

Project Title

Java API of Android Operating System for Ad-hoc networking

Team Members

Klaus Cipi - kcipi2015@my.fit.edu

Peter Banis - pbanis2015@my.fit.edu

Michael Kolar - mkolar2015@my.fit.edu

Robert Olsen - olsenr2015@my.fit.edu

Faculty Sponsor

Dr. Marius Silaghi - msilaghi@fit.edu

Client

Dr. Marius Silaghi - Associate Professor at College of Engineering & Science, FIT

Meetings with client

Wednesday, August 22, 2018

Monday, August 27, 2018

Goals and Motivation

Our goal is to complete the Android API for Ad-hoc networking and to create an integrity check file that will verify that the API is properly installed. API will allow easy connectivity between all main operating systems(Android, Mac OS, Windows, Linux). Currently, no unified API for Ad-Hoc networking exists, which creates a serious deficiency in what can be done for users that are not in only one ecosystem. The unifying nature of the Java API will allow for easy cross-platform communication. This will enable easier transfer of files and other information between devices, which will be convenient for their users. Ad-hoc networks will also allow different devices to connect to each other. This can allow users to give each other internet access, to create a distributed network or even form an intranet without being limited by what physical devices users have.

Approach / Features

1. Integrity check/installation validation.

The system must ensure that all components have been installed correctly. This is not limited to only ensuring the correct files exist in the correct directories, but also that all files have needed permissions(for example, a script might lack an execute permission which prevents further progress). The system will recognize when such a situation occurs, and if it cannot be fixed by the system alone, the system will provide useful feedback to the user so they may correct the error.

2. Automatic OS detection.

A single Java API package will include support for all major platforms like Android, Windows, Linux, and MacOS. The API will be able to automatically detect the platform that the developer is working on and provide the appropriate support for it. Furthermore, during the installation of any application that will use this API, only scripts for the proper platform will run.

3. Better utilization of Android's native hardware capability.

We plan on taking advantage of Android's native hardware capability for simultaneous dedicated internet connect and ad-hoc connection. Users will find this convenient since they won't need a second adapter to use both connections.

Novel Feature

Creating ability to use Android devices' dedicated ad hoc wifi connection. We are not aware of software that currently takes advantage of this feature.

Technical Challenges

One notable weakness is that half our teams lacks experience in network programming. Furthermore, our entire team is not familiar with android development. Finally, we all lack knowledge in scripting for operating system level operations.

Milestone I - Due October 1, 2018

- Decide between developing on MacOS, Windows, or Linux
- Install and configure Android Studio
- Decide whether to use Eclipse or IntelliJ for an IDE
- Install and connect to the SVN repository provided by Dr. Silaghi
- Successfully install the API
- Run an example program with the API(to verify correctness of install and behavior)
- Create Requirements Document
- Create Design Document
- Create Test Plan

Milestone II - Due October 29, 2018

- Integrity check/installation validation (IT&D).
- Automatic OS detection (IT&D).

Milestone III - Due November 26, 2018

- Create sample programs using our API (IT&D).

Workload Distribution

Task	Peter	Klaus	Michael	Robert
Decide between developing on MacOS, Windows, or Linux	Discussion	Discussion	Discussion	Discussion
Decide whether to use Eclipse or IntelliJ for an IDE	Discussion	Discussion	Discussion	Discussion
Install and configure Android Studio	Do	Do	Do	Do
Install and connect to the SVN repository provided by Dr. Silaghi	Do	Do	Do	Do
Successfully install the API	Linux	MacOS	Windows	Android
Run an example program with the API	Do	Do	Do	Do
Create Requirements Document	30%	30%	20%	20%
Create Design Document	20%	30%	20%	30%
Create Test Plan	30%	20%	30%	20%
Prepare demo	20%	20%	30%	30%

Faculty Sponsor's Approval

"I have discussed with the team and approve this project plan. I will evaluate the progress and assign a grade for each of the three milestones."

Signature: _____ Date: _____