

**CLOUD APPLICATION DEVELOPMENT**

Kavya Khandelwal

500084423

R214220608

B3 Hons

**SUBMITTED TO : HARVINDER SIR**

**WEEK-9**

**Understanding thread APIs:**

An operating system or programming language's thread creation and management are handled by thread APIs, which are collections of functions and data structures. Developers can create, launch, stop, and manage threads in their programmes by using thread APIs.

The two primary categories of thread APIs are user-level and kernel-level. Thread management for user-level thread APIs is handled entirely in user space, independent of the operating system kernel. On the other hand, kernel-level thread APIs are implemented as a component of the operating system's kernel and offer additional control over thread scheduling and management.

Several well-liked thread APIs are:

1. Pthreads (POSIX threads): Pthreads is a common thread API for operating systems based on Unix. The ability to construct and manage threads is provided via a set of functions and data structures.

2. Windows API: On the Windows operating system, the Windows API offers a set of operations for managing threads. It has tools for initiating, stopping, synchronising, and establishing new threads.

3. Java Thread API: The Java Thread API is a tool for managing threads in Java programmes and is a component of the Java Development Kit (JDK). For constructing, beginning, stopping, and synchronising threads, it offers classes and methods.

4..NET Thread API: To construct and manage threads in.NET applications, the.NET Framework offers the Thread class. The Thread class includes methods for creating, starting, stopping, and synchronizing threads.

**Understanding 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. 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 concerns as they are becoming more common due to the development of technology and social media. A person's everyday life can be significantly impacted by mental health conditions like depression and anxiety, making it challenging to finish activities or uphold good relationships. This is where a monitoring tool for mental health might be useful.

Monitoring and managing mental health is the goal of a mental health monitoring application. This type of application can offer insights on 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 certain level, it can also send notifications, letting them know they could be depressed. When necessary, the app may alert close friends or family so they can give 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 utilising 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 may 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 may 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 serious by offering insights regarding a person's stress levels, sleep quality, and activity levels. Close friends or family members may receive notifications via the app, enabling them to give support and assistance as required. Although careful preparation and execution are needed to create such an application, the potential advantages for those 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 individuals worldwide may have mental or neurological illnesses at some time in their life, 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 treatment for their mental health because of the stigma associated with it, and a lack of support networks can make matters worse. Additional pressures 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 assist 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 around mental health concerns by giving consumers a simple tool to check their mental health.

**Objectives:**

The following are the goals of the suggested project:

• To create a cloud-based application that uses replies to a mood log quiz to determine a person's stress and mood levels.

• To offer a user-friendly interface so that they may input their data, evaluate their mood and stress levels, and get notifications.

• To notify a person's friends or support network when their mood or stress levels are low so they may give encouragement and assistance.

**Methodology:**

• The suggested application would be created on a cloud-based platform like Microsoft Azure or Amazon Web Services. An easy mood journal quiz will be used by the programme to assess the user's emotional condition. The test will offer a number of questions on the subject's stress levels, general well-being, and how they would rank their mood on a scale of 1 to 10.

• The programme will notify the user's friends or support network when their mood or stress levels are low. The notifications can be delivered by email, text message, or directly through the programme. The warnings will contain details about the person's emotional condition as well as advice on how to help them.

**Project Plan:**

**Phase 1 (two weeks): Planning and Design**

* Determine the target audience's demands and the project's goals, objectives, and scope.
* Create a prototype of the user interface (UI) and select the cloud platform and technology stack (Amazon Web Services or Microsoft Azure).Milestones:
* The UI design mockup has been created, and the cloud platform and technology stack have been selected.

**Development (Phase 2): 3–4 weeks**

• Create the application's architecture

• Execute the UI design

• Create the functionality for the mood log quiz

• Execute the algorithm to determine the person's mood and stress levels

• Integrate the notification system to notify friends or other support systems

• Create the database to store user data

• Thoroughly test the application.

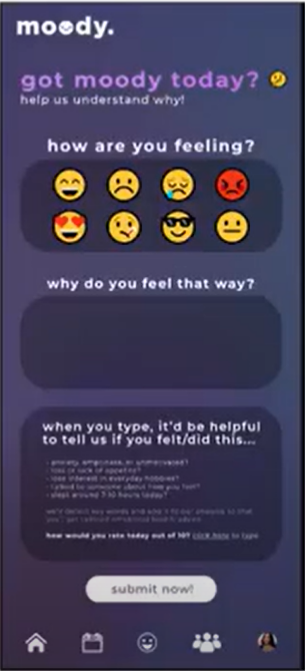
**Milestones:**

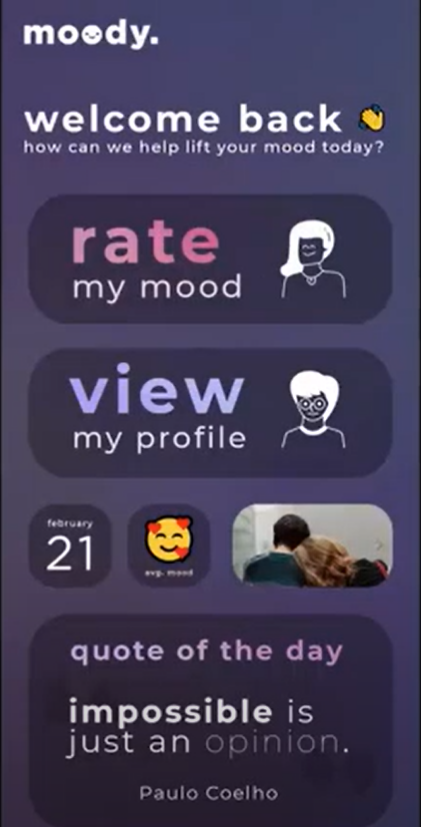
* The user interface has been built
* The functionality for the mood log quiz has been established.
* Implementation of a mood detection algorithm
* Integration of a notification system
* Completion of database construction
* Testing and deployment of the application.

**Functional Requirements:**

* User Registration: Users must be able to sign up for the application by giving a password, name, and email address.
* Mood Log Quiz: A mood log quiz that asks users to score their present mood on a scale of 1 to 10 must be included in the application.
* Mood Detection Algorithm: The programme has to contain an algorithm that examines the user's responses to the mood log quiz and determines whether they are happy, sad, nervous, or stressed.
* Notification System: The programme has to feature a notification system that notifies friends or other people who can help the user when they are feeling down or anxious. You may use push notifications, SMS, or email to create this functionality.
* Friend List Management: Users must be able to edit their friend list and add or delete people to whom alerts will be delivered.
* Dashboard: The programme needs a user-friendly dashboard that shows the user's stress levels, past moods, and trends. The alerts that have been sent to their friends should also be shown.
* Data Analytics: The programme must contain a data analytics component that offers perceptions into the inclinations and tendencies of the user. This will assist the user and others who are supporting them in comprehending the underlying causes of stress and poor mood.
* Security: The programme needs to be safe and safeguard the user's private information, including their name, email address, and password. Additionally, it must adhere to data protection laws like the CCPA and GDPR.
* Compatibility: The programme has to work with a variety of hardware and software, including web browsers, iOS, and Android. Additionally, it must to accommodate various regional and linguistic situations.

**Implementation:**







**Conclusion:**

In conclusion, we created a programme that tracks stress levels and notifies close pals when the user is having a bad day. We recognised the issue of mental health problems as well as the requirement for a tool to track and manage mental health. Data gathering, data processing, notification, and user interface were all identified as job dependencies. To create the various application components, we went with a task-based application paradigm, and we implemented the application using cloud computing. Our product seeks to give customers a tool to maintain their mental health and contact their support network when necessary.