Project Title
A Java API for unifying ad-hoc Wifi 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 Sponsor/Client
Thursday, November 1
Monday, November 5
Thursday, November 15

Progress of current Milestone (progress matrix)

Task	To Do (Completion %)	Peter	Klaus	Michael	Robert
Divorce API Networking code from Direct Democracy application	Extract sockets when P2P connections are complete (0% Complete)	0%	0%	0%	0%
Expand and create networking functions	Uncertain, depends on demo application (0% Complete)	0%	0%	0%	0%
Acquire DirectP2P capable adapters and Android phones for future testing	None (100% Complete)	25%	25%	25%	25%
Implement support for configuring devices in DirectP2P mode	MacOSX Multipeer, Windows Homegroups, Write Scripts for OSes (20% Complete)	30% of Linux P2P	20% of Mac P2P	25% of Windows P2P	25% of Windows P2P

Create more specific exceptions	None (100% Complete)	30%	30%	30%	10%
Debug and correct connection problems between Windows 7/10 and Mac/Linux	None (100% Complete)	15%	15%	35%	35%

Discussion (at least a few sentences, ie a paragraph) of each accomplished task (and obstacles) for the current Milestone

- Divorce API Networking code from Direct Democracy application: After more careful review we are pushing this task back until Direct P2P support is complete.
- Expand and create networking functions: As with above, this task was found to be unnecessary at the current time. We have prioritized more core responsibilities of the API for development and will return to this if needed in the future.
- Acquire DirectP2P capable adapters and Android phones for future testing: Each member of the team has purchased their own TP-Link N300 Wireless WiFi Network Adapter, which helps us avoid false conclusions of native incapability due to the fussy nature of the hardware on some devices. Store page for the adapter: https://www.amazon.com/TP-Link-TL-WN823N-Wireless-network-Raspberry/dp/B0088TKTY2. In addition to the adapters, we have obtained a Samsung Galaxy S5 (Android 6.X) for testing the Android side of the API in the future.
- Implement support for configuring devices in DirectP2P mode:
    - Linux: Linux is capable of supporting this with scripting, however handling the configuration through the script alone is not possible. We are working on a way to handle this through the java code by redirecting the input and output streams but the solution is still young.
    - Mac: Mac does not currently support the Direct P2P standard. Attempts to compile wpa_supplicant on Mac failed. Research is still ongoing for alternatives.
    - Windows: Windows doesn’t currently have generic P2P capabilities. Further research and experimentation is needed.
- Create more specific exceptions: New exceptions were created, and added to our source code. This includes a DeniedPermissionException, MissingArgumentsException, and ScriptMissingException. The DeniedPermissionException doesn’t find which permission was denied because we can deductively conclude that the only permission that would be relevant is the execute permission. The error message from the exception tells the user that they lack execute permission. The MissingArguments is thrown when the user

hasn't defined the Network Interface, SSID, Password, and Channel in their AdHocConfig object and then they try use a method which requires this data. Before the computer tries to run a script it will check to make sure that a script with the correct name is present in the /scripts or /scripts/Windows directory and if it's absent then the ScriptMissingException is thrown.

- Debug and correct connection problems between Windows 7/10 and Mac/Linux: Prior to this milestone, Windows 7 and 10 were deemed unable of making true Ad-hoc networks. Initially, the Windows scripts were creating infrastructure networks, which is not what was desired. A Windows user must create a profile in order to join or create a network. Whether this profile is for joining or creating a network is determined by whether or not it already exists. Windows will try to create a seperate network regardless, but it will be merged into one since their details are an exact match. Any issues from attempting to ping each other over our networks was resolved by making exceptions in our firewall settings or completely turning them off.

Discussion (at least a few sentences, ie a paragraph) of contribution of each team member to the current Milestone

- Peter Banis: Peter was in charge of working on the Linux part of the implementation of DirectP2P. He helped to add more exceptions to the API, expanding on its error reporting and clarity for its user base. He helped in research for the most appropriate yet cost-effective adapters and Android devices that were integral to further development of the API.
- Klaus Cipi: Klaus is responsible for the Mac implementation of DirectP2P. Klaus helped to clarify the exception handling components of the API by adding more exceptions. His research into wireless adapters and Android devices which both help assure that our toolset is appropriate for our budget yet representative of the standard user's experience.
- Michael Kolar: Michael, with the help of Robert, is working on the Windows implementation for DirectP2P. Michael helped Peter and Klaus with adding more exceptions to the API. He also resolved issues pertaining to Windows network connections, as well as making proper Ad-hoc networks on Windows. He too helped to select from the adapters and Android devices.
- Robert Olsen: Robert is assisting Michael in the DirectP2P implementation for Windows. Robert also helped Michael in resolving the connection issues on Windows, as well as finding out the proper method of making an Ad-hoc network through scripts. He too helped to select from the adapters and Android devices. Yet again, he helped to write the evaluation and presentation.

Plan for the next Milestone (task matrix)

Task	Peter	Michael	Klaus	Robert
Complete DirectP2P support	30%	25%	20%	25%
Complete DHCP option for Ad-Hoc networks	25%	25%	25%	25%
Implement Android P2P support	30%	17%	18%	35%

Discussion (at least a few sentences, ie a paragraph) of each planned task for the next Milestone

- Complete DirectP2P support:
  - The only platform that fully supports P2P is Linux, so during the next milestone we will try to work around Mac OSX Multipeer Connectivity and Windows Homegrouping to establish DirectP2P networks. We will work on writing scripts for each OS, if P2P is possible, and provide a way for developers to identify, join or create Wifi direct networks. After a connection is established, our API will provide sockets for the developers to exchange data.
- Complete DHCP option for Ad-hoc networks:
  - Currently we rely on a combination of static IPs and a feature of the 802.11 standard which specifies a device to default to an IP in the 169.254.X.Y range when attempting to get an IP via DHCP and there is no DHCP server available. Enabling DHCP support has undergone initial experimentation without success. There exist two main considerations that need to be solved: DHCP when starting a network, assigning a new member of the network as a DHCP server when the existing server leaves. Additionally, we need a method to determine all the peers on the network, as this is not reliably supported at the protocol level. Our current design approach is pinging all the IP's in the range and then returning a list of all the IPs that respond. This is not an ideal solution, but we will use it until we can devote more to researching the problem and solutions.
- Implement Android P2P support:
  - The Android API already has built in support for P2P networking through the WifiP2PManager class. This and related classes will enable us to progress quickly through the implementation of Android support.

Sponsor feedback on each task for the current Milestone

- Divorce API Networking code from Direct Democracy application:

- Expand and create networking functions:

- Acquire DirectP2P capable adapters and Android phones for future testing:

- Implement support for configuring devices in DirectP2P mode:

- Create more specific exceptions:

- Debug and correct connection problems between Windows 7/10 and Mac/Linux:

Sponsor Signature: _______________________________ Date: ________

Sponsor Evaluation

- Sponsor: detach and return this page to Dr. Chan (HC 322)
- Score (0-10) for each member: circle a score (or circle two adjacent scores for .25 or write down a real number between 0 and 10)

Peter Banis	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
Klaus Cipi	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
Michael Kolar	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
Robert Olsen	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10

- Sponsor Signature: ________________________________ Date: ________