International Journal of Multidisciplinary Engineering in Current Research

ISSN: 2456-4265, Volume 5, Issue 9, September 2020, http://ijmec.com/

"A JOURNEY INTO ANDROID APP DEVELOPMENT WITH ANDROID STUDIO: BUILDING YOUR FIRST APP"

Mrs. P RAMADEVI, MCA *1, Mr. C. SANTHOSH KUMAR REDDY , MCA *2 Mr. G. VENKATESHWARLU, MCA, M.Tech *3

*1 Faculty in Department of computer science, Siva Sivani Degree College

*2 Faculty in Department of computer science, Siva Sivani Degree College

*3 Faculty in Department of computer science, Siva Sivani Degree College

ABSTRACT

This research paper explores the comprehensive development process of a cutting-edge Android application aimed at optimizing the user experience. The study delves into the iterative phases of design, implementation, and testing, emphasizing user-centric principles and the integration of emerging technologies. The paper discusses key challenges encountered during development and proposes effective solutions. Furthermore, the research highlights the application's impact on user satisfaction, performance metrics, and potential avenues for future enhancements. This study contributes valuable insights into the dynamic landscape of Android application development, shedding light on best practices and strategies for creating user-centric, efficient, and innovative mobile applications.

KEY WORDS: activity, operating system, application

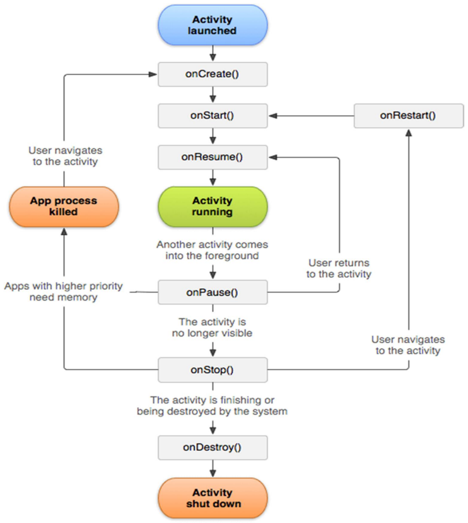
INTRODUCTION

Android is an open source and Linux-based Operating System for mobile devices such as smartphonesand tablet computers. In the ever-evolving landscape of mobile technology, Android applications continue to shape the way we interact with digital content. This article takes you on a step-by-step journey through the development of an Android application using the powerful Android Studio IDE. Whether you're a beginner or an experienced developer, this guide aims to provide insights and practical tips for creating a functional and user-friendly app.

Application frameworks specify how programs in a particular domain should be organized. A framework is basically a reusable component that establishes the architecture of applications by combining a collection of abstract classes and the collaboration of their instances. Google introduced Android, an open-source operating system built on the Linux kernel. Operating systems in mobile phones are limited by their hardware, storage capacity, power consumption, and mobility circumstances, in contrast to PC operating systems. Applications on mobile phone operating systems have some features that are different from those developed for PCs. This essay provides a thorough explanation of the fundamental design and application framework of the Android operating system, as well as an in-depth analysis of the primary organization of Android apps.

ACTIVITY

A single screen featuring a user interface is represented by an activity. While all of an application's operations complement one another to create a seamless user experience, they are all separate from one another. Therefore, any one of these actions can be initiated by a separate application. An implementation of an activity is a subclass of that activity. The way an application is designed by the developer affects both the specific form that an activity displays to users and the total number of activities in the application. When a multiple-activity application is launched for the first time, the user is usually presented with one activity designated as the "main" activity. After then, each activity can launch a new one to carry out a distinct task. Every time an action begins, the preceding


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

onCreate()

This is used to call when activity is created.

onStart()

This event is called when an activity starts working.

onResume()

This event is used to start when an activity is paused.

onPause()

This event will work when an activity is no longer working on foreground.

onStop()

This event is will be called when an activity stops working.

onDestroy()

This event will be called when an activity destroyed from the system.

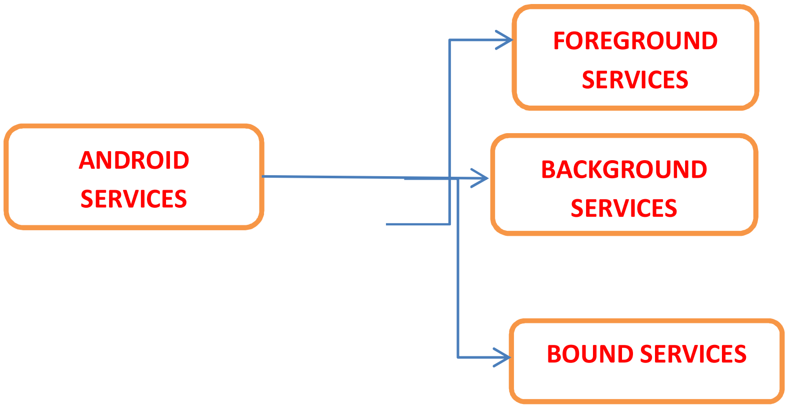
onRestart()

This event will be called when an activity again starts working.

ANDROID SERVICES:

Services are special component of an android which are useful to run the activities in the background for long running. A service will run in the background for long run continuously even if the application closed. An application components can bind itself to service to carry out **inter-process communication(IPC).**

TYPES OF ANDROID SERVICES



Foreground Services:

Foreground services are those that inform the user of their continuing activities. The notifications about the on going tasks that are delivered allow users to engage with the service. For example, when downloading a file, the user can pause and resume the process as well as monitor its progress.

Background Services:

Background services in Android play a crucial role in enabling apps to perform tasks even when the application is not actively in the foreground. These services are designed to run independently, allowing them to execute long-running operations without direct user interaction. Common use cases for background services include handling network requests, monitoring sensors, and updating data in the background.

Bound Services:

In Android, bound services are a type of service that allows other components, such as activities, to bind to them and interact with them through an interface. Here are some key points about bound services.

Application Development and Implementation

I. Getting Started:

Setting Up Your Development Environment

- Begin by installing Android Studio and configuring the necessary SDKs.

- Create a new project, selecting the appropriate template based on your application's requirements.

II. Understanding the Project Structure:

Unveiling the Anatomy of an Android Project

- Explore the key components, such as activities, layouts, and resources.

- Understand the importance of the manifest file in defining app configurations.

III. Designing the User Interface:

Creating an Intuitive and Responsive Layout

- Utilize the XML-based layout files to design the visual components of your app.

- Explore the various layout managers for arranging widgets dynamically.

- Integrate images, buttons, and text views to enhance the user experience.

IV. Implementing Functionality:

Bringing Your App to Life

- Write Java (or Kotlin) code to define the behavior of your app.

- Understand the activity lifecycle and implement methods for handling user interactions.

- Use intents to navigate between different activities and pass data.

V. Handling Data:

Persisting and Retrieving Information

- Integrate SQLite database or other storage options to manage app data.

- Implement data models and use content providers for seamless data access.

VI. Incorporating APIs:

Connecting Your App to the World

- Integrate web APIs to fetch real-time data or enable specific functionalities.

- Handle network requests asynchronously to ensure a smooth user experience.

VII. Testing Your App:

Ensuring Reliability and Quality

- Employ various testing methodologies, including unit tests and UI tests.

- Utilize Android Studio's testing tools to identify and resolve issues.

VIII. Enhancing User Experience:

Polishing Your App for Success

- Implement responsive design principles for different screen sizes and orientations.

- Incorporate animations and transitions to create a visually appealing interface.

IX. Deployment:

Preparing Your App for the World

- Generate a signed APK for release.

- Explore the process of publishing your app to the Google Play Store.

Developing an application using Android Studio comes with various challenges, including:

1. Device Fragmentation: Android runs on a wide range of devices with different screen sizes, resolutions, and hardware specifications. Ensuring a consistent user experience across this diversity can be challenging.

2. Version Compatibility: Supporting multiple Android versions and ensuring backward compatibility can be complex due to differences in APIs and features.

3. Performance Optimization: Achieving optimal performance on various devices requires careful optimization, considering factors like memory usage, CPU utilization, and battery consumption.

4. User Interface Design: Designing a responsive and appealing user interface that works well on different screen sizes can be challenging. Accommodating both smartphones and tablets adds an extra layer of complexity.

5. Security Concerns: Android applications need robust security measures to protect user data. Managing permissions, secure data storage, and preventing vulnerabilities are crucial aspects.

Conclusion:

Android application development using Android Studio offers a robust and versatile platform for creating innovative mobile solutions. With its powerful tools, extensive libraries, and active developer community, Android Studio facilitates the creation of feature-rich and user-friendly applications. Embracing best practices, staying updated with Android OS advancements, and considering user experience are pivotal for success in this dynamic field.

Embarking on the journey of Android app development can be both challenging and rewarding. Android Studio provides a robust platform, empowering developers to transform ideas into fully functional applications. By mastering the fundamental concepts outlined in this article, you'll be well-equipped to create innovative and impactful Android apps that resonate with users worldwide.

REFERENCES

1. **Official Android Documentation:**

opment of Android Applications”

- Android Developer Guide: [Android Developer](https://developer.android.com/guide)
- Android Studio Documentation: [Android Studio Guide](https://developer.android.com/studio/intro)

2. **Books:**

- "Android Programming: The Big Nerd Ranch Guide" by Bill Phillips and Chris Stewart.
- "Head First Android Development" by Dawn Griffiths and David Griffiths.

3. **Online Courses:**

- Udacity's "Android Basics Nanodegree" or "Android Developer Nanodegree."
- Coursera courses on Android development.

4. **Websites and Blogs:**

- Stack Overflow: [Android Questions](https://stackoverflow.com/questions/tagged/android)
- Medium articles on Android development.
- Android Developers Blog: [Android Developers Blog](https://android-developers.googleblog.com/)

5. Mrs. Prachi Sasankar Mrs. Usha Kosarkar , ” Research on Devel