Universal College of Engineering, Kaman

Department of Computer Engineering

Subject:Mobile Computing

Experiment No: 4

Aim: Implement an application that creates an alert upon receiving a message

Theory:

Implementing an Application that Creates an Alert upon Receiving a Message (Notification).

In mobile and desktop application development, it is crucial to inform the user about important events, updates, or messages in real time. An alert system or notification mechanism can be implemented within an application to ensure users are promptly notified whenever there is a new message or event that requires their attention. A notification system can trigger an alert, usually in the form of a pop-up message, sound, or icon update, to notify users of new information such as received messages, incoming calls, or application updates.

The process of implementing such an application involves integrating notification services with backend systems that can detect when a new message is received and trigger a corresponding notification. This is commonly achieved through integration with push notification services, such as Firebase Cloud Messaging (FCM), Apple Push otificationService (APNS), or other similar services, which allow real-time communication between the server and client devices.

Salient Features of Using Message Notification Alerts:

1. **Real-Time Communication:** Notifications allow users to be immediately informed about any updates, whether it's a message, alert, or event, without needing to open the app.

2. **User Engagement:** By sending timely alerts, an application can boost user engagement, ensuring that users don't miss important information.

3. **Customizable Alerts:** Alerts can be tailored to the type of message received, allowing different levels of urgency, sounds, and visuals based on the priority of the notification (e.g., urgent messages vs. casual ones).

4. **Background Functionality:** Notifications can be triggered even when the application is not actively running, allowing the app to keep the user informed at all times.

5. **Cross-Platform Compatibility:** Implementing push notifications ensures that users on different platforms (Android, iOS, etc.) receive alerts in a uniform manner.

6. **Increased User Retention:** By delivering timely and relevant notifications, users are more likely to return to the app, increasing retention rates.

Considerations for Implementing a Notification System:

1. **Privacy and Security:** Ensure that notifications do not expose sensitive information and are secured, particularly if they involve personal messages or alerts.

2. **Efficiency:** Notifications should be optimized to avoid excessive battery consumption or network usage on the user's device.

3. **User Preferences:** Allow users to manage their notification settings, including opting in/out, choosing alert types (sound, vibration, etc.), and controlling the frequency of alerts.

4. **Quality of Service (QoS):** Ensure that the notification system is reliable, delivers messages promptly, and operates effectively under varying network conditions.

5. **Customization:** Customize notifications to align with the user experience, including visual design, language, and timing of the alerts, to make the notifications more meaningful to the user.

Github Link :

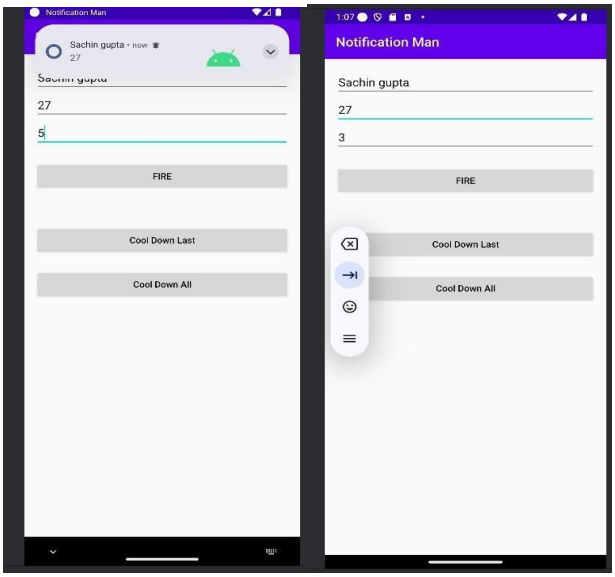
https://github.com/sachinskill/MC-EXPERIMENTS/tree/main/EXP-4%20Notification

Universal College of Engineering, Kaman

Department of Computer Engineering

Subject:Mobile Computing

Screenshots of the Output:



Conclusion: Hence, we have successfully implemented an application that creates real-time alerts upon receiving messages, enhancing user engagement and timely information delivery.