Requirements

1. Vision

Google's mobile app developers (or any mobile app developers) need a mobile app library and a web server to stream log files from a mobile device or emulator to a desktop web interface. This project aims to streamline the mobile app development process by cutting down on the time needed to look through logs to find error messages or check statuses while debugging. This project team will build an open source Android library, server application, and web app combination to allow the cordless debugging of mobile apps.

2. Definitions

- Logs: logs seen on the adb (Android debug bridge) output and iOS equivalent
- Mobile API: APIs which can be included in a mobile app to enable wireless logging
- Web Interface: the interface where users can view the logs coming from the device
- Web Server Library: the web server library that receives, parses, stores and sends logs to the Web Interface

3. Functional Requirements

- Allow for a wireless connection from a mobile device to a server over which logs can be streamed
- Mobile API that can be attached to app projects and streams logs to a specified server when the application is started
- Web server library that receives, parses and stores the logs from a mobile device, then sends them to the web interface
    - Error information will be sent to web interface
- Allow for session management (client side provide sessions by sending session ID, web server library keeps track of sessions, and web interface provides mechanism to select a session)
- Two test apps:
    - Android application to show use of the Android mobile API
    - App Engine instance to show use of the web server library
- Web interface allows the user to view logs in a readable format
    - View logs based on session ID
    - Real time streaming of logs
    - Real time filtering of logs
- Android support of mobile API

    - Documentation for transferring mobile API to iOS
- Mobile API must work on Android API level 21 and above

4. Non-functional Requirements

- Identify logs by user, device, and session
- Platform independent - ability to run the library on any server
- Web Server Library can run locally on a single computer
- Web Server Library can run on a server and can support multiple users and devices
- Documentation of the project for other developers to use and extend the library
    - API calls
    - Walkthrough of implementation
    - Sample App Engine instance
- Include how to extend the library to support iOS
- All code written according to Google's style guides

5. Potential Project Risks

5.1 Technology Risks

- Sharing a single App Engine instance between all team members

5.2 Skill Risks

- Inexperience in Android Development
- Inexperience in Web Application Development

6. Definition of Done

6.1 Minimum viable product

- An Android mobile API that sends log files from a device to a web server with the web server library imported
- A web server library that streams log files to a web interface
- An example Android application that demonstrates the functionality of the library
- An example App Engine instance that streams log files to a web interface
- Tests for all of the above software

- Documentation for all of the above software

6.2 Extensions of project

- Web Interface: Log file highlighting
- Web Interface: Support for other cloud hosting platforms/servers (AWS, Heroku, etc.)
- Mobile API: Support for iOS application logs
- Mobile API: Log streaming and searching from within the app
- Mobile API: Stream additional metrics
- Web Server Library/Web Interface: User authentication via an authentication framework
- Web Server Library: Storing log sessions (for historical purposes)
- Web Server Library: Associate client system health metrics with logs
- Web Server Library: Store sessions

6.3 Specifics for Field Session

- Project will be hosted on a private Google GitHub repository that will be owned by the client with the team members as contributors. Code reviews will be done by the client.
- Client would like to see 80% code coverage by tests of the project