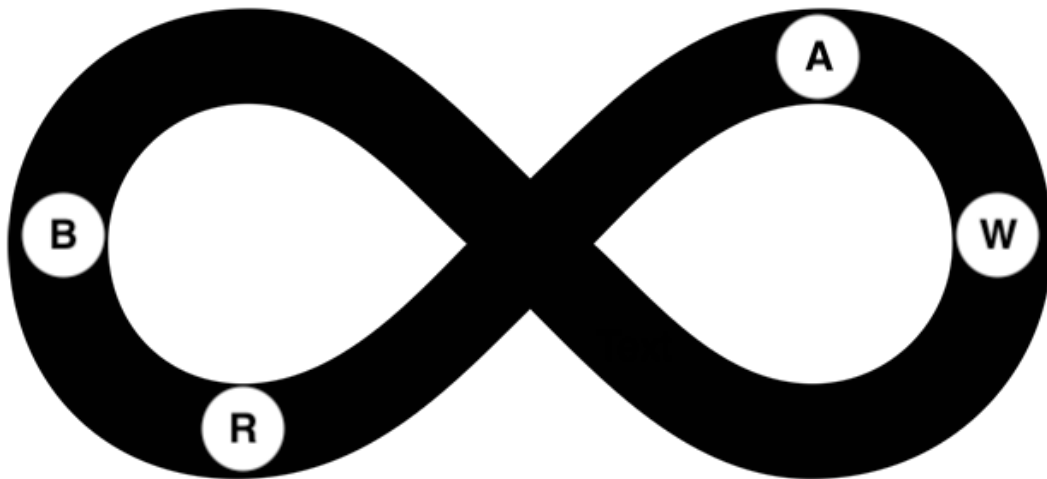


*Sprint Four Backlog*

**TEAM**

**BRAW**

**B**ekzod Tursunov | **R**adu Laudat | **A**lvin Tang | **W**esley Ma



**"Infinite Possibilities with BRAW"**

# Table of Contents

[Table of Contents](#)

[User Story 3 Tasks](#)

[User Story 5 Tasks](#)

[User Story 6 Tasks](#)

[User Story 7 Tasks](#)

[User Story 8 Task](#)

[User Story 10 Tasks](#)

[Sprint Plan](#)

[Sprint Report](#)

[Burndown Chart](#)

## User Story 3 Tasks

As Ivanka Tary, an employee of TEQ, I want to be able to generate specific reports using a graphical user interface, so that I can specify the data columns that will be used in the report

- Using a generalized checkbox to automatically add/remove columns the employee wants to include in report or graph

13. ~~Create new section in GUI that displays a list of all graphs to be generated, as well as allows us to add/remove; also a GUI that displays what is needed in the report to be added~~  
Work on a GUI for reports where you can add a query to modify the list of graphs/tables to be generated, view a list of graphs/tables to be generated, add graphs/tables to be generated, and remove graph/tables to be generated.

- Story Points: +6 (Adding 6 story points in addition to previous sprint)
- Dependencies: None
- Description: Using JSwing, show a list of all graphs to be generated (showing their titles) using the table feature in JSwing. Graphs should be created in such a manner that users will see the following when **adding** new graphs (adding graphs should be in another GUI pop up window):
  - Input main title, x title, y title
  - Radio buttons for deciding whether to graph or not
  - Dropdown selecting the type of graph (trend or distribution).  
If **distribution** is selected (**focus on this one for now**):
    - Dropdown of all available columns
    - Dropdown for style of graph (line or bar)
  - If **trend** is selected:
    - List of column entries to graph
    - Dropdown of all available columns
    - Dropdown of all available entries within the selected column

## User Story 5 Tasks

As Donald Tum, an employee of TEQ, I would like to make Bar graphs, line graphs, etc...

19. Modify the HTML template, NodeJS files and how the system handles ChartJS schemes such that a generated report can contain multiple pieces of information
- Story points: 3
  - Dependencies: 13
  - Description: Goal is to make it such that a generated report can contain multiple charts, basic information about the report (# of people, # of visits, years in report, report title, when the report was generated), as well as a print button.

## User Story 6 Tasks

As Jared Kuper, an employee of TEQ, I want to be able to generate specific reports using a query language similar to SQL, so that I can completely customize the data of the report

- Input data based on the columns similar to SQL (i.e. started in [date], aggregations) etc...

20. Design the query language in a separate document

- Story Points: 2
- Dependencies: None
- Description: Design a query language similar to SQL that takes into account the necessary operations to generate reports. This design should be made in a separate document with descriptions of each reserved word in the query and how it is used as well as examples of the implementation of a query. Also add an example of the output provided after query is implemented (e.g. list all the columns, and years to be put in report) Example of a query that could be implemented: *ALSO SELECT COLUMN "Referred by"* (*ALSO* adds on top of the requirements, *SELECT COLUMN* selects the column *Referred by* to be put in the requirements). Take in that some columns names cannot be added to the requirements simply by naming its column name in the query as there could be a naming difference that the user is not aware of, we have to manage how to search column names.

21. Create a class that takes in a query and implements operations

- Story Points: 5
- Dependencies: None
- Description: A class that has a method that takes in a query as a parameter, breaks the query down based off its reserved words, then call specific class methods to complete actions based off those reserved words.

22. Connect the GUI with the class that takes in a query

- Story Points: 2
- Dependencies: 13, 21
- Description: At the end of this task, the GUI in task 13 shall allow for the query to run and perform operations. These operations should focus on including adding a chart to a report.

## User Story 7 Tasks

As Jared Kuper, an employee of TEQ, I want to be able to view data that needs to be cleaned, or has been automatically cleaned in a graphical interface, so that I can select data or conflicts to be fixed or re-assessed

- The data that was automatically or manually fixed as a table (old data, new data, on which (column, row))

- View inconsistencies in red colours that need manual attention (validation errors i.e. texts in number only, etc.)
23. Implement logic to check each cell before it's added to make sure that it has proper formatting
- Story Points: 3
  - Dependencies:
  - Description: For now we check for only two cases
    - i. Auto: Trim spaces
    - ii. Manual: Type errors, for now, check for type errors in a specific column we know may contain type errors
    - iii. If these errors are present, don't add the row to masterData, instead, add them to a separate queue to be added after correction
24. Have a popup window appear after every upload to display the above info
- Story Points: 3
  - Dependencies: 23
  - Description: This window will not allow the upload to go through unless the errors are fixed. The window will display a table with columns [Column][Cell][Error Type]

## User Story 8 Task

As Jared Kuper, an employee of TEQ, I want to be able to select a conflict (current data, inputting data) and fix it manually, so that reports can be generated without issue

- Double visits to an agency in a month
25. Implement functionality in GUI to be able to fix the error upon double click or r-click ...
- Story Points: 3
  - Dependencies: 24
  - Description:

## User Story 10 Tasks

As an employee of TEQ, I want to be able to compare the data in our system with Public Census data, so that I can identify where/what services are lacking.

26. Research if it is possible to read public census data
- Story points: 3
  - Dependencies: None
  - Description: Research if the public census data can be integrated in the system.

27. Upon research, either use the public census data or create my own scrapped public census data

- Story points: 1
- Dependencies: 26
- Description: Create a json file with the public census data so we can attach it right away into the json payload when creating the html pages

## Sprint Plan

User Story	Tasks	Dependencies	User Points	Day 1	Day 2	Day 3	Day 4	Day 5
3	13		6		A:3	A:3		
5	19		3			A:3		
6	20		2	W:2				
6	21		5		W:3	W:2		
6	22	13, 21	2				W:2	
7	23		3			R:3		
7	24	23	3				R:3	
8	25	24	3					R:3
10	26		3					B:3
10	27	26	1					B: 1

## Sprint Report

User Story	Tasks	Dependencies	User Points	Day 1	Day 2	Day 3	Day 4	Day 5
3	13		6			A:6	A:1	
5	19		3				A:4	
6	20		2			W:2		

6	21		5				W:3	W:2
6	22	13, 21	2					W:3
7	23		3					R:4
7	24	19	3					R:4
8	25	24	3					
10	26		3					B:3
10	27	26	1					B:1

## Burndown Chart

	Day 0	Day 1	Day 2	Day 3	Day 4	Day 5
Provisional	31	29	29	12	7	0
Execution	31	31	31	29	20	2

