SOFTWARE ENGINEERING PROJECT REPORT

PROJECT NAME: AUTOMATIC EMPLOYEE APPRAISAL SYSTEM

TEAM BLITZ

NAME	SRN
BHARGAV M V	PES1UG20CS660
MAHESH SHRIPAD BHAT	PES1UG20CS661
KIRAN H KADEMANI	PES1UG20CS654
CHANNABASAV DANARADDI	PES1UG20CS637

Background:

This project is developed for large companies to make the process of appraisal of employees working in the company automatic, intuitive and easy, and notifying employees on being approved by both the manager and HR for Hike.

Key success factors of the project are adaptation to changes in the organization, promoting healthy work competition, and automatic appraisal.

Functions of the project:

It must be able to record responses of the employees for their appraisal

It must be able to allow managers to view the responses of the employees working under them

It must be able to allow managers to record their evaluation of the employees under him/her and select employees based on criteria set by the manager.

It must be able to allow the HR of the company to see the evaluation of the employee by the respective manager and approve or reject accordingly.

It must be able to allow managers to schedule meetings with the employee, and notify the employee when he is approved by the manager.

Working of the Project:

This system works in website form, which the employee, manager or the HR can login depending on their ID, and the dashboards are displayed respectively.

When an employee logs in, the website displays a form, which the employee can fill.

When the manager logs in, the website displays a form for each employee working under him, and he can fill respectively, and also view employee responses. He can then select a criteria according to which, employees will be selected and approved.

When the HR logs in, the website displays the manager evaluation of the employee, and can approve of reject accordingly

The notification of approval of the employees by the manager or the HR is sent through mail.

ADVANTAGES OF THIS SOFTWARE:

- Keeps track of all employees and managers working in the company, and simplifies their appraisal process.
- Automates appraisal of employees
- Very user friendly, simple, and efficient

System Requirements

Hardware Requirement

- macOS Sierra and above (If Mac setup is required) or
- Windows 7 or higher
- I3 processor system or higher
- 8 GB RAM or higher
- 100 GB ROM or higher

Software Requirement

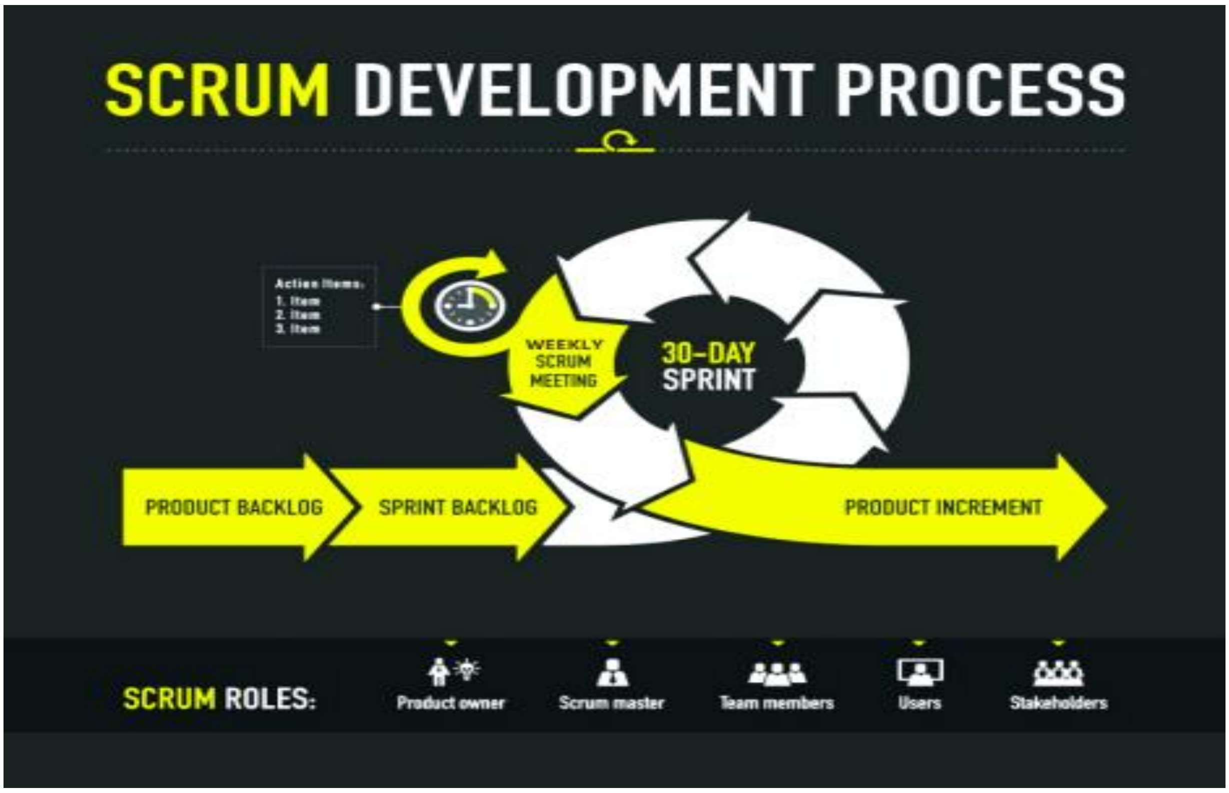
- Gmail access for notification
- SQL database access
- Streamlit web services module

a)

Agile scrum is the agile methodology used for the development of this project

Scrum is a framework that helps teams work together. Much like a rugby team (where it gets its name) training for the big game, scrum encourages teams to learn through experiences, self-organize while working on a problem, and reflect on their wins and losses to continuously improve.

1. SOFTWARE DEVELOPMENT LIFE CYCLE FOLLOWED



<u>How we follow the SCRUM model :</u>

● We gather the user experience from the user in the form of feedback. In our case, we carry out peer review to identify the direction in which our project is headed.

● We will have a weekly scrum meeting to ensure about the builds and fixes of user issues.

● We discuss what new things can be implemented to make our software stand out.

<u>Why SCRUM ?</u>

● Scrum is used when frequent changes in the software are required.

● We gathered the suggestions from our peers, and made frequent changes to better out software

● managing tasks in an organized way and have the planning necessary to reach them makes it a crucial tool to save time and money.

b)

SRS:

1. Introduction

1.1 Purpose

This SRS describes the software functional and non functional requirements for release 1.0 of the Employee Appraisal System. This document is intended to be used by the members of the project team that will implement and verify the correct functioning of the system. Unless otherwise noted, all requirements specified here are high priority and committed for release 1.0.

1.2 Intended Audience

This document is intended to be read by project manager, developers,. The rest of the SRS document contains the overall description, external interface, functional, non-functional and other requirements of the proposed project.

1.3 Product Scope

The Employee Appraisal System aims at automating the employee appraisal process in an organization with minimum effort from the employees. The section in this document titled "Product Functions" lists the features that are scheduled for full or partial implementation in this release.

Overall Description

Product Perspective

The proposed Employee Appraisal System aims at automating the employee appraisal process in an organization with minimum effort from the employees. The proposed system is designed to overcome the lieu pools and enhance the system on the basis of work performance automatically monitored.

All employees get response form, using which they can update their achievements which they have accomplished over a cycle.

Manager is provided with a user interface to view all the employees working under him using which he can see their name, email ID and other details. He is also provided with a capability to award the rating according to his progress tracking of the project.

HR is provided with a user interface for to see the rating by manager. HR appropriately allots the hike and selects the employees who he thinks deserves for the best for promotion to critical positions. This list is sent to his superiors and the employees selected by him are notified.

Product Functions

- Sending status forms made by manager for employees to fill, and collecting them for HR evaluation.
- Sending the evaluation made by the Managers to the HR.
- Pre-evaluation of the employees and managers based on the forms collected in the system and shortlisting of the employee list to determine which employees deserve a raise or promotion.
- Notification of the employees who are selected for a raise or promotion by the HR.

User Classes and Characteristics

Employee: All employees are required to fill GUI froms, by logging into the system, using which they can update their achievements which they have accomplished over a cycle. 6-7 months later, the responses are collected by the system and sent to HR.

Manager: Manager is provided with a user interface to view all the employees working under him using which he can see their name, email ID and other details. He is also provided with a capability to award the rating according to his progress tracking of the project.

HR: HR is provided with a user interface for to rating by manager. HR appropriately allots the hike and selects the employee who he thinks deserves for the best for promotion to critical positions. This list is sent to his superiors and the employees selected by him are notified.

Operating Environment

OE-1: The Employee Appraisal System shall operate with the following Web browsers: Chrome, Firefox and Microsoft Edge latest versions.
OE-2: The Employee Appraisal System shall operate on a server running the current corporate approved versions of Red Hat Linux and Apache Web Server.
OE-3: The Employee Appraisal System shall permit the Manager, Employees and HR to access from the corporate Intranet and access the goal sheet and forms. The responses to these forms submitted by the employees 6-7 months later after the form creation by the manager, are collected by the system and sent to HR.

Design and Implementation Constraints

CO-1: The system's design, code, and maintenance documentation shall conform to the Process Impact Intranet Development Standard, Version 1.3 [2].
CO-2: The system shall use the current corporate standard SQL database engine.
CO-3: All UI for managers and HR will be written in Python using gui libraries like Streamlit
CO-4: All the forms shall be shared by email to the employees and all the forms will be google forms.

2.6 Assumptions and Dependencies

AS-1: The Managers Track the progress of the project they are in charge of ,and track the employee performance of all the employees under them.
DE-1: The operations of the system depends on the updation of employees, managers and their positions
DE-2: The operations of the employee appraisal system depends on the opinions of the managers on the employees who work under him.

System Features

1. Sending status forms made by manager for employees to fill, and collecting them for HR evaluation.

5.1.1 Description and Priority

Every 6-7 months, the manager makes a form for the project he is responsible for, detailing the objectives that must be achieved by his team in the next 6 months, and The system sends it to every employee through mail. After 6-7 months, the system notifies all employees to fill the form , and collects all the forms from the employees and also informs the manager about the employees who haven't submitted the form.

Priority=High

5.1.2 Stimulus/Response Sequences

Stimulus: The manager makes a form and sends it to the system to send it to every employee under him
Response: The system shows the forms to every employee under the manager, and after a set time frame by the manager collects the form responses from all the employees and sends them to HR. It also notifies the manager and HR about the employees who haven't submitted the form.

2. Sending the evaluation made by the Managers to the HR.

5.1.1 Description and Priority

Managers evaluate their employees based on their progress tracking and, according to that, rates their employees based on their work, and their conduct in his/her UI interface, which lists all the employees working under him/ her. This is then sent to the system which then sends to the HR.

Priority=High

5.1.2 Stimulus/Response Sequences

Stimulus: The Manager rates his employees based on his opinions and his reports, and sends them to the system
Response: The system sends these reports to the HR, some preprocessing may be done to shortlist this report based on the ratings given by the manager, by the system

3. Pre-evaluation of the employees and managers based on the forms collected in the system and shortlisting of the employee list to determine which employees deserve a raise or promotion.

5.1.1 Description and Priority

The system shortlists the employee list sent by the Manager, based on some criteria set by the HR and sends the list to HR. HR evaluates this list along with the forms submitted by the employees and determines which employees deserve a raise.

Priority=High

5.1.2 Stimulus/Response Sequences

Stimulus: The Manager sends his report of the employees to the system
Response: The system sends these reports to the HR, some preprocessing is done to shortlist this report based on the ratings given by the manager, and the criteria set by the HR, by the system, and the shortlisted list is sent to the HR for further evaluation.

4. Notification of the employees who are selected for a raise or promotion by the HR.

5.1.1 Description and Priority

After HR evaluation, the list containing the names and IDs of employees who deserve a raise are sent to the superiors through mail, and the employees and the managers they work under are sent a notification by the system , informing them that hey are under the view of the HR for a raise.

Priority=High

5.1.2 Stimulus/Response Sequences

Stimulus: HR selects the employees who deserve a raise according to his evaluation, and sends this list to the system.

Response: The system sends a notification to the employees and the managers they work under, that they are selected by the HR for a raise and the list is sent to HR's superiors

5. **4.External interface Requirements :**
6. 4.1 User interfaces :
7. services : performance appraisal system is made up of components and services like command prompt,email address.
8.
9. Our system includes multiple employees,HR s, managers to rate and review the employees and the final decision is taken by the HR's after considering the reviews from managers.
10.
11. After logging in, the employees ,managers and HR will be redirected to different page, employees and managers will be asked to fill different set of forms ,while HR's are permitted to view both the forms .
12.
13. 4.2 Hardware Interfaces
14. Networks:
15. Internet, which is the global network used for communication among employees, HR's and the managers.
16. 4.3 Software Interfaces
17. 1)This will be written in Python
18. 2) HR-client software shall be written in Python GUI using Streamlit.
19. 3) Employees and managers shall be given url for google forms after logging in.
20. 4) Using python GUI ,there's a separate window which would be displayed for employees, managers and HR's based on their login credentials . Then for filling the form both employees and managers are given the url, this form is later viewed by HR's. SQL Server offers us the flexibility and scalability necessary to handle huge jobs without performance restrictions. This means a secure, stable database that protects information. We use SQL to store all the details from the form and only HR's are given permission to view.
21. 4.4 Communications Interfaces:
22. Communication is necessary in this system so communication tool s are requires including e-mail, web browser, network server communications protocols,GUI forms, and so on.

23. **5. Other Nonfunctional Requirements**
24. 5.1 Performance Requirements
25. PE-1: The system shall accommodate all employees,managers and HR managers.
26. PE-2: The forms filled by employees and managers are non visible to each other except the HR manager
27. PE-3: The system shall send mails to all the employees and managers regarding the current status of application.
28. 5.2 Safety Requirements
29. No safety requirements have been identified.
30. 5.3 Security Requirements
31. Only the authorized person will be able to access this system. Person will be able to see only his data. Only Admin is authorized to see other employees data on the basis of settings in security system of HR Manager.
32. 5.4 Software Quality Attributes
33. Availability-1: The Appraisal System shall be available to the employees for certain duration of time, then the form is available to the manager for certain interval of time for filling it. The form from them is later viewed by HR managers for confirmation.

user stories created:

As the HR of the company using this software, I want to make my process of evaluation and approval of employees working in the company automatic and easier

As the HR of the company using this software, I want to view the manager evaluation of the employees and evaluate accordingly.

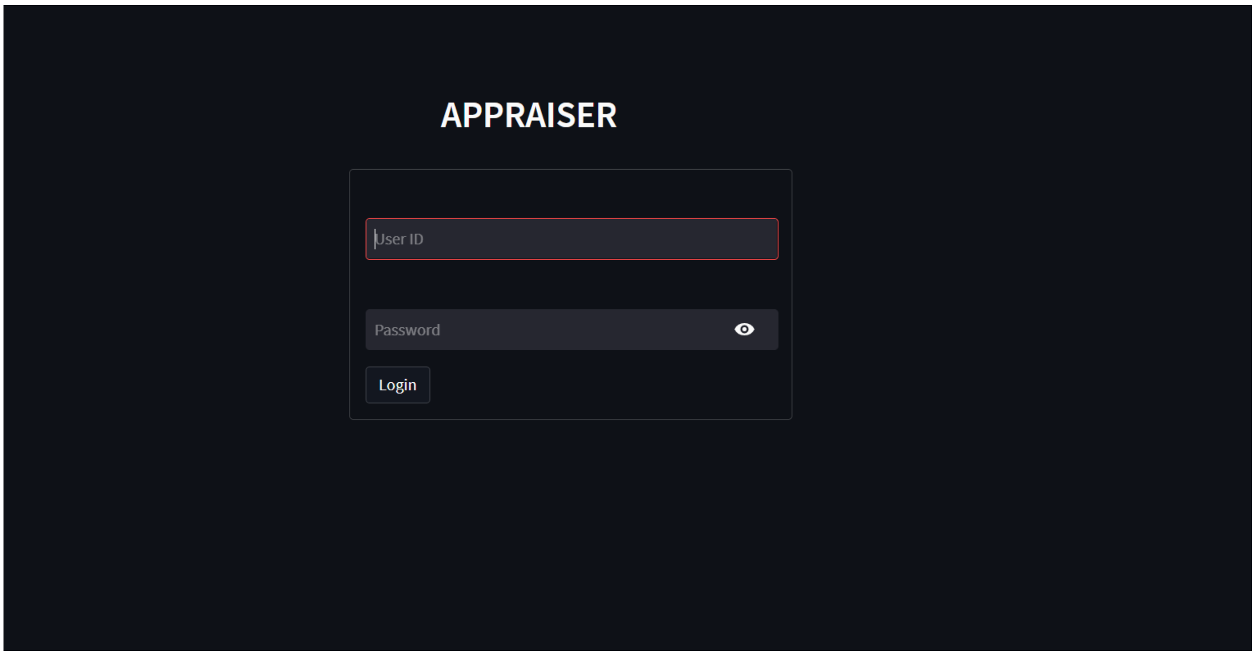
As one of the many managers working in this company I want the process of evaluation and approval of employees under me made automatic and easier

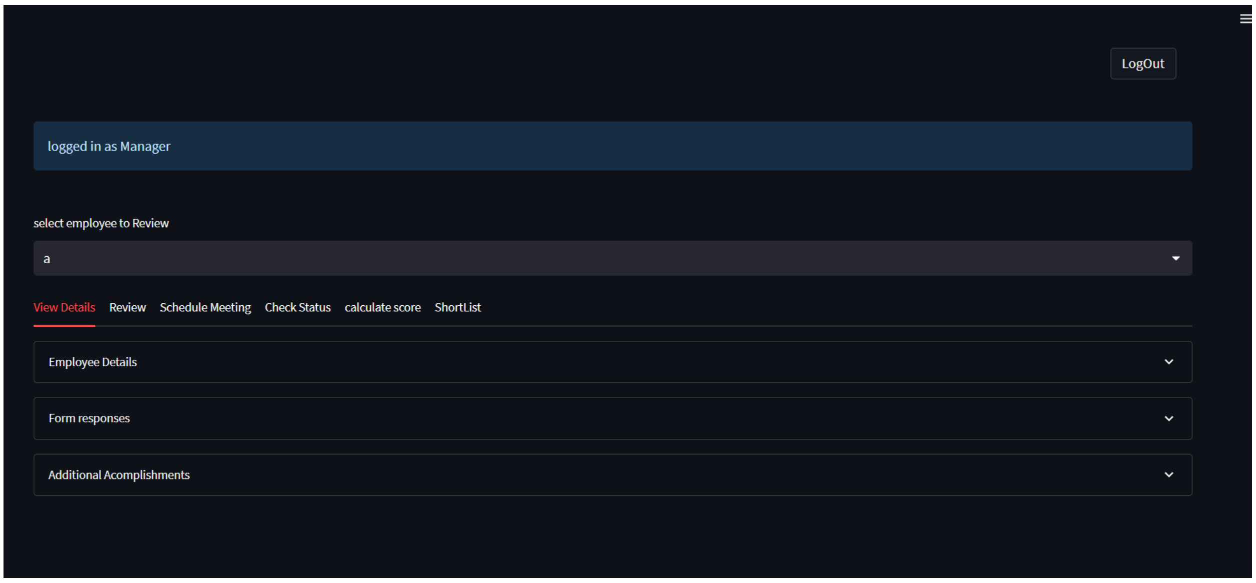
As one of the many employees working in this company, I want to record my responses and get evaluated for my work accordingly, and get notified regarding my appraisal.

As one of the managers of this company, I want to get notified when an employee under me is approved by the HR.

c)

Prototype built: Sample UI for manager and HR and login page done before the actual start of the project

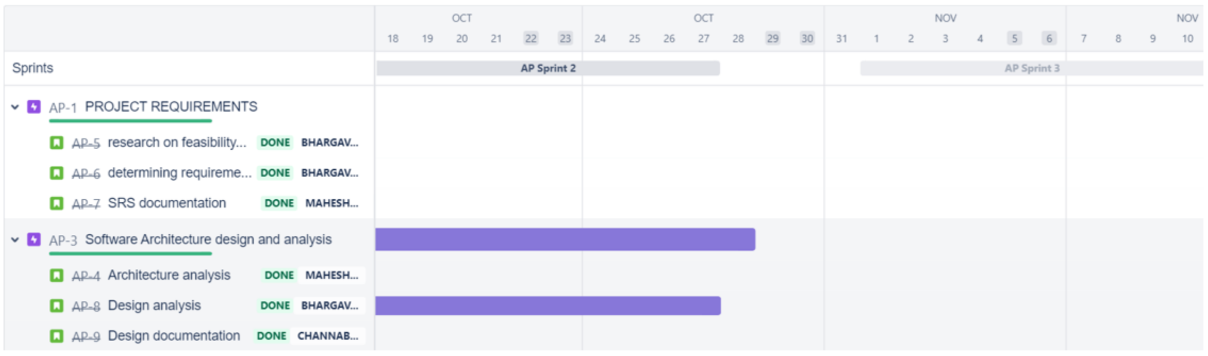




d)

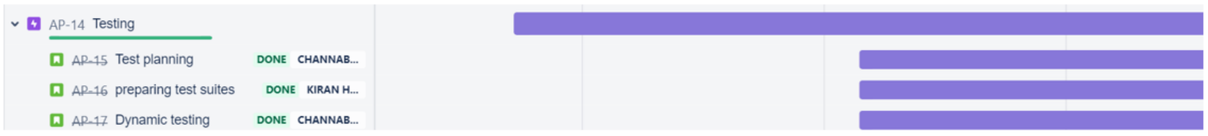
Breaking work into smaller tasks is a common productivity technique used to make the work more manageable and approachable. For projects, the Work Breakdown Structure (WBS) is the tool that utilizes this technique and is one of the most important project management documents. It singlehandedly integrates scope, cost and schedule baselines ensuring that project plans are in alignment.

WBS using GANTT chart in JIIRA

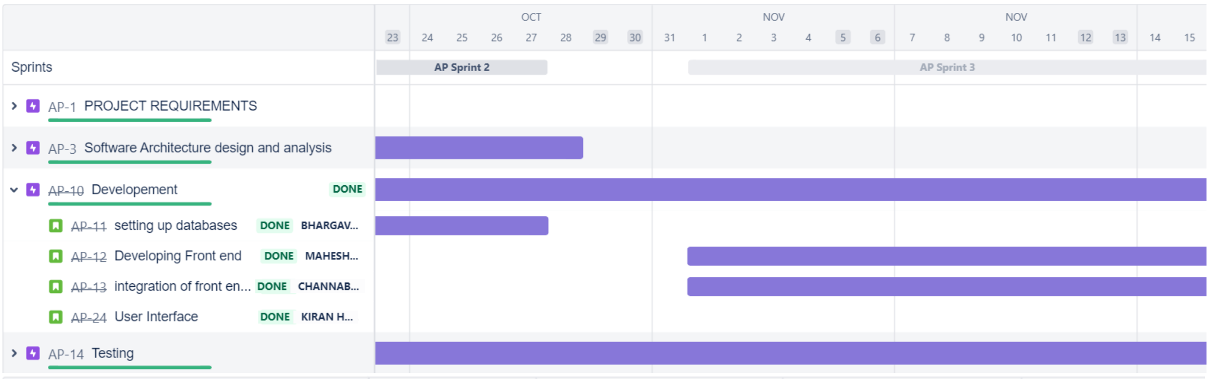


Sprints

AP-1	PROJECT REQUIREMENTS		
~~AP-5~~	research on feasibility...	DONE	BHARGAV...
~~AP-6~~	determining requireme...	DONE	BHARGAV...
~~AP-7~~	SRS documentation	DONE	MAHESH...
AP-3	Software Architecture design and analysis		
~~AP-4~~	Architecture analysis	DONE	MAHESH...
~~AP-8~~	Design analysis	DONE	BHARGAV...
~~AP-9~~	Design documentation	DONE	CHANNAB...

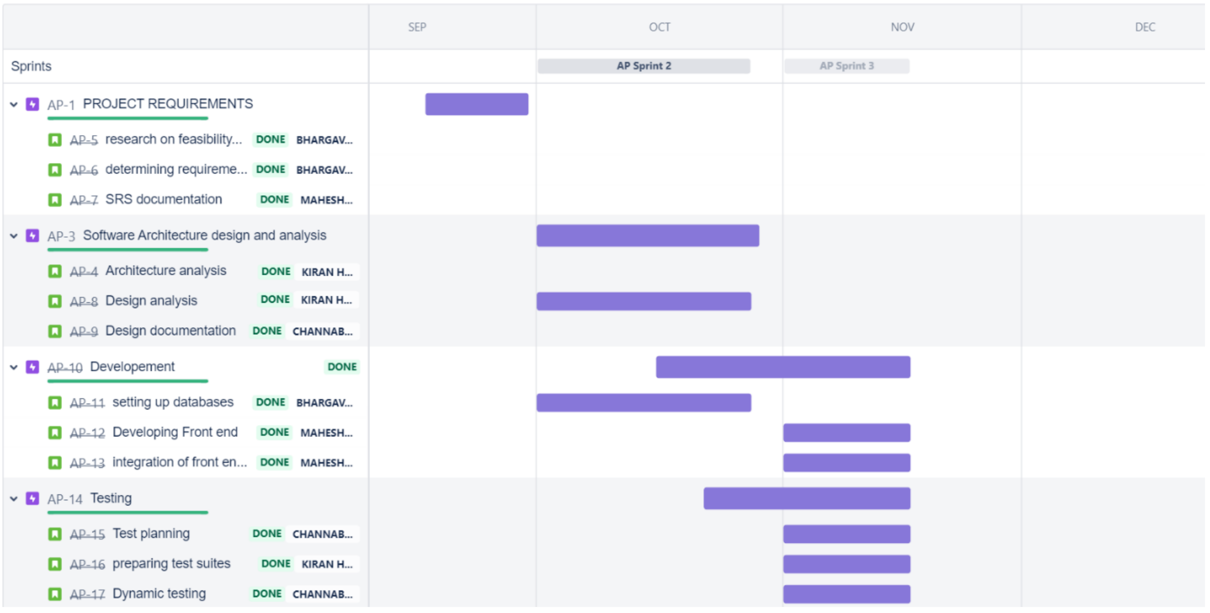


AP-14	Testing		
~~AP-15~~	Test planning	DONE	CHANNAB...
~~AP-16~~	preparing test suites	DONE	KIRAN H...
~~AP-17~~	Dynamic testing	DONE	CHANNAB...



Sprints

AP-1	PROJECT REQUIREMENTS		
AP-3	Software Architecture design and analysis		
~~AP-10~~	Developement	DONE	
~~AP-11~~	setting up databases	DONE	BHARGAV...
~~AP-12~~	Developing Front end	DONE	MAHESH...
~~AP-13~~	integration of front en...	DONE	CHANNAB...
~~AP-24~~	User Interface	DONE	KIRAN H...
AP-14	Testing		

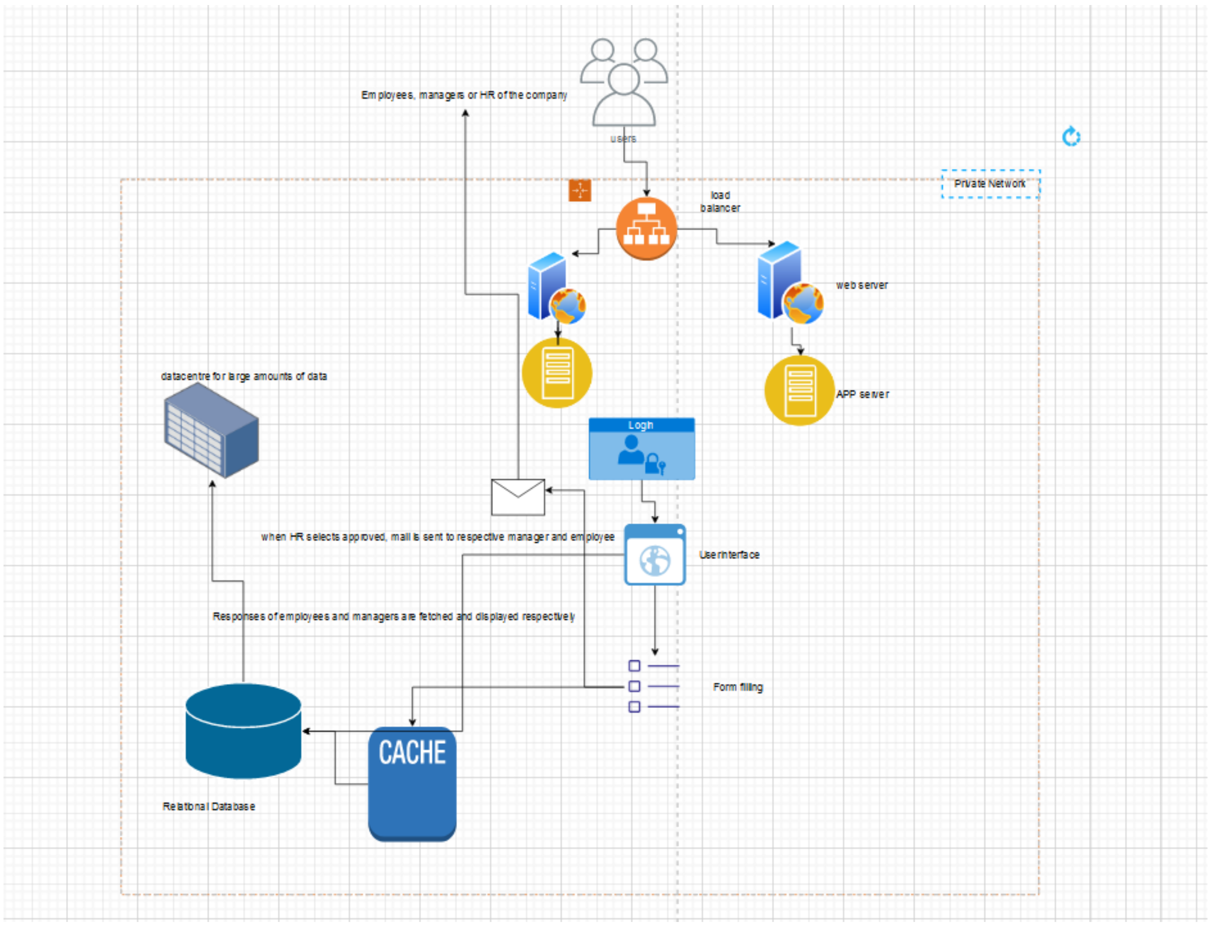


Sprints

AP-1	PROJECT REQUIREMENTS		
~~AP-5~~	research on feasibility...	DONE	BHARGAV...
~~AP-6~~	determining requireme...	DONE	BHARGAV...
~~AP-7~~	SRS documentation	DONE	MAHESH...
AP-3	Software Architecture design and analysis		
~~AP-4~~	Architecture analysis	DONE	KIRAN H...
~~AP-8~~	Design analysis	DONE	KIRAN H...
~~AP-9~~	Design documentation	DONE	CHANNAB...
~~AP-10~~	Developement	DONE	
~~AP-11~~	setting up databases	DONE	BHARGAV...
~~AP-12~~	Developing Front end	DONE	MAHESH...
~~AP-13~~	integration of front en...	DONE	MAHESH...
AP-14	Testing		
~~AP-15~~	Test planning	DONE	CHANNAB...
~~AP-16~~	preparing test suites	DONE	KIRAN H...
~~AP-17~~	Dynamic testing	DONE	CHANNAB...

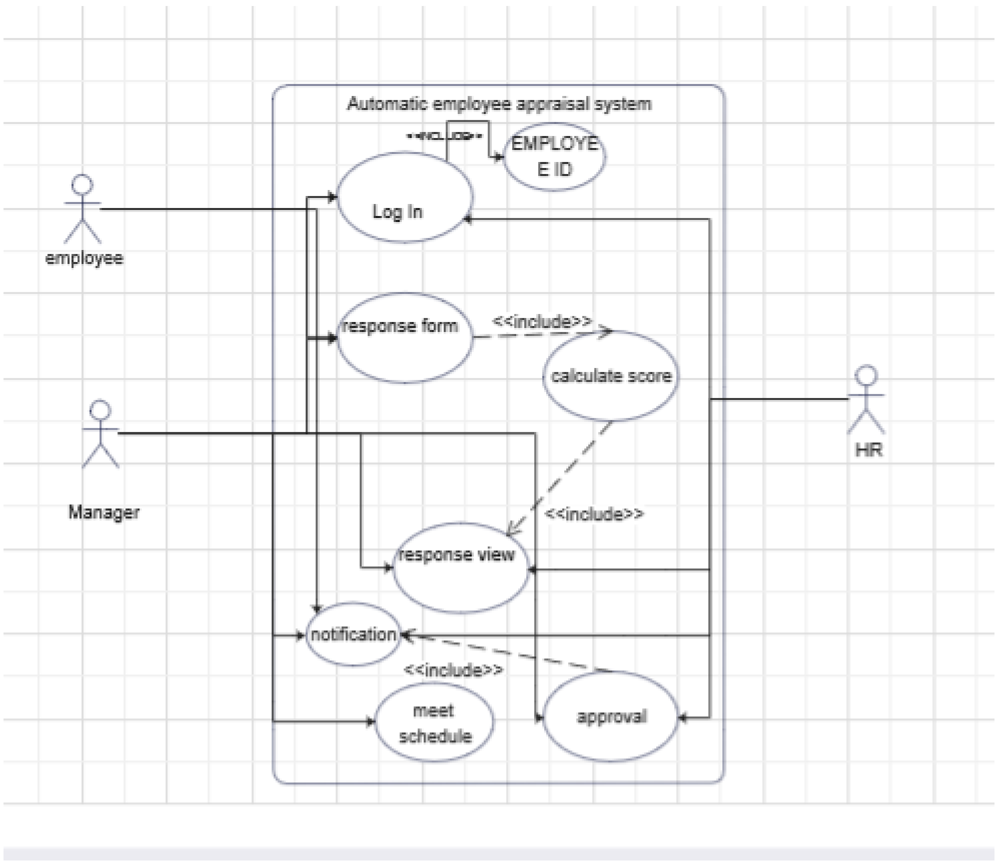
e)

Architecture diagram


Employees, managers or HR of the company
users
load balancer
Private Network
web server
APP server
datacentre for large amounts of data
Login
when HR selects approved, mail is sent to respective manager and employee
User interface
Responses of employees and managers are fetched and displayed respectively
Form filling
CACHE
Relational Database

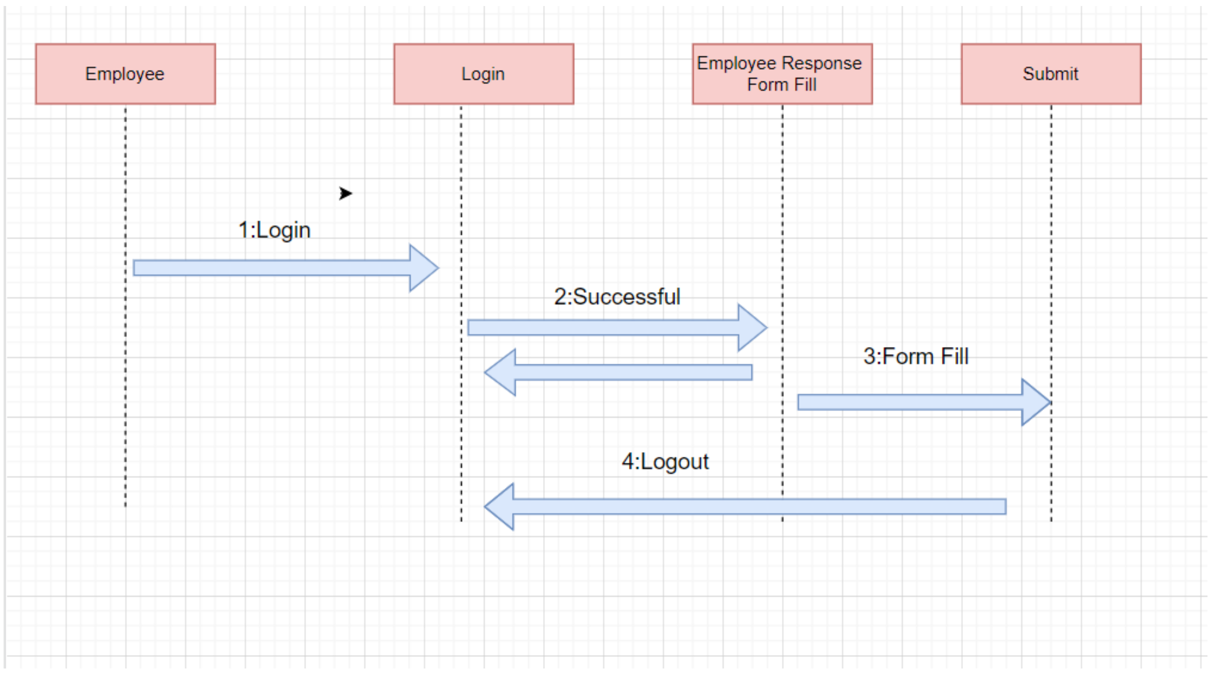
UML diagrams

use Case diagram

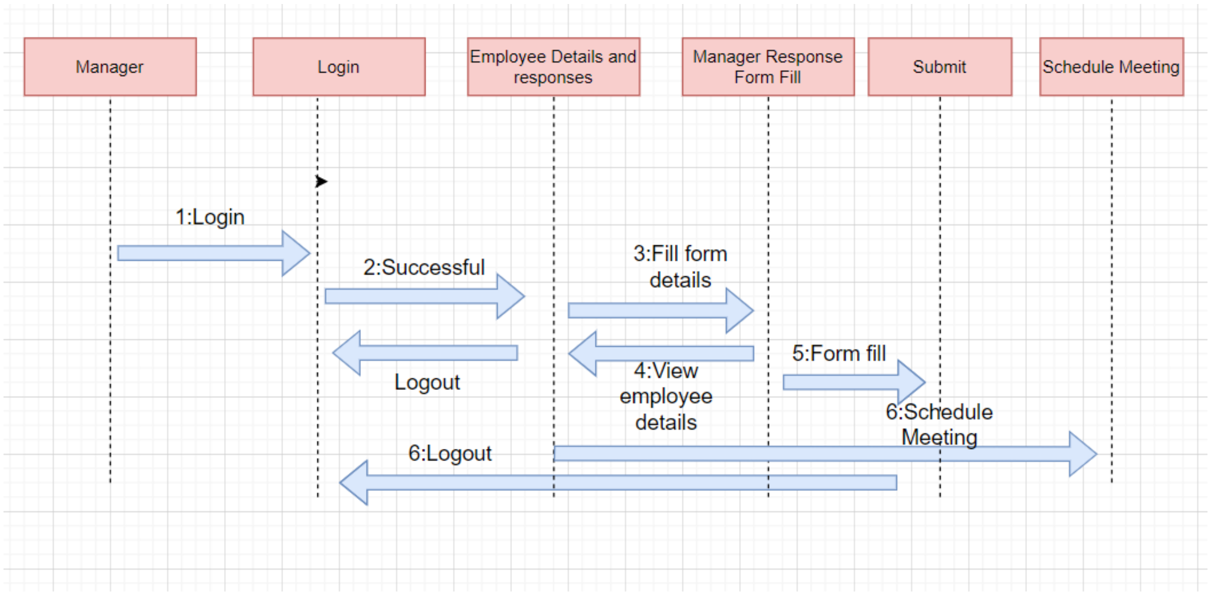


Sequence diagram

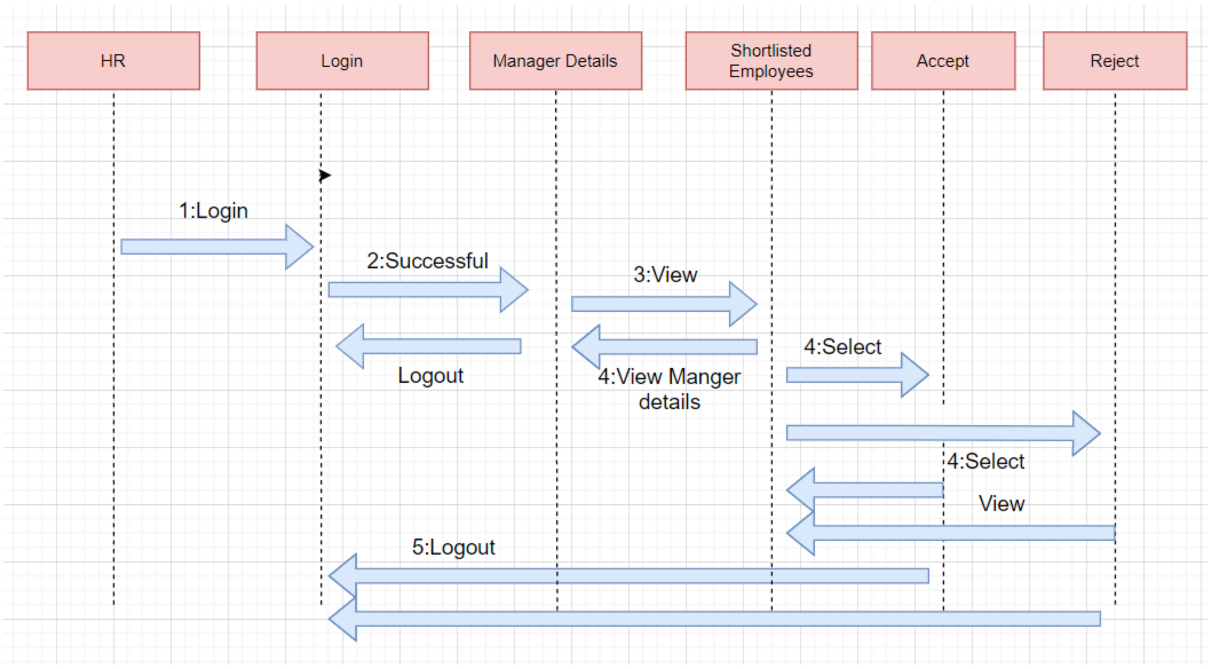
Employee:



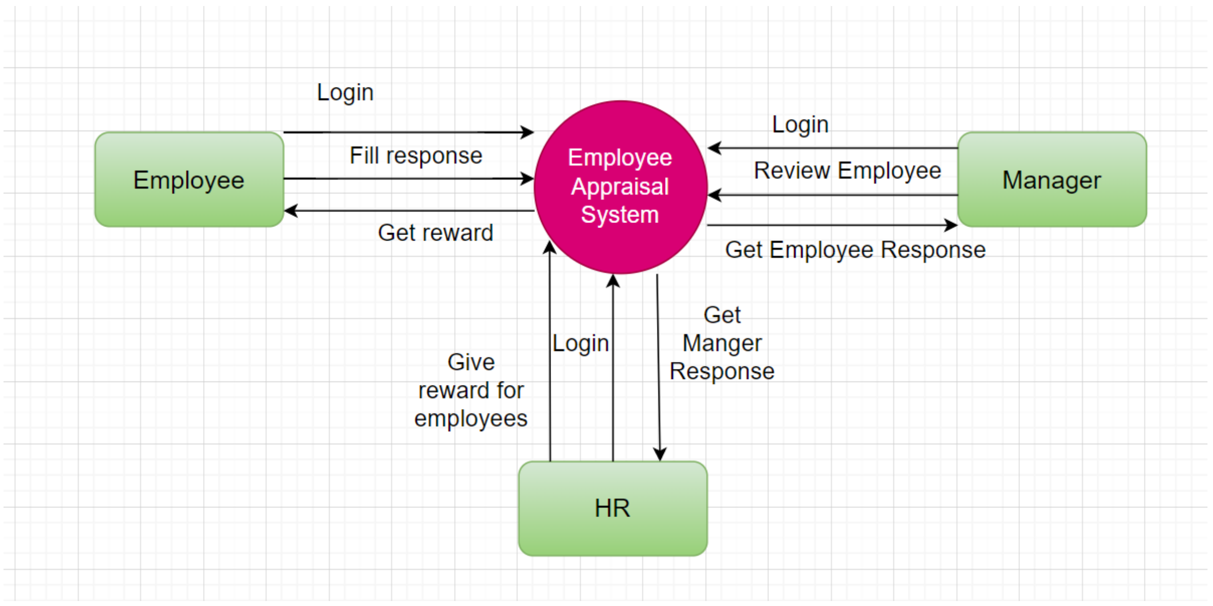
Manager:



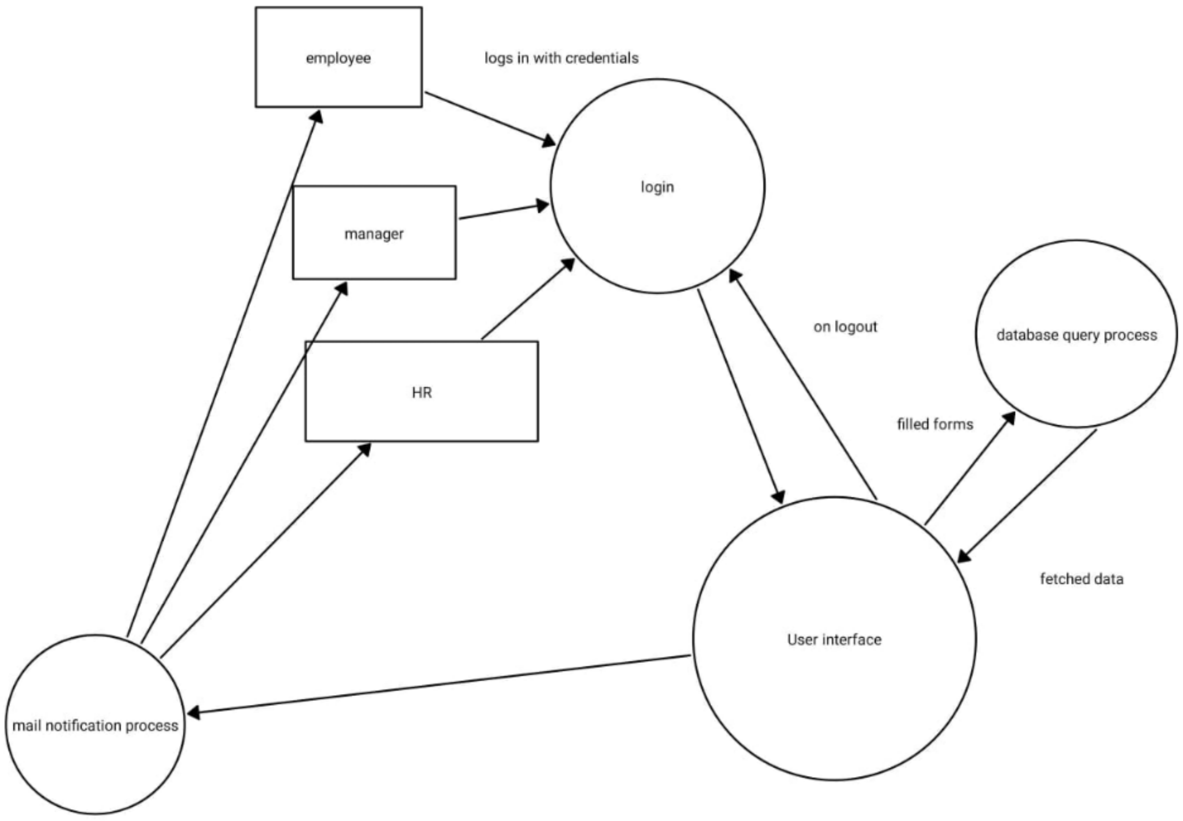
HR:



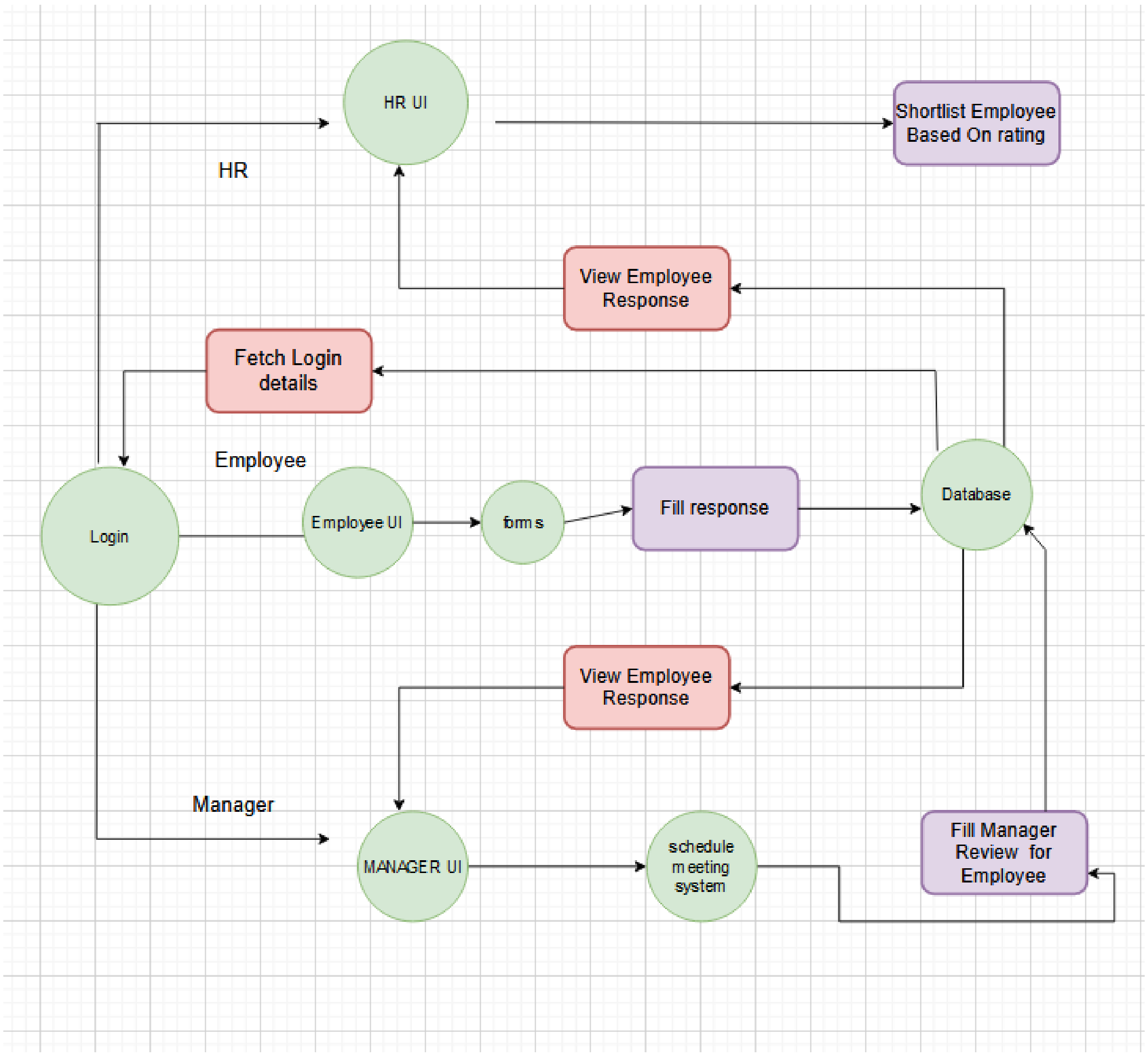
DFD level 0:



DFD LEVEL 1



DFD LEVEL 2



f)

Coding Standards used:

1. **Limited use of global variables**
   Avoid using global variables as far as possible.

2. **Standard headers for different modules:**
   For better understanding and maintenance of the code, the header of different modules should follow some standard format and information. The header format must contain below things that is being used in various companies:

   - Name of the module
   - Synopsis of the module about what the module does
   - Different functions supported in the module along with their input output parameters
   - Global variables accessed or modified by the module

3. **Naming conventions for local variables, global variables, constants and functions:**
   Some of the naming conventions are given below:

   - Meaningful and understandable variables name help anyone to understand the reason of using it.
   - It is better to avoid the use of digits in variable and function names.
   - The names of the function should be written in camel case starting with small letters.
   - The name of the function must describe the reason of using the function clearly and briefly.

4. **Indentation:**
   Proper indentation is very important to increase the readability of the code. For making the code readable, programmers should use White spaces properly. Some of the spacing conventions are given below:

   - There must be a space after giving a comma between two function arguments.
   - Each nested block should be properly indented and spaced.
   - Proper Indentation should be there at the beginning and at the end of each block in the program.
   - All braces should start from a new line and the code following the end of braces also start from a new line.

5. **Error return values and exception handling conventions:**
   All functions that encountering an error condition should either return a 0 or 1 for simplifying the debugging.

<u>Coding Guidelines used:</u>

1. **Avoid using a coding style that is too difficult to understand:**
   Code should be easily understandable. The complex code makes maintenance and debugging difficult and expensive.

2. **Avoid using an identifier for multiple purposes:**
   Each variable should be given a descriptive and meaningful name indicating the reason behind using it. This is not possible if an identifier is used for multiple purposes and thus it can lead to confusion to the reader. Moreover, it leads to more difficulty during future enhancements.

3. **Code should be well documented:**
   The code should be properly commented for understanding easily. Comments regarding the statements increase the understandability of the code.

4. **Length of functions should not be very large:**
   Lengthy functions are very difficult to understand. That's why functions should be small

enough to carry out small work and lengthy functions should be broken into small ones for completing small tasks.

g)

__Software Configuration Management__

We have used GitHub as a Software Configuration Management environment for our project.

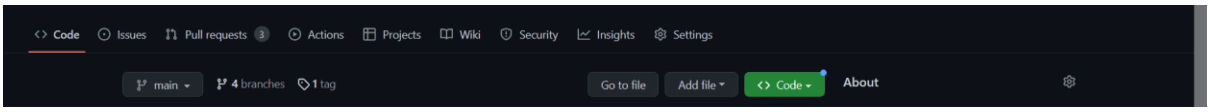
GitHub:

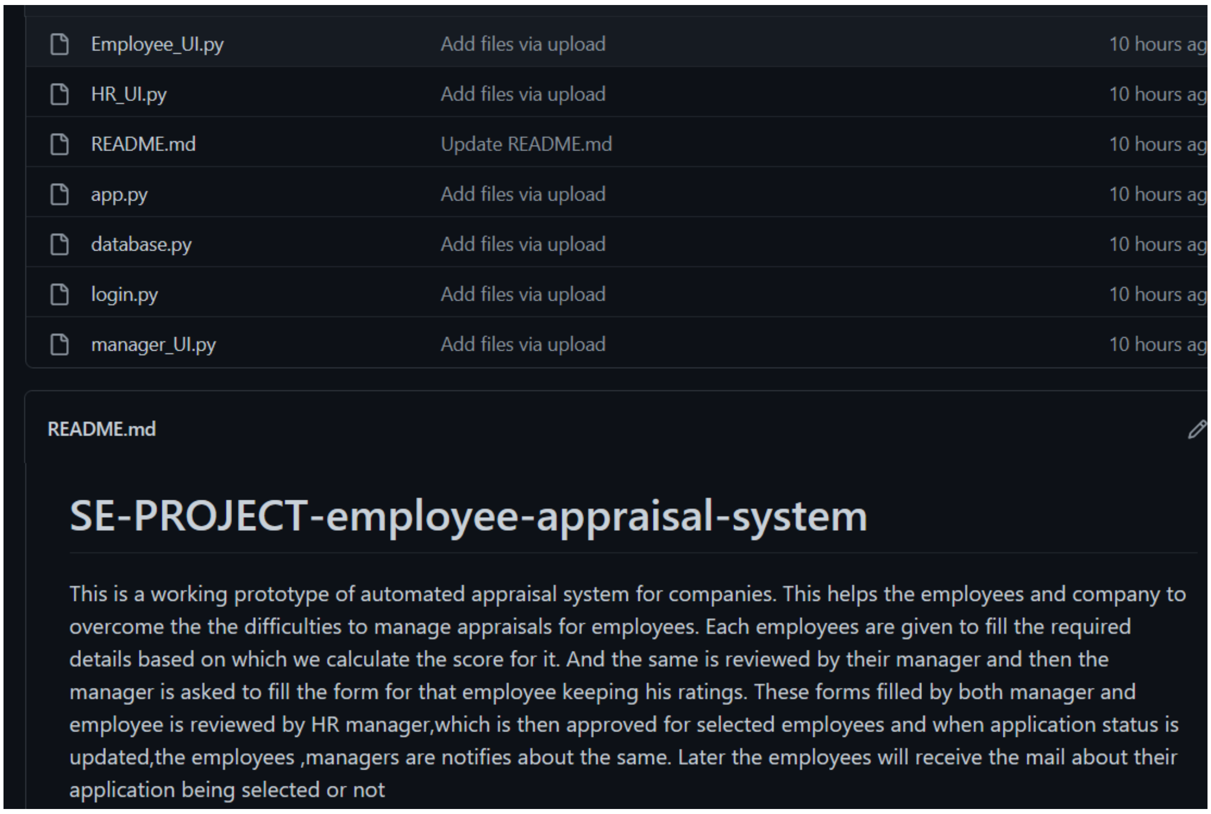
At a high level, GitHub is a website and cloud-based service that helps developers store and manage their code, as well as track and control changes to their code. It consists of 2 fundamental principles:

1) Branch management
2) Version control

Branch Management:

With branching, a developer duplicates part of the source code (called the repository). The developer can then safely make changes to that part of the code without affecting the rest of the project. Then, once the developer gets his or her part of the code working properly, he or she can merge that code back into the main source code to make it official. This is called branch management. We have used the in-built branch management system available on GitHub.



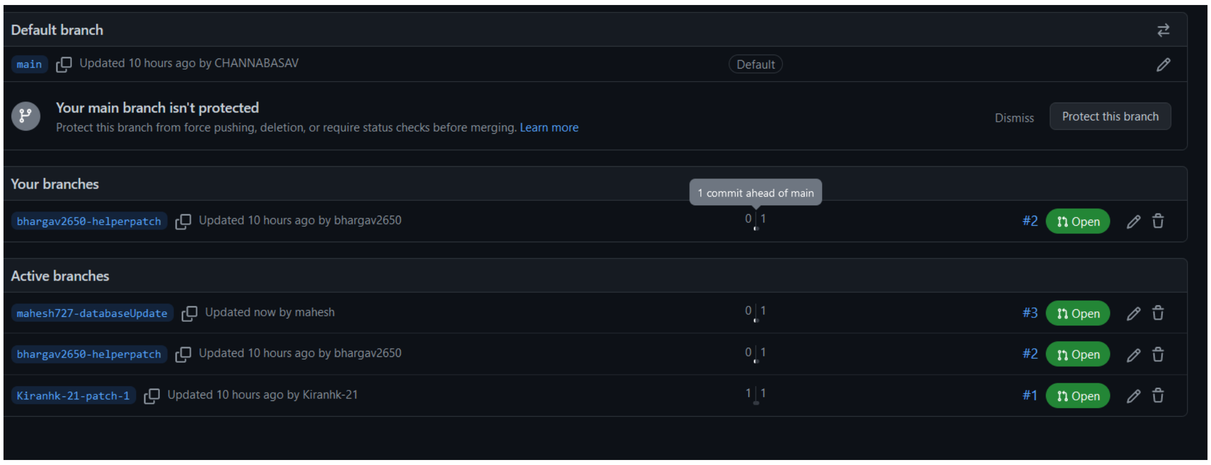


Employee_UI.py	Add files via upload	10 hours ag
HR_UI.py	Add files via upload	10 hours ag
README.md	Update README.md	10 hours ag
app.py	Add files via upload	10 hours ag
database.py	Add files via upload	10 hours ag
login.py	Add files via upload	10 hours ag
manager_UI.py	Add files via upload	10 hours ag

README.md

SE-PROJECT-employee-appraisal-system

This is a working prototype of automated appraisal system for companies. This helps the employees and company to overcome the the difficulties to manage appraisals for employees. Each employees are given to fill the required details based on which we calculate the score for it. And the same is reviewed by their manager and then the manager is asked to fill the form for that employee keeping his ratings. These forms filled by both manager and employee is reviewed by HR manager,which is then approved for selected employees and when application status is updated,the employees ,managers are notifies about the same. Later the employees will receive the mail about their application being selected or not


Default branch

main Updated 10 hours ago by CHANNABASAV Default

Your main branch isn't protected

Protect this branch from force pushing, deletion, or require status checks before merging. Learn more

Dismiss Protect this branch

Your branches

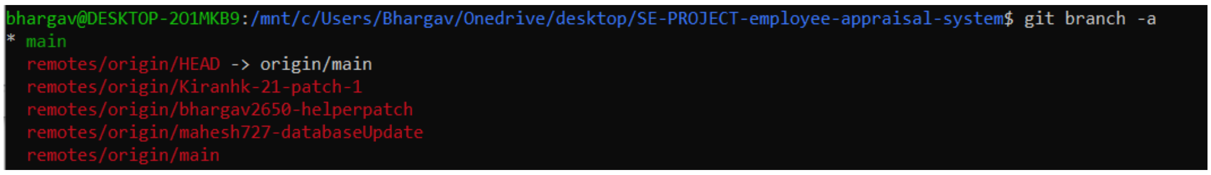
bhargav2650-helperpatch Updated 10 hours ago by bhargav2650 1 commit ahead of main 0 | 1 #2 Open

Active branches

mahesh727-databaseUpdate Updated now by mahesh 0 | 1 #3 Open

bhargav2650-helperpatch Updated 10 hours ago by bhargav2650 0 | 1 #2 Open

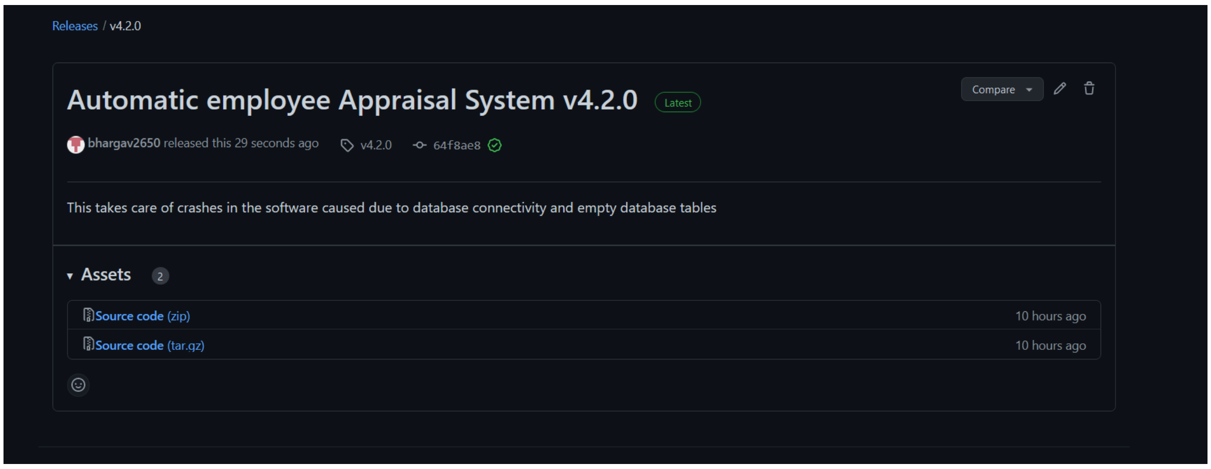
Kiranhk-21-patch-1 Updated 10 hours ago by Kiranhk-21 1 | 1 #1 Open


```
bhargav@DESKTOP-2O1MKB9:/mnt/c/Users/Bhargav/Onedrive/desktop/SE-PROJECT-employee-appraisal-system$ git branch -a
* main
  remotes/origin/HEAD -> origin/main
  remotes/origin/Kiranhk-21-patch-1
  remotes/origin/bhargav2650-helperpatch
  remotes/origin/mahesh727-databaseUpdate
  remotes/origin/main
```

Version control:

A version control system helps developers track and manage changes to files stored in the system. It protects the source code from any unintended human error and consequences. These files can be source code, assets, or other documents that might be part of a software development project. Teams make changes in groups called commits or revisions. Version control lets developers safely work through branching and merging.

Release:


Releases / v4.2.0

Automatic employee Appraisal System v4.2.0 Latest

bhargav2650 released this 29 seconds ago v4.2.0 64f8ae8

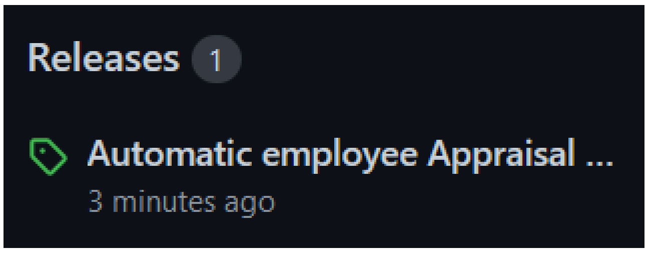
Compare

This takes care of crashes in the software caused due to database connectivity and empty database tables

Assets 2

Source code (zip) 10 hours ago

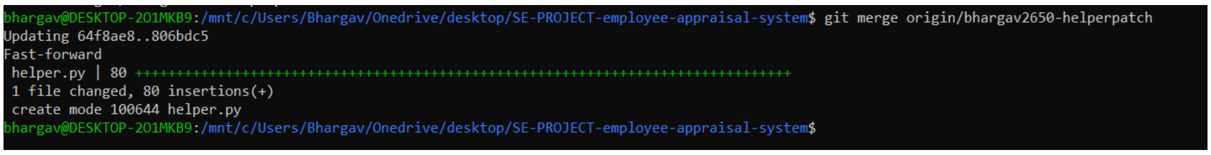
Source code (tar.gz) 10 hours ago


Releases 1

Automatic employee Appraisal ...

3 minutes ago

Merging of the helperpatch branch



```
bhargav@DESKTOP-201MKB9:/mnt/c/Users/Bhargav/Onedrive/desktop/SE-PROJECT-employee-appraisal-system$ git merge origin/bhargav2650-helperpatch
Updating 64f8ae8..806bdc5
Fast-forward
 helper.py | 80 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 1 file changed, 80 insertions(+)
 create mode 100644 helper.py
bhargav@DESKTOP-201MKB9:/mnt/c/Users/Bhargav/Onedrive/desktop/SE-PROJECT-employee-appraisal-system$
```

h)

1) Static testing
2) Unit testing
3) Integration testing
4) System testing
    - Functional testing
    - Regression testing
    - Usability testing
5) Acceptance testing

Static testing:

By static testing, we have removed the following:

(a) Unused variables

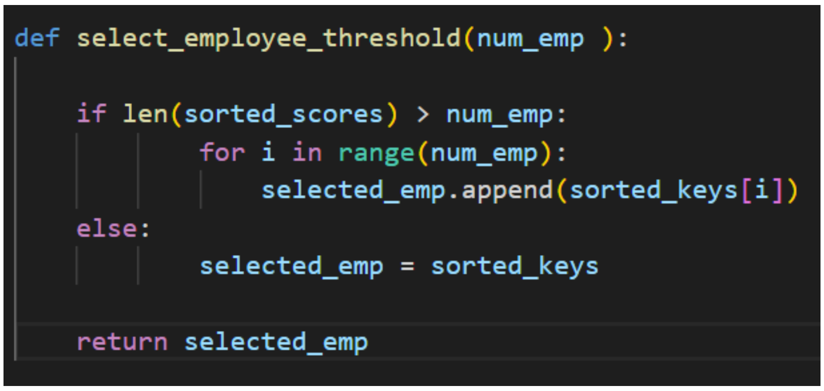
(b) Dead code

(c) Variable with undefined value

(d) Wrong syntax

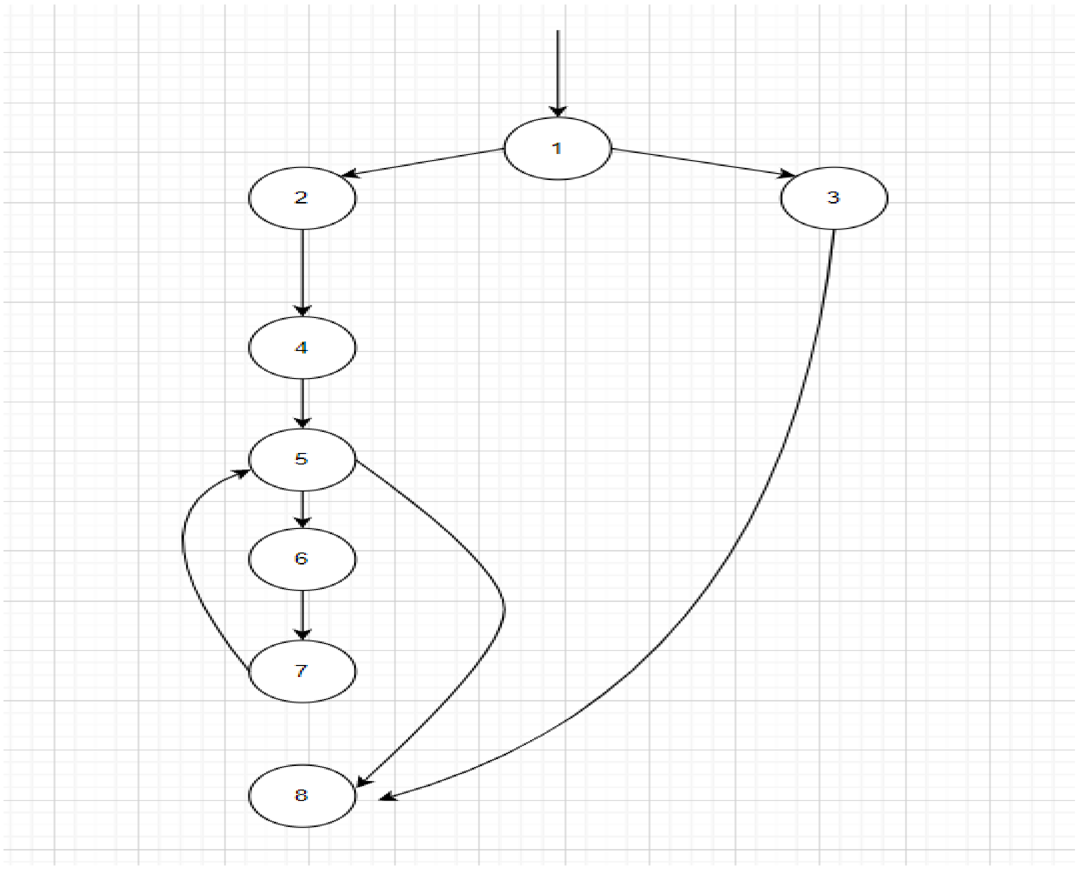
For example, we have chosen a snippet of code from the function select_employee_threshold which selects user_ID of employees whose score is above given threshold.

Code:



```python
def select_employee_threshold(num_emp ):

    if len(sorted_scores) > num_emp:
            for i in range(num_emp):
                selected_emp.append(sorted_keys[i])
    else:
            selected_emp = sorted_keys

    return selected_emp
```

Control flow diagram:



Conclusion:

Thus, the cyclomatic complexity of this unit of code is: Number of edges – number of nodes + (2*number of exit points)

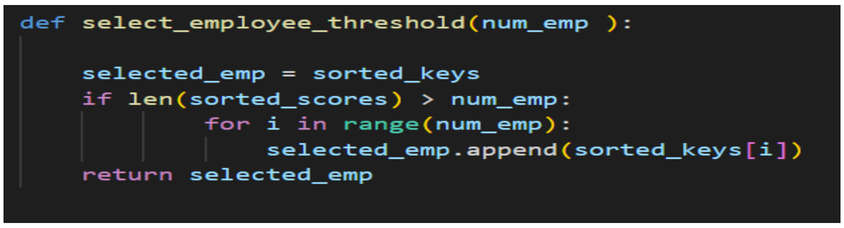
=E-N+2*P

= 9 – 8 + (2*1)

= 1 + 2 = 3

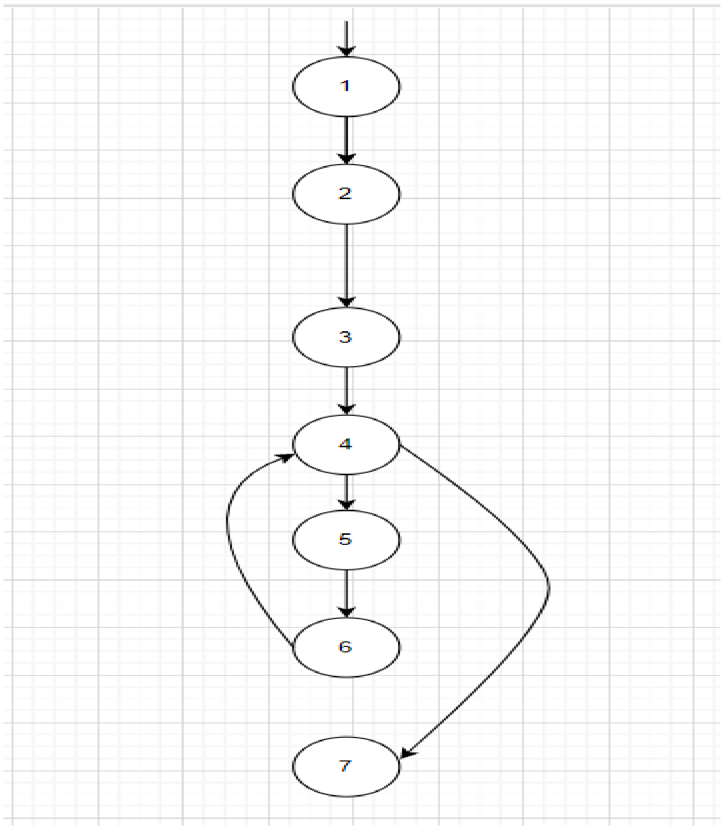
The cyclomatic complexity is 4, it is already pretty efficient and but can be optimized.

Modified code:



```
def select_employee_threshold(num_emp ):

    selected_emp = sorted_keys
    if len(sorted_scores) > num_emp:
            for i in range(num_emp):
                selected_emp.append(sorted_keys[i])
    return selected_emp
```

Control flow diagram:



Conclusion:

the cyclomatic complexity of this unit of code is: Number of edges – number of nodes + (2*number of exit points)

=E-N+2*P

= 7– 7 + (2*1)

= 0 + 2 = 2

Since the cyclomatic complexity is 2, it is already pretty efficient and cannot be optimized. By observing the control flow graph, we can see that if we remove any nodes, we will either disrupt the functioning or we will have to remove the same number of edges too thus not changing the overall cyclomatic complexity

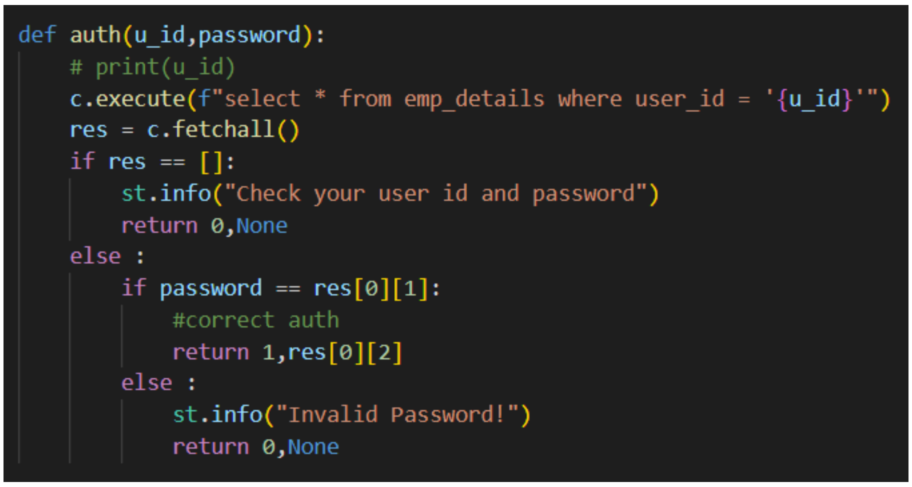
Unit testing:

Unit testing is a software testing method by which individual units of source code—sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures—are tested to determine whether they are fit for use. We have tested the various functionalities of the several pages we have in our application. The summary of the tests is given below:

Test Case ID	Name of Module	Test case description	Test Steps	Test data	Expected Result	Actual Result	Test Result
UT-01	login	Testing the login functionality	-Enter user_id, password -Click ok	User_id Password	-login of user on success -Route to main page	-login of user on success -Route to main page	Pass
UT-02	Get employe e name	Testing get name feature	-on login -fetch ename with the help of eID	User_iD	-Fetch corresponding ename after login	--Fetch corresponding ename after login	Pass
UT-03	Send mail	Sends a mail to given recv mailID and body	-enter recvID And body -click send mail	Email ID Body Attachtment (optional)	-Sends the mail to recv ID with body and attatchment embedded	-Sends the mail to recv ID with body and attatchment embedded	Pass
UT-04	Select employe e	Selects employee whose score is above given threshold	-enter threshold score - submit	-Threshold score -employee parameters	-sends list off all employees whose score is greater	-sends list off all employees whose score is greater	Pass

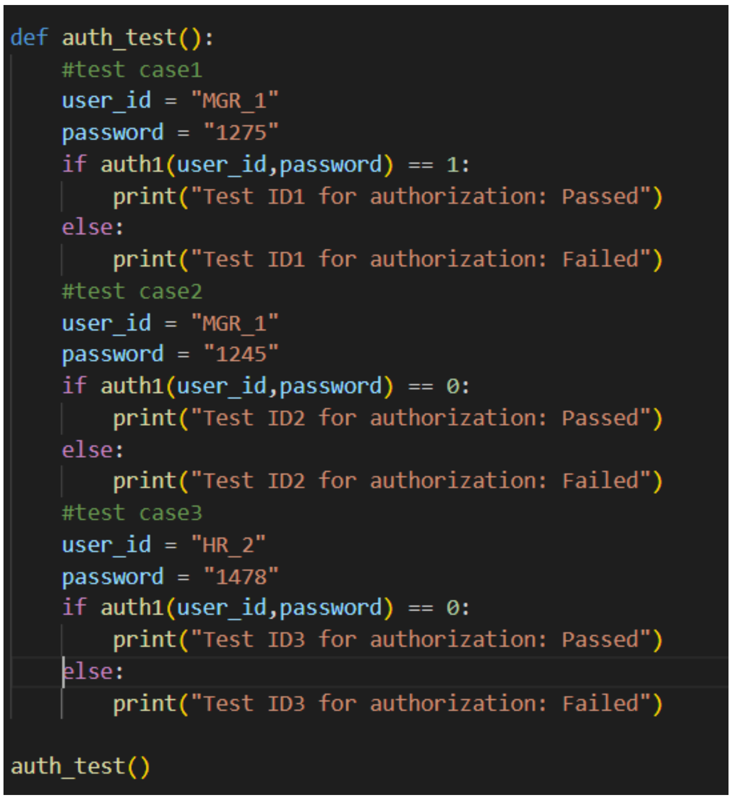
UT_01:

Function

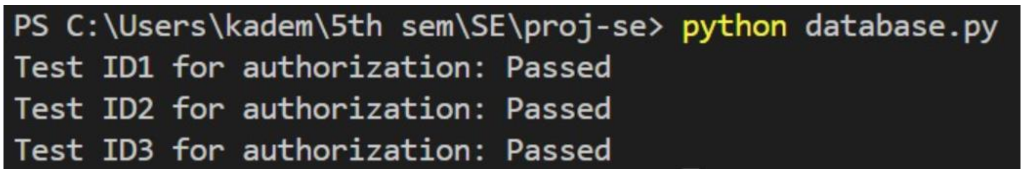


```python
def auth(u_id,password):
    # print(u_id)
    c.execute(f"select * from emp_details where user_id = '{u_id}'")
    res = c.fetchall()
    if res == []:
        st.info("Check your user id and password")
        return 0,None
    else :
        if password == res[0][1]:
            #correct auth
            return 1,res[0][2]
        else :
            st.info("Invalid Password!")
            return 0,None
```

Test cases



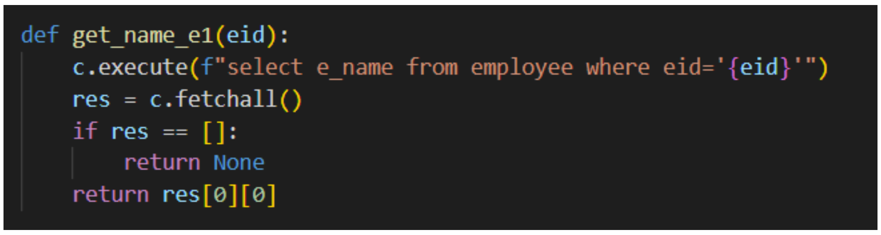
```python
def auth_test():
    #test case1
    user_id = "MGR_1"
    password = "1275"
    if auth1(user_id,password) == 1:
        print("Test ID1 for authorization: Passed")
    else:
        print("Test ID1 for authorization: Failed")
    #test case2
    user_id = "MGR_1"
    password = "1245"
    if auth1(user_id,password) == 0:
        print("Test ID2 for authorization: Passed")
    else:
        print("Test ID2 for authorization: Failed")
    #test case3
    user_id = "HR_2"
    password = "1478"
    if auth1(user_id,password) == 0:
        print("Test ID3 for authorization: Passed")
    else:
        print("Test ID3 for authorization: Failed")

auth_test()
```



```
PS C:\Users\kadem\5th sem\SE\proj-se> python database.py
Test ID1 for authorization: Passed
Test ID2 for authorization: Passed
Test ID3 for authorization: Passed
```

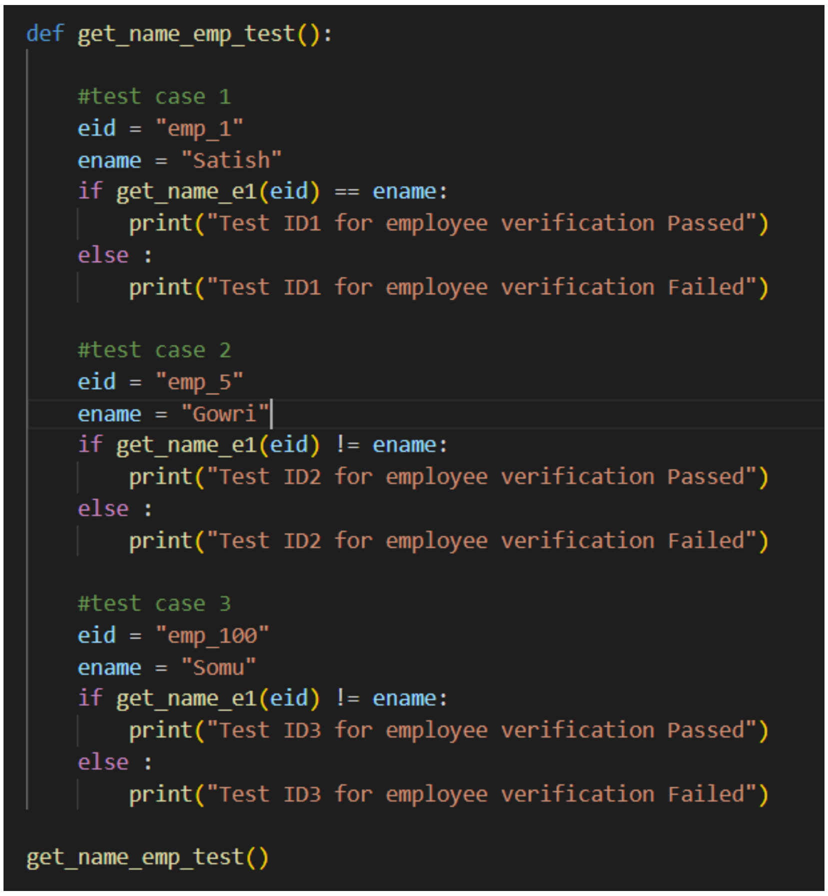
UT_02:

Function

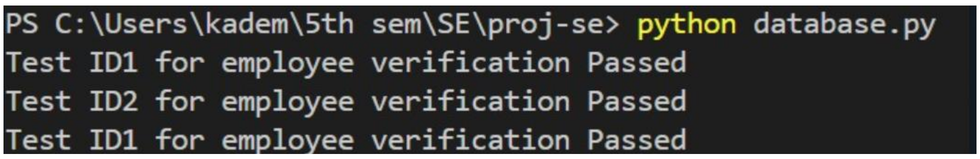


```python
def get_name_e1(eid):
    c.execute(f"select e_name from employee where eid='{eid}'")
    res = c.fetchall()
    if res == []:
        return None
    return res[0][0]
```

Test cases



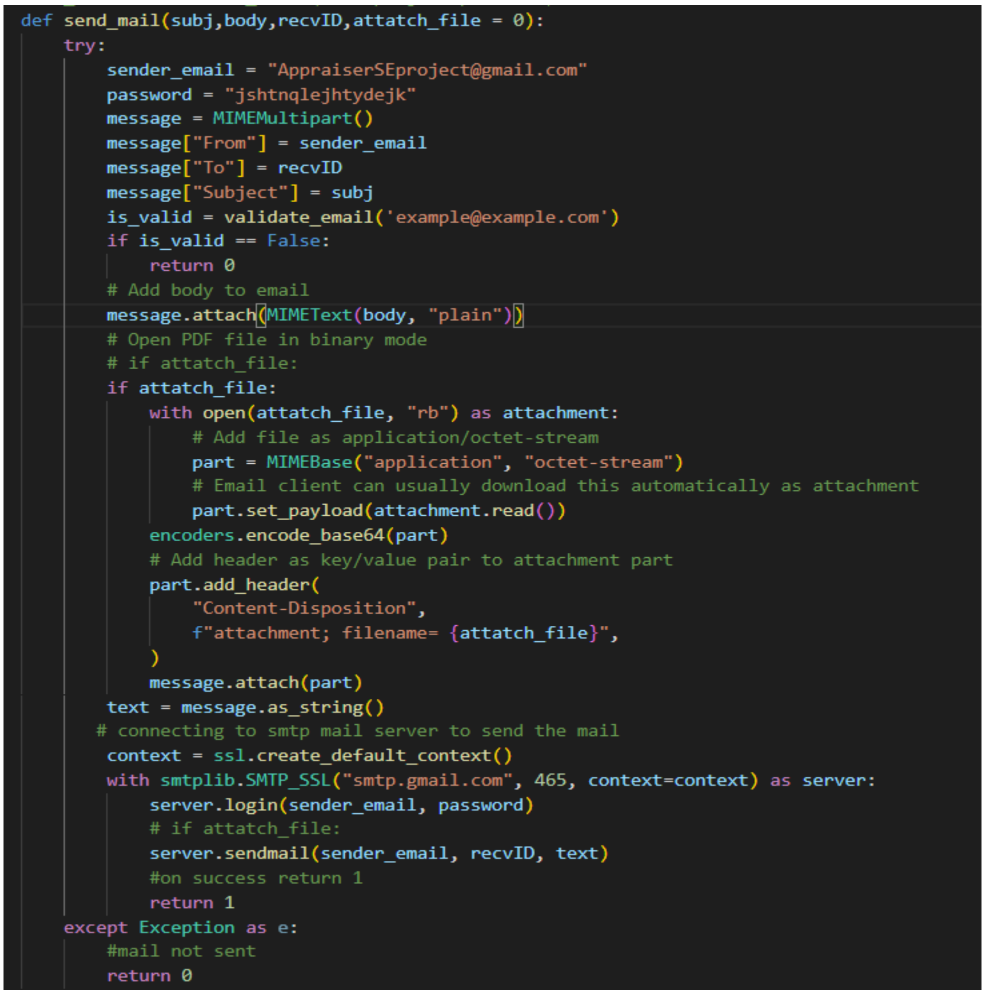
```python
def get_name_emp_test():

    #test case 1
    eid = "emp_1"
    ename = "Satish"
    if get_name_e1(eid) == ename:
        print("Test ID1 for employee verification Passed")
    else :
        print("Test ID1 for employee verification Failed")

    #test case 2
    eid = "emp_5"
    ename = "Gowri"
    if get_name_e1(eid) != ename:
        print("Test ID2 for employee verification Passed")
    else :
        print("Test ID2 for employee verification Failed")

    #test case 3
    eid = "emp_100"
    ename = "Somu"
    if get_name_e1(eid) != ename:
        print("Test ID3 for employee verification Passed")
    else :
        print("Test ID3 for employee verification Failed")

get_name_emp_test()
```



```
PS C:\Users\kadem\5th sem\SE\proj-se> python database.py
Test ID1 for employee verification Passed
Test ID2 for employee verification Passed
Test ID1 for employee verification Passed
```

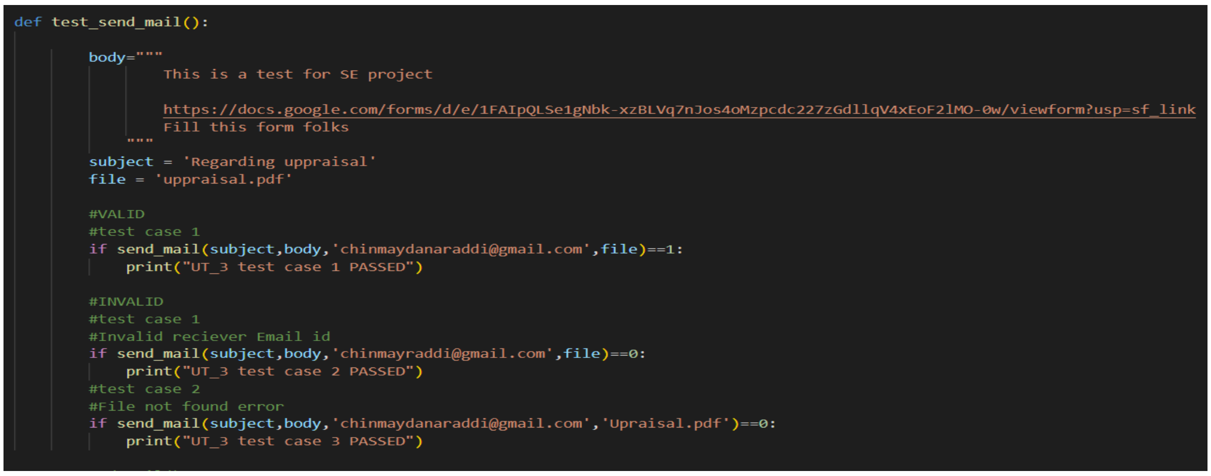
UT_03:

Function

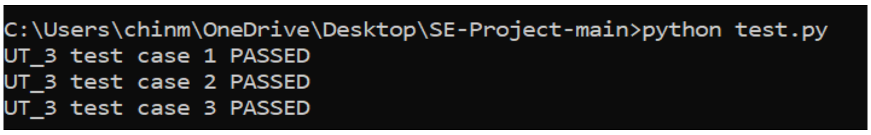


```python
def send_mail(subj,body,recvID,attatch_file = 0):
    try:
        sender_email = "AppraiserSEproject@gmail.com"
        password = "jshtnqlejhtydejk"
        message = MIMEMultipart()
        message["From"] = sender_email
        message["To"] = recvID
        message["Subject"] = subj
        is_valid = validate_email('example@example.com')
        if is_valid == False:
            return 0
        # Add body to email
        message.attach(MIMEText(body, "plain"))
        # Open PDF file in binary mode
        # if attatch_file:
        if attatch_file:
            with open(attatch_file, "rb") as attachment:
                # Add file as application/octet-stream
                part = MIMEBase("application", "octet-stream")
                # Email client can usually download this automatically as attachment
                part.set_payload(attachment.read())
            encoders.encode_base64(part)
            # Add header as key/value pair to attachment part
            part.add_header(
                "Content-Disposition",
                f"attachment; filename= {attatch_file}",
            )
            message.attach(part)
        text = message.as_string()
       # connecting to smtp mail server to send the mail
        context = ssl.create_default_context()
        with smtplib.SMTP_SSL("smtp.gmail.com", 465, context=context) as server:
            server.login(sender_email, password)
            # if attatch_file:
            server.sendmail(sender_email, recvID, text)
            #on success return 1
            return 1
    except Exception as e:
        #mail not sent
        return 0
```

Test cases



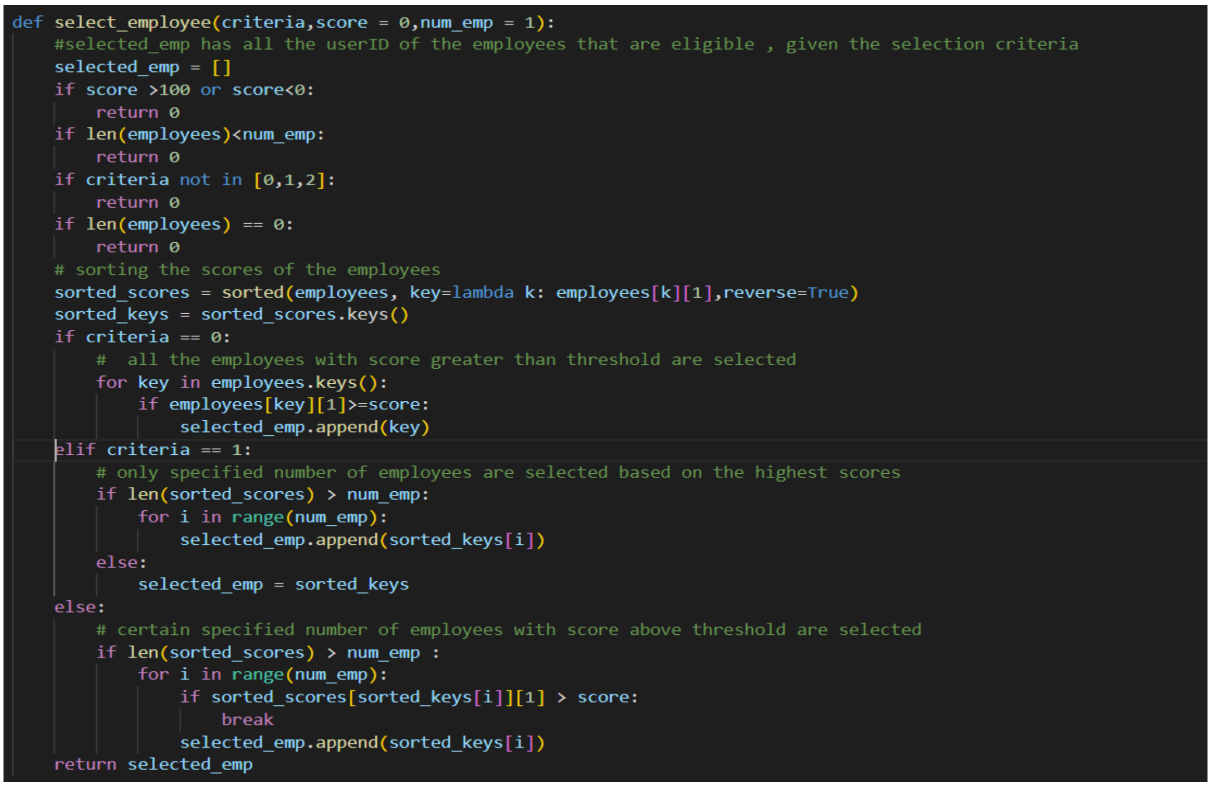
```python
def test_send_mail():

        body="""
                This is a test for SE project

                https://docs.google.com/forms/d/e/1FAIpQLSe1gNbk-xzBLVq7nJos4oMzpcdc227zGdllqV4xEoF21MO-0w/viewform?usp=sf_link
                Fill this form folks
            """
        subject = 'Regarding uppraisal'
        file = 'uppraisal.pdf'

        #VALID
        #test case 1
        if send_mail(subject,body,'chinmaydanaraddi@gmail.com',file)==1:
            print("UT_3 test case 1 PASSED")

        #INVALID
        #test case 1
        #Invalid reciever Email id
        if send_mail(subject,body,'chinmayraddi@gmail.com',file)==0:
            print("UT_3 test case 2 PASSED")
        #test case 2
        #File not found error
        if send_mail(subject,body,'chinmaydanaraddi@gmail.com','Upraisal.pdf')==0:
            print("UT_3 test case 3 PASSED")
```



```
C:\Users\chinm\OneDrive\Desktop\SE-Project-main>python test.py
UT_3 test case 1 PASSED
UT_3 test case 2 PASSED
UT_3 test case 3 PASSED
```

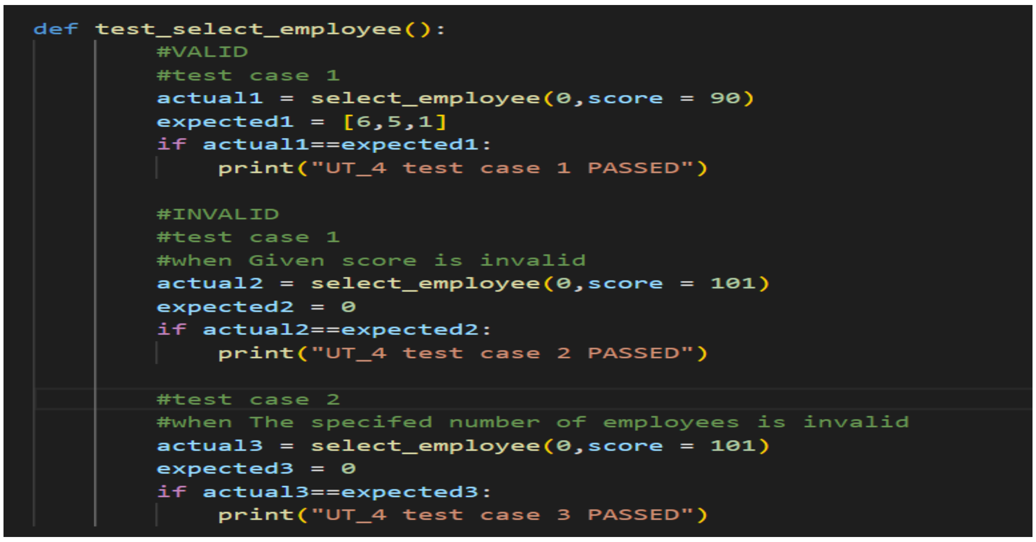
UT_04:

Function

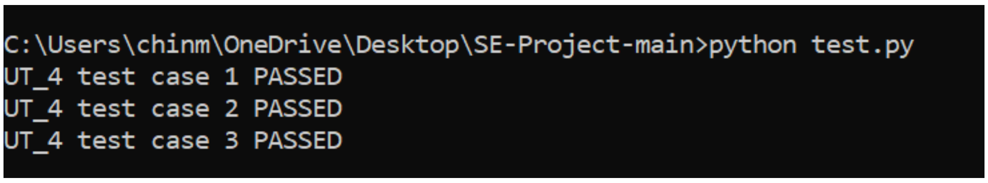


```python
def select_employee(criteria,score = 0,num_emp = 1):
    #selected_emp has all the userID of the employees that are eligible , given the selection criteria
    selected_emp = []
    if score >100 or score<0:
        return 0
    if len(employees)<num_emp:
        return 0
    if criteria not in [0,1,2]:
        return 0
    if len(employees) == 0:
        return 0
    # sorting the scores of the employees
    sorted_scores = sorted(employees, key=lambda k: employees[k][1],reverse=True)
    sorted_keys = sorted_scores.keys()
    if criteria == 0:
        #  all the employees with score greater than threshold are selected
        for key in employees.keys():
            if employees[key][1]>=score:
                selected_emp.append(key)
    elif criteria == 1:
        # only specified number of employees are selected based on the highest scores
        if len(sorted_scores) > num_emp:
            for i in range(num_emp):
                selected_emp.append(sorted_keys[i])
        else:
            selected_emp = sorted_keys
    else:
        # certain specified number of employees with score above threshold are selected
        if len(sorted_scores) > num_emp :
            for i in range(num_emp):
                if sorted_scores[sorted_keys[i]][1] > score:
                    break
                selected_emp.append(sorted_keys[i])
    return selected_emp
```

Test cases



```python
def test_select_employee():
        #VALID
        #test case 1
        actual1 = select_employee(0,score = 90)
        expected1 = [6,5,1]
        if actual1==expected1:
            print("UT_4 test case 1 PASSED")

        #INVALID
        #test case 1
        #when Given score is invalid
        actual2 = select_employee(0,score = 101)
        expected2 = 0
        if actual2==expected2:
            print("UT_4 test case 2 PASSED")

        #test case 2
        #when The specifed number of employees is invalid
        actual3 = select_employee(0,score = 101)
        expected3 = 0
        if actual3==expected3:
            print("UT_4 test case 3 PASSED")
```



```
C:\Users\chinm\OneDrive\Desktop\SE-Project-main>python test.py
UT_4 test case 1 PASSED
UT_4 test case 2 PASSED
UT_4 test case 3 PASSED
```

Integration testing:

Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing is conducted to evaluate the compliance of a system or component with specified functional requirements.

We have tested the 2 main pages of our application, each of the pages having several functionalities. Any errors detected were successfully corrected. Hence, we consider it to be successful.

IT-01	Select employee and send mail	Testing the viewing transaction details functionality	-select threshold and press send mail button	None	All the selected employees Should get mail	All the selected employees Should get mail	Pass
IT-02	Login and main page	Main page corresponding to role should be routed after login	-login with valid details	User_id password	-On successful login the corresponding main page should open	- On successful login the corresponding main page should open	Pass
IT-03	Login and get ename	After logging in , gets ename of corresponding emp_ID	- login -fetch ename with from eID	User_ID password	-Fetch corresponding ename after login	- Fetch corresponding ename after login	Pass

System testing:

After testing the various modules and the main pages of our application, we proceed to system testing.

System Testing is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications. System Testing is defined as a series of different tests whose sole purpose is to exercise the full computer-based system.

(a)Functional testing

Functional Testing is a type of Software Testing in which the system is tested against the functional requirements and specifications. Functional testing ensures that the requirements or specifications are properly satisfied by the application.

Functionalities	Status
- Login	Pass
-separate UI for manager, employee, HR	Pass
-employee form	Pass
-employee should get mail once the process starts	Pass
-manager should get reminders to review	Pass
-manager should have features to schedule meet	Pass
-Automatic score calculation	Pass
-Has to select employees based on scores	Pass
-HR has an option to approve or reject	Pass
-once approved by HR employee should get mail	Pass

(b) Regression testing

Every time a new module is added leads to changes in the program. This type of testing makes sure that the whole component works properly even after adding components to the complete program.

During the development phase, we kept adding new modules. Every time we added a new module, we checked to see if the new changes caused any trouble. We made sure that the whole system worked before moving forward by correcting any errors caused due to the new module. Hence, we can say that it was successful.

(c)Usability testing

Usability testing refers to evaluating a product or service by testing it with representative users. Typically, during a test, participants will try to complete typical tasks while observers watch, listen and takes notes. The goal is to identify any usability problems, collect qualitative and quantitative data and determine the participant's satisfaction with the product.

We showed our app to another team without telling them what it was. They could figure out what the app is for and could name the functionalities. This confirms that our app is easy to use for the average person. This means that the app can be used by more people and thus has more reach. Hence, we can say that our application has successfully passed the usability testing stage.

Acceptance testing:

Another team has reviewed our project.

Details of the other team**:**

Toukheer Baig PES1UG20CS616

LIkhith R PES1UG20CS659

Kushagra Singh PES1UG20CS657

Mohammad Anas Danish PES1UG20CS614

Review by the other team:

Their endeavor involved creating a tool for Automating performance appraisal . Has three main parts Manager ,HR ,Employee , once the appraisal process starts all the eligible employees get mail and have to fill a form on their portal which is later reviewed by manager and approved by HR . All the UI designed are user-friendly and easy to use .We realized that their project makes the entire process of appraisal smooth and allows minimum human intervention.

Bugs detected:

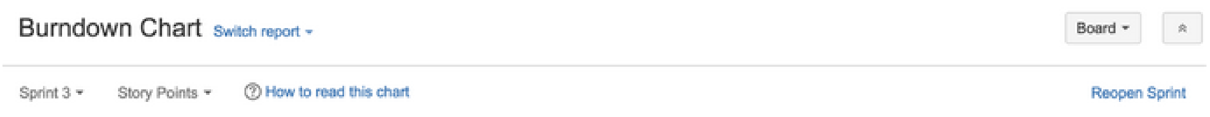
1) Logout button has to be clicked twice.
2) Duplicate submission accepted.

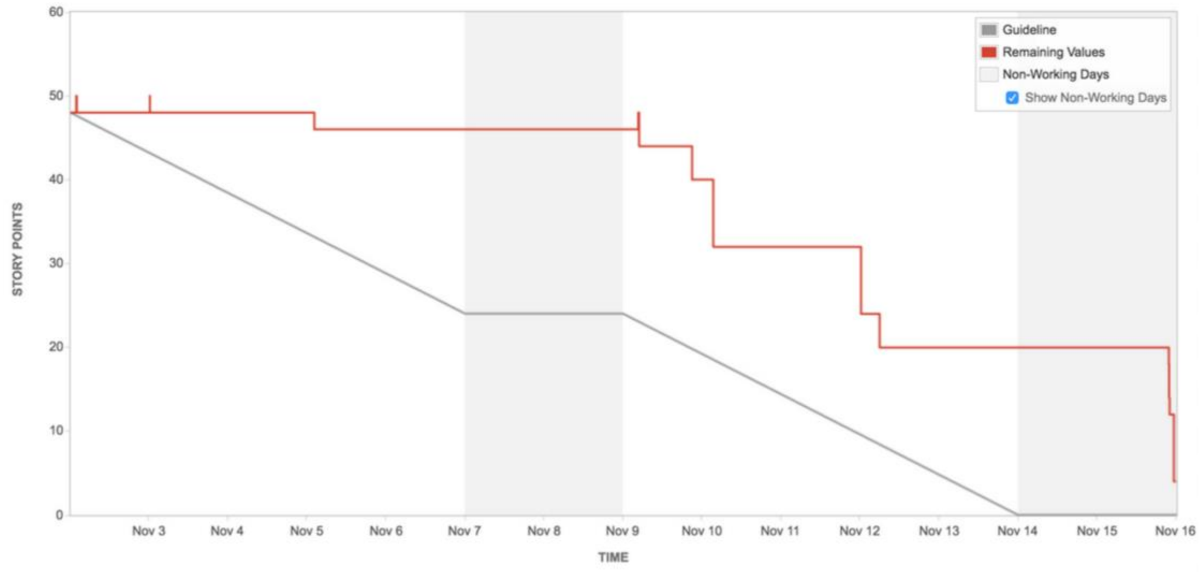
   These bugs were then corrected.

Suggestions given by them:

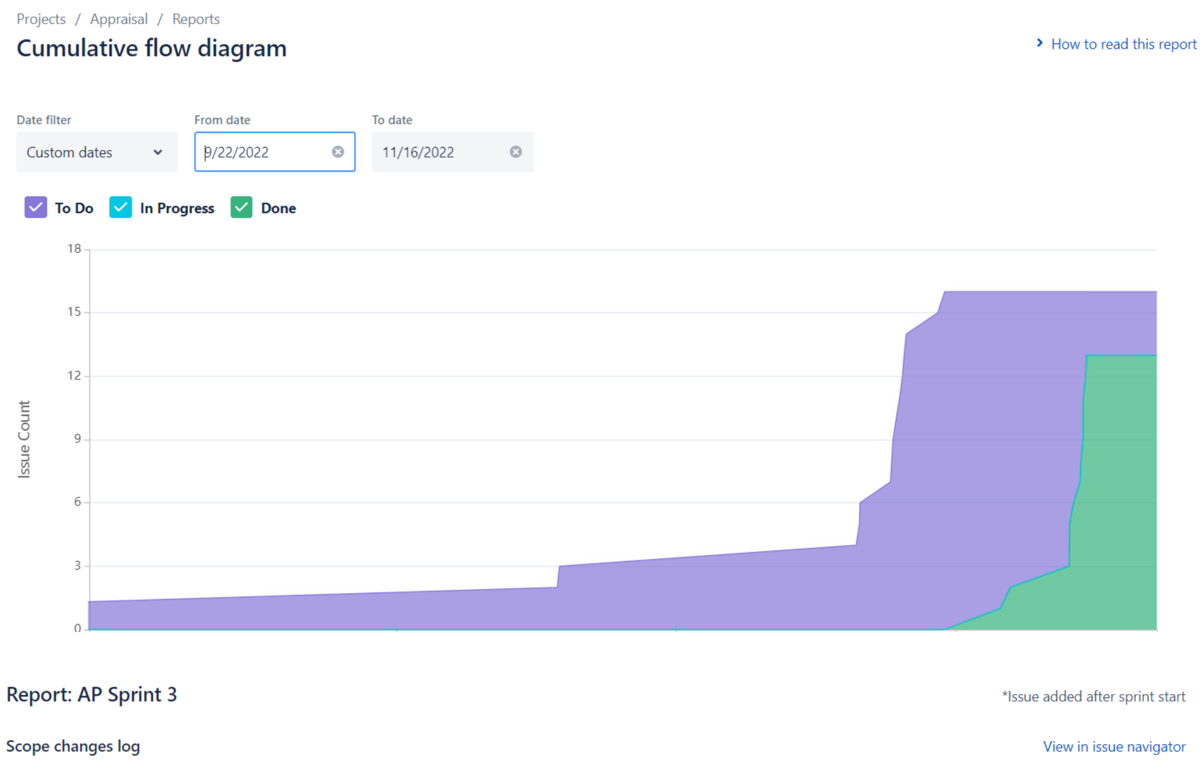
1) Send email option in manager UI
2) Automatic meeting link generation
3) Google forms instead of employee UI

i)





j)report accessed on JIIRA from report tab on the left side of the website


Projects / Appraisal / Reports

Cumulative flow diagram

How to read this report

Date filter: Custom dates

From date: 9/22/2022

To date: 11/16/2022

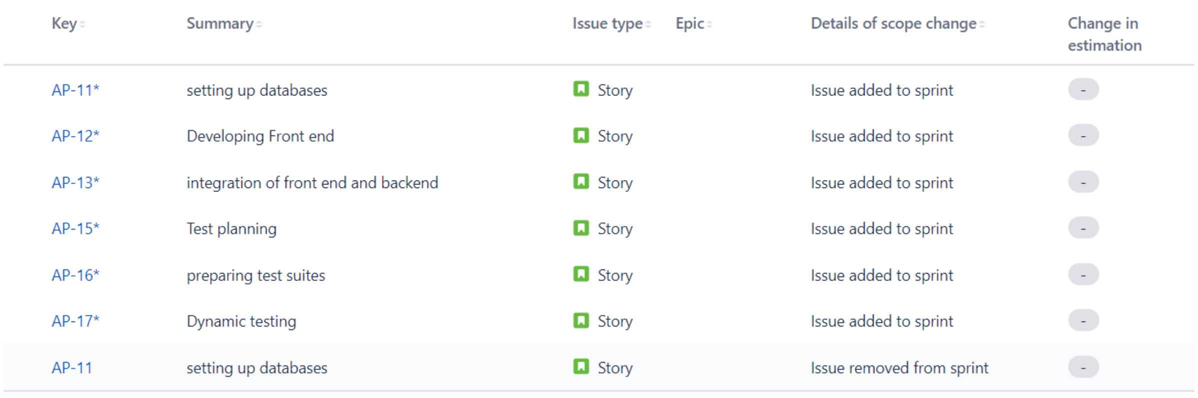
✓ To Do ✓ In Progress ✓ Done

Report: AP Sprint 3

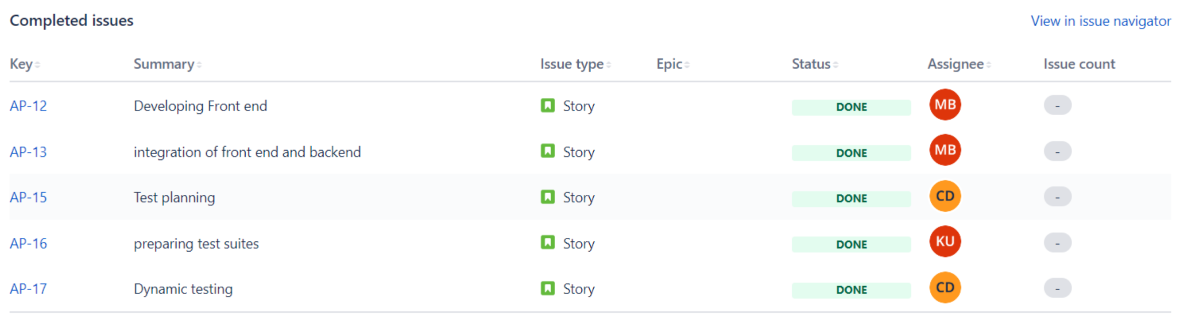
*Issue added after sprint start

Scope changes log

View in issue navigator



Key	Summary	Issue type	Epic	Details of scope change	Change in estimation
AP-11*	setting up databases	Story		Issue added to sprint	-
AP-12*	Developing Front end	Story		Issue added to sprint	-
AP-13*	integration of front end and backend	Story		Issue added to sprint	-
AP-15*	Test planning	Story		Issue added to sprint	-
AP-16*	preparing test suites	Story		Issue added to sprint	-
AP-17*	Dynamic testing	Story		Issue added to sprint	-
AP-11	setting up databases	Story		Issue removed from sprint	-



Completed issues

View in issue navigator

Key	Summary	Issue type	Epic	Status	Assignee	Issue count
AP-12	Developing Front end	Story		DONE	MB	-
AP-13	integration of front end and backend	Story		DONE	MB	-
AP-15	Test planning	Story		DONE	CD	-
AP-16	preparing test suites	Story		DONE	KU	-
AP-17	Dynamic testing	Story		DONE	CD	-

Gihub link:

https://github.com/bhargav2650/SE-PROJECT-employee-appraisal-system/blob/main/README.md