l

**Lab 4 – Proximity Connect User Manual**

Team Yellow

Old Dominion University, Computer Science Dept.

CS411, Professional Workforce Dev. II

Professor J. Brunelle

April 30, 2024

Final Version

Table of Contents

[**1. Welcome to Proximity Connect 3**](#_heading=h.dsdtsct3b97m)

[**2. Getting Started 3**](#_heading=h.xp7jbrqmcvcn)

[2.1. Android Kotlin Application Starting Prompt Screen (O: Jeffrey) 3](#_heading=h.8v4illoyrlli)

[2.2. Android Kotlin Application Account Creation Screen (O: Jeffrey) 4](#_heading=h.ryl3lv9n28i)

[2.3. Android Kotlin Application Log In Screen (O: Jeffrey) 5](#_heading=h.huinle6lqpbi)

[**3. In App 6**](#_heading=h.hx0dqpoxnezw)

[3.1. Android Kotlin Application In App Home Screen (O: Rossy) 6](#_heading=h.w6kfmyxifu7s)

[3.2. Android Kotlin Application User Create Meeting Screen (O: Rossy) 7](#_heading=h.4eow2g4d7yfi)

[3.3. Android Kotlin Application Successful Meeting Created Screen (O: Rossy) 8](#_heading=h.3v357k3nckn1)

[3.4. Android Kotlin Application In Meeting Screen (O: Jeffrey) 9](#_heading=h.1rp9xegow4k)

[3.5. Android Kotlin Application Join Meeting Screen (O: Rossy) 10](#_heading=h.l9rdqyacondg)

[3.6. Android Kotlin Application Successful Meeting Joined Screen (O: Rossy) 11](#_heading=h.x6i4v6d5wpe5)

[3.7. Android Kotlin Application Bluetooth Screen (O: Milroy) 12](#_heading=h.kixlksn2nbjl)

[3.8. Android Kotlin Application Enable Bluetooth Screen (O: Milroy) 13](#_heading=h.t562zhwbi0wj)

[**4. Python Application 14**](#_heading=h.tt910215uqui)

[4.1 Python Application Starting Prompt (O: Spigner) 14](#_heading=h.tfor0d566di3)

[4.2 Python Application Processing and Insertion into the Database (O: Spigner) 15](#_heading=h.ajssr2ldm28z)

[**5. Django Website 16**](#_heading=h.79klvvc3uv7l)

[5.1. Django Website User Page (O: Milroy & Fulbright) 16](#_heading=h.aekv6j9pn1nk)

[5.2. Django Website Group Page (O: Milroy & Fulbright) 17](#_heading=h.3110jyferczg)

[5.3. Django Website Meeting List (O: Milroy & Fulbright) 17](#_heading=h.5dbgv3tzp758)

[5.4. Django Website Meeting Transcript (O: Milroy & Fulbright) 18](#_heading=h.b4k2ki6hshju)

[5.5. Django Website Analytics Page (O: Fulbright) 18](#_heading=h.77rdy7v4vq9)

[**6. Proximity Connect Site Map 19**](#_heading=h.7wp80u3ka5e8)

[**7. Glossary 20**](#_heading=h.v5st2qenb8fi)

# Welcome to Proximity Connect

Proximity Connect is a comprehensive solution that facilitates seamless and inclusive face to face experiences. This user manual details how to properly use and navigate throughout the Proximity Connect Application.

# Getting Started

The *Getting Started* is a description of all the screens a user shall encounter when attempting to create an account or log in. The *Getting Started* only applies to the Android Kotlin Application.

## Android Kotlin Application Starting Prompt Screen (O: Jeffrey)

1. Log In - The user clicks the *Log In* button and shall be taken to the[*Android Kotlin Application User Login*](#_heading=h.4eow2g4d7yfi) screen. (see section 2.3)

2. Join Now - The user clicks the *Join Now* button and shall be taken to the *Android Kotlin Application Account Creation* screen. (see section 2.2)

3. Continue as guest - The user clicks the *Continue as guest* button and shall be taken to the *Android Kotlin Application In App Home* screen. (see section 3)

## Android Kotlin Application Account Creation Screen (O: Jeffrey)

1. Enter Email - The user enters the email address they want to register to their account.

2. Enter Phone Number - The user enters the phone number they want to register to their account.

3. Create Password - The user enters the password they want to register to their account. Passwords require at least 1 uppercase letter, 1 number, and 1 special symbol.

4. Re-Type Password - The user enters the password they entered into the *Create Password* field.

5. Sign In - The user clicks the *Sign In* button and shall be taken to the *Android Kotlin Application In App Home* screen. (see section 3)

## Android Kotlin Application Log In Screen (O: Jeffrey)

1. Email - The user enters the email address they want to log in to their account with.

2. Password - The user enters the password they want to log in to their account with.

3. Log In - The user clicks the *Log In* button and shall be taken to the[*Android Kotlin Application In App Home*](#_heading=h.4eow2g4d7yfi) screen. (see section 3)

4. Continue as Guest - The user clicks the *Continue as guest* button and shall be taken to the *Android Kotlin Application In App Home* screen. (see section 3)

5. Don’t have an Account? Sign Up - The user clicks the *Sign Up* button and shall be taken to the *Android Kotlin Application Account Creation* screen. (see section 2.2)

# In App

The *In App* is a description of all the screens a user shall encounter after logging in to the app via log in or account creation. The In App only applies to the Android Kotlin Application.

## Android Kotlin Application In App Home Screen (O: Rossy)

1. Create Meetings - The user clicks the *Create Meetings* button and shall be taken to the *Create Meetings* screen (see section 3.1)

2. Create Groups- The user clicks the *Create Groups* button and shall be taken to the *Create Groups* screen

3. Bluetooth - The user clicks the *Bluetooth* button and shall be taken to the *Bluetooth* screen (see section 4.3)

4. Join Meetings - The user clicks the *Join Meetings* button and shall be taken to the *Join Meetings* screen (see section 3.2)

5. Join Group - The user clicks the *Join Group* button and shall be taken to the *Join Groups* screen

6. Transcripts - The user clicks the *Transcripts* button and shall be taken to the *Transcripts* screen

7. Logout - The user clicks the *Logout* button and shall be logged out of their account and is taken back to the *Android Kotlin Application Starting Prompt* screen (see section 2.1)

## Android Kotlin Application User Create Meeting Screen (O: Rossy)

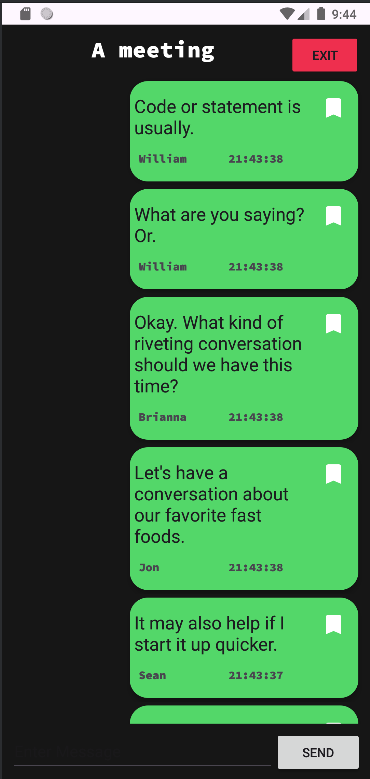
1. Enter Meeting Name - The user enters a meeting name for the meeting they want to create

2. Create Meeting - The user clicks the *Create Meeting* button and shall be taken to the [*Android Kotlin Application User Successful Meeting Created*](#_heading=h.l9rdqyacondg) screen (see section 3.3)

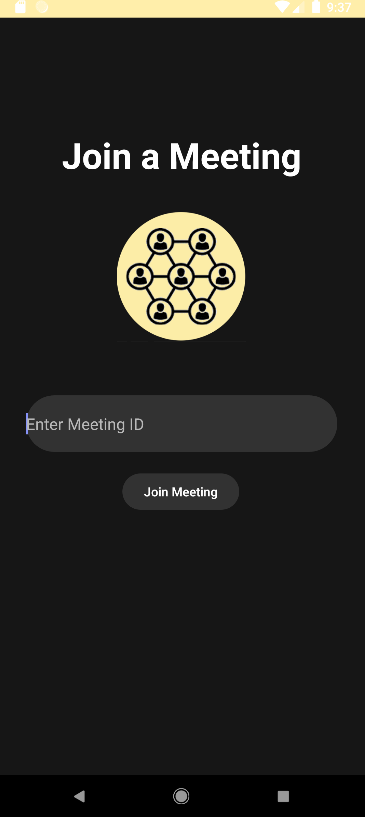
## Android Kotlin Application Successful Meeting Created Screen (O: Rossy)

1. To In Meeting - The user clicks the *To In Meeting* button and shall be taken to the *Android Kotlin Application In Meeting* screen. (see section 3.4)

## Android Kotlin Application In Meeting Screen (O: Jeffrey)

1. Exit - The user clicks the *Exit* button and shall exit the meeting room and be taken back to the *Android Kotlin Application Home* screen (see section 3.1)

## Android Kotlin Application Join Meeting Screen (O: Rossy)

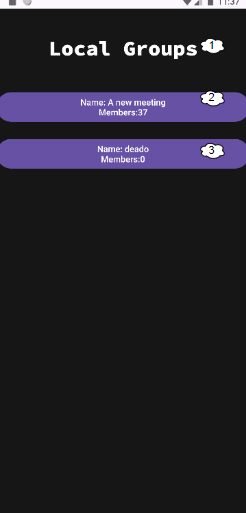
1. Enter Meeting ID - The user enters a meeting ID for the meeting they want to join.

2. Join Meeting - The user clicks the *Join Meeting* button and shall be taken to the [*Android Kotlin Application User Successful Meeting*](#_heading=h.l9rdqyacondg) Joined screen (see section 3.6)

## Android Kotlin Application Successful Meeting Joined Screen (O: Rossy)

1. To In Meeting - The user clicks the *To In Meeting* button and shall be taken to the *Android Kotlin Application In Meeting* screen. (see section 3.4)

## Android Kotlin Application Bluetooth Screen (O: Milroy)

1. Local Groups - This screen allows the user to see local meetings being held nearby via a bluetooth low energy mesh network. Provides a selectable list of detected meetings for an easy way to find and join local meetings. If Bluetooth is not enabled on the device it prompts the user to re-enable bluetooth. (see section 3.8)
2. Name - Displays the name of the discovered meeting. The user clicks the *Name* button and shall be taken to the *Android Kotlin Application In Meeting* screen corresponding to the meeting name in the button. (see section 3.4)
3. Members - Displays the number of users currently within the meeting being held.

## 3.8. Android Kotlin Application Enable Bluetooth Screen (O: Milroy)

1. Disabled Bluetooth Status - The status of the bluetooth on the smart device will be displayed should it be disabled.
2. Enable Button - The Enable Button allows the user to press the button to re-enable the bluetooth on their device to see available local meetings. (see section 3.7)

# Python Application

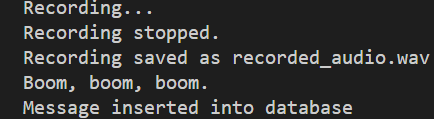
The *Python**Application* is a supportive tool for the Proximity Connect App. This program is used to create a real-time transcript by recording the user’s voice and inserting a text version of their speech into the database.

## 4.1 Python Application Starting Prompt (O: Spigner)

# 

1. User ID- Upon first launching the program, the user will be prompted for their User ID. The user types their User ID and presses enter to submit.
2. Meeting ID- The user is then prompted to enter the meeting ID of the meeting they wish to participate in. The user types their Meeting ID and presses enter to submit.
3. Press And Hold Spacebar- The user is prompted to press the spacebar to start recording from their microphone. The user is notified that when the spacebar is released, the recording ends.

## 4.2 Python Application Processing and Insertion into the Database (O: Spigner)



1. Recording Captured- The application notifies the user when the recording is stopped, and when the recording is saved as recorded\_audio.wav within the terminal.
2. Message Body- After a variable amount of time, usually a few seconds, the audio has been processed by the transcription API, and a text version of the user’s speech is displayed to the terminal for quick user review.
3. Database Confirmation- If the message text is not empty, the terminal displays a confirmation of the message’s database insertion.

# 

# Django Website

The *Django Website* is a description of all the pages a user shall encounter after logging in to the Django Website.

## Django Website User Page (O: Milroy & Fulbright)

1. Users List- The List displays each registered user located within the database.
2. User - Each user is displayed with their username and the associated user\_ID located within the parenthesis.
3. User Hyperlink - By selecting a user, the link will redirect to the user’s group page. (see section 5.2)

## Django Website Group Page (O: Milroy & Fulbright)

1. Group List - List displays each group the user is in
2. Group - Each group is displayed with their associated group name and group\_ID located in the database.
3. Group Hyperlink - By selecting a group, the link will redirect the to the group’s meeting page. (see section 5.3)

## Django Website Meeting List (O: Milroy & Fulbright)

1. Meeting List - List displays each meeting the group has had.
2. Meeting - Each meeting is displayed with their associated meeting name and the meeting\_ID located in the database.
3. Meeting Hyperlink - By selecting a meeting, the link will redirect to the meeting transcript page. (see section 5.4)

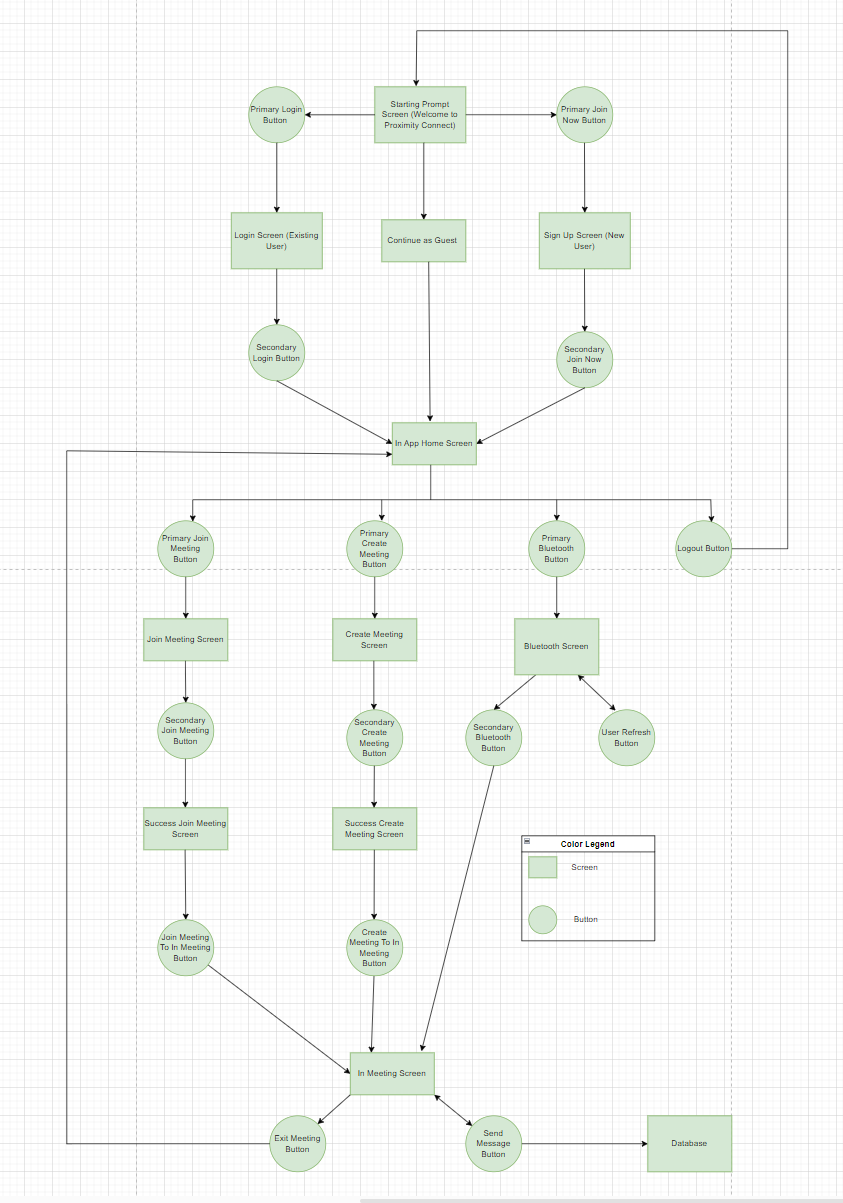
## Django Website Meeting Transcript (O: Milroy & Fulbright)

1. Meeting Transcript - Contains all messages that have been transcribed and sent to the database.
2. Message - Each message contains each of the following:
   1. Date - The date the message was sent
   2. Time - Time the user started speaking down to seconds.
   3. User - Who sent the message
   4. Message - What was said by the user.

## Django Website Analytics Page (O: Fulbright)

1. Analytics - Contains information about the usage of the application. The information is broken down into:
   1. Messages - The number of messages sent based on time scale.
   2. Meetings - The number of meetings made based on time scale.
   3. Groups - The total number of groups made.
   4. Accounts - The total number of accounts made based on time scale.

# Proximity Connect Site Map



# 

# Glossary

**ANDROID**: An operating system developed primarily for mobile devices by Google, characterized by its open-source nature, diverse app ecosystem, and customizable user interface.

**ANDROID STUDIO**: An integrated development environment (IDE) designed by Google for

Android app development, providing tools, emulators, and a comprehensive environment for coding, debugging, and testing applications for the Android platform.

**API**: A software intermediary that enables applications to access and utilize specific functions or

services provided by another software or platform.

**BLUETOOTH**: A wireless technology standard used for short-range communication between

devices, enabling data exchange and connectivity without cables.

**C#:** A programming language developed by Microsoft, primarily used for building a wide range of applications on the .NET framework.

**Coqui Ai**: an open-source initiative providing accessible tools and models for speech recognition

and natural language processing.

**CSS**: A style sheet language used to define the presentation and formatting of HTML (Hypertext

Markup Language) and XML (Extensible Markup Language) documents.

**Deaf**: Lacking the power of hearing or having impaired hearing.

**GIT**: A distributed version control system that enables multiple developers to collaborate on projects by tracking changes to source code, managing revisions, and facilitating the merging of different versions, enhancing team collaboration and software development workflows.

**GitLab**: A web-based platform for DevOps lifecycle management, providing a comprehensive set of tools for version control, issue tracking, continuous integration, and deployment.

**Hard of Hearing**: Not able to hear well.

**HTML**: The standard markup language used for creating web pages and structuring their content by employing a system of tags to define elements such as text, images, links, and multimedia.

**IDE**: A software application consolidating various tools and features necessary for software

development, typically including a code editor, debugger, build automation tools, and a compiler or interpreter.

**iOS**: An operating system developed by Apple for use on mobile devices like iPhones, iPads, iPod touch, etc. It provides the interface and framework for running applications and managing the device’s hardware and software.

**JAVA**: A programming language known for platform independence, allowing developers to write

code that can be run on various devices and commonly used for building applications, from mobile apps to web applications and large-scale enterprise systems.

**JAVASCRIPT**: A programming language used primarily to create interactive effects and dynamic website content, enabling functionality such as forms, animations, and user interactions with web browsers.

**JUnit**: A tool used in programming for testing Java code to ensure it works as expected, helping

developers identify and fix errors or bugs in their programs.

**MySQL**: A type of database software that helps store and organize information, commonly used for websites and applications to manage data efficiently.

**Open API**: A specification for building APIs that facilitates easy comprehension and interaction.

with web services. Enables developers to define, document, and describe Restful APIs using a structured and standardized format.

**REACT**: A popular JavaScript library used for building user interfaces in web applications. Enables developers to create interactive and reusable UI components.

**RNNoise**: An open-source, real-time noise suppression software that uses signal processing and

deep learning.

**TRELLO**: Web-based project management tool that uses boards, lists, and cards to organize tasks and facilitate collaboration among team members.

**VISUAL STUDIO**: An IDE made by Microsoft that supports various programming languages.