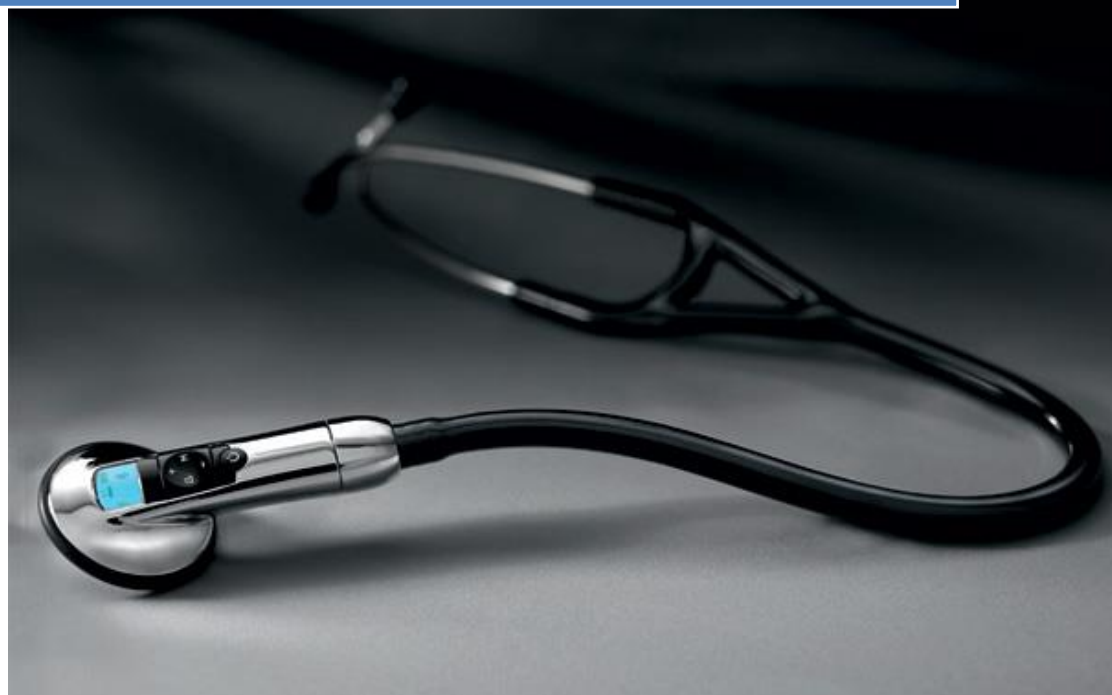


Advisor:
Dr. Akhilesh Tyagi

Client:
Joshua Gossett (3M)

Listen-In Design Document
Group Cardio (Dec13-12)
12/09/2013



Tyler Treat
Sean Strickland
Luke Pint
Alex Swanson
Daniel Heiniger
Juan Mozqueda

Table of Contents

| | |
|-----------------------------------|----|
| Problem Statement | 3 |
| System Description | 3 |
| Specification | 3 |
| Functional Requirements | 3 |
| Non-functional Requirements | 4 |
| Use Cases..... | 4 |
| Technology | 4 |
| Challenges..... | 5 |
| Deliverables | 5 |
| System Block Diagram | 6 |
| Module Descriptions..... | 6 |
| Android Client..... | 6 |
| UI Layer | 6 |
| Service Layer | 7 |
| Data Access Layer | 8 |
| Device Interface Layer | 8 |
| Sound File Server..... | 9 |
| User Interface Specification..... | 9 |
| Action Bar | 9 |
| Menus..... | 9 |
| Splash Screen | 9 |
| License..... | 9 |
| Dashboard | 10 |
| About..... | 10 |
| Stethoscope Manager | 10 |
| Pair/Unpair Stethoscope | 10 |
| Sound Recorder | 10 |
| Sound Manager | 11 |
| Phonocardiogram..... | 11 |
| Stethoscope Details..... | 11 |
| Test Specification..... | 11 |

Problem Statement

Many times, doctors are working away from their office and have a need for analyzing stethoscope data. In other cases, they are working with patients as a telepresence, which may require a nurse to listen with a stethoscope. Lastly, doctors often need to train medical students and provide guidance on how to accurately diagnose conditions.

In all of these cases, doctors and other medical professionals need a way to capture data with a phonocardiograph and a way to analyze it, possibly from a remote location. There is also a need for the playback of educational sound files so that students or other individuals being trained can hear what certain conditions sound like to accurately diagnose them.

System Description

This project consists of implementing an application targeting Android devices, specifically mobile phones and tablets. The system is geared towards healthcare educators as well as healthcare providers who would like to make it easier to analyze and diagnose patients on the go.

The application must be able to receive data from a Littmann 3200 Bluetooth stethoscope in a manner that minimizes ambient noise. With the data transmitted from the Bluetooth stethoscope, the application will render a phonocardiogram. Additionally, it will come with educational sound files so that users of various backgrounds can learn to better diagnose a patient's conditions. There is also a need for some integration with social media so that healthcare educators and providers can share sound files with websites like Facebook and Google+.

Specification

This section outlines the specification for the Listen-In project, including use cases, functional, and non-functional requirements. Use cases are presented as user stories since the project will be developed in an Agile environment.

Functional Requirements

1. Can pair a stethoscope to android device
 - a. Establishes a Bluetooth connection with Android Tablets
2. Can plot a phonocardiogram
 - a. Graphical representation of the sound data with axes, timestamps, guidelines
3. Can record stethoscope audio data

4. Can play sound files from tablet to stethoscope
 - a. Transmits data from the Tablet to the Stethoscope via Bluetooth
5. Can share sound files on social media
 - a. Can access Google+, Facebook profiles
 - b. Ability to upload/post information

Non-functional Requirements

1. Reduces noise
2. Gracefully disconnects the stethoscope
3. Works on Android 4.0 or greater

Use Cases

- As a healthcare professional, I want to pair a stethoscope to the tablet
- As a healthcare professional, I want to see a phonocardiogram so that I can better diagnose the patient with a condition
- As a healthcare provider, I want to play sound files, both audio recordings and educational sound files, from the tablet to stethoscope
- As a healthcare professional, I want to share non-confidential sound files on Google+ and Facebook for academic purposes.
- As a marketer, I want to change sound files via a server rather than through an application update

Technology

- **Java 1.6:** The Android client will be run on the Dalvik Virtual Machine, so the program will be written in Java, which is compiled to Dalvik bytecode.
- **Android SDK (API level 14):** The Android client will be built against API level 14 (Ice Cream Sandwich) of the Android SDK.
- **OpenGL ES:** The phonocardiogram will be rendered using the graphics API provided by OpenGL ES.
- **3M Stethoscope SDK:** A proprietary API for interfacing with the Littmann Bluetooth stethoscope provided by 3M.

- **Infinitum Framework:** An open source application framework for Android which provides facilities for dependency injection, aspect-oriented programming, and object-relational mapping.
- **JUnit:** A unit testing framework for Java.
- **Robolectric:** An open source unit testing framework that integrates with JUnit to enable test-driven development for Android.
- **Mockito:** An open source mocking framework that allows software components to be unit tested in isolation.
- **Apache Maven:** A software comprehension tool for managing builds and dependencies.

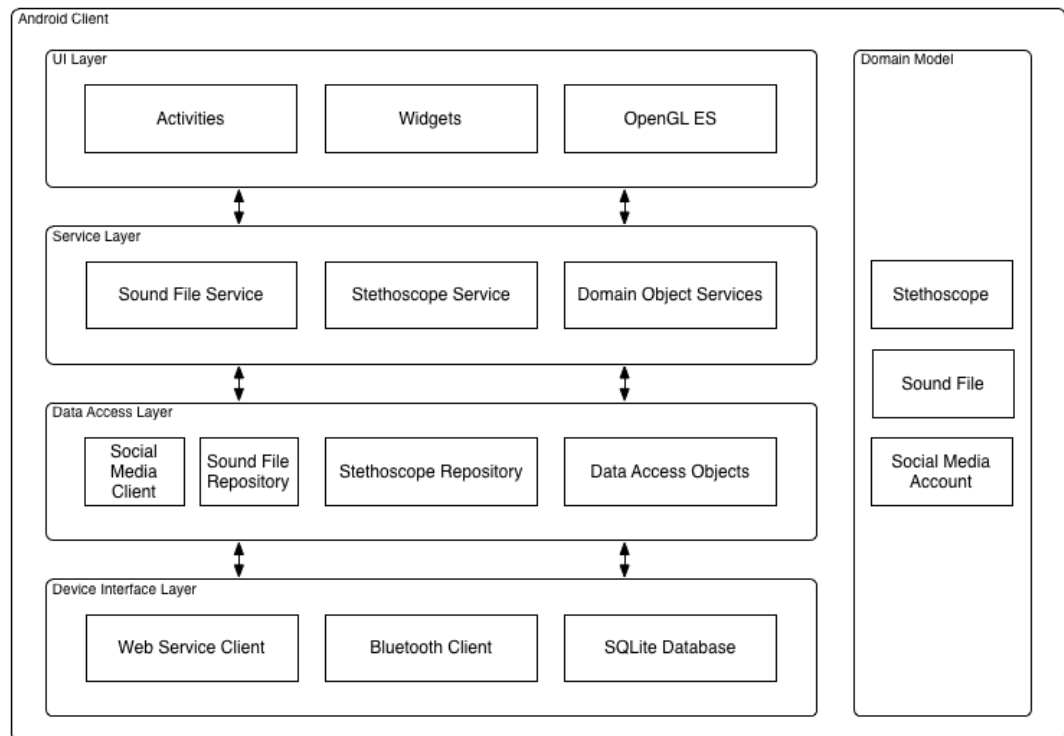
Challenges

- The customer has requested that the phonocardiogram be implemented using OpenGL, which is a 3D graphics library that does not natively support the drawing of text. This makes drawing a 2D phonocardiogram with axis labels non-trivial.
- The project was provided with an array of code already implemented. Some of it needs to be debugged, and other needs to be rewritten or removed.
- Stethoscope data comes in as a stream, which causes latency issues during debugging.
- Keeping incoming data and the graphical representation of the phonocardiogram synchronized.
- Sharing incoming data between multiple components of the application.

Deliverables

Our final expected product is a working Android application that will be used for mobile phones and tablets. The 3M Littmann Stethoscope will need to pair up with the user's device of choice. Once successfully paired, the user (such as a doctor) should be able to analyze the patient's phonocardiogram via the user interface. In addition, they should be able to record and share this information for later analysis and educational purposes.

System Block Diagram



Module Descriptions

Android Client

UI Layer

Activities

An Android activity is a single, focused unit of user interface. It provides a GUI in which the user will interact with and experience the application in a variety of ways.

Listen-In will include several activities which will be navigable by the user. As previously described, each activity will provide discrete pieces of functionality. The activities which are currently planned are listed below, along with a brief description of each.

Splash Screen: Displays version and branding information to the user and is used to initialize the application.

License: Displays an end-user license agreement to the user, requiring them to

accept the agreement before proceeding (once the agreement has been accepted, the user will not be prompted in the future).

Dashboard: Provides a main menu to the user, allowing them an easy way to navigate the application and quick access to Bluetooth device pairing.

About: Displays information about the application.

Stethoscope Manager: Provides functionality for pairing, unpairing, and nicknaming Bluetooth stethoscopes.

Phonocardiogram: Renders a phonocardiogram plot for the active stethoscope in real time. Also provides other functionality such as recording and saving audio from the stethoscope.

Sound Manager: Allows the user to manage (e.g. delete, rename, play, and share) audio files, including those recorded from a stethoscope or downloaded from the server.

Widgets

Widgets are reusable UI components used in activities. This includes dialog windows, audio player controls, stethoscope views, sound file lists, etc. These elements are critical to the maintainability of the application's UI as they are used in many different places.

OpenGL ES

OpenGL ES is a subset of the OpenGL API designed for embedded systems, including mobile phones and tablets. This module receives the data from the active stethoscope or recorded file and renders a phonocardiogram plot.

Service Layer

Sound File Service

Manages audio files on the device. This module handles operations such as audio recording and playback. The Sound File Service communicates with the Sound File Repository when it needs information from the database.

Stethoscope Service

Sends queries to the Stethoscope Repository module requesting data such as stethoscope pairings and readings from actively transmitting stethoscopes. This module includes operations for filtering, manipulating, and performing computations on the data to pass along to the UI Layer.

Domain Object Services

Responsible for calling the data access objects for access to the database. The operations should return the data requested or the status of request.

Data Access Layer

Social Media Client

Manages social media interaction between the UI Layer and Web Service Client. This module handles operations such as sharing sound files to Google+ and Facebook.

Sound File Repository

Manages audio files within the database. This module handles operations such as update and retrieval. The Sound File Repository fulfills requests made by the Sound File Service.

Stethoscope Repository

Uses the Bluetooth Client module to retrieve data for the Stethoscope Service module to query from. This module will act as a middleman for communication between the Service Layer and the Device Interface layer.

Data Access Objects

These objects will provide an interface to interact with the database. They will handle requests such as adding, updating, retrieving, and deleting data.

Device Interface Layer

Web Service Client

Provides a low-level HTTP client for interacting with RESTful web service APIs, including “external” social media APIs, such as Google+ and Facebook, and “internal” Listen-In APIs, such as the sound file storage service.

Bluetooth Client

Provides an interface between the application and external Bluetooth devices, such as the Littmann 3200 Stethoscope. The Bluetooth client provides facilities for scanning for devices, querying for paired devices, establishing RFCOMM channels, and transferring data to and from devices.

SQLite Database

Provides data about previously connected stethoscopes such as the nickname, serial number, and most recent connect time. The database will also contain a list of existing sound files. It will provide information such as the length, name, and size of individual sound files.

Sound File Server

File storage

A repository on a remote server for audio files. The files in the repository are accessible by the application via the Web Service Client. Files in the repository can be updated without requiring the application to be update

User Interface Specification

Action Bar

- An Action Bar is designated real estate at the top of a screen to display consistent navigation or tools in an application.
- Each view except License and About will contain an Action Bar.
- Items Include links and icons for:
 - Stethoscope Manager
 - Sound Manager
 - Sound Recorder
 - Phonocardiogram

Menus

- All menu type screens will span the entire view like tiles. This includes the Dashboard, Stethoscope Manager, Sound Manager, and Sound Recorder views.
- No menu will require scrolling to access extra elements.
- Each “Menu page” will implement its own settings button menu for common functionality on that view.

Splash Screen

- First screen on application launch/loading.
- Displays the version number, 3M’s logo, and other information to be decided.

License

- Displayed on initial launch.
- Contains an end-user license agreement for the user.
- Must click “Accept” button in the bottom left side of the screen.

- If the user clicks “Deny” in the bottom right hand side, the application will pop up an alert for confirmation.
 - Displays notice the application will exit
 - User can select from 3 options
 - Yes - Continue to Dashboard
 - No - Exit application
 - Cancel - Exit the alert and return to License view

Dashboard

- Displayed after the splash screen after application launch and the initial accept of the end-user agreement.
- Acts as a main menu with buttons covering all available screen space in tiles.
 - About Tile
 - Stethoscope Manager Tile
 - Phonocardiogram Tile

About

- Displays information about the application and other client-specified information.

Stethoscope Manager

- Contains tiles relating to stethoscope management displayed similar to the Dashboard.
 - Pair/Unpair Stethoscope Tile
 - View Stethoscope Details Tile

Pair/Unpair Stethoscope

- Will contain a list of all previously paired devices.
- Option to unpair a currently paired stethoscope.
- Option to pair a connected device.

Sound Recorder

- Contains tiles allowing the user to record and save audio from a stethoscope.
 - Record button
 - Will have a finish button and cancel button

- ◆ On finish the user will be prompted to enter a filename and save the recording or cancel

Sound Manager

- Contains tiles relating to sound file management.
 - Share Sounds
 - Listen to educational sounds on a server

Phonocardiogram

- OpenGL graphically created display of the sound streaming from the stethoscope to the tablet.
- Will contain a button to begin recording during the viewing of the phonocardiogram.
- Will contain a button to navigate to the sound manager.

Stethoscope Details

- View and change the Nickname of the currently paired stethoscope.
- View diagnostic information for the stethoscope.

Test Specification

The following features will be tested once they have been implemented:

- Verify user can connect/disconnect and pair/unpair stethoscope to/from tablet.
- Verify application can connect and pair automatically to the most recently connected stethoscope without user input.
- Verify database contains the most up-to-date information regarding recently connected stethoscopes such as nicknames, serial numbers, last connection date.
- Verify user can retrieve sound file information such as name, length, and size.
- Verify user can record and save sounds from stethoscope.
- Verify user can retrieve existing sound files through the application and play them on stethoscope.
- Verify user can share sound files on Google+ and Facebook.
- Verify application successfully receives the stethoscope's raw data, and plots a phonocardiogram using OpenGL.