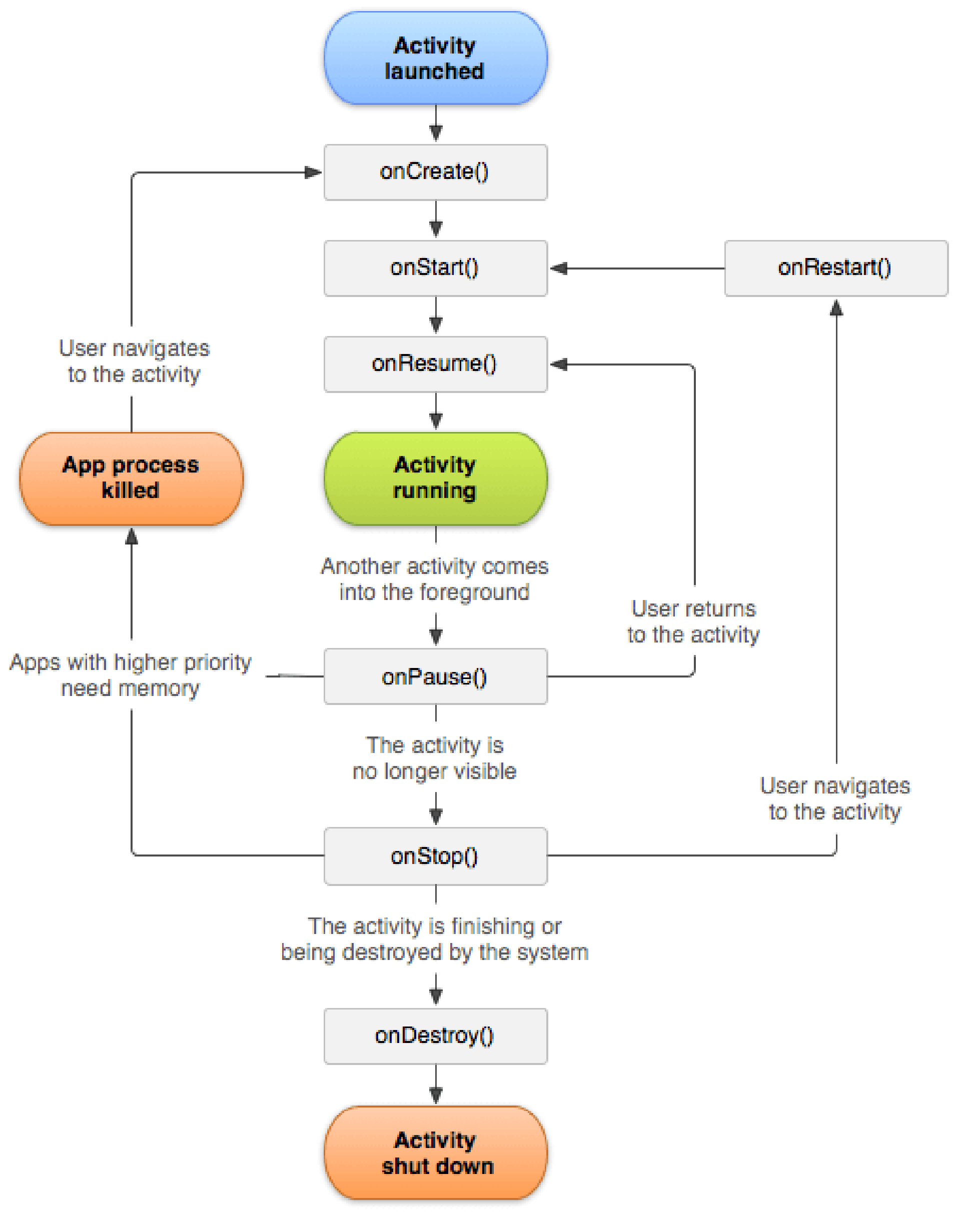
ACTIVITY LIFECYCLE OF ANDROID APP



Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

Activity

It is one of the four main components of android and is generally the App's Entry Point.

Unlike other programming paradigms where most of the programs are started using a main() method, Android launches code in an Activity instance through specific callback methods during various stages of its lifecycle.

onCreate()

This fires when the system first creates the activity. On activity creation, the activity enters the *Created* state. In the onCreate() method, you perform basic application startup logic that should happen only once for the entire life of the activity. For example, binding layout or your ViewModel.

onStart()

When the activity enters the Started state, the system invokes this callback. The onStart() call makes the activity visible to the user, as the app prepares for the activity to enter the foreground and become interactive. Use this where the app initializes the code that maintains the UI, like starting animation or attaching listeners or receivers.

onResume()

When the activity enters the Resumed state, it comes to the foreground, and then the system invokes the onResume() callback. This is the state in which the app interacts with the user. The app stays in this state until something happens to take focus away from the app. Such an event might be, for instance, receiving a phone call, the user navigating to another activity, or the device screen turning off.

onPause()

The onPause() callback method is called when an activity is no longer in the foreground, either because another activity has taken focus or because the user has navigated away from the app. In this method, you should perform any necessary actions that should happen before the activity becomes partially or completely obscured, like stopping animation or persisting UI states.

onStop()

This method is called when an activity is no longer visible to the user and is in the process of being stopped. In this method, you should perform any necessary actions that should be done when the activity is no longer visible. Like, releasing the resources, and unregistering listeners or receivers.

onPause() is called when an activity is losing focus, while onStop() is called when the activity is no longer visible to the user.

onDestroy()

onDestroy() is called before the activity is destroyed. The system invokes this callback either because:

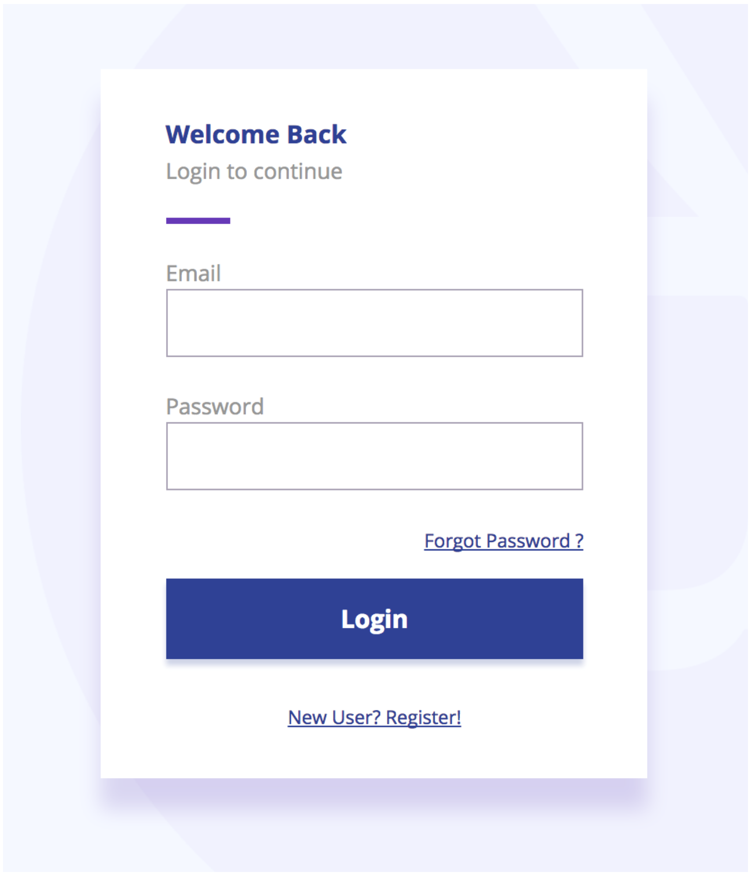
1. the activity is finishing (due to the user completely dismissing the activity or due to finish() being called on the activity), or
2. the system is temporarily destroying the activity due to a configuration change (such as device rotation or multi-window mode).

Activity Life Cycle of Phonepe:

Imagine tapping on the PhonePe app icon, opening a gateway to a the financial activities.

As you enter, the app get into Main Activity (MainPhonePeActivity), where the application starts.

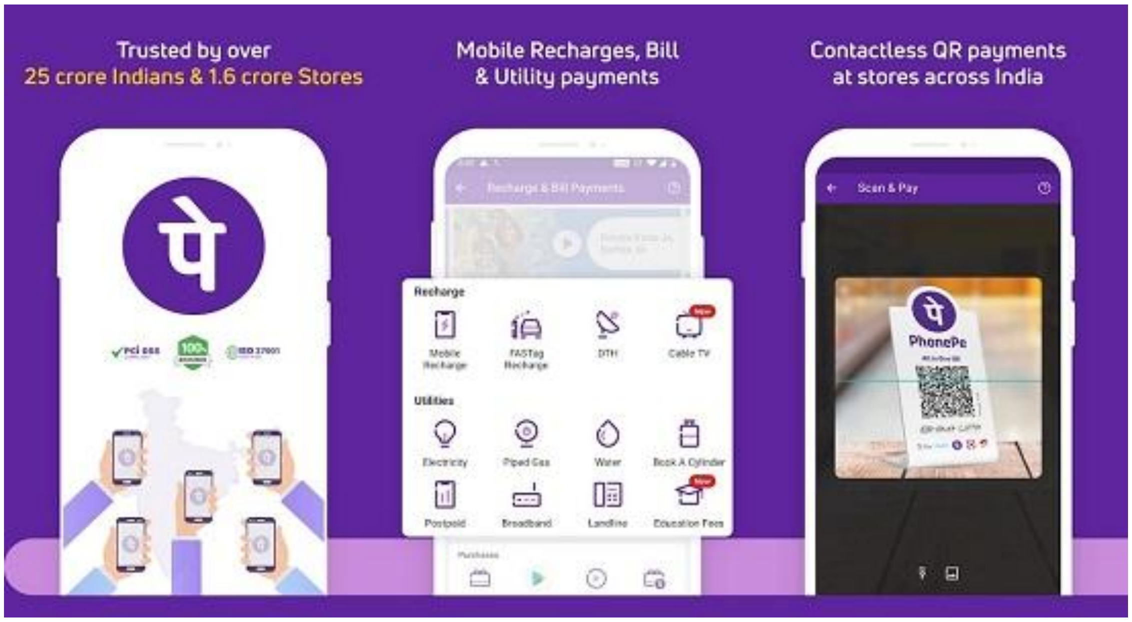
The onCreate() method serves as the catalyst, initializing this digital domain.



Onstart() Event:

Users traverse through a digital landscape within PhonePe, navigating of financial transactions, bill payments, and money transfers, interacting with its features.

With each interaction, the digital activities of PhonePe responds, facilitating transactions, providing insights, and unlocking rewards.



OnResume() Event:

In the Phonepe app users navigate through different financial activities such as transactions, bill payments, and money transfers, with each tap and swipe revealing new possibilities. The app responds to their actions, providing a smooth experience tailored to their needs. When users come back to the app after taking a break, onResume() refreshes it, empowering users to continue their financial journey . They can once again explore features and unlock hidden conveniences, making the PhonePe app a lively realm full of opportunities.

OnPause() Event:

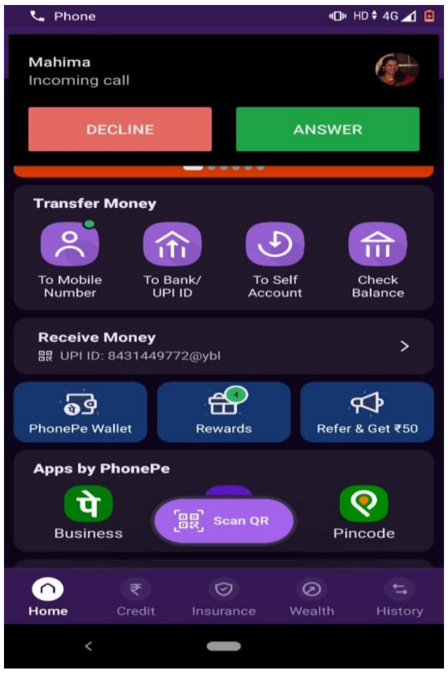
In the PhonePe app, the **onPause()** event is triggered when the user navigates away from the app or switches to another app while using PhonePe. This event indicates that the app is about to lose focus and move into the background. Here's how **onPause()** might be utilized in the context of PhonePe:

1. **Saving State**: When **onPause()** is called, PhonePe may save the current state of the app. This could include saving any data entered by the user, such as partially completed transactions or preferences set within the app. Saving this state ensures that the user's progress is not lost when they return to the app later.

2. **Pausing Processes**: PhonePe may use **onPause()** to pause any ongoing processes or animations within the app. For example, if there are any background tasks running, such as fetching transaction updates or processing payments, these tasks may be paused to conserve system resources and battery life while the app is not in focus.

3. **Releasing Resources**: **onPause()** might also be used to release resources that are no longer needed while the app is in the background. This could include releasing memory allocated for UI components or stopping any sensors or services that are no longer necessary.

4. **Handling Interruptions**: PhonePe may implement logic within **onPause()** to handle interruptions gracefully. For example, if a phone call or notification interrupts the user's interaction with the app, **onPause()** could be used to pause any ongoing tasks and display a message to the user indicating that the app is temporarily paused.



OnStop() Event:

After **onPause()** is executed, the app moves into the stopped state when users fully exit the app. This triggers the **onStop()** event.

As the user navigates away from the app, either by pressing the home button or switching to another app, the onStop() method is invoked.

Phonepe main activity stops its visible operations, transitioning to a background state. This phase marks the point where the app is no longer actively visible to the user but remains in memory, ready to resume when needed.

OnDestroy() Event:

In the destroy state, PhonePe may perform additional cleanup tasks, such as stopping background services, releasing system resources, and preparing the app for potential destruction by the system if memory resources are low.

App's main activity performs final cleanup tasks, releasing any remaining resources and closing database connections.