SE UNIT 4 ASSIGNMENT

TEAM 9

SUMANTH.C(PES1UG20CS711)

ROHAN T.P (PES1UG20CS682)

VINOD KUMAR E(PES1UG20CS702)

NIRANJAN T S (PES1UG20CS709)

Problem Statement – 1: Unit Testing

A unit is the smallest block of code that functions individually. The first

level of testing is Unit testing and this problem statement is geared

towards the same.

• Discuss with your teammates and demarcate units in your code

base

o Note: discuss why the code snippet you have chosen can

be classified as a unit

• Develop test cases for both valid and invalid data

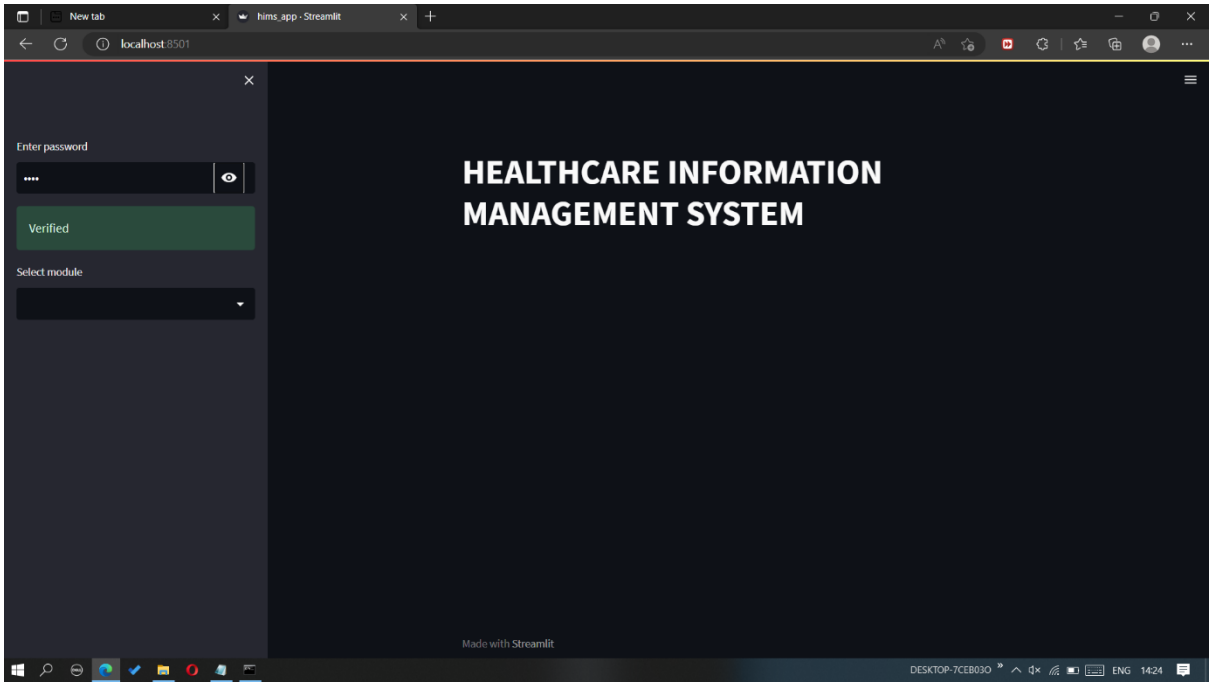
• Ideate how you could further modularize larger blocks of code

into compact units with your teammates

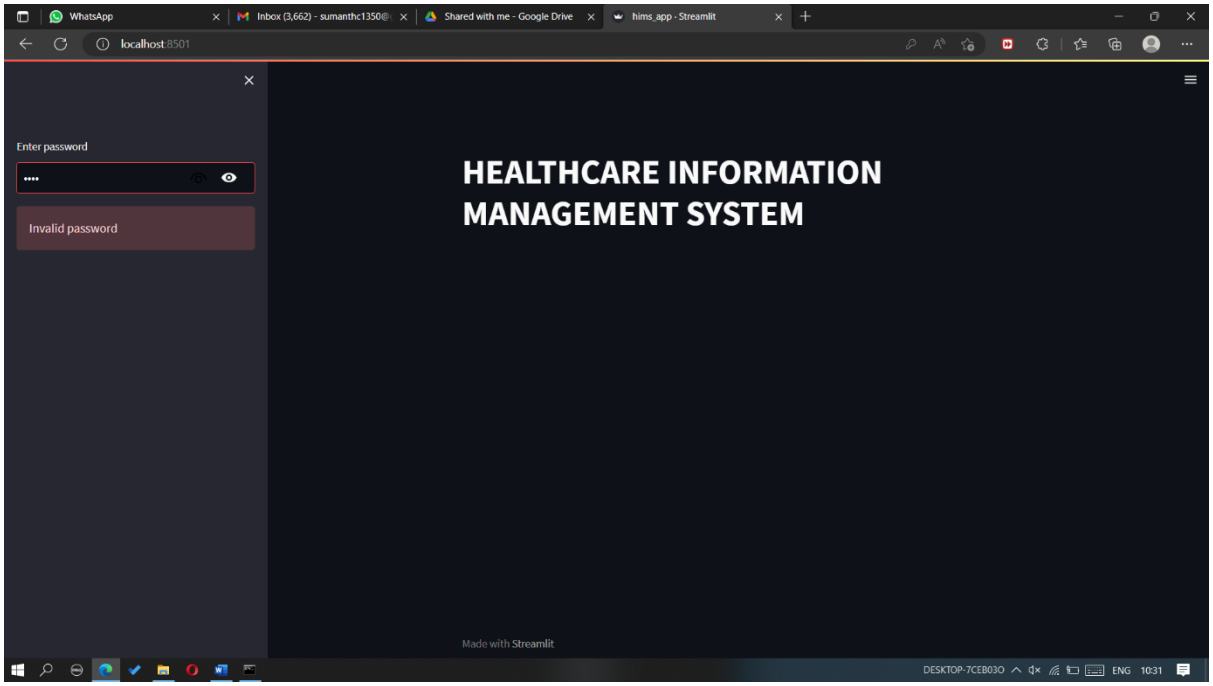
OUTPUT:

1.

Valid case:

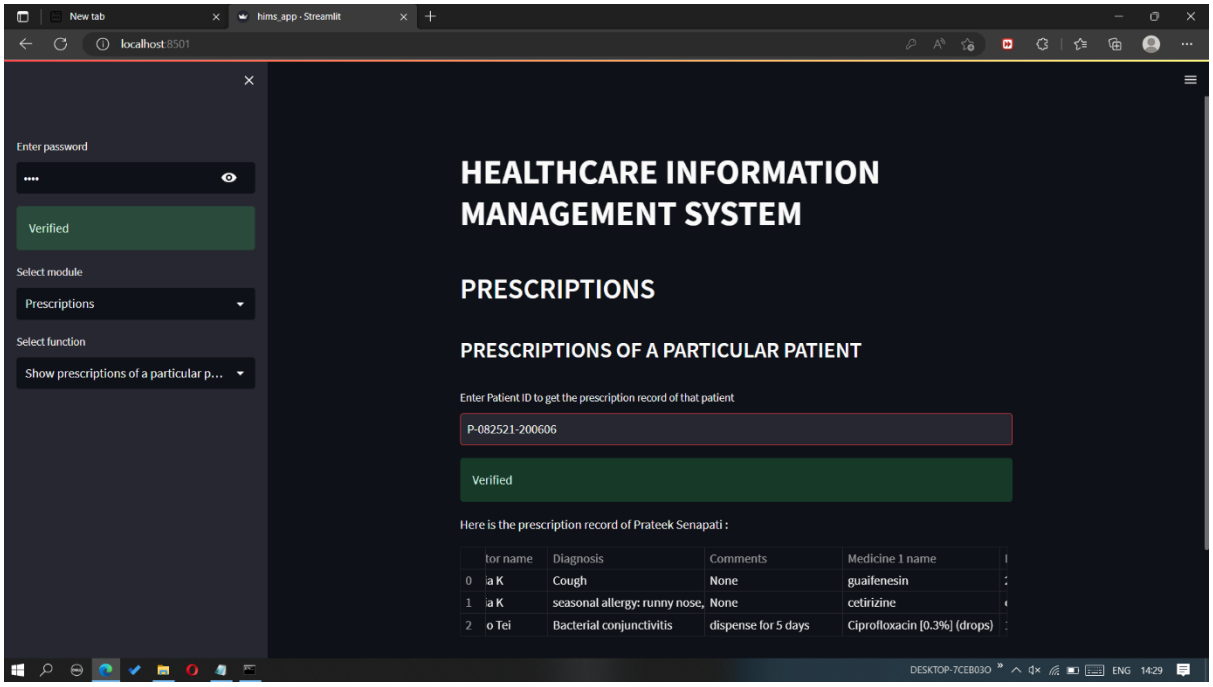


Invalid case:

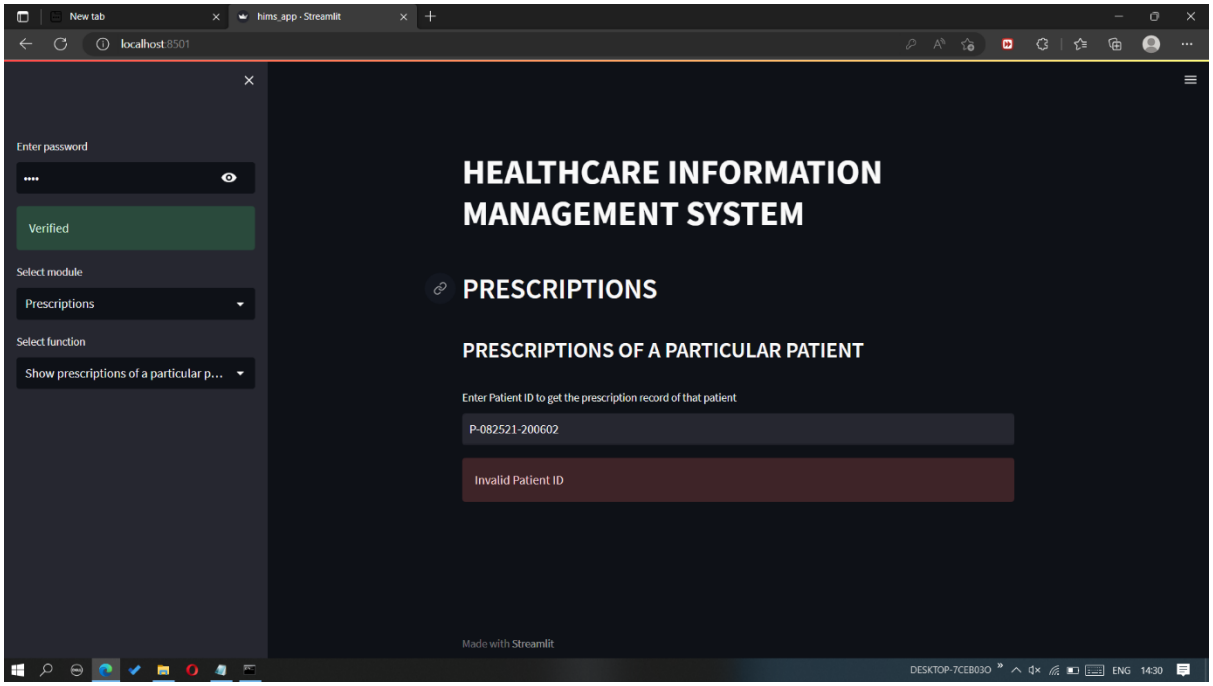


2.

Valid case:



Invalid case:



Problem Statement – 3: Static Testing

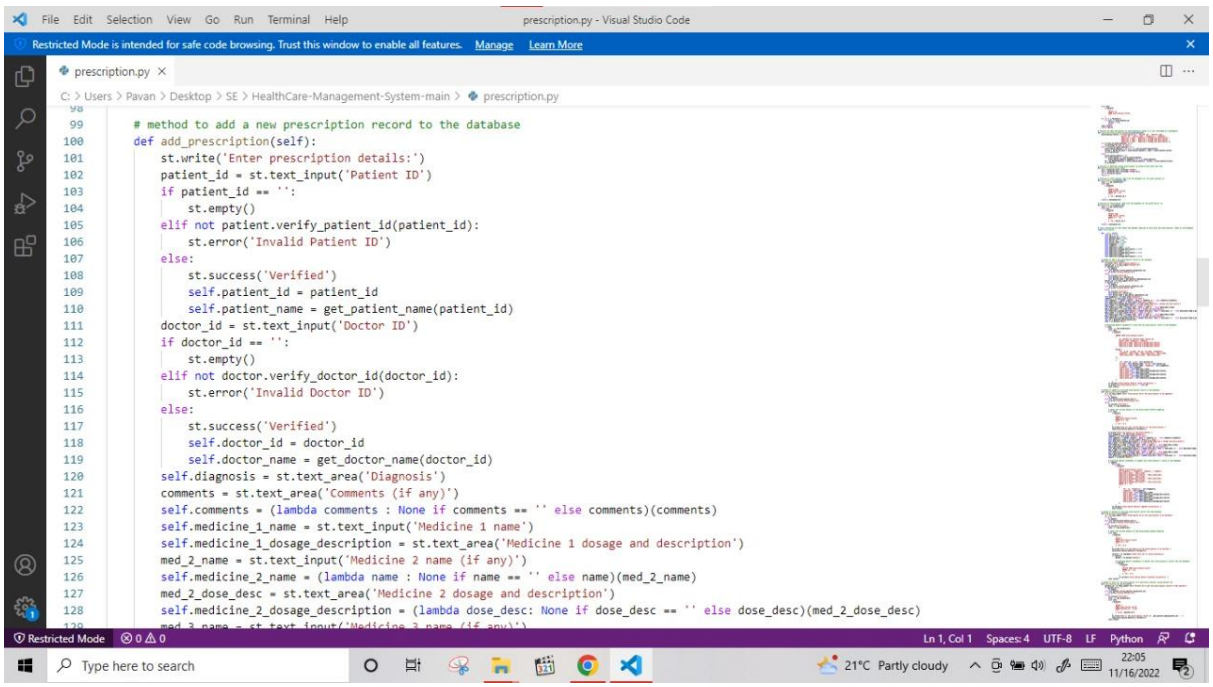
Static testing involves validating your code without any execution. Under this problem statement, you will be expected to analyse and calculate the cyclomatic complexity of your code.

• Using the unit you selected in the first problem statement as an example, develop the control flow graph of your problem statement.

• Using the Control flow graph, calculate the cyclomatic complexity of your code.

• Using the cyclomatic complexity as an indicator, Ideate and code your unit again to reduce complexity

OUTPUT:



File Edit Selection View Go Run Terminal Help prescription.py - Visual Studio Code

Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

prescription.py

C: > Users > Pavan > Desktop > SE > HealthCare-Management-System-main > prescription.py

```python
        self.comments = (lambda comments : None if comments == '' else comments)(comments)
        self.medicine_1_name = st.text_input('Medicine 1 name')
        self.medicine_1_dosage_description = st.text_area('Medicine 1 dosage and description')
        med_2_name = st.text_input('Medicine 2 name (if any)')
        self.medicine_2_name = (lambda name : None if name == '' else name)(med_2_name)
        med_2_dose_desc = st.text_area('Medicine 2 dosage and description')
        self.medicine_2_dosage_description = (lambda dose_desc: None if dose_desc == '' else dose_desc)(med_2_dose_desc)
        med_3_name = st.text_input('Medicine 3 name (if any)')
        self.medicine_3_name = (lambda name : None if name == '' else name)(med_3_name)
        med_3_dose_desc = st.text_area('Medicine 3 dosage and description')
        self.medicine_3_dosage_description = (lambda dose_desc: None if dose_desc == '' else dose_desc)(med_3_dose_desc)
        self.id = generate_prescription_id()
        save = st.button('Save')

        # executing SQLite statements to save the new prescription record to the database
        if save:
            conn, c = db.connection()
            with conn:
                c.execute(
                    """
                    INSERT INTO prescription_record
                    (
                        id, patient_id, patient_name, doctor_id,
                        doctor_name, diagnosis, comments,
                        medicine_1_name, medicine_1_dosage_description,
                        medicine_2_name, medicine_2_dosage_description,
                        medicine_3_name, medicine_3_dosage_description
                    )
                    VALUES (
                        :id, :p_id, :p_name, :dr_id, :dr_name, :diagnosis,
                        :comments, :med_1_name, :med_1_dose_desc, :med_2_name,
```

Restricted Mode ⊗ 0 ⚠ 0 Ln 1, Col 1 Spaces: 4 UTF-8 LF Python

Type here to search 21°C Partly cloudy 22:08 11/16/2022

File Edit Selection View Go Run Terminal Help prescription.py - Visual Studio Code

Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

prescription.py

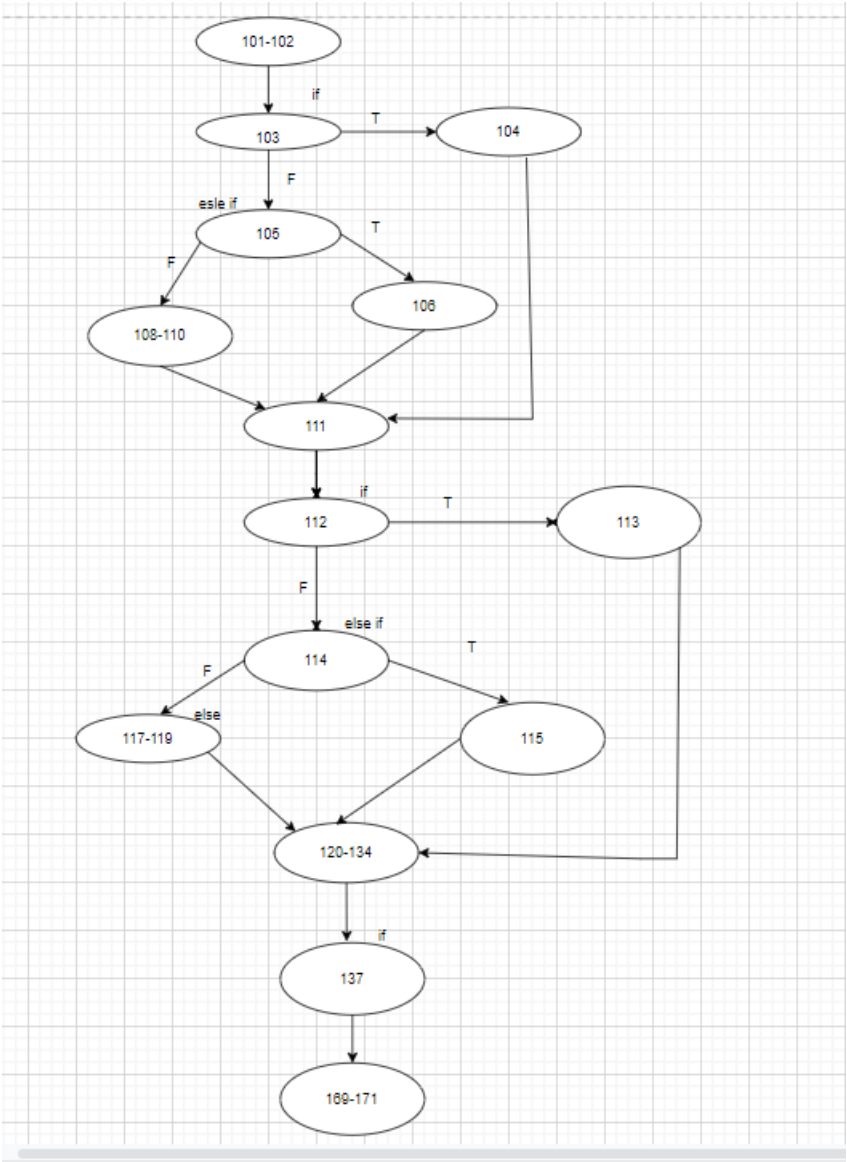
C: > Users > Pavan > Desktop > SE > HealthCare-Management-System-main > prescription.py

```python
                    (
                        id, patient_id, patient_name, doctor_id,
                        doctor_name, diagnosis, comments,
                        medicine_1_name, medicine_1_dosage_description,
                        medicine_2_name, medicine_2_dosage_description,
                        medicine_3_name, medicine_3_dosage_description
                    )
                    VALUES (
                        :id, :p_id, :p_name, :dr_id, :dr_name, :diagnosis,
                        :comments, :med_1_name, :med_1_dose_desc, :med_2_name,
                        :med_2_dose_desc, :med_3_name, :med_3_dose_desc
                    );
                    """,
                    {
                        'id': self.id, 'p_id': self.patient_id,
                        'p_name': self.patient_name, 'dr_id': self.doctor_id,
                        'dr_name': self.doctor_name, 'diagnosis': self.diagnosis,
                        'comments': self.comments,
                        'med_1_name': self.medicine_1_name,
                        'med_1_dose_desc': self.medicine_1_dosage_description,
                        'med_2_name': self.medicine_2_name,
                        'med_2_dose_desc': self.medicine_2_dosage_description,
                        'med_3_name': self.medicine_3_name,
                        'med_3_dose_desc': self.medicine_3_dosage_description,
                    }
                )
            st.success('Prescription details saved successfully.')
            st.write('The Prescription ID is: ', self.id)
            conn.close()

    # method to update an existing prescription record in the database
```

Restricted Mode ⊗ 0 ⚠ 0 Ln 1, Col 1 Spaces: 4 UTF-8 LF Python

Type here to search 21°C Partly cloudy 22:09 11/16/2022



CYCLOMETRIC COMPLEXITY=e-n+2p

=5.

The given function can be further modularised by breaking it into two functions ,which can reduce cyclomatic complexity.