Report for Principles of Software Development – Java Android app seminar project

My name: Erik Moravec

App name: Super Proximity Messenger

Technologies and API – For the purpose of implementing this application and especially the networking part I chose to use the Nearby Connection API designed by google. This API is designed as a peer-to-peer network model and utilizes multiple connectivity technologies, mainly Bluetooth. For the UI part I used the new AndroidX package that provides new, enhanced and modern UI components and is the recommended way today.

Application – As shown in the presentation, I have created multiple activities and designed corresponding screens and their layouts. The main screen displays a welcome text for the user and contains two big buttons that are used to start the communication. One of them is used to begin searching for nearby devices that already started advertising and are considered “Hosts”. The other one starts the advertising and allows “Clients” to connect.

The chat room is then activated by pressing the floating action button with the chat icon if you are a host and some clients are connected. If you are a client, you open the chat room by clicking on the discovered host room. Once the chat room is opened, you can write a message by clicking to the edit box and then send it by clicking the send button on the right. On the top there is a toolbar with the title consisting of names of participants in the chat room (there can be more of them). Your messages are displayed on the right side of the screen and other messages are on the left. You can leave the chat by pressing either the system back button or the back button on the toolbar.

Actually, all of the screens present in the application contain a toolbar at the top, which is a typical design element in Android and helps to provide a common “Look and Feel” for Android applications. On all screens where it makes sense (except the main screen) this toolbar also contains a back button arrow to help with navigation in the application. Aside from that, the toolbar always contains a title for the user to know where in the app he is right now. Moreover, on multiple screens, where it makes sense, there is also an overflow menu (another typical aspect of a native android application) on the top right corner of the screen, in the toolbar that contains additional functionality.

This overflow menu is especially important on the main screen of the app. It can be used to access other activities by clicking on its items.

First of these activities is the “About you” activity, which is a fancy name for a profile information screen. Here the user can set information about himself. This information is then visible to other nearby users when they discover nearby devices and connect with them in a chat room. The information that can be set here is the name of the user (the one he wants other users to see), the sex of the user and a short profile info/description of the user called Bio. This screen also contains an overflow menu where the user has an option to reset all the information to the default values.

The second activity accessible from the main screen through the overflow menu is the settings activity. Here the user can change the look of the application by choosing a completely custom color to be used as the main color theme of the application. This color not only determines the color of the toolbar in the app, but also the color of the android status bar while the app is open. Again, these settings can be reset to the default values in the overflow menu of this screen.

The third activity is called "Message yourself". It allows the user to enter the chat mode without the need to first initialize a connection. It can be used for testing purposes of the UI and for the user to play around with the app and try how it works.

Future Plans (possible extensions and improvements):

- Better visualization of other users profiles (show their bio, …)
- Possibility to upload a profile picture and make others see it
- Night mode for the application UI
- Database utilization for better message and chat persistence
- Ability to send Files and Images

Context – Well, I understood that we were supposed to develop an android application that works as a messenger/chat but does not use the internet connection as usually. It should only use other means of communication such as Bluetooth that require proximity of the users, but don't need internet connection. We were supposed to find, choose, study and integrate the API all on our own. As far as I know, there were no other specifications or requirements given for the application.

Implemented lecture concepts – 1. Activities – There is a bunch of activities created and designed in the application.

2. Views (Buttons, TextFields, Edittexts, RadioButtons, ActionFloating Buttons, Toolbars, …) – All the UI is designed using the view elements, as the components are called on android. Most of the is implemented using the standard and modern AndroidX, however also some custom (chat message bubbles) and third party views (color picker) are used.

3. Intents – used for the navigation and for starting new activities.

4. Shared Preferences – used to store some user data, such as the profile information and settings.

5. Manifest – Used to define permissions, activities, …

6. Composite, Observer and Adapter design patterns

7. Layouts – Relative and Constraint Layouts are used

8. Services and Threads – I have experimented and played with them quite a lot, but finally decided not to use them as I found them unnecessary and maybe even counter-effective.

Erik Moravec