Outline

I’ll be implementing “*Stock Analysis System*” App.

The application is designed to make it easier for investors to check the stock market on their mobile devices.

Description:

1. When the user opens the application, a splash screen that lasts three seconds displays and three seconds after that goes to the stock list and search screen. The method used is PostDelay() in the Handler class(Handler is the UI thread that receives information and updates the UI interface).
2. The main interface consists of a list, a stock code input box, a search button.

(1). The list can be used to add several of my stocks, each item consisting of the stock code, the stock name, the current price, and today's increase (today's drop). And the interface will refresh every 10 seconds.

(2). The input box can be used to input the code of the stock that you want to add to complete the adding of the new stock. The search button can communicate the input content of the text box with the background data and complete the adding. Each piece of code that is entered is validated before the query to prevent the user from entering the wrong code or malicious input, such as multiple Spaces or line breaks, or the user entering code that does not exist.

1. Of course, you can delete the stock that you do not need, but the dialog will pop up before the deletion to remind you whether to delete or not. This is done by long pressing on the selected item
2. When you click on each item in the list, a session pops up that displays detailed information about the stock, including the current price, the stock code, today's opening price, yesterday's closing price, today's lowest price and today's highest price. And the corresponding K diagram.
3. There are two buttons in the session, one is to refresh, you can refresh the current stock information by sending a query request again; The other is return, which can be returned to the list.
4. Clicking on the K diagram will enlarge the image.
5. After each pause and delete, the information of the stock code and all the information of the last refreshed stock will be stored in the local file through internal storage. After each re-opening of the application, the data in the file will be reloaded into the corresponding ArrayList and displayed in the ListView. According to the characteristics of Chinese stocks, a rising stock will be in red, a falling stock will be in green.
6. There is also a tabHost in the interface, and there are two tabLabels. Page one is used to place the list and query box. Page two is used to display the total points of Shanghai stock index, Hong Kong Hang Seng index and Nasdaq index, the current price and the rise rate. And the corresponding chart. At the same time, the image can be clicked to enlarge, and the enlarged image is displayed with dialog
7. Notifications can be turned on or off by clicking on menu in the system. When the number of stocks in the list increases by more than 2% and notifications are turned on, the user is notified, "XX increases by 2%." Click the notification content to go directly to the stock list page.
8. Widgets can be added, and widgets refresh every 20 seconds to get the two stocks with the highest current price increase and display their stock code, stock name, current price and price increase. The widget also has an "enter" button that takes you directly to the application
9. All the data are based on Sina stock data source, which needs to be parsed. Also use internal storage to save the correct stock code information as a local file.
10. When the screen is portrait, the interface will be displayed as tabHost, while when the screen is landscape, the fragment will be displayed, and when the list items in the list are clicked, the stock details will be displayed on the right side instead of a dialog box. And portrait can achieve the functions, such as: add and delete, can be achieved in landscape.

Schedule:

Week9:

Complete the basics of creating an application interface, using previous knowledge, complete the list, buttons, input boxes, custom sessions, and Adapter(using BaseAdapter) code. Create a class (called EachStock) to hold the data for each stock, along with a List to hold each new object (belonging to EachStock) together. Parse the stock data and store the stock code in the local file. Access to Sina stock data resources

Week10:

Complete the operation of adding and deleting the stock code, and set the timing reminder through Service and notification.

Week11:

Achieve 10 seconds to refresh the interface, Add a widget

Week12:

Use internal storage to achieve permanent storage, complete the design of different models (mobile phones and tablets, Land and Port), and optimize the UI interface, do Land screens with fragment, do Port screens with tabHost.

Week13:

Click on the image to enlarge the event. Fifty input tests were performed, and dozens of bugs that caused the system to crash were changed and fixed. Stability tests (notifications and timers and widgets) were performed 10 times, in units of 30 minutes.