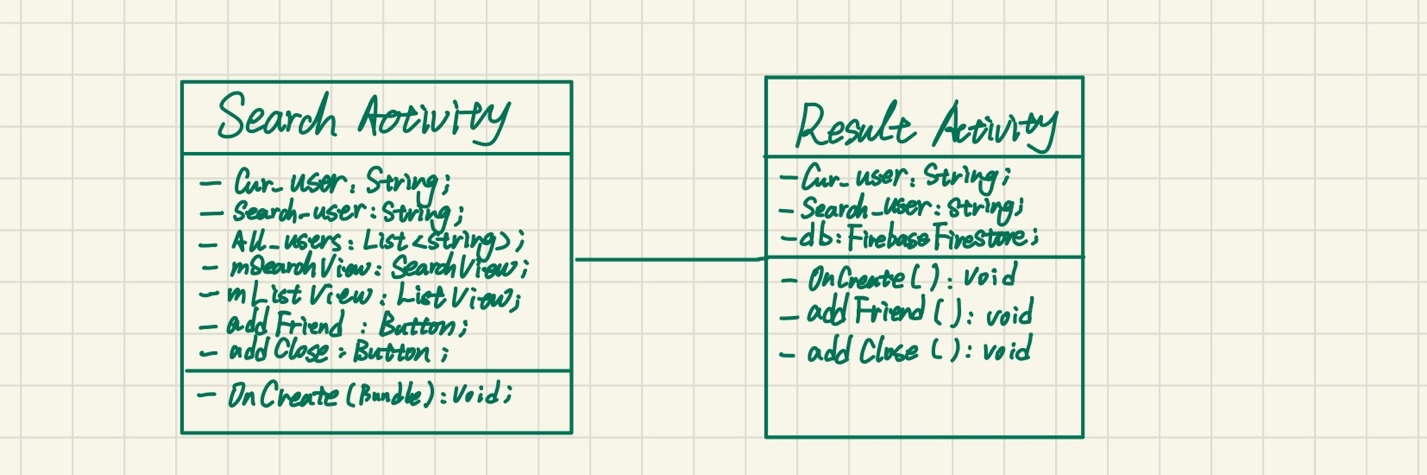
**Design**

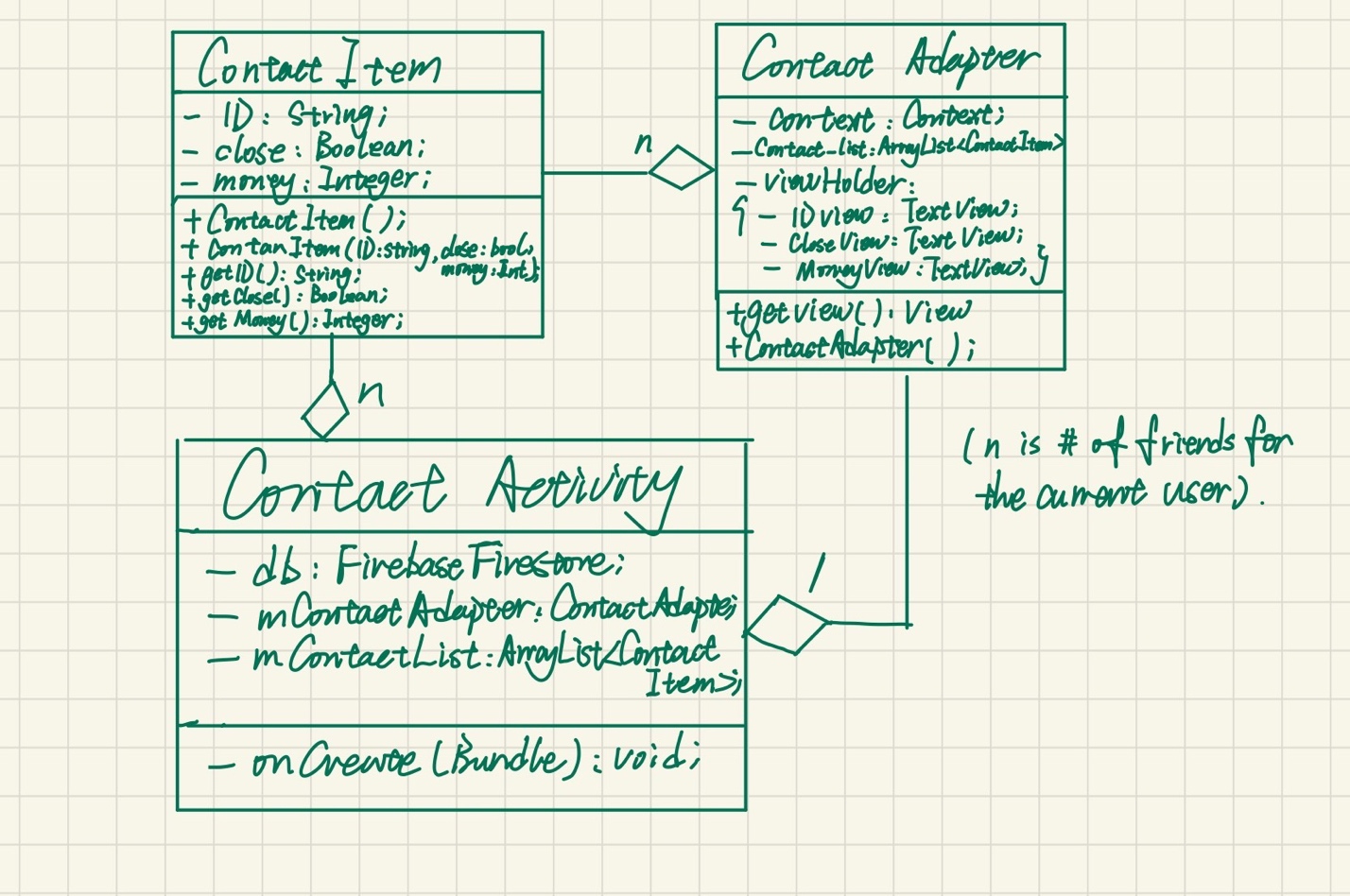
**Search Activity**

****

There are two classes for search, Search Activity and Result Activity. In the Search Activity class, there is the listview adapted by all user names in database, which will update depends on the standard input. The interface for search activity contains the search bar and the dynamic listview. Clicking the username on the list or clicking search button will bring user to the result activity which displays corresponding user portfolio page. The current user can see some accessible information about searched user. There are two functions in the result activity which can add the friend or the close friend. The attribute values in database will be updated. If the current user clicks to add the searched user as his or her friend, both contact lists for these two users will be updated in the database, same for clicking add close friend.

To map the layered architecture, the search bar, result portfolio page, and the add friend buttons are accessible for users as the presentation layer. On the business logic layer, search activity class will adapt the listview with the database collection for user. And result activity can update the database values by setting the click listener. The database layer is the cloud firestore which contains the accessible contact collection and user collection.

**Contact List Activity**

****

Since all the main functional properties of the application are achieved base on the social network for the user, I design the contact list activity. There are three classes, Contact Activity, Contact Item, and Contact Adapter. In the Contact Item class, each object represents a specific friend of the user. There are some attributes such as the friend’s name, the settled debt to this specific friend, and if this friend is the close friend. To achieve encapsulation, I set these attributes private and provide the getters for them. In the Contact Adapter class, I use the adapter (decorator) design pattern with the view holder to wrap the data for the listview to display. This class corresponding with the externally-visible interface which displays each item on the list view (contact list). The Contact Activity get the current user name from previous activity and adapt all his or her friends’ information from the database (cloud firestore) to the listview using contact adapters. For each friend, there are two buttons for remove the friend from contact list and set off the close friend. For display, clicking the “contact list” button from previous page will bring the user to his or her contact list.

In terms or the layered architecture applied, the presentation layer contains the contact list and user can remove the friend or close friend by click the button. The business logic layer corresponding to the contact activity which fetches friend information from the database and change the attribute values in database using click listener. The database layer is the cloud firestore which contains the contact collection for all users and their friends’ information.