

GSUZapApp- Send Notifications In A Single Click

-Bhavya Kilari
Advisors- Dr. Raj & Dr. Li

Agenda

- Part "A"
 - Introduction to Android
 - Project and Features
- Part "B"
 - Demonstration
 - Future Scope

The New Mobile Era

The General Purpose Computing Club

Mainframe Server Workstation Laptop New Kid on the Block

Apps on Mobile

- Mobile Web Apps
- Native Mobile Apps
- Hybrid Apps

ANDROID-History

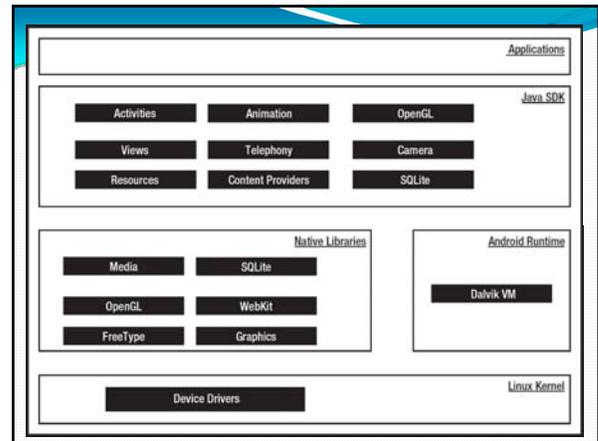
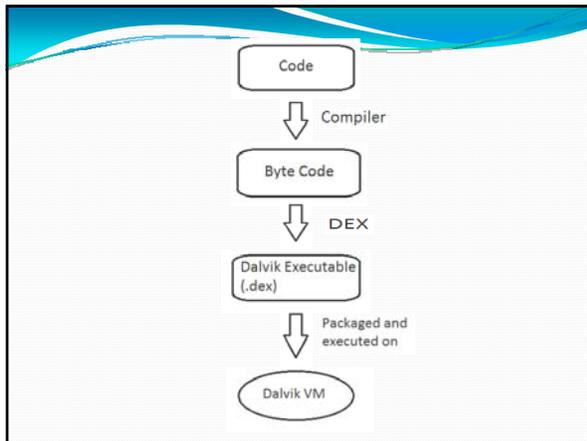
- The Android platform promised openness, affordability, open source code, and a high-end development framework.

2008	2008	2008	2008
	T-Mobile G1 Announced	SDK 1.0 Released	Android Open Sourced
2007	2007	2007	
	OHA Announced	Early Link SDK	
2005	2005	2005	
	Google Buys Android Inc.	Work on Dalvik VM Starts	

Virtual Machine: Dalvik

```

    graph LR
      A[Java Program] --> B[Compiler]
      B --> C[Java Bytecode Program]
      C --> D[Java Interpreter for Mac OS]
      C --> E[Java Interpreter for Windows]
      C --> F[Java Interpreter for Linux]
      C --> G[Dalvik VM for Android]
  
```

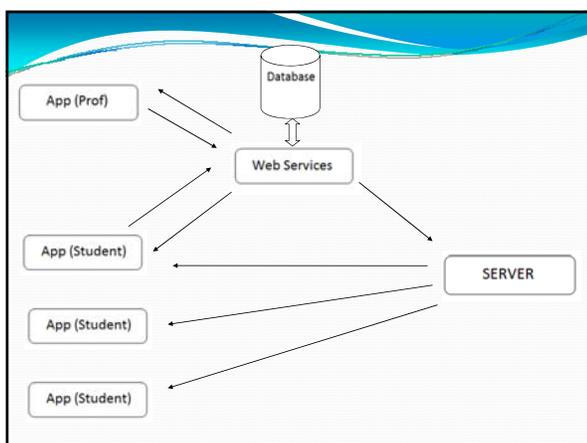


ZapApp

- Goal: To built an application that allows a user to log in and broadcast notifications to a group of registered users.
- Technologies: Android SDK in Eclipse, SQL, .NET
- App Details:
 1. End Application is designed for Android phones.
 2. Web services written in .Net are loaded live in a server.
 3. Database is loaded in the server.

Notifications

- **Poll**- Periodically poll the server for new messages from a background local or remote service. The more often you poll the closer you get to the real-time push.
- **SMS** Android allows you to intercept SMS messages. Your server sends a specially encoded SMS to your phone, whenever there is something new. Your app intercepts all messages, looks for the ones from the server, then pops up a notification.
- **Persistent TCP/IP(Push)** The phone initiates a long-lived mostly idle TCP/IP connection with the server and maintains it by occasionally sending keep-alive messages. Whenever there is something new on the server, it sends a messages to the phone over the TCP connection.



Flow

- App inputs login credentials from the user.
- App calls the web services for login verification.
- Web services verify data with database values and return the role.
- App decides the layout and flow based on the role returned by the web services
- If role is "professor"- app will allow the user to send a message with a subject as a notification.
- If role is "student" – app look like GSU computer science website.

