

**CLOUD APPLICATION DEVELOPMENT**

**Kavya Khandelwal**

**500084423**

**BATCH B3 HONS**

**R214220608**

**SUBMITTED TO : HARVINDER SIR**

**WEEK -6**

**Thread programming:**

Multiple threads of code can run concurrently within a single process by using thread programming. A thread is a quick procedure that can run independently within another process, enabling the execution of numerous tasks at once. By utilising the capability of contemporary CPUs with many cores, thread programming enables programmers to design more effective and scalable code.

Each thread is operated independently in thread programming, and each thread has its own programme counter, stack, and registers. Since threads can share memory, they can access the same variables and data structures. This enables programmers to create programmes that can handle a variety of simultaneous operations, including computations, reading and writing to files, and interacting with network services.

Improved performance, greater resource utilisation, and more responsive applications are all advantages of thread programming. Concurrently running threads allow the CPU to be utilised more effectively, speeding up the processing of computationally demanding activities. The use of thread programming, which enables threads to wait for input or output activities without obstructing the CPU, can also increase resource utilisation.

However, there are difficulties with thread programming as well. Making sure that many threads don't access shared resources concurrently, which could cause data races or deadlocks, is one of the biggest challenges. When two or more threads attempt to access and alter the same memory address at the same time, a data race occurs and unexpected behaviour results. When multiple threads are waiting for each other to release a lock, a deadlock results. Deadlocks happen when two or more threads are awaiting the release of a shared resource, preventing any of the threads from moving further.

Thread programming needs careful design and implementation to address these issues. Threads can access shared resources securely and under control by using synchronisation methods like locks, semaphores, and condition variables. Data races and deadlocks can be prevented by employing thread-safe programming techniques including minimising shared mutable state and using immutable data structures.

In conclusion, thread programming is an effective method for creating scalable, concurrent applications. Thread programming has become crucial for maximising speed and resource utilisation as multi-core CPUs have grown in popularity. The necessity to ensure secure access to shared resources is one of the difficulties of thread programming. Thread programming can be a useful tool for developers aiming to increase the speed and scalability of their systems with careful design and implementation.

**Introduction:**

The importance of mental health has grown over the past several years. It is crucial to treat mental health issues as they are becoming more common due to the development of technology and social media. A person's daily life can be significantly impacted by mental health conditions like depression and anxiety, making it challenging to finish activities or uphold healthy relationships. This is where a monitoring tool for mental health can be useful.

Monitoring and managing mental health is the goal of a mental health monitoring application. This type of application can offer insights into a person's stress levels, sleep quality, and activity levels by utilising wearable technology and machine learning algorithms. When a person's stress levels exceed a predetermined level, it can also send notifications, letting them know they might be depressed. When necessary, the app can alert close friends or family so they can offer support and assistance.

A task-based application architecture must be chosen, the problem must be identified, the task dependencies must be established, and the project must be developed and implemented using cloud computing. All of these responsibilities are essential to the application's success and must be appropriately carried out.

Understanding the effects of mental health problems on a person's life and the need for a tool to monitor and manage mental health are necessary for identifying the issue.

By segmenting the application's development into manageable, standalone activities, the task dependencies can be identified. Selecting the best model to construct the programme's many components is part of choosing a task-based application model. The application will be hosted and managed by cloud-based services like Amazon Web Services (AWS) as part of the project development and implementation using cloud computing.

In general, a monitoring app for mental health can be a useful tool for people to track and manage their mental health. It can assist people in identifying possible problems and taking action before they become severe by offering insights regarding a person's stress levels, sleep quality, and activity levels. Close friends or family members may receive alerts via the app, enabling them to offer support and assistance as needed. Although careful preparation and execution are needed to create such an application, the potential advantages for people who are dealing with mental health problems are substantial.

**Identifying the Problem:**

An important area of public health that has been growing in importance globally is mental health. Approximately one in four people worldwide will experience mental or neurological illnesses at some point in their lives, according to the World Health Organisation. A person's relationships, productivity, and quality of life can all be impacted by mental health disorders. People are frequently reluctant to seek help for their mental health because of the stigma associated with it, and a lack of support networks can make matters worse. Additional stressors brought on by the COVID-19 pandemic include social isolation, unstable economic conditions, and health issues.

Our team saw the need for a mental health monitoring tool to help users keep track of their stress levels and receive notifications when they are experiencing poor moods in reaction to these difficulties.

We seek to raise awareness and lessen the stigma surrounding mental health concerns by giving consumers a simple tool to check their mental health.

**Determining Task Dependencies:**

We had to figure out all the different tasks that were necessary to make the mental health monitoring programme work before we could start constructing it. The duties were divided into four main parts:

1. Data Gathering: In order to deliver correct insights, the application must gather information about the user's stress levels. We made the decision to leverage wearable technology, such as smartwatches and fitness trackers, to gather information on the user's heart rate, level of exercise, and sleeping habits. This information will provide light on the user's stress levels.Data Processing: The collected data needs to be processed to generate meaningful insights and alerts. We decided to use machine learning algorithms to analyze the data and generate insights on the user's stress levels. Additionally, we created a system of alerts that would let the user's close friends know when they were feeling down.
2. Notification System: When a user is depressed, the app needs to alert their close friends. We made the decision to employ a push notification system that would notify the user's friends when their stress levels exceeded a predetermined level.
3. User Interface: The programme must have an intuitive user interface that enables users to track their levels of stress and view insights. We made the decision to create a smartphone application that people could use.

**Choosing a Task-Based Application Model:**

We choose to utilise a task-based application paradigm after determining the numerous tasks necessary to make the mental health monitoring application operational. This paradigm separates the application into various parts, each of which is in charge of a particular duty. This method enables scalability, easy maintenance, and modular development.

1. The first component is data collection.

The data collection element is in charge of gathering information on the user's stress levels. We made the decision to leverage wearable technology, such as smartwatches and fitness trackers, to gather information on the user's heart rate, level of exercise, and sleeping habits. This information will provide light on the user's stress levels. We selected these wearable gadgets because they are practical, reasonably priced, and capable of delivering precise data. We employed the device's sensors, including the heart rate monitor, accelerometer, and gyroscope, to gather the required data. The data was then processed and kept in a database. Since Amazon S3 is a scalable and affordable alternative for storing big volumes of data, we employed it to store and retrieve data.

2. Data Processing Component: This component processes the data that has been gathered to produce insights and warnings. In order to analyse the data and produce insights on the user's stress levels, we choose to apply machine learning techniques. We used a sizable dataset of stress levels and associated physiological characteristics, like heart rate variability and sleep quality, to train our machine learning models.

The gathered data is fed into our machine learning algorithms, which produce insights on the user's stress levels. Since AWS Lambda is a serverless computing technology that enables the scalable and economical processing of substantial amounts of data, we employed it to create the machine learning models.

1. Notification Component: When a person is depressed, the notification component informs their close friends. We made the decision to employ a push notification system that would notify the user's friends when their stress levels exceeded a predetermined level. Using Amazon SNS, a fully managed push notification service, we created the notification system.

When the user's stress levels exceed a predetermined threshold, the notification system is activated, letting them know they might be depressed. The system then sends a push notification to the user's close friends, alerting them to check on the user.

1. User Interface Component:

A user-friendly interface for tracking stress levels and viewing insights is provided by the user interface component. We made the decision to create a smartphone application that people could use. Since React Native makes it simple to create cross-platform mobile applications, we used it to design the user interface component.

Users can check their stress levels, get alerts, and view insights produced by the machine learning models thanks to the user interface component. A dashboard with numerous graphs and charts that show the user's stress levels over time, sleep quality, and exercise levels is provided by the application.

**Development and Implementation:**

We decided to create and build the mental health monitoring programme using cloud computing. The devices and technology we employed were as follows:

1. AWS Lambda: To create the serverless application components, we used AWS Lambda.

2. Amazon S3: To store and retrieve data, we made use of Amazon S3.

3. Amazon API Gateway: To give a RESTful API to the user interface component, we made use of Amazon API Gateway.

4. React Native: To create the user interface component, we used React Native.

**Conclusion:**

This study details the creation and deployment of a mental health monitoring app that tracks stress levels and notifies close friends when a user is having a bad day. In order to build and deploy the programme, we used cloud computing after identifying the issue, determining task dependencies, and selecting a task-based application model. Our product seeks to give customers a tool to maintain their mental health and contact their support network when necessary.