CSC 3220 – Applications Programming, Spring 2019 – Lab Assignment #3 Dialer App

Due: Tuesday, February 11 by 11:59pm

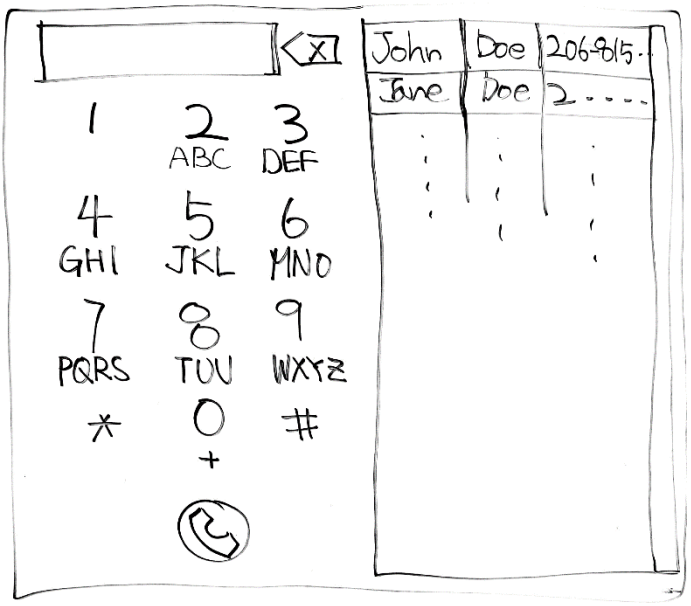
Dialer has been a default app for smartphones and most people do not care how it works, as it's like an air. This project will remind us the importance of air by a dialer design and development, as well as help us to understand what is more 'serious' MVC implementation.

Your job: read the specifications below and go for the tutorial. Return to the specifications below and implement the remaining functionalities.

The sketch shows how your dialer app is supposed to look.

Note: your app may have a little different look and feel, but should follow the similar layout.

Read the following feature specifications and check out the necessary video tutorials. It will show you some important implementation tips and tricks, how to structure your GUI app, and how to implement the number lookup functionality.



Required Features:

No.	Weight	Specification
1	20%	**[COMMON, MANDATORY]** ❖ Create the user interface described in the first page.
2	25%	**[COMMON, MANDATORY]** ❖ Once a user clicks a keypad button, the dialer should add the corresponding digit to the label on top. ➢ The phone number showing on the top label should follow **the dash format**, so if a user pressing 2068151234, it should show in the following sequence: ▪ 2 -> 20 -> 206 -> 206-8 -> 206-81 -> 206-815 -> 206-815-1 -> 206-815-12 -> 206-815-123 -> 206-815-1234 ➢ Deletion should support the same presentation format. ❖ 'x' button next to the label on top deletes the last digit. ❖ If the label is empty, nothing happens. ❖ Once the call button clicked, ➢ Pop up a quick dialog box with a message like: 'Calling 206-815-1234…' ➢ And after dialog closes, reset (empty) the label on top.
3	25%	**[COMMON, MANDATORY]** ❖ Have a menu item named 'Open Address Book' ❖ Let it read an address book file (e.g., 'us-500.csv') and load the address information in the address book QTableView. ❖ Once a name (first or last) or phone number clicked, load the phone number to the label on top-left.

4	**30%**	**[OPTION #1] Design Challenge** ❖ Make a 'cool' look and feel. ➢ Choose a model design from dribbble.com or somewhere over the internet. ➢ Follow the design at your best effort. You may need to use icons or specific fonts. ➢ **Note: your own design is not permitted. You should pick a model and follow it.** ▪ **To make this challenge count, you should get an instructor approval for your model design, at least 3 business days before the deadline.** ❖ Your UI design will be evaluated by the following criteria: ➢ If the color choices are same to the model design ➢ If the font type, size and placement are same to the model design ➢ If icon or specific graphics are used, they should be same or close enough. ❖ In your demo video, you **should** show which model design you have been pursuing.
5	**30%**	**[OPTION #2] Implementation Challenge** ❖ Real-time address book lookup ➢ As a user clicks the keypad buttons, the address book list (on the right side) should change, showing all possible name/phone # combinations from the digits. ❖ Rules for address records to be displayed: ➢ All numbers starting with the current digits typed ➢ All (first/last) names starting with corresponding alphabets (2-ABC, 3-DEF, 4-GHI, …). ➢ For example, for '63' typed, the following should show up. ▪ Phone numbers starting with 63 • 631-335-3… Abel Maclead, 631-957-7… Latrice Tolfree, … ▪ Names starting with 'me', 'ne', 'od', 'of', ...: • Me-: Melissa Wiklund, Meaghan Garufi, Oretha Menter, … • Ne-: Nelida Sawchuk, Matthew Neither, … ❖ I know that the above functionality might be a little challenging but will be fun to do! ➢ There's a cool solution that will make your life easier, you are encouraged to share the idea in the class channel. ❖ Then show your app is working as expected in your demo video. ➢ Show many random patterns of number digits and make sure that it does not crash.

Deliverables: `(Double Check Canvas Assignment Page for additional information)`

1) Your Demo Video. Make sure to include a narration. It should show:
   a. The UI design and functionality you have chosen to implement.
   b. **[Design Challenge]** Show your model design and your design side-by-side, then explain the design details.
   c. **[Implementation Challenge]** Show the phone number lookup feature working correctly without crash.
2) Your .cpp, .h, .ui files
   a. Be careful: now I am asking .ui file to be submitted together this time.