SLEEP CYCLE ANALYZER SYSTEM

PRESENTATION

Presentation by : Group 8

OUR TEAM

JELIZA JUSTINE A/P SEBASTIN - A21EC0034

VIBHUSHA A/P SAMPASIVA RAO - A23CS0194

YEO WERN MIN - A23CS0285

NUR SYAKIRAH ADILAH BINTI AZRI - A23CS0159

INTRODUCTION

The Sleep Cycle Analyzer System is a comprehensive tool designed to help individuals monitor and improve their sleep quality. This system is a C++-based project that provides a user-friendly interface for users to input their sleep data, analyze their sleep patterns, and receive personalized recommendations for improvement.

Purposes

1. To provide users with a platform to track and monitor their sleep patterns.
2. To analyze sleep data and provide insights into sleep quality, duration, and stages (deep sleep, light sleep, and REM sleep).
3. To offer personalized recommendations for improving sleep quality based on the user's sleep data.
4. To educate users on the importance of sleep and provide resources for better sleep habits.

Input

1. Personal information: name, age, weight, and height.
2. Sleep session data: session ID, date, start time, end time, and sleep data entries (timestamp, duration, and estimated sleep stages).

Output

1. **Sleep Report**: A comprehensive report summarizing the user's sleep patterns, including average sleep duration, deep sleep, light sleep, and REM sleep percentages.
2. **Sleep Quality Assessment**: A score indicating the user's overall sleep quality, along with a description of their sleep quality.
3. **Improvement Plan**: Personalized recommendations for improving sleep quality, including suggestions for sleep duration, sleep stages, and lifestyle changes.

MODULE 1 : SLEEP DATA

1 Class SleepData

The SleepData class is designed to represent and manage data related to an individual's sleep patterns.

Private members:
timestamp, duration,
deepSleepDuration,lightSleepDuration
REMduration

Public members:
SleepData (constructor),
getTimestamp, getDuration,
getDeepSleepDuration,
getLightSleepDuration,
getREMduration

```cpp
//Module 1 :  Sleep Data
// Sleep Data Class
class SleepData {
private:
    string timestamp;
    double duration;
    double deepSleepDuration;
    double lightSleepDuration;
    double REMduration;

public:
    SleepData(string t, double d, double ds, double ls, double rem)
        : timestamp(t), duration(d), deepSleepDuration(ds), lightSleepDuration(ls), REMduration(rem) {}

    string getTimestamp() const { return double SleepData::duration
    double getDuration() const { return duration; }
    double getDeepSleepDuration() const { return deepSleepDuration; }
    double getLightSleepDuration() const { return lightSleepDuration; }
    double getREMduration() const { return REMduration; }
};
```

Encapsulation:
The SleepData class encapsulates sleep-related data with private member variables and provides public methods to access these variables, ensuring that the internal state of the object can only be modified through its methods,

MODULE 1 : SLEEP DATA

2 Class SleepSession

The SleepSession class is designed for managing and analyzing sleep data that spans multiple segments within a single sleep period

Private members:
sessionId, startTime, endTime, sleepData

Public members:
SleepSession (constructor), getSessionDuration, getSleepData

```cpp
// Sleep Session Class
class SleepSession {
private:
    string sessionId;
    string startTime;
    string endTime;
    vector<SleepData> sleepData;

public:
    SleepSession(string id, string start, string end, vector<SleepData> data)
        : sessionId(id), startTime(start), endTime(end), sleepData(data) {}

    double getSessionDuration() const {
        double totalDuration = 0;
        for (const auto& data : sleepData) {
            totalDuration += data.getDuration();
        }
        return totalDuration;
    }

    vector<SleepData> getSleepData() const { return sleepData; }
};
```

Composition:
The SleepSession class contains a vector of SleepData objects. This indicates that a sleep session is composed of multiple sleep data entries, but SleepData objects can exist independently of SleepSession.

MODULE 2 : SLEEP ANALYSIS

- analyze and assess sleep data, providing insights into the quality of sleep based on various parameters such as deep sleep, light sleep, and REM sleep.

1 class SleepQualityAssessment

Encapsulation
- SleepQualityAssessment: Encapsulates the qualityScore and description attributes and provides getter methods to access these values.

- To represent the quality of sleep based on a computed score and a corresponding description.
- Focuses solely on representing the quality of sleep.

- Constructor: Initializes the qualityScore and description.
- getQualityScore(): Returns the quality score.
- getDescription(): Returns the description of the sleep quality.

```
//Module 2 : Sleep Analysis
// Sleep Quality Assessment Class
class SleepQualityAssessment {
private:
    double qualityScore;
    string description;

public:
    SleepQualityAssessment(double score, string desc)
        : qualityScore(score), description(desc) {}

    double getQualityScore() const { return qualityScore; }
    string getDescription() const { return description; }
};
```

MODULE 2 : SLEEP ANALYSIS

2

class SleepCycleAnalyzer

- To analyze sleep sessions and assess the overall quality of sleep.
- Focuses on analyzing sleep sessions and computing sleep quality

- analyzeSleep(const vector<SleepSession>& sessions): Analyzes sleep sessions and prints the duration of each session
- qualityAssessment(const vector<SleepSession>& sessions) const: Computes and returns a SleepQualityAssessment object based on the quality of sleep data across multiple sessions.

Encapsulation

- SleepCycleAnalyzer: Encapsulates the logic for analyzing sleep data and assessing sleep quality within its methods.

Composition

- The SleepCycleAnalyzer class uses the SleepQualityAssessment class to return the results of its analysis, demonstrating the "has-a" relationship.

```cpp
// Sleep Cycle Analyzer Class
class SleepCycleAnalyzer {
public:
    void analyzeSleep(const vector<SleepSession>& sessions) {
        cout << "Analyzing sleep data..." << endl;
        for (const auto& session : sessions) {
            cout << "Session Duration: " << fixed << setprecision(2) << session.getSessionDuration() << " hours" << endl;
        }
    }

    SleepQualityAssessment* qualityAssessment(const vector<SleepSession>& sessions) const {
        double totalScore = 0;
        int dataCount = 0;

        for (const auto& session : sessions) {
            for (const auto& data : session.getSleepData()) {
                double sessionQuality = 0;
                sessionQuality += data.getDeepSleepDuration() * 1.5;
                sessionQuality += data.getLightSleepDuration();
                sessionQuality += data.getREMduration() * 1.2;
                totalScore += sessionQuality;
                ++dataCount;
            }
        }

        double averageScore = (dataCount > 0) ? (totalScore / dataCount) : 0;
        string description;

        if (averageScore >= 8.0) {
            description = "Excellent";
        } else if (averageScore >= 6.0) {
            description = "Good";
        } else if (averageScore >= 4.0) {
            description = "Fair";
        } else {
            description = "Poor";
        }

        return new SleepQualityAssessment(averageScore, description);
    }
};
```

MODULE 3 : SLEEP SUMMARY

1 Class Sleep Report

The SleepReport class generates a comprehensive report based on the user's sleep data

It has private member variables such as reportDate, summary, and recommendations.

The class also has methods like generateSummary, generateRecommendations, and printReport.

The SleepReport class encapsulates the report data and provides methods to generate and print the report, demonstrating the concept of encapsulation.

```
class SleepReport {
private:
    string reportDate;
    string summary;
    string recommendations;

public:
    SleepReport() : reportDate("2024-06-01") {}

    void generateSummary(const vector<SleepSession>& sessions) {
        summary = "Summary of sleep report:\n";
        double totalDuration = 0;
        double totalDeepSleep = 0;
        double totalLightSleep = 0;
        double totalREM = 0;

        for (const auto& session : sessions) {
            for (const auto& data : session.getSleepData()) {
                totalDuration += data.getDuration();
                totalDeepSleep += data.getDeepSleepDuration();
                totalLightSleep += data.getLightSleepDuration();
                totalREM += data.getREMduration();
            }
        }

        double averageDuration = totalDuration / sessions.size();
        double averageDeepSleep = totalDeepSleep / sessions.size();
        double averageLightSleep = totalLightSleep / sessions.size();
        double averageREM = totalREM / sessions.size();
        summary += "Average Sleep Duration: " + to_string(averageDuration) + " hours\n";
        summary += "Average Deep Sleep Duration: " + to_string(averageDeepSleep) + " hours\n";
        summary += "Average Light Sleep Duration: " + to_string(averageLightSleep) + " hours\n";
        summary += "Average REM Duration: " + to_string(averageREM) + " hours\n";
    }

    void generateRecommendations(const vector<SleepSession>& sessions) {
        recommendations = "Recommendations for better sleep:\n";

        double totalDuration = 0;
        for (const auto& session : sessions) {
            totalDuration += session.getSessionDuration();
        }
        double averageDuration = totalDuration / sessions.size();

        if (averageDuration < 7.0) {
            recommendations += "- Aim for at least 7 hours of sleep per night.\n";
        } else if (averageDuration > 9.0) {
            recommendations += "- Aim for 7-9 hours of sleep per night. Sleeping more than 9 hours can lead to
                sleep inertia and negatively impact your daily routine.\n";
        } else {
            recommendations += "- Your sleep duration is within the recommended range of 7-9 hours. Keep up the
                good habits!\n";
        }


        void printReport(const vector<SleepSession>& sessions) {
            generateSummary(sessions);
            generateRecommendations(sessions);
            cout << "Sleep Report - " << reportDate << endl;
            cout << summary << endl;
            cout << recommendations << endl;
        }

        void printRecommendations() {
            cout << recommendations << endl;
        }
};
```

MODULE 3 : SLEEP SUMMARY

2 Class Suggestion & ImprovementPlan

The Suggestion class is an abstract base class that provides a pure virtual method provideSuggestion to be implemented by derived classes

The ImprovementPlan class is a derived class of Suggestion and has a private member variable planDetails to store the details of the improvement plan

The class has a constructor to initialize plan details and a method createPlan to generate a customized improvement plan based on user's sleep quality assessment.

```cpp
// Forward declaration of User class
class User;

class Suggestion {
protected:
    string suggestionText;

public:
    virtual string provideSuggestion() const = 0;
};

class ImprovementPlan : public Suggestion {
private:
    string planDetails;

public:
    ImprovementPlan(string details) : planDetails(details) {}

    void createPlan(const User& user);

    string provideSuggestion() const override {
        return "Improvement Plan: " + planDetails;
    }
};
```

The ImprovementPlan class inherits from the Suggestion class, demonstrating the concept of inheritance.

The createPlan method is a standalone function that can be accessed through the provideSuggestion method, demonstrating the concept of polymorphism.

MODULE 3 : SLEEP SUMMARY

3 ImprovementPlan : CreatePlan

The createPlan method is a standalone function that:

- Analyzes the user's sleep quality assessment, sleep duration, and sleep stages
- Identifies areas for improvement (e.g. sleep duration, sleep quality, etc.)
- Provides personalized recommendations for improvement (e.g. establishing a consistent sleep schedule, creating a relaxing bedtime routine, etc.)
- Generates a customized improvement plan based on the user's specific needs and goals

```
void ImprovementPlan::createPlan(const User& user){
    const SleepQualityAssessment* assessment = user.sleepQualityAssessment;

    if (assessment) {
        double score = assessment->getQualityScore();
        planDetails = "Generated Plan based on your sleep quality assessment:\n";

        // Suggestions based on overall sleep quality score
        if (score < 4.0) {
            planDetails += "- Your sleep quality score is low. Consider consulting a healthcare provider for
                potential sleep disorders such as insomnia or sleep apnea.\n";
            planDetails += "- Focus on improving sleep quality through relaxation techniques such as
                meditation or deep breathing exercises before bedtime to improve your score.\n";
        } else if (score < 6.0) {
            planDetails += "- Your sleep quality score is fair. Focus on improving sleep quality through
                relaxation techniques such as meditation or deep breathing exercises before bedtime to
                improve your score.\n";
        } else {
            planDetails += "- Your sleep quality score is good. Keep up the good habits!\n";
        }

        // Suggestions based on sleep duration
        double totalDuration = 0;
        for (const auto& session : user.sleepSessions) {
            for (const auto& data : session.getSleepData()) {
                totalDuration += data.getDuration();
            }
        }

        double averageSleepDuration = totalDuration / user.sleepSessions.size();
        planDetails += "- Your average sleep duration is " + to_string(averageSleepDuration) + " hours.\n";
        if (averageSleepDuration < 7.0) {
            planDetails += "- Aim for at least 7-9 hours of sleep per night to improve overall sleep quality
```

```
    } else {
        planDetails += "- Your average sleep duration is excellent. Keep up the good habits!\n";
    }

    // Suggestions based on sleep stages
    double totalDeepSleep = 0;
    double totalLightSleep = 0;
    double totalREM = 0;
    double totalSleep = 0;

    for (const auto& session : user.sleepSessions) {
        for (const auto& data : session.getSleepData()) {
            totalDeepSleep += data.getDeepSleepDuration();
            totalLightSleep += data.getLightSleepDuration();
            totalREM += data.getREMduration();
            totalSleep += data.getDuration();
        }
    }

    double deepSleepPercentage = (totalDeepSleep / totalSleep) * 100;
    double lightSleepPercentage = (totalLightSleep / totalSleep) * 100;
    double REMPercentage = (totalREM / totalSleep) * 100;

    planDetails += "- Your average deep sleep percentage is " + to_string(deepSleepPercentage) + "%.\n";
    if (averageSleepDuration < 7.0) {
        planDetails += "  - It's difficult to get quality sleep with such a short sleep duration. Aim for
            at least 7-9 hours of sleep per night to improve overall sleep quality.\n";
    } else if (deepSleepPercentage < 20 ||deepSleepPercentage > 25) {
        planDetails += "  - Consider creating a consistent sleep schedule, reducing caffeine intake, and
            ensuring a dark, quiet, and cool sleep environment to improve deep sleep.\n";
        planDetails += "  - Learn relaxation techniques such as progressive muscle relaxation,
            mindfulness meditation, or deep breathing exercises from websites like <https://www.headspace
            .com/> or <https://www.calm.com/> to improve deep sleep.\n";
        } else {
            planDetails += "  - Your REM sleep percentage is good. Keep up the good habits!\n";
        }
    }
```

The OO concepts employed in the program include: Encapsulation , Inheritance, Polymorphism , Composition Friendship

The ImprovementPlan class is a friend class of the User class, demonstrating the concept of friendship

```
friend class ImprovementPlan;
```

MODULE 4 : USER

1 Module 4 Purpose

Manage User Profiles and Sleep Data:

- handle all functionalities related to user profile
- manages sleep data including adding, updating, and storing sleep sessions for each user

Perform Sleep Analysis and Generate Reports:

- generate reports to provide insights into user's sleep patterns based on analysis that the system has made

Provide Personalized Improvement Suggestions:

- generate tailored suggestions for improving sleep quality based on user data

User Class Overview:

- manage all functionalities related to user profiles, sleep analysis, report generation, and personalized suggestions

MODULE 4 : USER

```
class User {
private:
    string userID;
    string name;
    int age;
    double weight;
    double height;
    vector<SleepSession> sleepSessions;
    SleepReport* sleepReport;
    SleepQualityAssessment* sleepQualityAssessment;
    vector<Suggestion*> suggestions;
    SleepCycleAnalyzer sleepCycleAnalyzer;

public:
    User(string id, string n, int a, double w, double h)
        : userID(id), name(n), age(a), weight(w), height(h),
        sleepQualityAssessment(nullptr), sleepReport(nullptr) {}

    void updateProfile(string n, int a, double w, double h) {
        name = n;
        age = a;
        weight = w;
        height = h;
    }

    string getName() const { return name; }
    int getAge() const { return age; }
    double getWeight() const { return weight; }
    double getHeight() const { return height; }
```

```
public:
    void generateReport() {
     if (sleepReport) {
         delete sleepReport;
     }
     sleepReport = new SleepReport();
     sleepReport->printReport(sleepSessions);
     qualityAssessment();
}


    void displaySleepQuality() const {
        if (sleepQualityAssessment) {
            cout << "Sleep Quality Score: " << fixed << setprecision(2)
            << sleepQualityAssessment->getQualityScore() << endl;
            cout << "Description: " << sleepQualityAssessment->getDescription() << endl;
        }
    }

    void displaySuggestions() const {
        for (const auto& suggestion : suggestions) {
            cout << suggestion->provideSuggestion() << endl;
        }
    }

    ~User() {
        if (sleepQualityAssessment) {
            delete sleepQualityAssessment;
        }
```

MODULE 4 : USER

```
        for (auto suggestion : suggestions) {
            delete suggestion;
        }
        delete sleepReport;
    }

    friend class ImprovementPlan;
};

void ImprovementPlan::createPlan(const User& user){
    const SleepQualityAssessment* assessment = user.sleepQualityAssessment;

    if (assessment) {
        double score = assessment->getQualityScore();
        planDetails = "Generated Plan based on your sleep quality assessment:\n";

        // Suggestions based on overall sleep quality score
        if (score < 4.0) {
            planDetails += "- Your sleep quality score is low. Consider consulting a healthcare provider for potential
            sleep disorders such as insomnia or sleep apnea.\n";
            planDetails += "- Focus on improving sleep quality through relaxation techniques such as meditation or deep
            breathing exercises before bedtime to improve your score.\n";
        } else if (score < 6.0) {
            planDetails += "- Your sleep quality score is fair. Focus on improving sleep quality through relaxation techniques
            such as meditation or deep breathing exercises before bedtime to improve your score.\n";
        } else {
            planDetails += "- Your sleep quality score is good. Keep up the good habits!\n";
        }
```

```
// Suggestions based on sleep duration
double totalDuration = 0;
for (const auto& session : user.sleepSessions) {
    for (const auto& data : session.getSleepData()) {
        totalDuration += data.getDuration();
    }
}

double averageSleepDuration = totalDuration / user.sleepSessions.size();
planDetails += "- Your average sleep duration is " + to_string(averageSleepDuration) + " hours.\n";
if (averageSleepDuration < 7.0) {
    planDetails += "- Aim for at least 7-9 hours of sleep per night to improve overall sleep quality.\n";
} else {
    planDetails += "- Your average sleep duration is excellent. Keep up the good habits!\n";
}

// Suggestions based on sleep stages
double totalDeepSleep = 0;
double totalLightSleep = 0;
double totalREM = 0;
double totalSleep = 0;
```

MODULE 4 : USER

```
for (const auto& session : user.sleepSessions) {
    for (const auto& data : session.getSleepData()) {
        totalDeepSleep += data.getDeepSleepDuration();
        totalLightSleep += data.getLightSleepDuration();
        totalREM += data.getREMduration();
        totalSleep += data.getDuration();
    }
}

double deepSleepPercentage = (totalDeepSleep / totalSleep) * 100;
double lightSleepPercentage = (totalLightSleep / totalSleep) * 100;
double REMPercentage = (totalREM / totalSleep) * 100;

planDetails += "- Your average deep sleep percentage is " + to_string(deepSleepPercentage) + "%.\n";
if (averageSleepDuration < 7.0) {
    planDetails += "  - It's difficult to get quality sleep with such a short sleep duration. Aim for at least
    7-9 hours of sleep per night to improve overall sleep quality.\n";
} else if (deepSleepPercentage < 20 ||deepSleepPercentage > 25) {
    planDetails += "  - Consider creating a consistent sleep schedule, reducing caffeine intake, and ensuring a
    dark, quiet, and cool sleep environment to improve deep sleep.\n";
    planDetails += "  - Learn relaxation techniques such as progressive muscle relaxation, mindfulness meditation,
    or deep breathing exercises from websites like <https://www.headspace.com/> or <https://www.calm.com/> to improve deep sleep.\n";
} else {
    planDetails += "  - Your deep sleep percentage is good. Keep up the good habits!\n";
}
```

```
        planDetails += "- Your average light sleep percentage is " + to_string(lightSleepPercentage) + "%.\n";
        if (averageSleepDuration < 7.0) {
            planDetails += "  - It's difficult to get quality sleep with such a short sleep duration. Aim for at least 7-9 hours of sleep
            per night to improve overall sleep quality.\n";
        } else if (lightSleepPercentage < 50) {
            planDetails += "  - Consider engaging in regular physical activity, maintaining a comfortable and supportive mattress and pillows,

             and learning about sleep-conducive habits and routines to improve light sleep.\n";
            planDetails += "  - Learn about the importance of light sleep and how to improve it from websites like <https://www.sleepfoundation.org/>
            or <https://www.sleep.org/>.\n";
        } else {
            planDetails += "  - Your light sleep percentage is good. Keep up the good habits!\n";
        }

         planDetails += "- Your average REM sleep percentage is " + to_string(REMPercentage) + "%.\n";
        if (averageSleepDuration < 7.0) {
            planDetails += "  - It's difficult to get quality sleep with such a short sleep duration. Aim for at least 7-9 hours of sleep per
             night to improve overall sleep quality.\n";
        } else if (REMPercentage < 20 || REMPercentage > 25 ) {
            planDetails += "  - Consider reducing stress through mindfulness practices, avoiding alcohol and heavy meals before bedtime,
             and learning about stress-reducing techniques and sleep tips to improve REM sleep.\n";
            planDetails += "  - Learn about the importance of REM sleep and how to improve it from websites like <https://www.verywellmind.com/>
            or <https://www.healthline.com/>.\n";
        } else {
            planDetails += "  - Your REM sleep percentage is good. Keep up the good habits!\n";
        }
    }
}
```

MODULE 4 : USER

2 Attributes and Methods

Attributes:

- **userID, name, age, weight, height**
- **SleepSessions:** a vector of SleepSessions objects representing the user's sleep sessions
- **SleepReport:** a pointer to a SleepReport object that stores the sleep report for the user
- **SleepQualityAssessment:** a pointer to a SleepQualityAssessment object that stores the sleep quality assessment for user
- **Suggestions:** a vector of pointers to Suggestion objects that store various suggestions for the user
- **SleepCycleAnalyzer:** an instance of the SleepCycleAnalyzer class used to analyze the sleep data of the user

Methods:

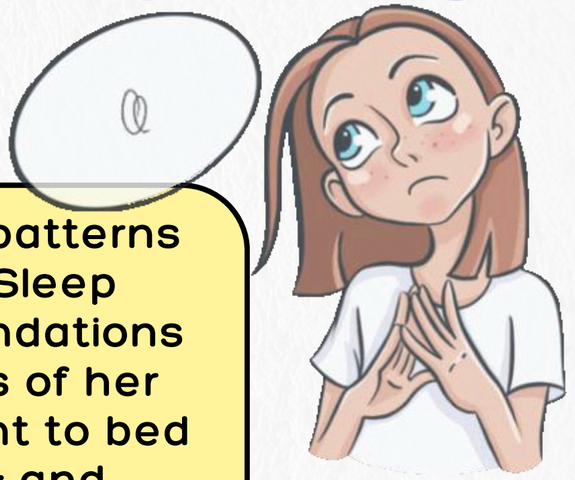
- **addSleepSession:** adds a sleep session
- **analyzeSleep:** analyzes sleep data
- **qualityAssement:** Assesses sleep quality
- **generateReport:** generates and prints a sleep report
- **generateImprovementPlan:** creates personalized improvement plans
- **displaySleepQuality:** displays sleep quality score and description
- **displaySuggestions:** displays personalized suggestions

MODULE 4 : USER

3 Object - Oriented Concepts

- Encapsulation:
  Bundling data and method in ‘User’ class
- Composition: ‘sleepSession’ ans
  ‘sleepCycleAnalyzer’ within ‘User’
- Aggregration: ‘SleepReport’ ,
  ‘SleepQualityAssessment’ ,vand ‘Suggestions’
  in ‘User’
- Array of Objects: vector manage 'sleepSession'
  objects

DEMONSTRATION



Emma, a 20-year-old individual, has been struggling with her sleep patterns lately. She is 160 cm tall and weighs 45 kg. She wants to use the Sleep Analyzer System to track her sleep and get personalized recommendations for improvement. To input her sleep data, Emma enters the details of her two recent sleep sessions: session1 on June 21, 2024, where she went to bed at 22:30 and woke up at 06:00, with a sudden awakening at 02:30; and session2 on June 22, 2024, where she went to bed at 23:00 and woke up at 07:00, with a sudden awakening at 03:45. She estimates that she slept for 7.0 hours in the first session, and spent 7.5 on bed, as she was awake for 30 mins after the sudden awakening and 8 hours sleep, and 8.5 hours, as she was also awake for 15 mins after the sudden awakening in the second session. Emma is eager to know her average sleep duration, sleep quality score, and personalized recommendations for improvement from the system. She wants to know if her sleep duration is within the recommended range, and if there are any areas she can improve on to get better sleep.

THANK YOU

Presented by : Group 8