**MCIT 591 Final Project: Timezone Converter**

**Class–Responsibility–Collaborator (CRC) Cards**

|  |  |
| --- | --- |
| **Class: GUI**\* | |
| Responsibilities: | Collaborators: |
| * has Map (city | location)   Methods:   * presents user with a selectable map * has search function to captures user input on city|location selection. * imports and places DatePicker and TimeSpinner components in appropriate locations * Has sidebar containing simple instructions for user on how to use the program. | * Passes selected cities|locations user input to TimeConverter Class * Passes selected date|time user input to TimeConverter Class * Gets city information from City class. * DatePicker * TimeSpinner |

|  |  |
| --- | --- |
| **Class: DatePicker (User Input)**\* | |
| Responsibilities: | Collaborators: |
| * presents user with a selectable calendar * captures user input on date for conversion * outputs resulting date after conversion\*\* | * Passes selected date user input to TimeConverter Class and receives updated time and date as a result of conversion |

|  |  |
| --- | --- |
| **Class: TimeSpinner (User Input)**\* | |
| Responsibilities: | Collaborators: |
| * presents user with a scrollable time spinner with the option to manually type in time. * captures user input on time for conversion * outputs resulting time after time conversion\*\* | * Passes selected time user input to TimeConverter Class and receives updated time and date as a result of conversion |

\*We have included a mock-up of Classes representing the user interface. We will be creating this interface using JavaFx.

\*\*Once a date/time has been changed, it will auto-update in all other locations added to the map to show output.

|  |  |
| --- | --- |
| **Class: TimeConverter** ^ | |
| Responsibilities: | Collaborators: |
| * has ZoneId (source | input) * has ZoneId (destination | output) * has user Input String * has LocalDateTime * has ZonedDateTime   Methods:   * Converts time to another timezone (e.g. pass in date/time, timezone 1, timezone 2) | * Takes in user input from GUI:  1. date/time to be converted 2. Timezone converting from 3. Timezone converting to  * Output passed to a display output (GUI) |

^ Currently the TimeConverter.java file is not yet linked to GUI to take in user input - these aspects of the Class will be further developed in the final project submission (details as specified above under “Collaborators”).

|  |  |
| --- | --- |
| **Class:** ZoneId (Java in-built) | |
| Responsibilities: | Collaborators: |
| Variables:   * has Name * has DateTime (from java.time) | * Called in the TimeConverter Class to support and facilitate the time conversion. * City (each timezone linked to multiple cities / locations) [?]^ |

^ Another issue on which we are currently researching is how to convert user-selected cities or geographical locations to Java-recognized timezones (as detailed below under the City Class).

|  |  |
| --- | --- |
| **Class: City** ^^ | |
| Responsibilities: | Collaborators: |
| * Has Name * Has Country * Has Timezone | * Pass in city information captured by GUI (user input) - for both the “From” and “To” cities in the conversion. * Timezone (each city linked to one java in-built timezone) |

^^Currently researching on whether Java provides in-built functions that can connect cities to timezones. If not, manual data needs to be supplied and parsed to establish the connection between a user-inputed city, a timezone recognized by Java (ZoneID), and displaying to approximate location on a map.

\*\*\* End of CRC \*\*\*

**List of Classes and Collaboration Plan**

(approved at initial TA advisory meeting with Anvi, Sun 14 April 2019, 6:30pm)

* + Develop Input-Parser Class - **John Caton**
    - Potential: Multiple people inputs
  + Time Converter Class - **Chelsea**
  + Calendar GUI Class - **Sai**
  + Timespinner GUI Class - **Sai**
  + Map GUI Class - **John Caton**
    - Potential: Drawing selection option
  + City (?) - **Chelsea**
  + Output (?)
* Create PNG mock-up of finished product - **Sai**

**Other Resources and Research**

* JavaFx Tutorial Links:
  + <https://www.youtube.com/watch?v=9YrmON6nlEw&list=PLS1QulWo1RIaUGP446_pWLgTZPiFizEMq&index=1>
  + <https://www.youtube.com/watch?v=FLkOX4Eez6o&list=PL6gx4Cwl9DGBzfXLWLSYVy8EbTdpGbUIG>