

Calorie Counter

Ryan Hamacher

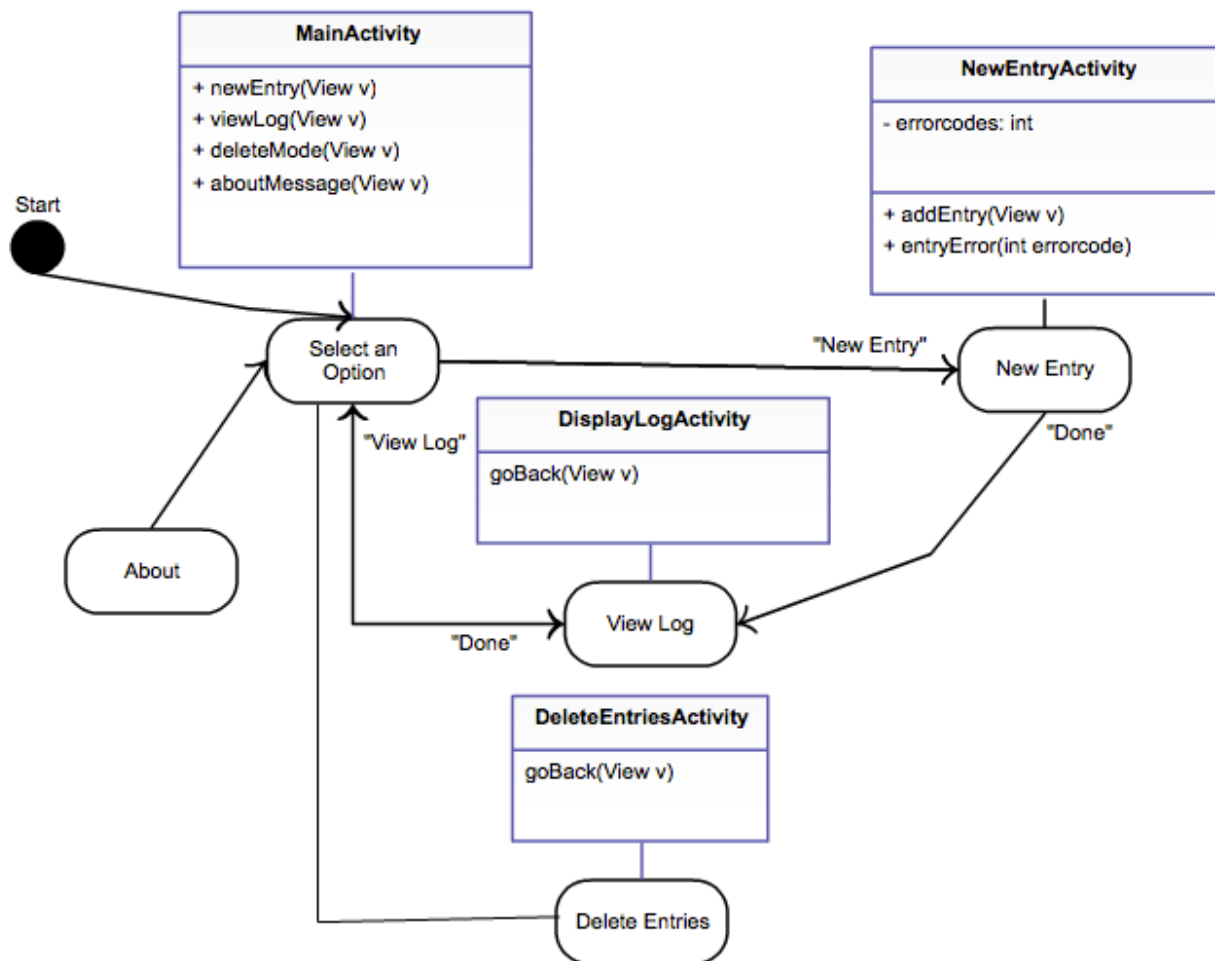
rhamache

1278626

Calorie counter is a simple calorie logging app. You can use it to maintain a basic log of calories you have consumed. Calorie counter can track the date and size of meals you have consumed, and provides some simple statistics about your overall consumption.

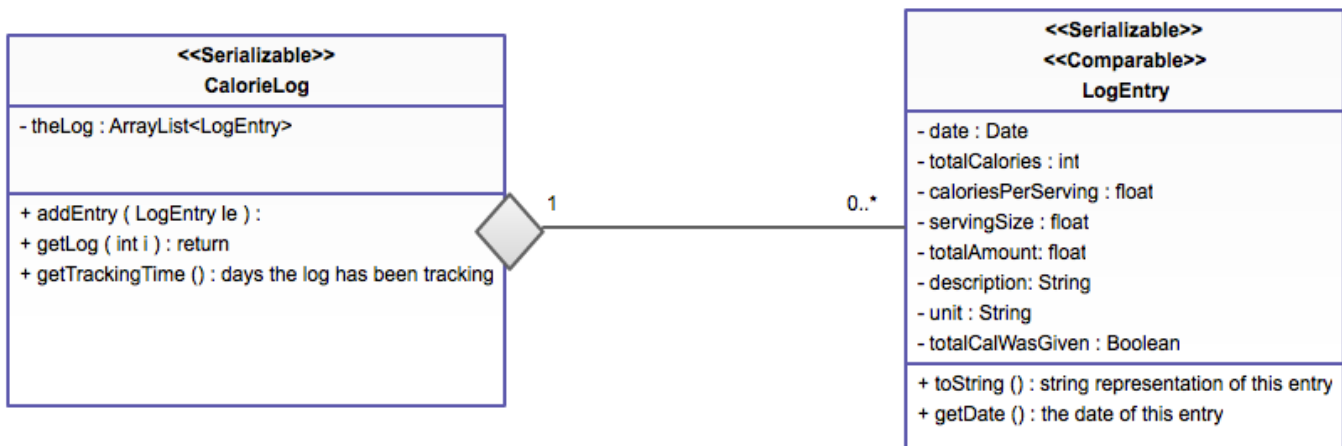
ACKNOWLEDGEMENTS: During the creation of this application I consulted the android developer's guide at <http://developer.android.com/>. I also had some help using the SimpleDateFormat class from Motiejus Osipovas. Please see the source code for the specific code he helped me with.

User Interface



The UI for this app begins by loading MainActivity. From here the user is presented with the main menu where they can select 'New Entry', 'View Log', 'Delete Entries', or 'About'. 'New Entry', 'View Log', and 'Delete Entries' all have activities and views associated with them. 'New Entry' brings up several EditText fields in order to create a new log entry. It also does error checking and assists the user in providing proper input. 'View Log' uses an ArrayAdapter to adapt the calorie log to a ListView object in order to be displayed on the screen. 'Delete Entries' brings up a view similar to 'View Log' with the difference being that an entry is deleted upon being clicked. 'About' brings up a dialog box with authorship information. One important note about the UI classes is that MainActivity contains the methods save() and load() for reading and writing the log to a file on disk.

Data Model



CalorieLog is responsible for storing the log entries. It implements an ArrayList to accomplish this. CalorieLog also contains methods for adding/removing entries and calculating statistics about the log, such as total calories across all entries or the total tracking time.

LogEntry is used to represent one log entry. Each entry stores all its information such as date, calories per serving, or description. LogEntry also contains methods for getting each attribute as well as toString() to easily represent a single entry in String form. As can be seen above, each CalorieLog object can be associated with any number of LogEntry objects, but each LogEntry object should only belong to one CalorieLog.

LogEntry implements comparable so that the entries can be sorted within the CalorieLog class. Entries are compared on the basis of date, with later dated entries being sorted to the front or lowest elements of the calorie log. Both these classes implement serializable as well so that they can be saved as objects.