) Diagram may be a style of flowchart that illustrates how “entities” like people, objects or concepts relate to every other within a system. In the fields of software engineering, business information systems and research the ER Diagrams are mostly used to design or debug relational databases.

ER Diagram uses an outlined set of symbols like rectangles, diamonds, ovals and connecting lines to symbolize the interconnected entities, relationships and also their attributes. They depict grammatical structure, in which entities are nouns and the relationships as verbs.

3.4 Class Diagram(UML):



The purpose of class diagram is to model the static view of an application.

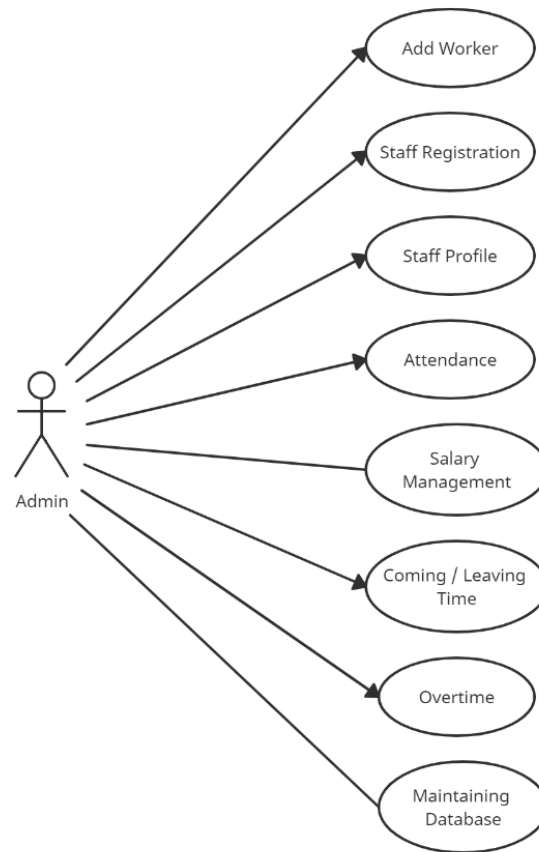
Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction.

UML diagrams like activity diagram, sequence diagram can only give the sequence flow of the application, however class diagram is a bit different. It is the most popular UML diagram in the coder community.

The purpose of the class diagram can be summarized as –

- Analysis and design of the static view of an application.
- Describe responsibilities of a system.
- Base for component and deployment diagrams.
- Forward and reverse engineering.

Use Case Diagram (UML):



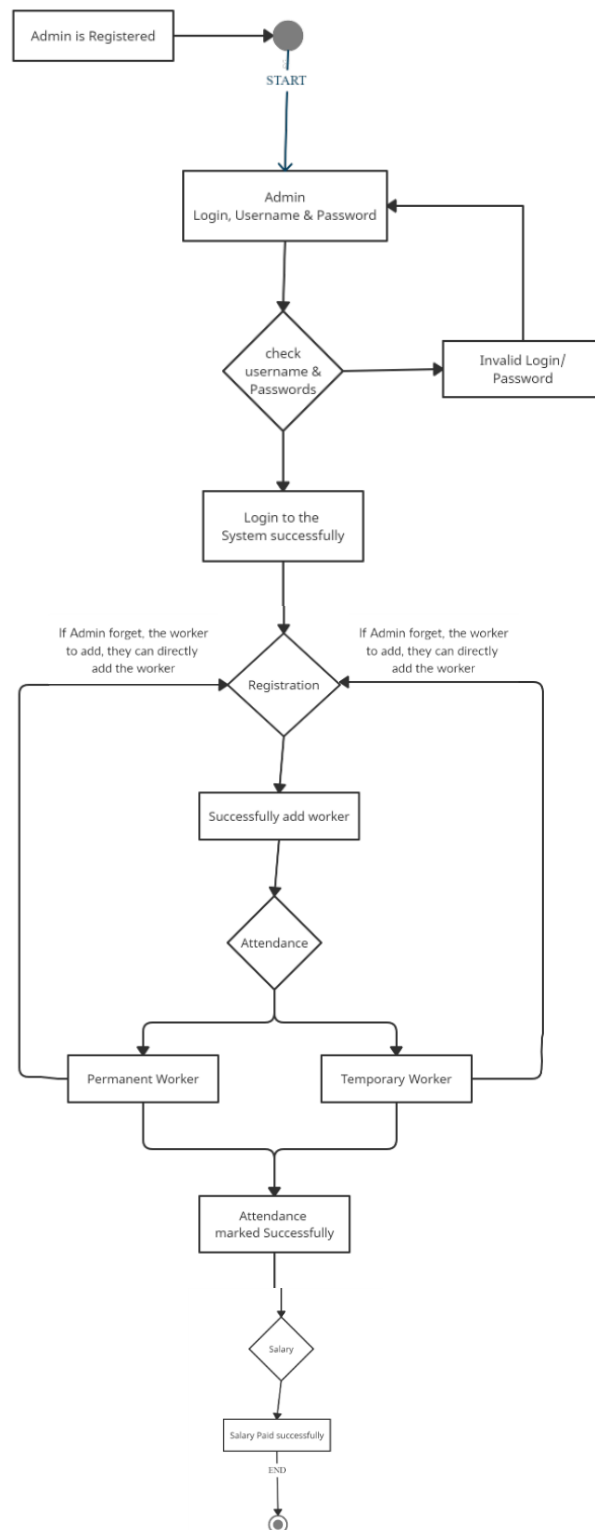
Use case diagram is to capture the dynamic aspect of a system. However, this definition is too generic to describe the purpose, as other four diagrams (activity, sequence, collaboration, and (State chart) also have the same purpose. We will look into some specific purpose, which will distinguish it from other four diagrams.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified.

When the initial task is complete, use case diagrams are modelled to present the outside view
In brief, the purposes of use case diagrams can be said to be as follows –

- Used to gather the requirements of a system.
- Used to get an outside view of a system.
- Identify the external and internal factors influencing the system.
- Show the interaction among the requirements are actors.

Activity Diagram (UML):



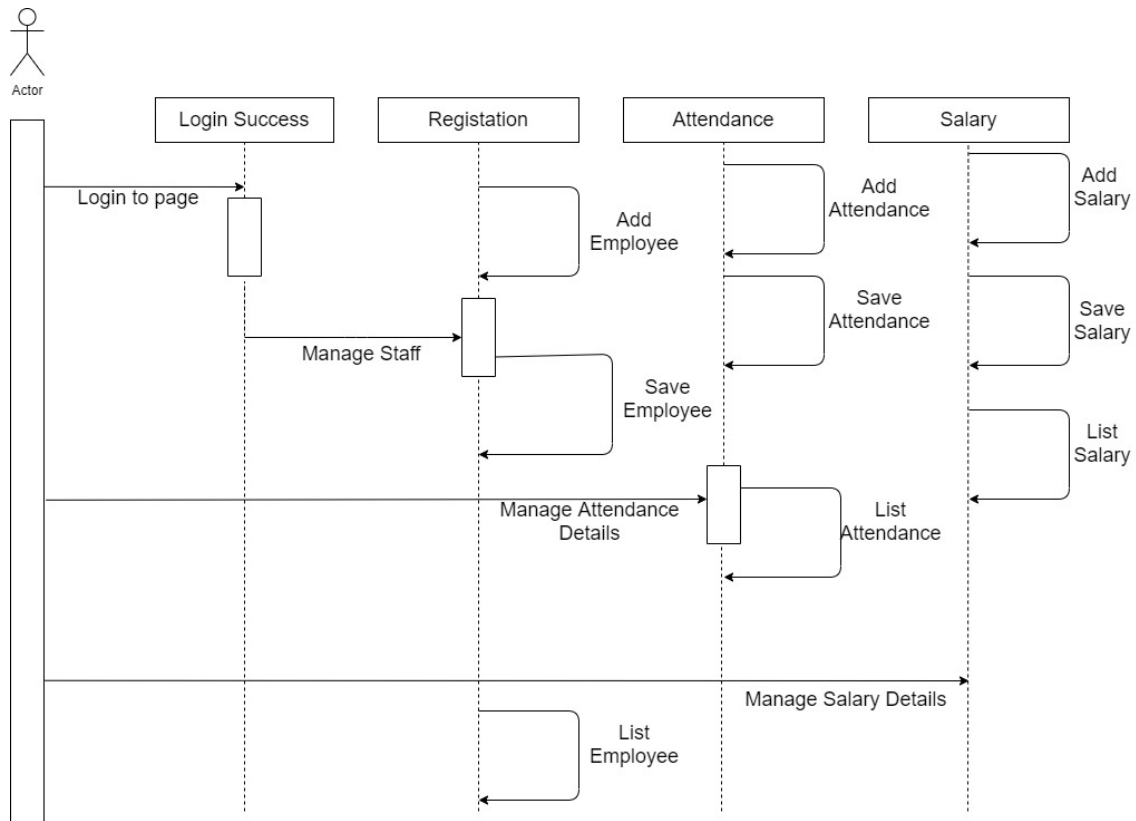
The basic purposes of activity diagrams is similar to other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.

It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not. It shows different flows such as parallel, branched, concurrent, and single.

This is the activity diagram of Staff Management System, which shows the flows of login activity, where admin will be able to login using their username and password After login user can manage all the operations on, Staff, Attendance, Salary. All the pages such as Registration , attendance and Salary are secure and user can access these page after login. The upper diagram helps demonstrate how the logon page works in a staff management system.

Sequence Diagram(UML):



Sequence Diagram is an interaction diagram that shows how processes operate with one another and in what order. It shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and as horizontally arrows, the message exchange between them in the order which they occur. It is used to show interaction between objects in the sequential order that occur. Interaction means sending or receiving messages. So we have designed the following Sequence diagram for recruitment portion:

Purpose of Sequence Diagram

- Model high-level interaction between active objects in a system.
- Model the interaction between object instances within a collaboration that realizes a use case.
- Model the interaction between objects within a collaboration that realizes an operation.
- Either model generic interactions (showing all possible paths through the interaction) or specific instances of a interaction (showing just one path through the interaction)

Chapter 4

Implementation and Testing

4.1 Code (Place Core segments)

```
def login(self):
    if self.name.get()==" " or self.pass_.get()==" ":
        tkinter.messagebox.showerror(title='Error',message='All fields are required!!')
    elif self.name.get()=="a" and self.pass_.get()=="a":
        tkinter.messagebox.showinfo(title='Successfully', message='Welcome '+ self.name.get())
        self.root.destroy()
        HomePage()
    else:
        tkinter.messagebox.showwarning(title='Error', message='invalid credentials !!')
```

Admin Login

```
class HomePage:
    def __init__(self):
        top = Tk()
        top.geometry('1530x790+0+0')
        top.resizable(False,False)
        top.title('STAFF MANAGEMENT SYSTEM')
        top.configure(background='gray')
```

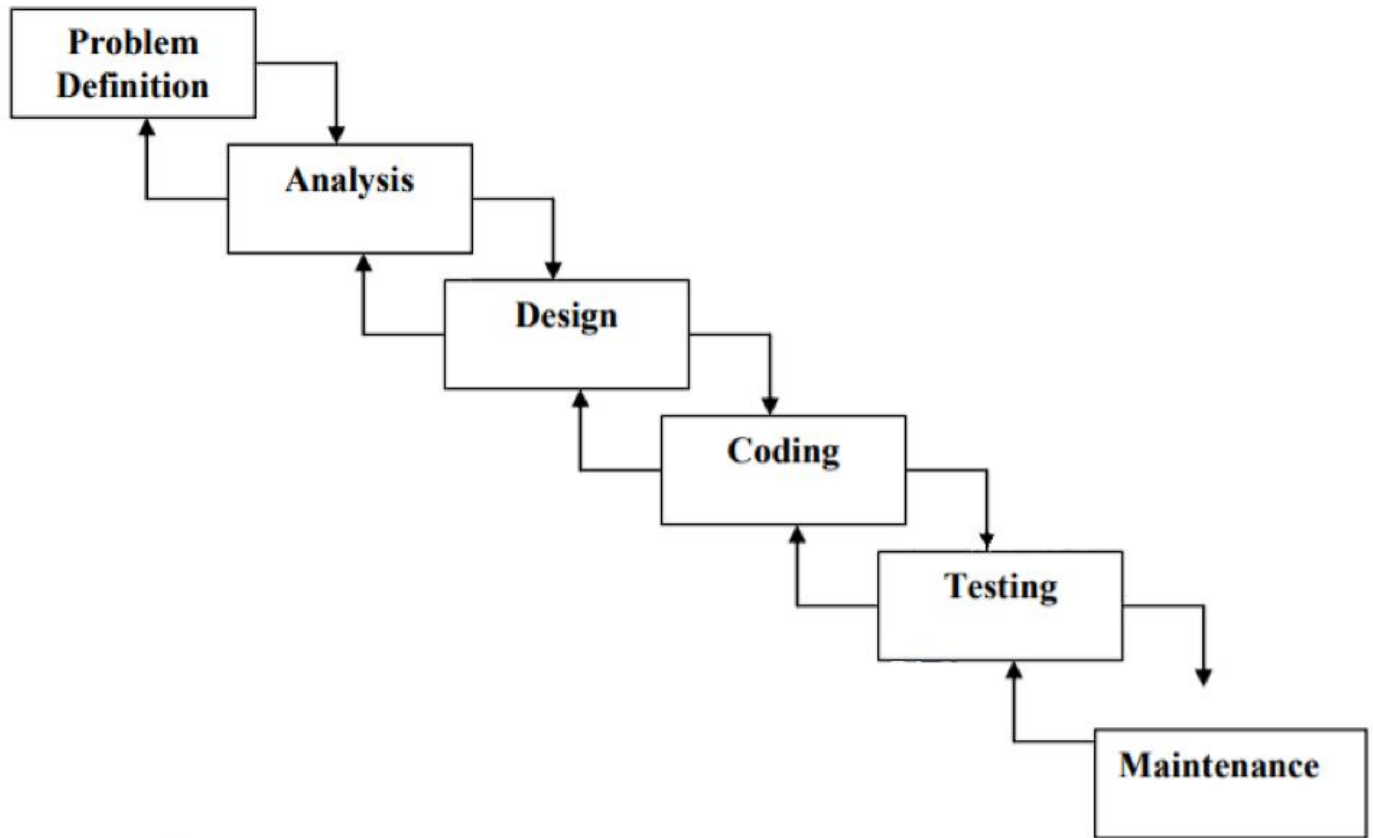
Main Home Page

```
class StaffRegistration:
    def __init__(self):
        self.window = Tk()
        self.window.title('STAFF REGISTRATION')
        self.window.configure(background='black')
        self.window.geometry('900x650+300+135')
        self.window.resizable(False,False)
        #====Initalizing Variables=====
        self.Name = StringVar()#label1
        self.Email = StringVar()#label2
        self.Mobile = IntVar()#label3
        self.Address = StringVar()#label4
        self.Dob = StringVar()#label5
        self.Age= IntVar()#label6
        self.Gender = None
        self.Salary_type = StringVar()#label8
        self.Salary = IntVar()#label9
        self.SalaryCycle = StringVar()#label10
```

Staff registration Page

4.2 Testing Approach

ITERATIVE WATERFALL MODEL



Software development life cycle process specifies a method of developing the software. Each software development projects starts with some needs and ends with some software that satisfies those needs. A software development life cycle specifies the set of activities that should be performed to go from user needs to final products. There are different models of SDLC process and each model specifies the activities and the order in which they should be performed. Depending on the nature of project, a suitable model is chosen and the entire process of software requirement analysis, design, coding, testing and maintenance is preformed accordingly

Various life cycle models are present, but our system is based on **WATER FALL MODEL**, which is most widely used in procedure oriented development. This model attempts to break up the identifiable activities into series of action, each of which must be completed before the next begins. The activities include Problem definition, Requirement Analysis, Design, Coding, Testing and maintenance.

An initial investigation culminates in a proposal that determines whether a system is feasible or not. It determines its workability, impact on the organization, ability to meet user needs, and effective user resources. The objective of feasibility study is not solve the problem but to acquire a sense of its scope. During the study, the problem definition is crystallized and aspects of the problem to be included in the system are determined. Consequently, cost and benefits are estimated with greater accuracy at this stage. This is a bridge in between the User Requirements and the output that he can avail under a set of given constraints, inputs and outputs.

The main steps are:

- Statement of constraints
- Identification of specific system objectives
- Description of outputs

COCOMO model:

COCOMO (Constructive Cost Model) is a regression model based on LOC, i.e number of **Lines of Code**. It is a procedural cost estimate model for software projects and often used as a process of reliably predicting the various parameters associated with making a project such as size, effort, cost, time and quality.

The key parameters which define the quality of any software products, which are also an outcome of the COCOMO are primarily Effort & Schedule:

Effort: Amount of labor that will be required to complete a task. It is measured in person-months units.

Schedule: Simply means the amount of time required for the completion of the job, which is, of course, proportional to the effort put. It is measured in the units of time such as weeks, months.

Different models of COCOMO have been proposed to predict the cost estimation at different levels,

based on the amount of accuracy and correctness required. All of these models can be applied to a variety of projects, whose characteristics determine the value of constant to be used in subsequent calculations. These characteristics pertaining to different system types are mentioned below.

Organic – A software project is said to be an organic type if the team size required is adequately small, the problem is well understood and has been solved in the past and also the team members have a nominal experience regarding the problem.

Semi-detached – A software project is said to be a Semi-detached type if the vital characteristics such as team-size, experience, knowledge of the various programming environment lie in between that of organic and Embedded. The projects classified as Semi-Detached are comparatively less familiar and difficult to develop compared to the organic ones and require more experience and better guidance and creativity. Eg: Compilers or different Embedded Systems can be considered of Semi-Detached type.

Embedded – A software project with requiring the highest level of complexity, creativity, and experience requirement fall under this category. Such software requires a larger team size than the other two models and also the developers need to be sufficiently experienced and creative to develop such complex models. This project is an organic system.

COCOMO consist of a hierarchy of three increasingly detailed and accurate forms. Any of the three forms can be adopted according to the requirement. The types of COCOMO models are as follows:

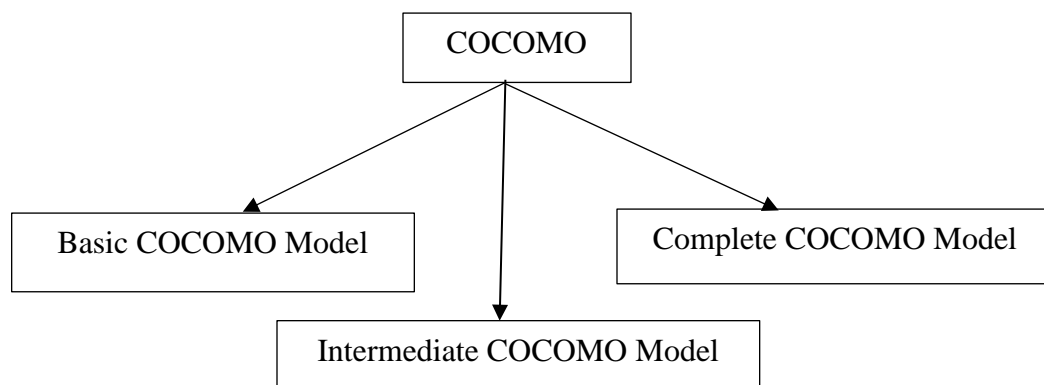


Figure: Types of COCOMO Model

The first level, **Basic COCOMO** can be used for quick and slightly rough calculations of Software Costs. Its accuracy is somewhat restricted due to the absence of sufficient factor considerations. This Project uses basic COCOMO to estimate the cost.

Estimation of Effort Calculations:

Basic Model: $E = a(KLOC)^b$

The above formula is used for the cost estimation of for the basic COCOMO model, and also is used in the subsequent models. The constant values a, b, c and d for the Basic Model for the different categories of system:

SOFTWARE PROJECTS	A	B
Organic	2.4	1.05
Semi Detached	3.0	1.12
Embedded	3.6	1.20

Table: Software Project System Type

Project	SLOC%	SLOC	Comments	Blank Lines	Total
All Source	100.00%	412	31	90	510

Now applying COCOMO Model formula:

$$E = a (KLOC)^b$$

$$= 2.4 (0.510)^{1.05}$$

$$= 2.4 (0.4931)$$

$$= 1.1834$$

Which is approximately 1 months.

$$\text{Development time} = c (\text{Effort Applied})^d$$

$$= 2.5 (1.1834)^{0.38}$$

$$= 2.5 (1.0661)$$

$$= 2.6652$$

Which is approximately 3 months.

$$\text{Productivity} = KLOC / \text{Effort}$$

$$= 0.510 / 1$$

$$= 0.51$$

4.2.1 Unit Testing

In Unit testing the structure of the program is not considered. Test cases are solely determined on the basis of requirement or specification of the program or module. Internals of the modules and the programs are not considered for selection of test cases. Due to this nature it is also called "black box testing". The most obvious functional testing procedure is "exhaustive testing", which is impractical. The other criteria for generating the test case are to generate them "randomly" this strategy has a little chance of resulting in a test case that is close to optimal.

The techniques are:

- Black Box Testing - Using which the program, input and output are tested.
- Exhaustive Box Testing - All possible data combinations are used and Exploratory testing includes implicit data combinations present in the state of the software at the start of testing
- Glass Box Testing - Testing technique that examines the program structure and derives test data from the program logic/code..

4.2.2 Integration Testing

Incremental testing is a way of Integration testing. In this type of testing method, you first test module of the software individually and then continue testing by appending other modules to it then another and so on.

Incremental integration is the contrast to the big bang approach. The program is constructed and tested in small segments, where errors are easier to isolate and correct. Interfaces are more likely to be tested completely, and a systematic test approach may be applied.

There are three types of Integration testing

- Top down testing
- Bottom-up testing
- Big – Bang testing

Top down Integration Testing –

Top-down integration testing is the technique where the utilization of behavior is stimulated for the lower-level modules that do not seem to be yet or not been integrated. during this integration testing, testing takes place from top to bottom. First, high-level modules are tested then low-level modules and at last integrating the low-level modules to a high level to confirm the system is functioning as intended.

Bottom-Up Integration Testing –

Bottom-up integration begins construction and testing with modules at the lowest level in the program structure. In this process, the modules are integrated from the bottom to the top. In this testing processing required for the modules subordinate to a given level is always available and the need for the stubs is eliminated.

As integration moves upward, the need for separate test drivers lessons. In fact, if the top two levels of program structure are integrated top-down, the number of drivers can be reduced substantially, and integration of clusters is greatly simplified. Integration follows the pattern illustrated below. As integration moves upward, the need for separate test drivers lessons.

Big-Bang Integration Testing –

Top-down integration testing is the technique where the utilization of behavior is stimulated for the lower-level modules that do not seem to be yet or not been integrated. during this integration testing, testing takes place from top to bottom. First, high-level modules are tested then low-level modules and at last integrating the low-level modules to a high level to confirm the system is functioning as intended.

Test Cases:

Login -

Test Case	Test Purpose	Test Condition	Expected Outcome	Actual Result
Login	Check username And Password	If user details are not correct, display error message	Grant Access to the applicable main system	User successfully logs into the system upon submission of correct login credentials.

In this testing, it basically use to test the functionality of the system to check whether it working correctly or not. However describes that a test case is “an input, action, or event and an expected response to determine if a feature of a software application is working correctly” A test case might consist of test case name, Test purpose, Test Condition, Expected Outcome and Actual Result.

User Acceptance Testing:

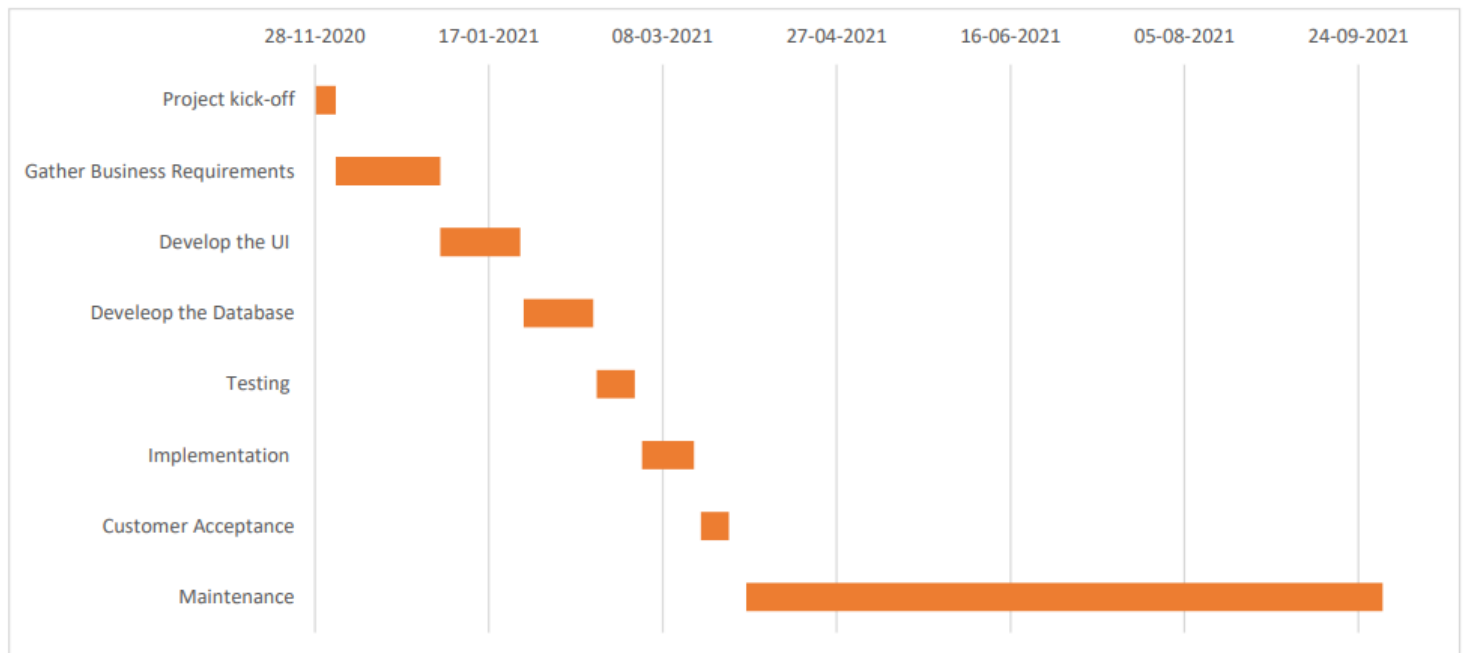
User acceptance testing is a very important testing because it can determine whether the system developed is success or not. User such as manager or the owner of the Business manager’s is important people to test the system because they are the end user. Therefore, their feedback and comments for the developed system are the most important factor deciding the success of the system. Users will be asked to use the system to perform the task and they will validate the GUI based on their first time experience using the system.

Gantt Chart :

Staff Management System

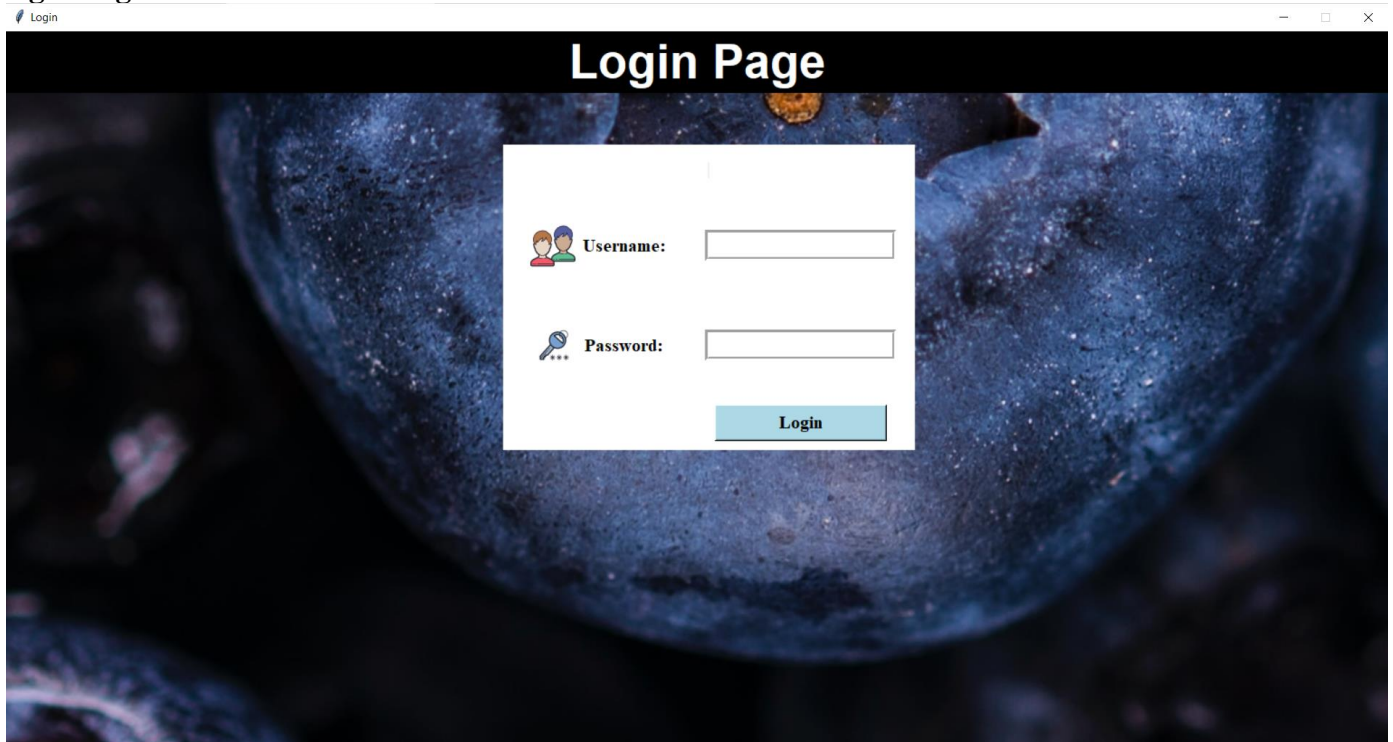
Project Lead - Yash Panchal

<u>Task</u>	<u>Start Date</u>	<u>End Date</u>	<u>Duration</u>
Project kick-off	28-11-2020	04-12-2020	6
Gather Business Requirements	04-12-2020	03-01-2021	30
Develop the UI	03-01-2021	26-01-2021	23
Develeop the Database	27-01-2021	16-02-2021	20
Testing	17-02-2021	28-02-2021	11
Implementation	02-03-2021	17-03-2021	15
Customer Acceptance	19-03-2021	27-03-2021	8
Maintenance	01-04-2021	01-10-2021	183



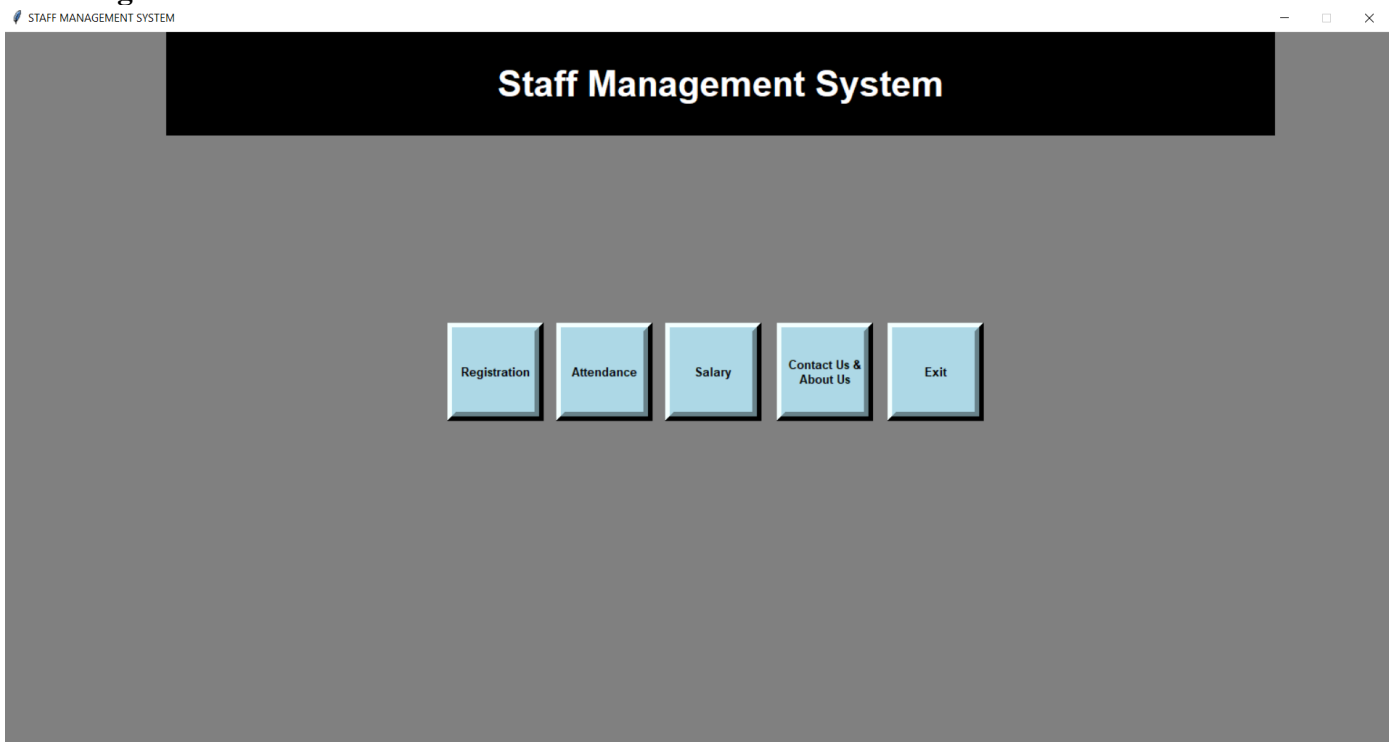
Chapter 5

Login Page :



The screenshot shows a web browser window titled "Login". The page has a black header with the text "Login Page" in white. The background is a dark, textured image of a blue, rocky surface. In the center, there is a white login form. The form contains two input fields: "Username:" with a small icon of two people to its left, and "Password:" with a small icon of a key to its left. Below the password field is a blue button labeled "Login".

Home Page :



The screenshot shows a web browser window titled "STAFF MANAGEMENT SYSTEM". The page has a black header with the text "Staff Management System" in white. The background is a solid gray. In the center, there are five blue buttons arranged horizontally: "Registration", "Attendance", "Salary", "Contact Us & About Us", and "Exit".

Add Staff :

STAFF REGISTRATION

+ADD STAFF

NAME:	<input type="text"/>
EMAIL:	<input type="text"/>
MOBILE:	<input type="text"/>
ADDRESS:	<input type="text"/>
DOB(DD-MM-YYY):	<input type="text"/>
AGE:	<input type="text"/>
GENDER:	<input type="text" value="Male"/> <input type="text" value="Female"/> <input type="text" value="Other"/>
SALARY TYPE:	<input type="text"/>
SALARY:	<input type="text"/>
SALARY CYCLE:	<input type="text"/>

Attendance Page :

ATTENDANCE

WORKER ATTENDANCE PAGE

<input type="button" value="PERMENT
WORKER"/>	<input type="button" value="TEMPORARY
WORKER"/>
<input type="button" value="BACK"/>	

Permanent Worker Attendance :



PERMENT WORKER ATTENDANCE



NAME:	<input type="text"/>	MOBILE NO:	<input type="text"/>
COMING TIME:	<input type="text"/>	LEAVING TIME:	<input type="text"/>
WORK DETAIL:	<input type="text"/>	DATE:	<input type="text"/>
		LATE FINE:	<input type="text"/>
		PAID HOLIDAY:	<input type="text"/>
OVERTIME	<input type="text"/>		
SUBMIT		BACK	STAFF REGISTRATION

Temporary Worker Attendance :



TEMPORARY WORKER ATTENDANCE



NAME:	<input type="text"/>	MOBILE NO:	<input type="text"/>
COMMING TIME:	<input type="text"/>	LEAVING TIME:	<input type="text"/>
LUNCH PROVIDED:	Yes No		
OVERTIME:	<input type="text"/>		
SUBMIT		BACK	STAFF REGISTRATION

Salary :

Salary Page

Worker's Salary

NAME:	<input type="text"/>
EMAIL:	<input type="text"/>
MOBILE:	<input type="text"/>
TOTAL SALARY:	<input type="text"/>
SALARY PAID OR NOT:	<input type="button" value="Paid"/> <input type="button" value="Not"/>
BONUS:	<input type="text"/>
DEDUCTION (PF,ESI,etc.):	<input type="text"/>
LOAN ENTRY:	<input type="text"/>

Contact us :

ABOUT US & CONTACT US

YASH's STAFF MANAGEMENT

Created by: Yash Panchal

Yash Panchal
Contact Number: 9004168049
Email: panchalyash089@gmail.com

Chapter 6

Conclusion and Future Work

This project built keeping in mind that it is to be used by only one user that is the admin. It is built for use in small scale organization where the number of employees is limited. According to the requested requirement the admin can add worker, mark attendance and pay the salary all worker data in his organization. The admin can also mark permanent worker's attendance and Temporary worker's attendance them. The required records can be easily viewed by the admin anytime he wants in an instant. The payment of the worker's is based on monthly basis, Per hour basis, Daily basis, Work basis and Weekly basis.

Numerous validations implemented would enable the admin to enter accurate data. The main objective of this framework is to save time, make the system cost effective and management records efficiently. Some additional stuff could be implemented and integrated into the application code making it more reliable and flexible; especially what concerns apay-roll module, for instance.

Apparently, the role of such systems is basic and essential with each company that wants to keep a really good control and record concerning its personal data, functionality and performance on all levels in its structure. Every organization, in now days has the necessity of managing its staff on a really good level as the staff has definitely the greatest merit of building up a company as such as it is.

The well managed worker means giving the appropriate financial award-ness and all kind of benefits as such as they have been deserved.

Future scope of the work:-

- The option to print the records in future.
- I intend to add a leave structure in the future.
- I would like to implement a regular backup mechanism to back up the worker's database to avoid disasters.
- The system can be developed in such a way that its existing features can be modified to better versions.

Chapter 7

References

- [1] Learn python the Hard way by Zed A/Shaw (3rd Edition).
- [2] Python programming by John Zelle (3rd Edition), published in 2004.
- [3] Python Cookbook by David Beazley and Brian K. Jones
- [4] Python in Easy Steps by Mike McGrath.
- [5] www.geeksforgeeks.com
- [6] www.stackoverflow.com
- [7] www.learnpython.org
- [8] www.realpython.com
- [9] www.tutorialspoint.com
- [10] www.programiz.com
- [11] <http://www.slideshare.com>
- [12] www.doc.python.org

Chapter 8

Glossary

ADMINISTRATIVE SERVICES ONLY(ASO):

The hiring of a firm to handle certain administrative tasks. The firm does not assume any risk but merely carries out the specialized functions that the staff cannot or does not want to do . For example an staff funds its own dental insurance claim payments but pays the ASO firm to process the claims.

ACCESSIBILITY:

The extent to which a contractor's or worker's facility is readily approachable and does not inhibit the mobility of individuals with disabilities, particularly such areas as the personal office, worksite and public areas.

AGILE ORGANIZATION:

Also known as agile manufacturing, this is a term applied to an organization that has created the processes, tools an training to enable it to respond quickly to customer needs and market changes while still controlling costs and quality.

ALTERNATE DISPUTE RESOLUION(ADR):

An informal process to resolve disputes. Involved parties meet with a trained third party who assists in resolving the problem by arbitration, mediation, judicial settlement conferences, conciliation or other methods. Though usually voluntary, ADR is sometimes mandated by a judge as a first step before going to court.

BYOD(bring your own device):

A term used to describe the growing trend of worker's – owned devices within a business such as smartphones tablets, laptop and other devices. Many worker's have policies that govern the use of worker-owned devices in the workplace.

GENERAL AGENTS:

General agents are in middleman for carriers and brokers and usually focus on the 250 staff market. Usually an individual appointed by a life or health insurer to administer its business in a given territory. GAs are important for companies who sell to small workers or brokers e.g., benefits administration software providers.

STAFF ENGAGEMENT:

Staff engagement, also called worker engagement, is a business management concept. An “engaged staff” is one who is fully involved in, and enthusiastic about their work, and thus will act in a way that further their organization’s interests.

STAFF RELATIONS:

Developing, maintaining and improving the relationship between staff and boss by effectively and proactively communicating with workers, processing/disputes, etc.

STAFFING:

A method of finding , evaluating and establishing a working relationship with future workers. They may be current worker or future worker

PERFORMANCE MANAGEMENT:

The process of maintaining or improving worker job performance through the use of performance assessment tools, coaching and counseling. The ultimate goal is to better meet organizational objectives.

QUALITY MANAGEMENT:

A system to make sure that a product or service meets standards of excellence and that the process by which the product or service is created is efficient and effective as well. The three key components of this system are quality control, quality assurance and improvement.

UNION:

Workers who organize a united group, usually related to the kind of work they do, to collectively bargain for better work conditions pay or benefit increases, etc.