Optimal State

5/5/18

CEO:	Dr.A.I Concepcion
Prepared By:	David Cruz Eric Blasko

Section 1: File Structures

Our file structure is well kept and managed using Git through the terminal. All of our files are saved on a repository on Bitbucket in order to make it possible for all group members to freely access and branch off to make changes without jeopardizing the master files.

LoginActivity.java

This Activity is the starting activity for the Optimal State Application. This Activity allows for existing users to login to their accounts. If a user does not have an account they will select the register text which is displayed at the bottom of this Activities interface. This will redirect them to the Register User Activity.

RegisterActivity.java

This screen display multiple blank text fields which then the User is supposed to input their information to the corresponding field. This information is then used to create an account for the user.

UserMain.java

This activity is the main interface for general users. The screen will greet the user with their first and last name. The user will then have four options from which they can choose From options will include exercises, history and settings. The user may also choose to log out of their account.

AssessmentActivity.java

This activity allows the user to choose from four different section based on how they currently feel. The user may select any combination of the three outside sections or the center region by itself.

ExerciseActivity.java

This activity provides users with information that can help improve or maintain their current mental state. The interface provides a three tier expandable list which provides information based on the users recent Assessment value. Different assessment values will provide different results.

ViewHistory.java

This screen displays all of the users previous assessments and post assessments which they have taken.

Settings.java

This screen displays four buttons to choose from, those buttons being : Change User Info, Coming Soon, Share Settings, and Main. The buttons redirect the User to their Corresponding screens.

ChangeUserInfo.java

The ChangeUserInfo screen provides the User with the ability to input new information for FirstName, LastName, and Weight. The screen also displays three buttons: Reset Password, Save Changes, and Back.

ShareSettings.java

The ShareSettings screen displays a list of other users for the User to choose from. Those which are chosen are given the ability to view the User's information which is limited to only assessments.

ProviderMain.java

This activity is the main interface for providers. The screen will greet the provider with their first and last name. The provider will then have four options from which they can choose from. These include add client, remove client, view client history, and settings. The provider may also choose to log out of their account.

ProviderAddClient.java

The ProviderAddClient screen displays three buttons: Search for Client, Add Client, and Back. Each of the buttons takes the Provider to their corresponding interfaces, the screen also displays a blank text field in which the Provider can input a UserID to search and add Clients.

ProviderRemoveClient.java

The ProviderRemoveClient screen displays a list of currently available clients, with each client having a corresponding checkbox allowing the provider to select one or multiple clients. The screen also displays two buttons: Remove, Back.

ViewClientHistory.java

This screen displays assessment and post assessment history for all clients which the provider chooses to view.

Section 2: Instructions

The application can be run using one of the following options listed below:

1. Using Android-Studio emulator
2. By downloading the APK and then installing it onto an Android device.

To Run the app on the android studio emulator, the following steps must be taken:

1. First step is you must have access to the team repository in order to download all necessary files.
2. Once files are downloaded, ensure all files are saved into one folder.
3. Once all files are saved and stored in a folder now you can go on Android Studio and go to File > Open - and select the folder in which you saved all the files.
4. Ensure once the folder is opened that all files are properly opened and ensure the gradle was built properly.
5. Ensure that google play services is installed and updated. This is done through tools-> apk -> apk tools.If google play services is checked the next step can be taken. If it's not checked, select it and select install.
6. Finally, Click the PLay button on the top right corner and then select the device emulator you wish to run the app on.

To Run the app on an Android Device, the following steps must be taken:

1. To create a new APK, in android studio's, click build-> build signed APK. Select location to save.
2. If using existing APK, Download the APK for the project
3. Open your Apps Tray. It's usually a button with a grid of dots ⋮⋮⋮ at the bottom center of your screen.
4. Tap File Manager.
5. Tap Downloads.
6. Tap the APK file you downloaded.
7. Tap INSTALL in the lower-right. The APK file you downloaded will be installed on your device.

Section 3: Good Implementations

Positive and novel implementations of the app:

- The app is well thought out and has a pleasant appearance.
- The app is fast and responsive, with a simple layout
- The app is well-designed and easy to use without much guidance or instructions
- The app has large amounts of research to support its features and content
- The app's source code contains error handling cases to prevent unexpected crashes.
- The app contains help menus to help users who might be confused about how the app works.

Section 4: Needs Improvements

The following are weak points of the current implementation:

- The app currently does not have an iOS counterpart.
- The app currently has a limited database as we are using a Firebase from google

Those weak points will be addressed by our team as we intend to continue development.

Section 5: Overall Recommendations

Regarding future development, our team is already currently addressing that matter. New features will be added and an iOS version will also be developed. That being said, some recommendations for future development are the following:

- Increased time spent on development and design.
- Well thought out and functional interfaces.
- Implementation of features which would provide users with a great and easy experience while using the app.
- More team members to help with workload.