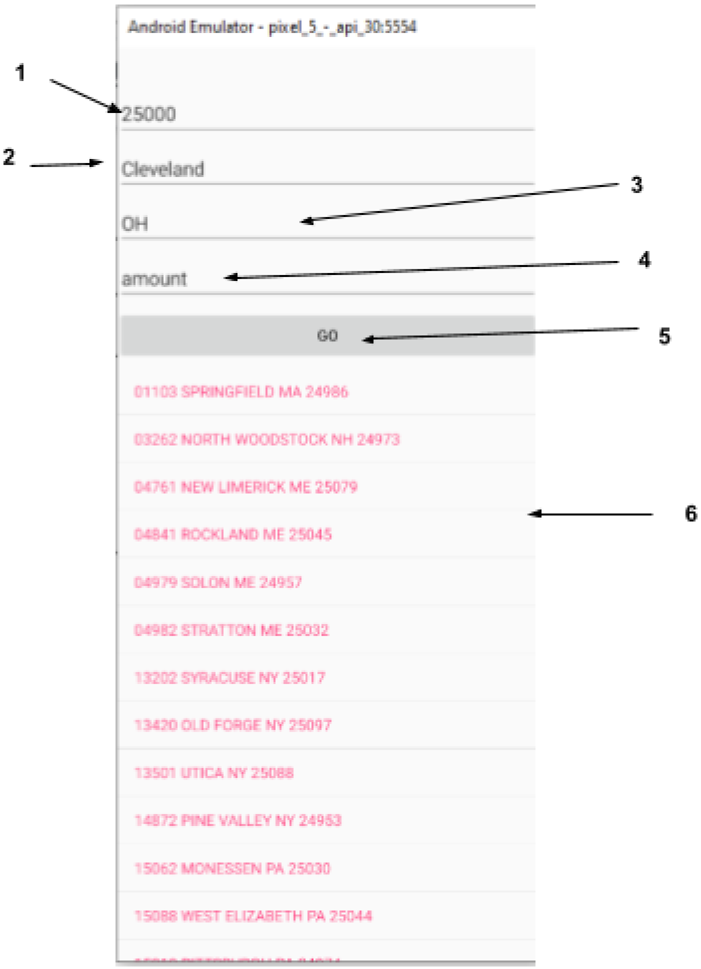
**CSE 382**
**Project #2 - Mobile Application: US Tax Information**
**Fall 2022**

You will create a mobile app, `USTaxes`, using Xamarin that executes queries regarding tax returns filed in the US. The mobile app will be based on the following screenshots:

1. A dollar amount provided by the user
2. The name of a city provided by the user
3. The name of a state provided by the user
4. The type of query to perform (provided by the user). There are two types of queries:
    a. The first, termed *amount*, finds all zip codes that have an average tax return equivalent to the dollar amount provided by the user (i.e., box #1) ±$100.
    b. The second, termed *city state*, finds all zip codes that are in the city-state provided by the user.
5. A button to initiate the query.
6. The scrollable list of zip codes that are produced by the query. The results are sorted by zip code, with city, state, and average tax return shown too.

**Get zip codes with average tax return of $25000 +- $100**



Android Emulator - pixel_5_-_api_30:5554

25000

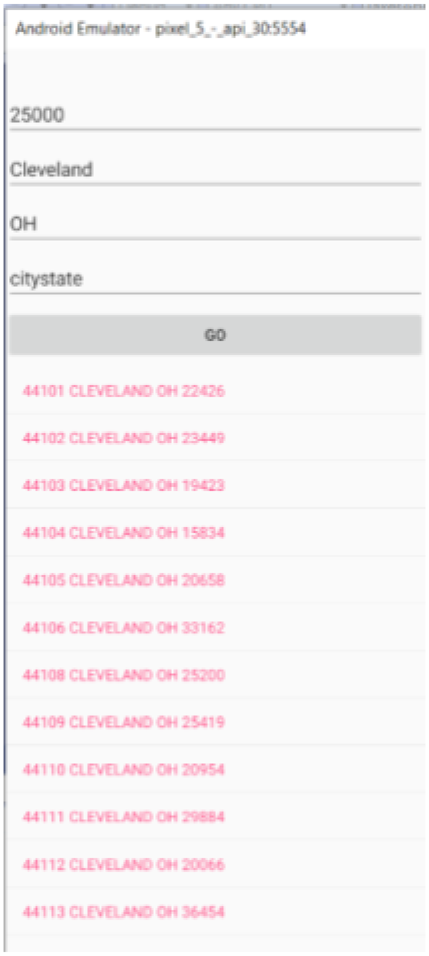
Cleveland

OH

amount

```
01103 SPRINGFIELD MA 24986
03262 NORTH WOODSTOCK NH 24973
04761 NEW LIMERICK ME 25079
04841 ROCKLAND ME 25045
04979 SOLON ME 24957
04982 STRATTON ME 25032
13202 SYRACUSE NY 25017
13420 OLD FORGE NY 25097
13501 UTICA NY 25088
14872 PINE VALLEY NY 24953
15062 MONESSEN PA 25030
15088 WEST ELIZABETH PA 25044
```

**Get all Cleveland, OH zip codes**



Android Emulator - pixel_5_-_api_30:5554

25000

Cleveland

OH

citystate

```
44101 CLEVELAND OH 22426
44102 CLEVELAND OH 23449
44103 CLEVELAND OH 19423
44104 CLEVELAND OH 15834
44105 CLEVELAND OH 20658
44106 CLEVELAND OH 33162
44108 CLEVELAND OH 25200
44109 CLEVELAND OH 25419
44110 CLEVELAND OH 20954
44111 CLEVELAND OH 29884
44112 CLEVELAND OH 20066
44113 CLEVELAND OH 36454
```

**Notes.**

- For a given zip code, the data file can include multiple options. For example, 45056 is associated with six options:

  | | | | | |
  |---|---|---|---|---|
  | … 45056 | STANDARD | OXFORD | OH | ... |
  | … 45056 | STANDARD | MIAMI UNIV | OH | … |

  You are to discard all but the first instance of a given zip code.
- You can repurpose your `USLocations` class from a previous project (copy it into your new project and revise it); you may also find it necessary to add methods/fields to the class. If the asynchronous portion of the code causes runtime problems, please drop back to doing regular, synchronous, IO.
- You are expected to alter the appearance of the app in order to make it more appealing and usual than the one displayed above. This includes, but is not limited to:
  - Colors
  - Font sizes
  - Labels to identify purpose of controls
  - Positions/layout of controls
  - Type of controls used for user input for effective, and resilient, user input
  - Enabling/disabling controls
  - You can make any changes you wish, as long as the spirit of the requirements are still met.
- Your program should not crash, regardless of the input entered by the user.
- The zipcodes.tsv must be treated as an embedded resource. This is necessary so that the mobile app has the original data file when it is deployed to an actual mobile device.
- Your program should not crash.
- Average tax return is to be computed as total wages in that zipcode divided by the total number of tax returns filed. A zip code (e.g, 45056) can appear multiple times in the file under multiple names (e.g., Oxford, Miami University, etc). For computing purposes, use the first one that appears in the file.
- XAML is not required for this project.

**Submission.**

- The submission for this assignment differs from the prior assignment.
- Here are the submission instructions to use when ready to submit to Canvas.
- Some extra notes for the video
  - At a minimum do the following searches:
    - amount: $25000
    - amount: $40000
    - citystate: Cleveland, OH
    - citystate: Detroit, MI
  - Make sure that you illustrate that your program does not crash. Do this by entering illegal information into any place the user can enter information. Any error checking that you do should be exhibited in your video.
  - The maximum length of video should be 4 minutes.

**Scoring.**

- **(75) Correctness**. Your mobile application must operate correctly.
- **(10) Robustness**. Your program should not crash under any circumstances.
- **(15) Appearance**. Your application will be subjectively evaluated for visual and operational appeal. Implementing a basic application, such as the one displayed above, will receive 0 points on this criteria.
  - 13-15: Professional appearance/operation.

- 10-12: Nice appearance/operation. Significantly better than the default.
- 3-9: Some attempt was made to improve on the default application but the appearance/operation has an obvious need for improvement.
- 0-3: Approximately the same quality as the default application.