

CS 5150 Project Feasibility Study

OpenComm iPhone App Development
Team 18 | 9/20/2012

Table of Contents

⌘ APPLICATION OVERVIEW	1
CLIENT OVERVIEW	1
PROJECT OVERVIEW	2
⌘ TEAM OVERVIEW	2
MEMBERS AND CONTACT INFORMATION.....	2
PRIMARY TA AND CONTACT INFORMATION	3
INDIVIDUAL RESPONSIBILITIES AND SPECIALTIES	3
⌘ DELIVERY TIMELINE	3
MILESTONE 1 - INITIALIZATION.....	4
MILESTONE 2 - UI.....	4
MILESTONE 3 - FUNCTIONALITY	4
MILESTONE 4 - TOUCHUPS	4
⌘ TECHNICAL FEASIBILITY	5
DESIGN	5
MANAGEMENT	5
TESTING	5
OPENCOMM TECHNOLOGY	6
⌘ VISIBILITY	6
⌘ RISK ANALYSIS	6
TECHNICAL.....	6
SOCIAL RISK.....	7
LEGAL	7
⌘ CONCLUSION	7

~*~

⌘ Application Overview

Client Overview

(Description from <http://opencomm.github.com/home.html>)

OpenComm is currently an Android application that strives to make mobile conference calling as intuitive as possible via an advanced user experience and sound spatialization.

While face-to-face interaction is the preferred method of conversation, sometimes it is simply impossible to have everyone together in the same room. The standard solution is to hold meetings through conference calls. However, a conference call

creates many problems that are not inherently noticeable. Users being unable to discuss matters confidentially or to split into group discussions before making a final decision, are just two such problems.

Features Supported on the Android Platform

Sound Spatialization - Virtually control the layout of the conference and listen to the spatially oriented voice of individuals as if they're seated at their designated positions in the room/table. In simpler terms, simulate a real world conference room, with room layout and each individual's seating impacting their voice levels.

Side Chats - Start simultaneous private conversations to separately discuss the conference, plan ahead with a partner, or just for fun.

Control - A user can have complete and unrestricted control over the conference with our user-friendly layout.

Client Contact Information

Risa Naka risanaka@gmail.com

Website <http://opencomm.github.com/home.html>

Project Overview

Our main goal is to build an iOS application to complement the existing Android client built by the OpenComm research team. iOS users should support all core functionalities through an interface which strongly resembles the current Android design.

Functionality Objectives

The current OpenComm model can be boiled down to a set of core functionalities, such as the ability for users to control the layout of their conference and hold private conversations on the side while the original meeting is still going on. We will first implement the basic capability of establishing phone conferences using OpenComm on iOS platform. After that, we will move onto enabling more advanced functionalities on iOS that are currently supported on Android platform. We promise to implement these functionalities in a clean, elegant and efficient manner. This means that all complex routines will be clearly commented, making it easy for code maintenance in the future.

Business Impact

iOS captures around 33.4% of the current US. mobile market. The lack of an iOS compatible model renders OpenComm not impacting more than one in three users. We believe that our software engineering project can have a huge impact on the future of OpenComm's business model and further its growth.

Design Objectives

OpenComm is a small niche software targeted towards everyday users who would like to have access to additional features while conference calling. As a result, the user experience needs to be simple, clean and intuitive. Our team will cooperate regularly with the design and user interface teams to come up with the most optimal design for the end users.

~*~

⌘ Team Overview

We are a group of senior computer science majors who come from various compute science backgrounds. Although none of us have previous iOS development experience, we are very confident in our abilities to pick it up very quickly. In addition, three of us are enrolled in the iOS development class, which serves as a great resource to help us combat our technical difficulties.

Members and Contact Information

Kevin Chen	kfc35@cornell.edu
Qiming Fang	qf26@cornell.edu
Shihui Song	ss2249@cornell.edu
Sauhard Bindal	sb673@cornell.edu
Yiyi Chen	yc523@cornell.edu

Primary TA and Contact Information

Myle Ott

mao37@cornell.edu

Individual Responsibilities and Specialties

Following our initial discussion about the implementation details with the client we have come up with the following assignments, based on each individual's interests and skill set:

1. Kevin Chen

Kevin will be primarily responsible for debugging and testing of our application components. He is fast at catching bugs and is willing review other people's code to find any errors that it may have. He will also help Sweet with the backend components of the application. Additionally, he will ensure that everyone is in sync with the communications so that all members of the group are on the same page. Kevin is enrolled in the iOS app design course, so he will have a basic understanding and knowledge of iOS app development.

2. Qiming Fang

Qiming is pursuing the systems vector at Cornell, and he will be responsible for iOS UI development together with Sauhard. He is well versed in backend component of software design and is more than willing to lend a hand to solving server side and database issues. He has proven to be a good leader of the group and will be leading the group.

3. Shihui Song

Shihui, better known as Sweet, will be responsible for the backend integration of the project along with Kevin. Through her internships she has gained much experience with backend software development and familiarity with backend solutions. She is a stickler for detail and adept at multitasking. She would like to be involved in communication with the client. Sweet is also the only one to have prior mobile experience, having done some Android app programming in the past.

4. Sauhard Bindal

Sauhard will work on the UI and frontend part of the app. Having taken some graphics courses at Cornell, as well as having a good eye for design, he is confident that he will do a great job with the UI implementations. Sauhard will be the devil's advocate and take the point of view of the customer and client from time to time. Additionally, he will also be involved in communication with the client, especially with the UI team. Like Kevin, he is also taking the iOS app design class giving him a good foundation to work on this project.

5. Yiyi Chen

Yiyi leads the user experience and will be the product manager of the group. She is currently taking HCI, exemplifying her interest and passion for user experience and design. Furthermore, she is also involved in two consulting clubs at Cornell, gaining valuable project management experience and client communication skills. Her abilities in algorithms will be employed often in problem solving situations.

Although we are all Computer Science majors, we each bring our individual skills to the table. Through our internships, research, classes, interests and other experiences, we have developed unique abilities and together we form a cohesive team. We are all dynamic, passionate and enthusiastic about the project and look forward to enhancing our knowledge by venturing into a new field for us and have fun while doing so.

~ * ~

⌘ Delivery Timeline

We chose the incremental software development process for our project. We have broken the major client requirements into two separate increments (Milestone 2 and 3). During every increment, we will all work on the same major issue at hand in an effort to push out a feature as quickly as possible. We decided to use an incremental approach because we value the Agile methodologies, particularly its ability to repeatedly identify crucial problems and quickly pivot the focus of the entire team to solve them. In addition, incremental processes is the least likely to waste code, which certainly helps us in this semester long project. Finally, we are not very experienced iOS programmers, which means that sometimes we will run into problems that are very hard to solve. The rapid incremental approach lets us take a step back and look to solve the problem from a completely different angle. This, in turn, reduces the amount of wasted time as well.

Please note that the both the CS 5150 delivery deadlines as well as the OpenComm delivery deadlines are noted below.

Milestone 1 - Initialization

Important Dates

Sept 21 | CS 5150 Feasibility Report due

Tasks

Obtain git repo access, and clone existing git repo into local machines
Setup access to Base Camp
Finalize weekly meeting schedule, and determine the team reps for team meetings

Deliverables

iOS project committed to OpenComm Git repository

Milestone 2 - UI

Important Dates

Oct 7 | OpenComm Milestone
Oct 11 | CS 5150 Milestone 2 Presentation
Oct 12 | CS 5150 Milestone 2 Report Due
Oct 24 | OpenComm Milestone

Tasks

Create basic UI of iOS application with basic buttons in place
Look into possible solutions for data exchange between Red Hat server & iOS device
Determine what would be a good format to transmit audio byte stream (iOS may not be compatible with android byte[])

Deliverables

Functional iOS application with basic UI structure
Document proposal depicting the different types of data transfer solutions available, and the solution we plan to adhere to
Obtain consent from OpenComm team regarding said solution

Milestone 3 - Functionality

Important Dates

Nov 4 | OpenComm Milestone
Nov 8 | CS 5150 Milestone 3 Presentation
Nov 9 | CS 5150 Milestone 3 Report
Nov 18 | OpenComm Milestone

Tasks

Enhance the design of UI by optimizing user interaction panels
Implement efficient solution to data passing
Verify that all audio data is received correctly
Begin phase 2 of user testing - Users will be giving us feedback on what they think about the conference call experience (can they hear what other people are saying?)

Deliverables

Implement simple data transfer protocol to allow for byte stream transfer

Milestone 4 - Touchups

Important Dates

Nov 30 | CS 5150 Final Presentation and Demo
Dec 2 | OpenComm Milestone
Dec 7 | CS 5150 Project Delivery

Tasks

UI should be perfected to give the iOS app the same *look and feel* of the Android app

Audio should be transmitted fully and flawlessly
Security features should be verified

Deliverables

Project should be fully functional with clean UI

~*~

⌘ Technical Feasibility

We describe below a list of tools we will use to enhance the development of the project. We believe that there exists many technically feasible solutions to developing this project. However, the numerous ways in which we can implement the solutions must be feasible to the rest of OpenComm, as they are our clients.

Programming - iOS development

- *MAC OSX* - To develop for iOS, MAC OSX is essential. The development and business model under which Apple operates is closely intertwined iOS development with the MAC OSX. Using MAC OSX will then facilitate much of the development hurdles otherwise. Particularly, XCode as an IDE is only available on MAC OSX. 4 of us will be developing on a MAC, while the other will be running MAC OSX image as a virtual machine on linux.
- *XCode IDE* - The developers at Apple have released XCode as the development environment for iPhone. It's built for iPhone specific development and is extremely crucial to the App development process as it runs the iPhone simulator. XCode will be the IDE we will use to work with for the entire project.
- *Objective-C* - The programming language for iPhone is Objective C. While none of the software engineering developers programmed in the language, all of us have worked with different languages of the C family. Furthermore, most of us are in the iPhone Development course and are confident that we will be able to use the language to suit our needs.
- *XMPP* - The OpenComm Android App utilizes XMPP as its communication protocol. Currently, there is a code base (<https://github.com/robbiehanson/XMPPFramework>), which has integration potential with the iOS App.
- *Other Open Source Plugins and Code* - The growing iOS community has already written much code for various issues that other teams ran into. In the world of software engineering, the open source community remains supportive and sharing of their technologies. For this reason, we look to Open Source projects that might have different solutions to various problems that we will tackle.

Design

- *XCode IDE* - XCode has a UI design tool which has a drag-and-drop functionality to directly design the iPhone App in the IDE. However, the UI design team in the original OpenComm group will design the App and we will only need to implement it, so this design tool can provide direct translation from the OpenComm visions to iPhone application.

Management

- *Git* - Git will be the version control choice for our code. It is arguably the leading source control at the moment, and allows for smooth parallel development. It is also the revision control already used by OpenComm, so integration is easy.
- *GitHub* - GitHub is the host for our code. OpenComm had been using GitHub, and the GitHub desktop component is very much advocated by OpenComm.
- *Google Docs* - Similar to git, but on a more flexible and interactive level, Google Docs are the tools for which we will be writing documentation and reports. They are hosted on the cloud like Git, but it is much easier to contribute to a document at the same time than can be managed by Git.
- *Base Camp* - We will be using a tool called BaseCamp for project management. BaseCamp is what OpenComm currently uses, and it is effective in enhancing communication among multiple members of the group.

Testing

- *iPhones* - As this project is to develop an iPhone application, testing must be conducted on real iPhones, not just the iPhone simulator in XCode. There are 4 iPhones in the group, from 3G to 4S, and we believe that this sample of iPhones is a good representation of the iPhone market. In addition, having this many iPhones will be good for testing concurrently. One of our team members is a part of the iOS Developer Program, so all the phones will be registered with him to test on.

OpenComm Technology

- *Server and Database* - OpenComm has their own XMPP and HTTP servers and MYSQL databases that we will need to be working with for the project. Because the servers and databases are already working with the Android App, we expect that some of the configuration portions will need to be tailored to be compatible with the iPhone App.

~ * ~

⌘ Visibility

Our team will attend weekly meetings with the OpenComm research team. In addition, one member will serve as the *group leader* and will attend an additional weekly meeting to help with iteration planning and contribute to high-level discussions.

Internally, we plan to adopt an incremental process. We plan to have a weekly meeting on Tuesdays 6:30pm - 8:00pm to go over individual progress, and to address any technical problems that may have arisen. We will use project management software, Basecamp, to keep track of tasks, and to assign tasks to members. Additional communication will be done through email and google docs. We will be using git as our version control system.

Our team has a bi-weekly release cycle. This means that we will demo our progress to the OpenComm team every two weeks. We guarantee that our releases are bug-free and well documented. In addition, we also get new tasks every two weeks. In the event that we have trouble finishing all of the tasks by the two-week deadline, we must contact the OpenComm management team by the end of week one.

~ * ~

⌘ Risk Analysis

Below is a list of risks that we anticipate from the project. For the most part, our client understands the risks for our project. Because of this, most risks will have low interference with the basic requirements of our project if those risks happen to occur. However, the risk on finding appropriate libraries must be addressed as soon as possible so that work can be done in creating said libraries if necessary.

Technical

- Unable to connect conversations between users of different devices (Android and iPhone).
 - *How to mitigate* - The risk has been discussed with the client, and it suffices that all iPhone users can have satisfactory conversations within their base.
 - *Risk level* - LOW - Because the client has decided to forgo this risk
- Lack of sufficient security enforcements concerning users data and conversations.
 - *How to mitigate* - Because the focus of the project is a functioning iPhone application, the client has agreed to delay secure transmissions.
 - *Risk level* - LOW - Because the client has decided to forgo this risk
- Missing appropriate libraries for protocols like XMPP and RTP to transfer communications for iOS
 - *How to mitigate* - If possible, we will strive to create much of the library as possible by ourselves
 - *Risk level* - HIGH - Because this risk entails a main portion of the project (particularly, the transfer of sound)

- Important layout/GUI difference between different platforms due to different technologies
 - *How to mitigate* - The client has stated that subtle deviations depending on iPhone capabilities from the provided design given by the design team is OK
 - *Risk level* - LOW - Because the client has addressed this risk
- Missing specific functionalities enabled in the Android environment, e.g. voice specialization, on iOS development, either due to hardware reasons or because of the difficulty of the task
 - *How to mitigate* - The client understands this, and will accept simple sound transfers between phones at the minimum
 - *Risk level* - LOW - Because the client has addressed this risk

Social Risk

- Users dissatisfied with the interface we have created on the iPhone side for this application.
 - *How to mitigate* - The client has agreed that user testing load will be moved away from us, and that we only need to implement designs that the Design team will give to us.
 - *Risk level* - LOW - Because the client has addressed this risk

Legal

- Ideally we would like to retain the copyright upon project handover. We have contacted our client with regard to that and are waiting for her response.
- Any copyright associated with used libraries will be notified to OpenComm.

~*~

⌘ Conclusion

We believe that our project is feasible after the extensive analysis. Our inception is very strong for a number of reasons.

1. Application of what we learned in class:

We will use agile development process so our progress will be steady and constant. We place strong focus on having effective communication with client and within the team to understand their real needs. We also try hard to follow a good software process.

2. Clear requirement and expectation:

The requirements and expectations on this project from our client have been thoroughly discussed and clarified. As reflected in our timeline, we have a pretty clear idea of what should be done at each time period. We will also keep a close communication with our client, so we can always make sure we successfully implement the desirable functionalities.

3. Supportive client:

Our client (Risa) and the whole design team are very supportive and well organized.

4. Careful consideration of potential challenges:

We also discussed internally and with our client about the potential technical difficulties we might run into, and have come up with different ways to mitigate them. We also give careful thought into legal and design aspect of the project.

5. Highly capable and responsible team:

All the team members have decent amount of programming experience. Most also have client facing experience or business background. We also have high self-standards in delivering quality product on time.

~*~