**README**

**Emergency Contact:**

Jagtej Sodhi (Software Development Lead)

(510) 709-7669

jsodhi@ucsd.edu

Shazzy Gustafson (Project Manager)

(619) 823-4255

sagustaf@ucsd.edu

**Phone Specifications:**

\*Please run our app, WineKnow on the following type of phone:

Samsung Galaxy i9300 SIII

Android version 4.1 Jellybean

**Installation**

1. Open up the terminal
2. Prep cs110s
3. Run eclipse: **eclipse &**
4. Choose your default workspace
5. Under eclipse, select File -> Import -> Android -> Existing Android code
6. Click 'Browse' and navigate to the folder where WineKnow is located (should be under the TITS -> Source Code)
7. Single click on the WineKnow folder to select it and click OK. You should see “WineKnow” checked off under Project to Import.
8. Click Finish and wait for the project to show up under the package explorer in eclipse.

**Running the application on your Android Phone**

1. Plug your phone into your computer
2. Wait 5-10 seconds for the machine to recognize your phone.
3. Right click the WineKnow project under the package explorer and select Run As -> Android Application.
4. Under this screen, be sure to select your Samsung Android device and press Okay.
5. Allow Eclipse 10-15 seconds to install the application onto the phone. Make sure your phone is awake and the screen is on at this time.
6. Once installation is complete, WineKnow should automatically open up on your device.
7. Enjoy WineKnow!

**Model View Controller/Layering**

Our application implements model view controller in a sense that each different Activity is a separate view and the DatabaseHelper is out controller. The Activity views then make calls to the controller as they are also Observers looking for a change. In reality, they use call-backs, meaning the Android API is the observer that then calls the controller based on changes to a view. Our controller also acts as a mediator to the views as it was most logical for layering to keep the database functionality in one separate layer from the other moving parts. And as our controller only acts a mediator when the View changes, this doesn't result in parts that our overly coupled. Our database in then the model as it contains the data we want to change and move around our application, ratings use the controller to update the model (in the database), and other activities generally only change what is viewed from the model and the controller helps to mediate getting the information needed from the model.

**Information to know when running our app**

-The android back button will not take you back to the previous screen; it will exit out of the app.

-View Ratings only allows you to view the first 5 wines you rate, so if you rate more than 5 they will not appear