**MILESTONE 3:**

**Changes:**

**User Interface/design:**

* Decided to use Java FXML w/Scene Builder
  + Used anchor panes to organize layout
    - VBox and HBox to organize buttons and text areas
  + Used a tab pane to display tableview and statistics summary
  + Utilized cascading stylesheets to assist in the appearance of the interface
  + One controller for one screen

**Features:**

**Option to choose a CSV File to input:**

* User can either browse for a specific CSV file using the browse button
  + Uses the java IO File collection and FileChooser to accomplish this
* User can enter the absolute path of the CSVfile
  + Verifies the file in the specified path exists and is a CSV file
  + Grabs the file name and parses it

**Filter Data:**

* All data from CSV is stored initially (this is to prevent having to obtain the data again by parsing it)
* Modified(filtered) data is stored in another variable that is going to be used to update the tableview
* Filters work on a case by case basis
  + If there is text in account number return that property if found
    - There can only be one account number so filter stops after run
  + If there is text in address return that property if found
    - There can only be one address so filter stops after run
  + If there is text in neighbourhood only filter by neighbourhood
  + If there is text in assessment class only filter by assessment class
  + If there is text in BOTH assessment class and neighbourhood, use all property data to filter selected filters for both

**Summarize Statistics – Using Charts \*\*\*TODO\*\***

* Use a bar chart to summarize assessed values by neighbourhood of given data

**Known Bugs/Flaws:**

* Does not check whether CSV file is in the correct format (will accept any CSV file)
* BarChart is incomplete

**REFERENCES:**

**Background Image:**Mellish, B. (2016, June 3). House Lights Turned On. https://www.pexels.com/