Ad Auction Project

Deliverable 4 - Project Increment 3
GROUP 27

Badea, Sebastian (**sb1g19**)
Milanov, Miroslav (**mm3u19**)
Prodaniuc, Pavel-Ruben (**prp1u19**)
Kanapinskas, Mantas (**mk3g19**)
Dixon, Nathan (**nd2g19**)
Olubummo, Imisioluwa Kehinde (**iko1g18**)

Response to feedback

The following points explain how we have addressed the feedback relating to our program and documentation.

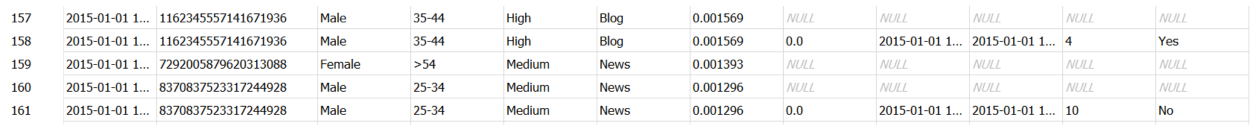
Response to lack of Unit testing:

We have implemented Junit 4 tests to ensure that our SQL queries are correct and that the totals shown for each line/metric remain correct. We have also used them to test different parts of our program such as date ranges and bounce definitions.

Furthermore, we added more images to show the stream of actions that go into completing a certain scenario, as opposed to before where we only showed the final result.

Response to database configuration:

Our database consists of one table (mainTable). This is done to allow for faster querying as it removes the need to perform left/inner joins when querying and applying filters. Impression entries are put into the database with empty click information. Once an impression is found with the same ID as a click entry, we input an entry that contains information from both of them. As the two inputs are done separately, impression costs are duplicated where a matching ID is found in the server/click log. This is why we had to adjust the queries to accommodate for this. The image below demonstrates how impression costs are duplicated.



157	2015-01-01 1...	1162345557141671936	Male	35-44	High	Blog	0.001569	NULL	NULL	NULL	NULL	NULL
158	2015-01-01 1...	1162345557141671936	Male	35-44	High	Blog	0.001569	0.0	2015-01-01 1...	2015-01-01 1...	4	Yes
159	2015-01-01 1...	7292005879620313088	Female	>54	Medium	News	0.001393	NULL	NULL	NULL	NULL	NULL
160	2015-01-01 1...	8370837523317244928	Male	25-34	Medium	News	0.001296	NULL	NULL	NULL	NULL	NULL
161	2015-01-01 1...	8370837523317244928	Male	25-34	Medium	News	0.001296	0.0	2015-01-01 1...	2015-01-01 1...	10	No

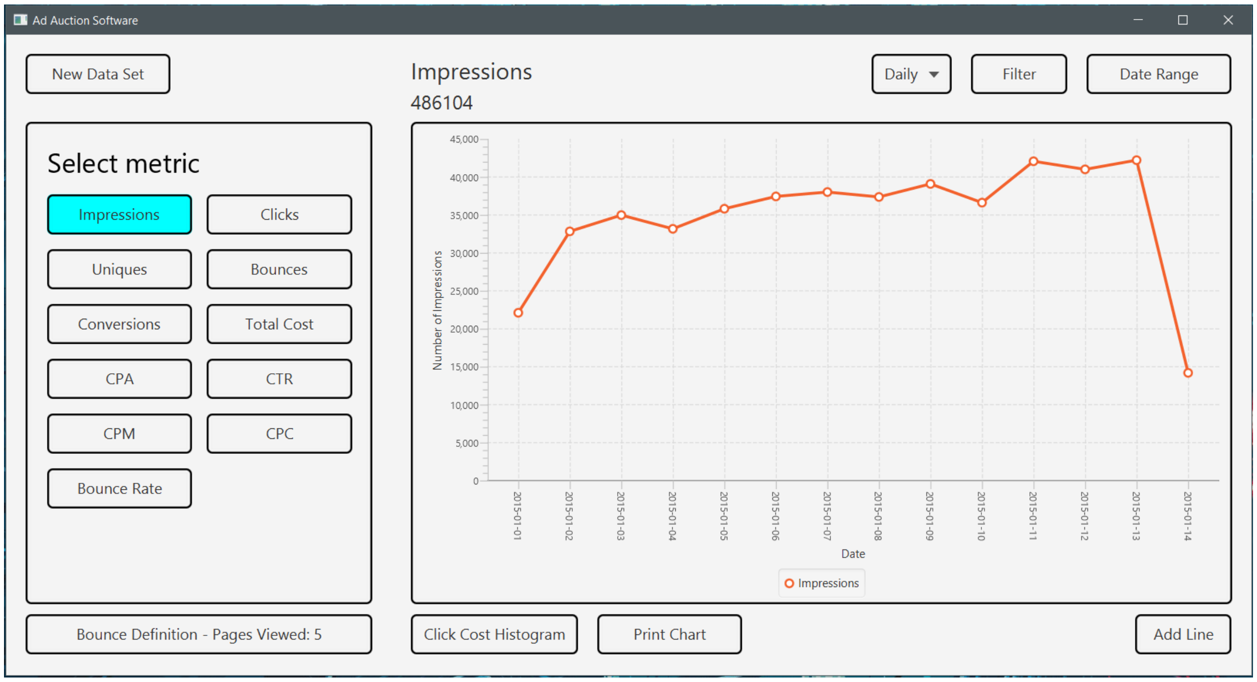
Arrangements of buttons:

We change the locations of the buttons to group them better. Now the buttons above the chart relate to the lines currently shown on the chart and better fit under the category of filter options.

Scenarios:

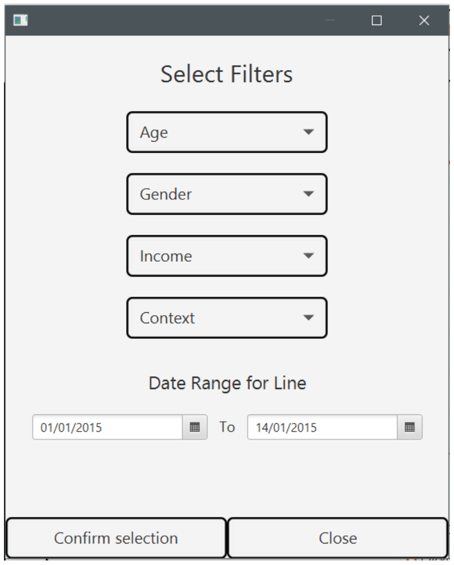
We have adapted our scenarios to the feedback. Now our scenarios mimic real interaction that users are likely to have and cover more than just a single user story.

Final design



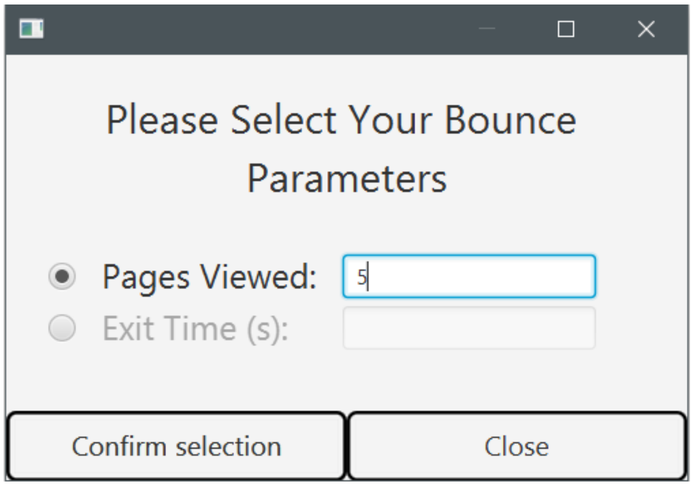
Our final program delivers an intuitive UI which offers a simple yet powerful experience to the user. It's not cluttered thus not overwhelming to the user and the theming remains consistent throughout. Furthermore it's responsive to the user, since every interaction with a button elicits some visual feedback, be it changing colour or changing the cursor.

<u>Adding date range to individual filters:</u>

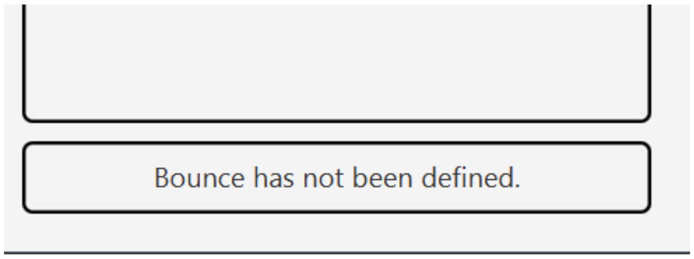


One key change in this iteration is the addition of the date range in the filter popup. This was done to allow the user to plot multiple lines of varying date ranges on the chart as opposed to them all being the same range. Doing this also ensures that requirement 7 is met as any metric can be replicated however many times over different date ranges on the same chart. This will be shown later in testing.

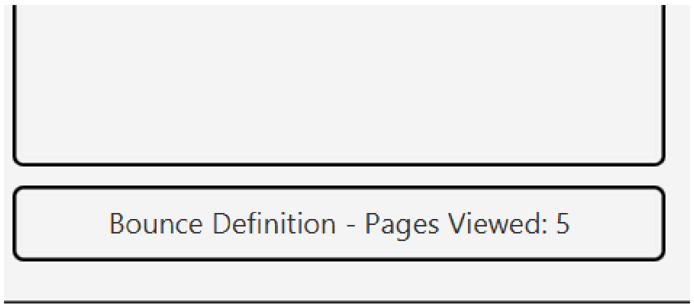
<u>Bounce Parameter Pop up:</u>



The program now also allows the user to select the specificities of their bounce definition. Users are allowed to specify the amount of pages viewed or seconds viewed, for the bounce definition. This is a massive improvement on the last sprint as it gives the user far more insight into the data.



Moreover, we made it so that the button inside the main menu updates as the bounce has been redefined.



That way the user can clearly see what the current bounce is, instead of having to reopen the popup.

Scenarios:

<u>User creates a database and looks at the impressions metric over 2 weeks:</u>

- Julius opens the software
- He clicks on “new data set”
- He chooses the correct logs and clicks confirm
- He selects on the “Impressions” metric
- The chart will show the impressions metric spanning 2 weeks

<u>User selects bounce definition:</u>

- Julius opens the software
- Clicks on a bounce rate metric button
- Clicks on the bounce definition button
- Selects pages viewed radio button
- Enters 10
- Clicks confirm selection
- The chart is adjusted to accommodate the chosen bounce definition

<u>User Adds 3 lines and adjusts date range:</u>

- Julius opens the software
- Selects a impressions metric
- Selects add line
- Selects metrics
- He repeats last 2 steps 2 more times with different metrics
- The chart will have 4 lines, 3 of them will have a total value next to their chart legend
- User selects date range
- User enters a date range
- The chart updates all lines to accommodate for the chosen range

<u>User views Unique clicks over 2 different date ranges:</u>

- Julius opens the software
- Clicks on “Uniques” metric button
- Clicks on the filter option
- Selects range of 02/01/2015 to 06/01/2015 and clicks confirm
- The chart is adjusted to accommodate this range

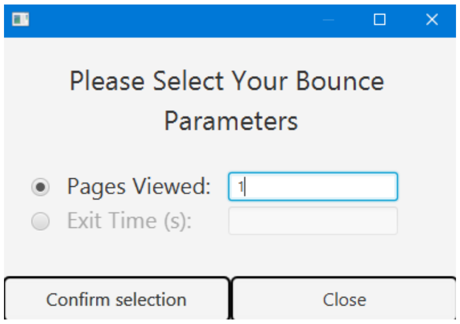
- He selects add line
- Selects range of 08/01/2015 to 12/01/2015 and clicks confirm
- The chart is adjusted and now shows 2 lines over 2 different date ranges

User compares metric with different filters applied over 1 day:

- Julius opens the software
- Clicks on a impressions metric button
- Clicks on time granularity drop down and selects hourly
- The chart adjusts for this granularity
- He then adds a line with the <25 filter applied and clicks confirm
- Chart add this as a new line
- He then clicks on date range button and applied the range 05/01/2015 to 05/01/2015 and click confirm
- The chart adjusts and shows 2 lines over a period of 1 day

Testing:

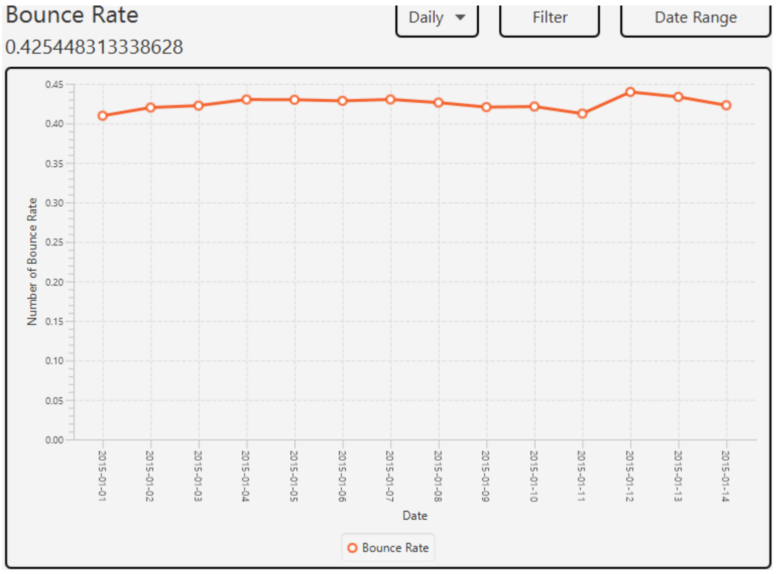
Adjusting bounce definition:



Please Select Your Bounce Parameters

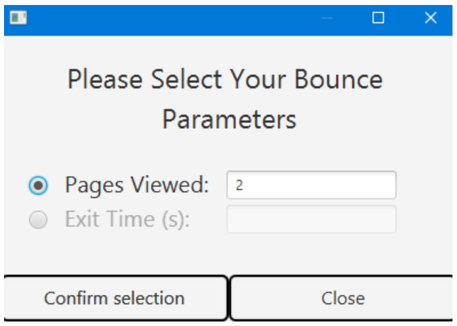
[✓ Pages Viewed:] 1

Exit Time (s):



Bounce Rate

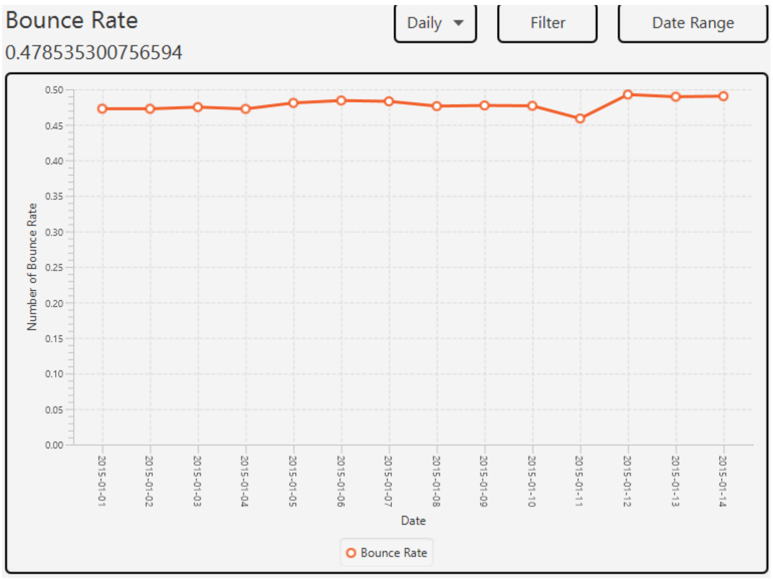
0.425448313338628



Please Select Your Bounce Parameters

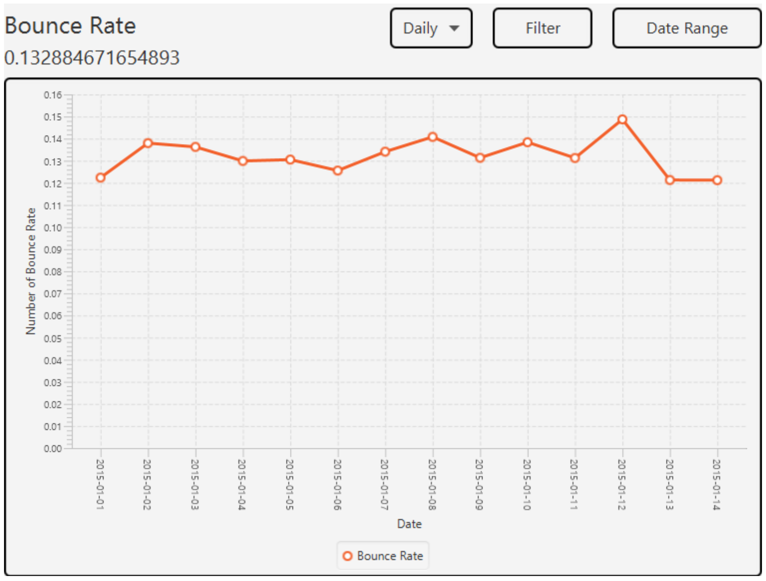
[✓ Pages Viewed:] 2

Exit Time (s):

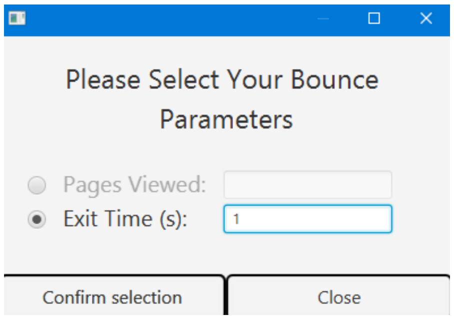


Bounce Rate

0.478535300756594

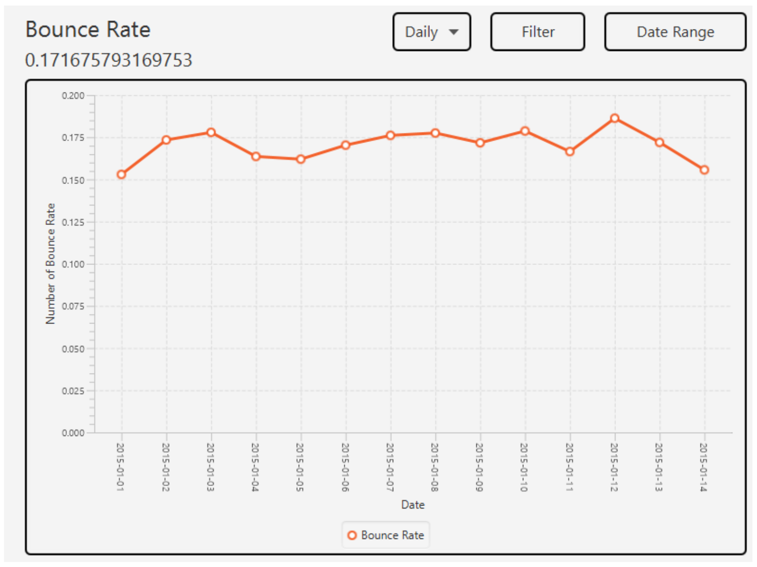


Bounce Rate
0.132884671654893

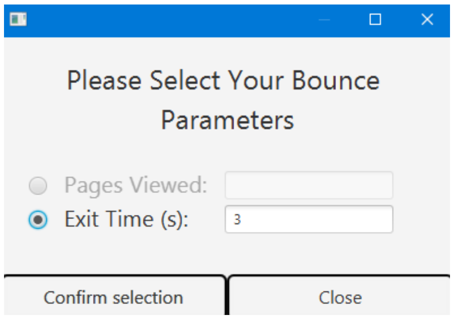


Please Select Your Bounce Parameters

[ ] Pages Viewed:
[✓ Exit Time (s):] 1



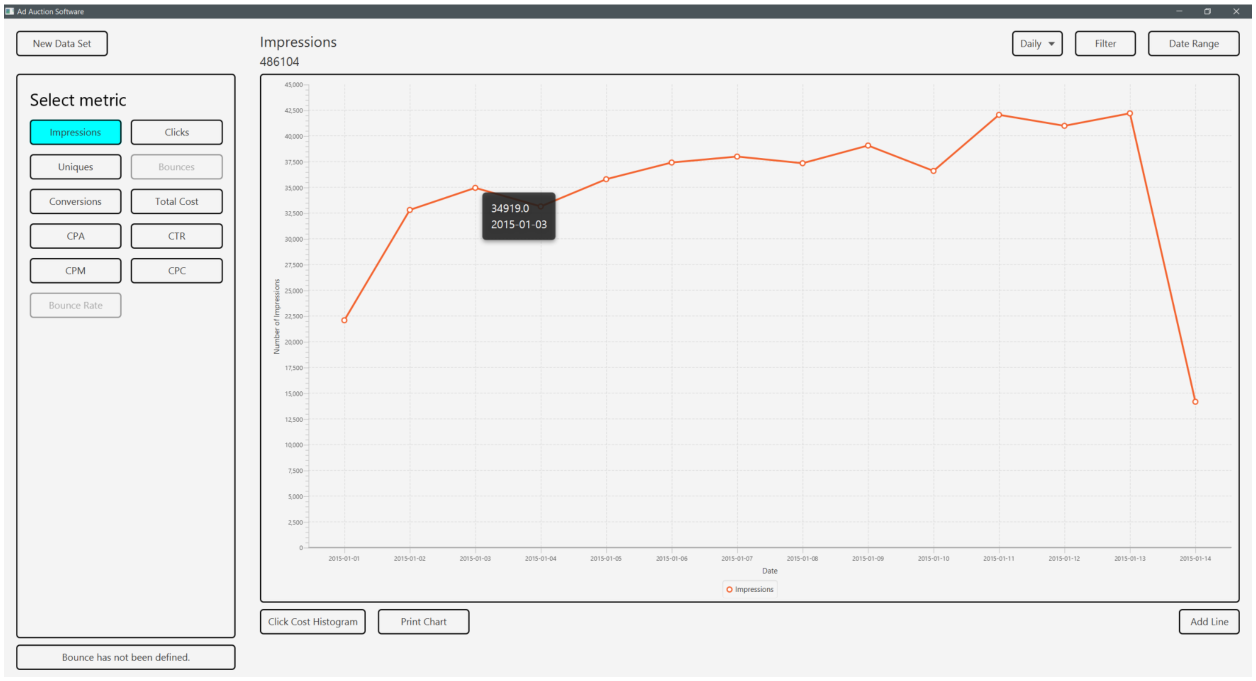
Bounce Rate
0.171675793169753



Please Select Your Bounce Parameters

[ ] Pages Viewed:
[✓ Exit Time (s):] 3

Hovering over data points reveal their value:

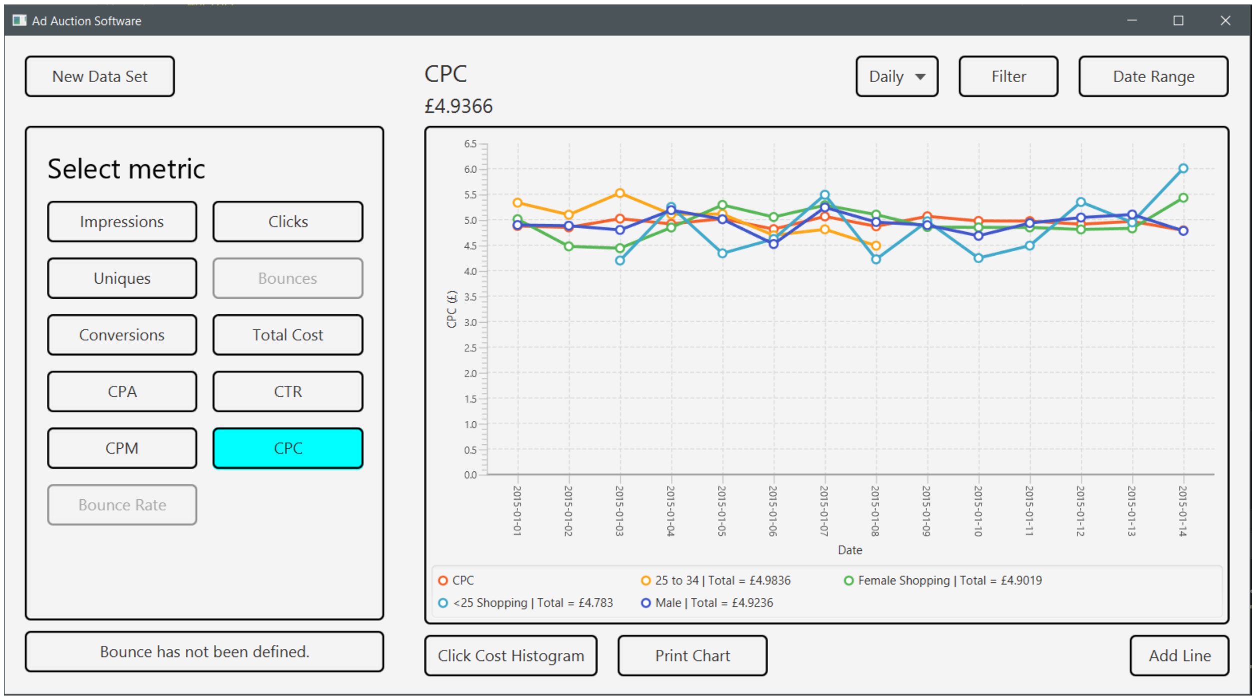


Impressions
486104

34919.0
2015-01-03

Bounce has not been defined.

Total for each line is shown:

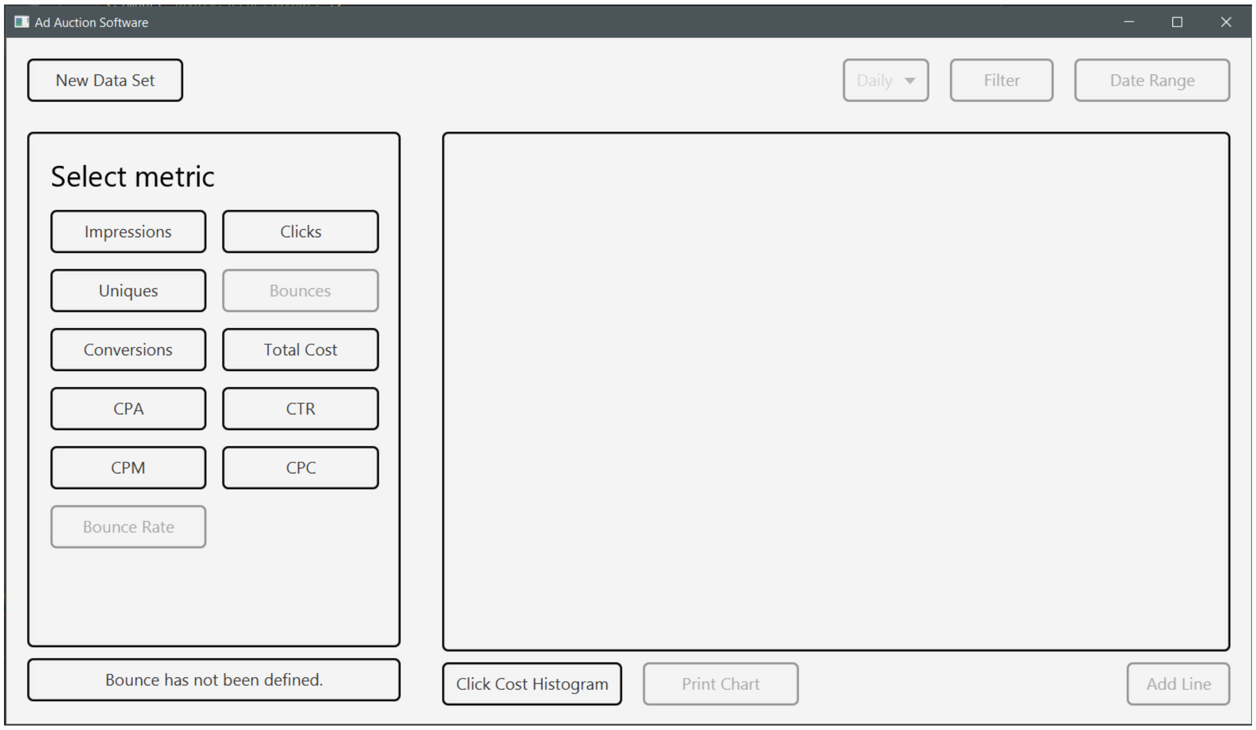


CPC
£4.9366

CPC
25 to 34 | Total = £4.9836
Female Shopping | Total = £4.9019
<25 Shopping | Total = £4.783
Male | Total = £4.9236

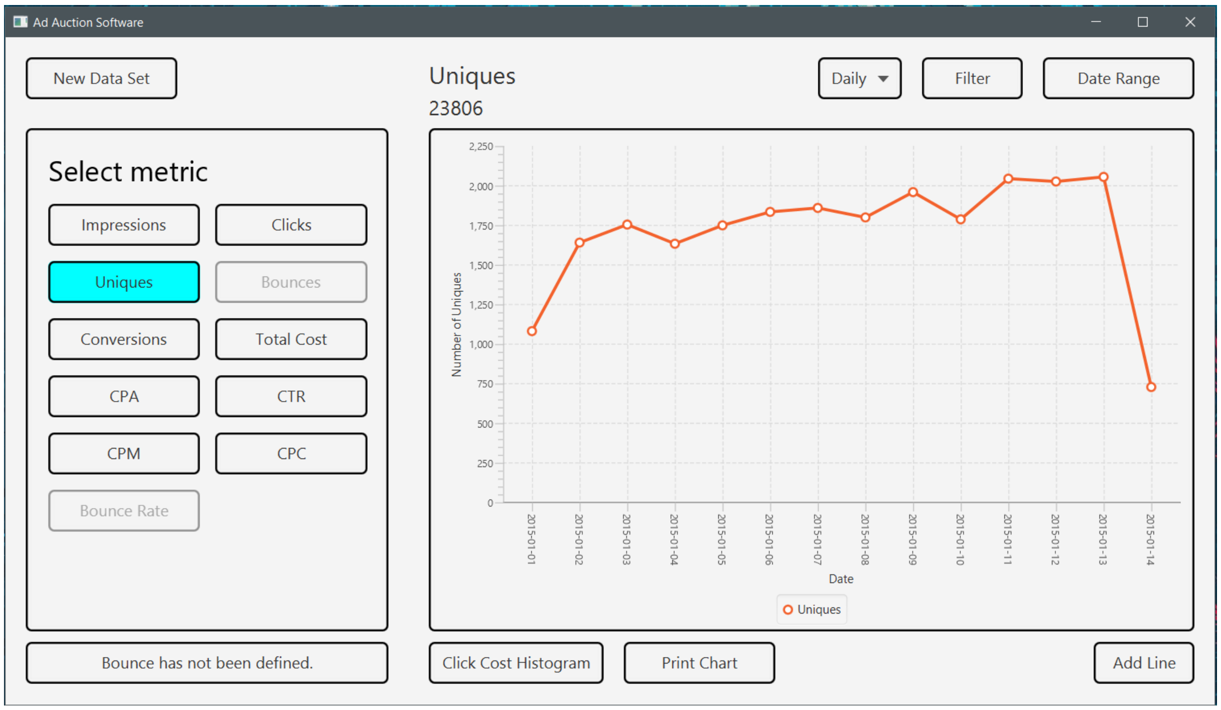
Bounce has not been defined.

Buttons which relate to chart are disabled if chart is empty:

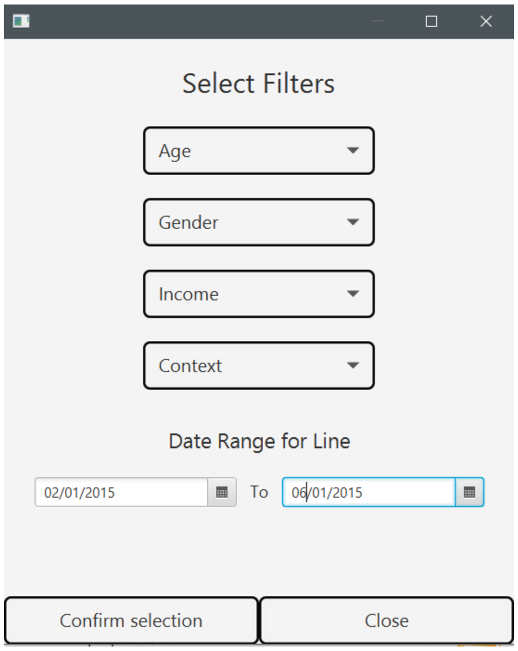


Bounce has not been defined.

Compare uniques over 2 distinct time periods:



Bounce has not been defined.


Select Filters

Age

Gender

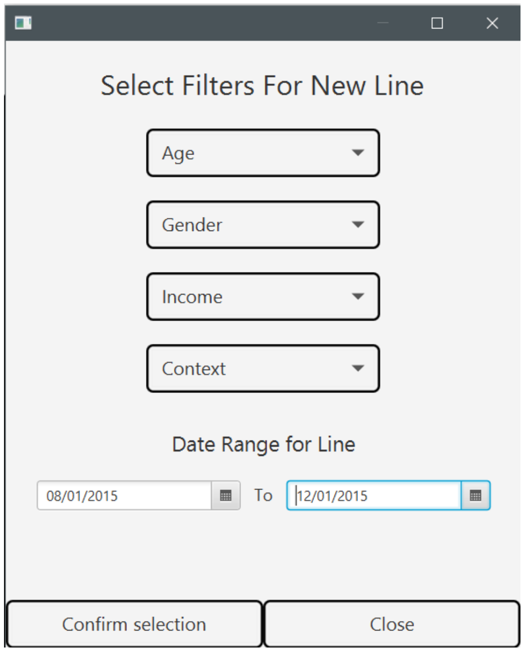
Income

Context

Date Range for Line

02/01/2015 To 06/01/2015

THEN


Select Filters For New Line

Age

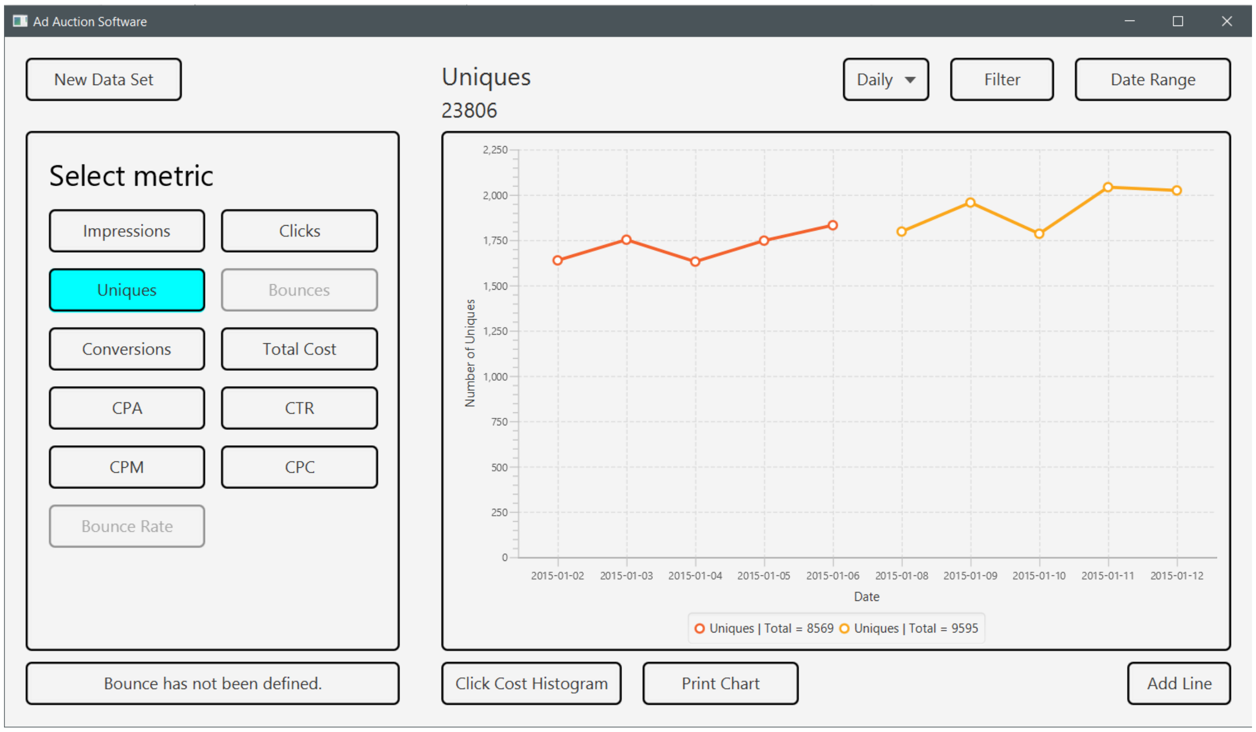
Gender

Income

Context

Date Range for Line

08/01/2015 To 12/01/2015


Ad Auction Software

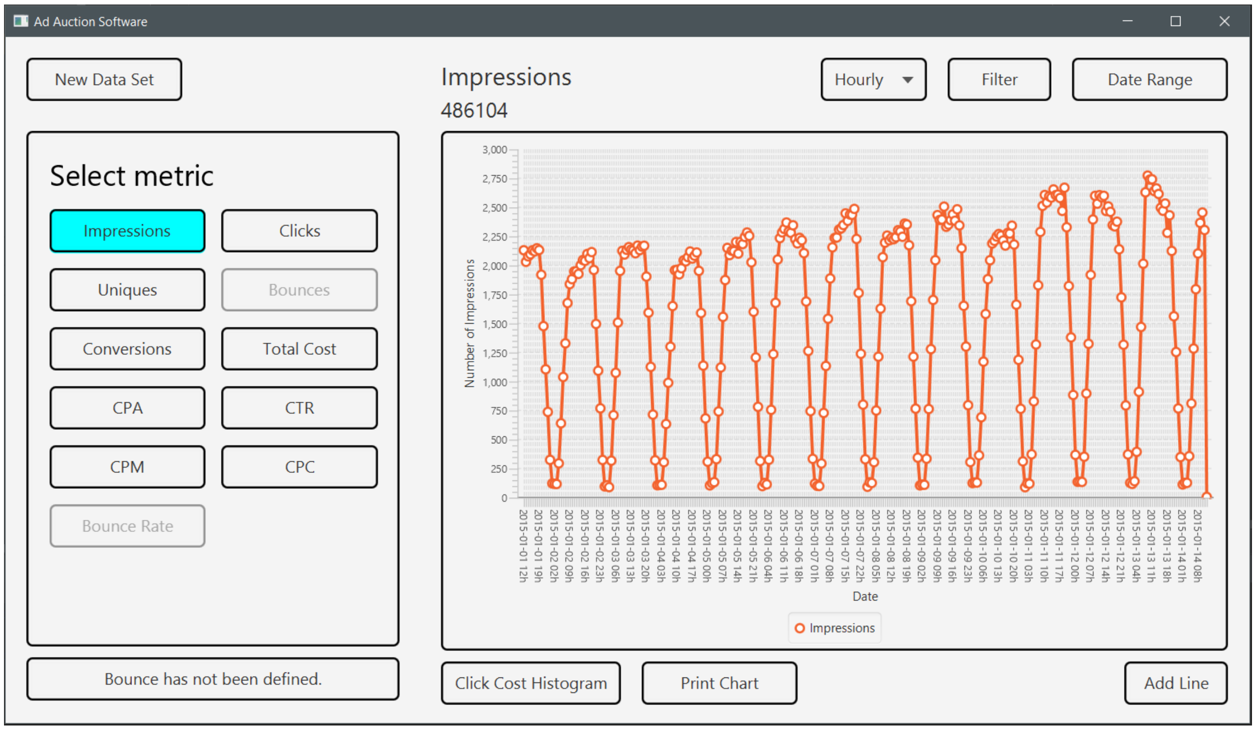
Uniques

23806

Bounce has not been defined.

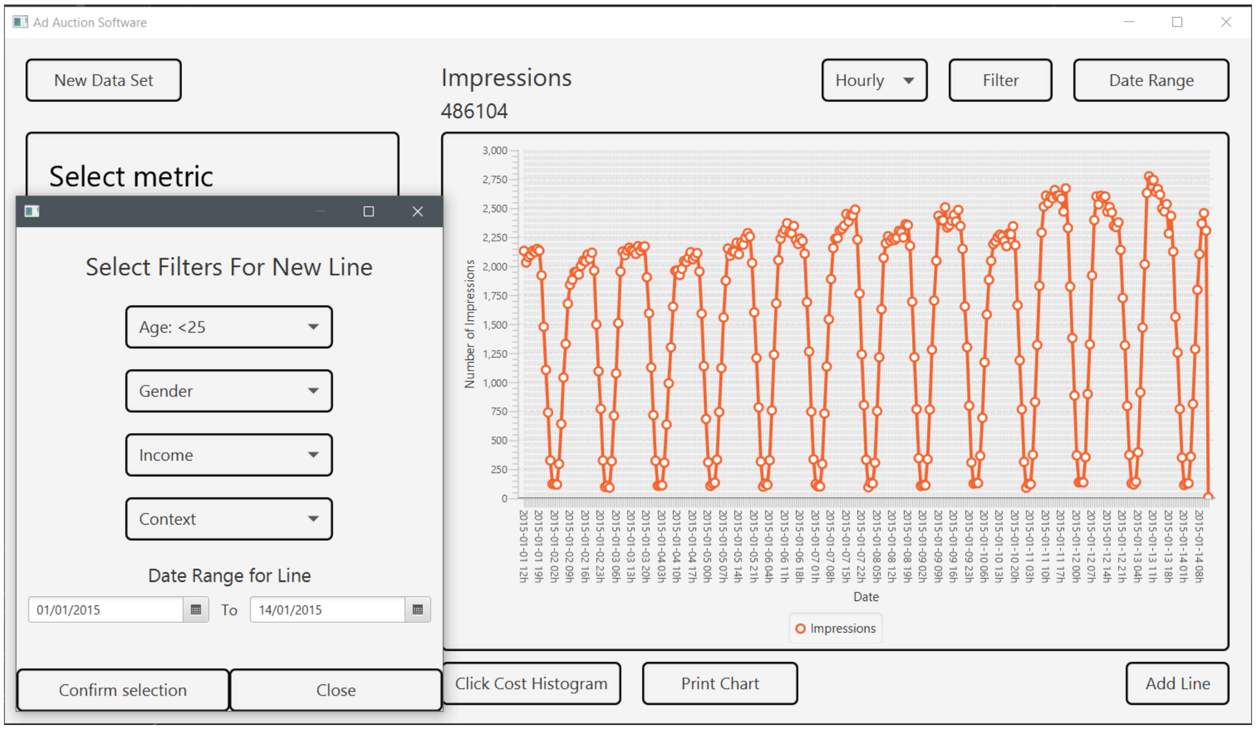
Uniques | Total = 8569 Uniques | Total = 9595

User compares metric with different filters applied over 1 day:



Impressions
486104

Bounce has not been defined.

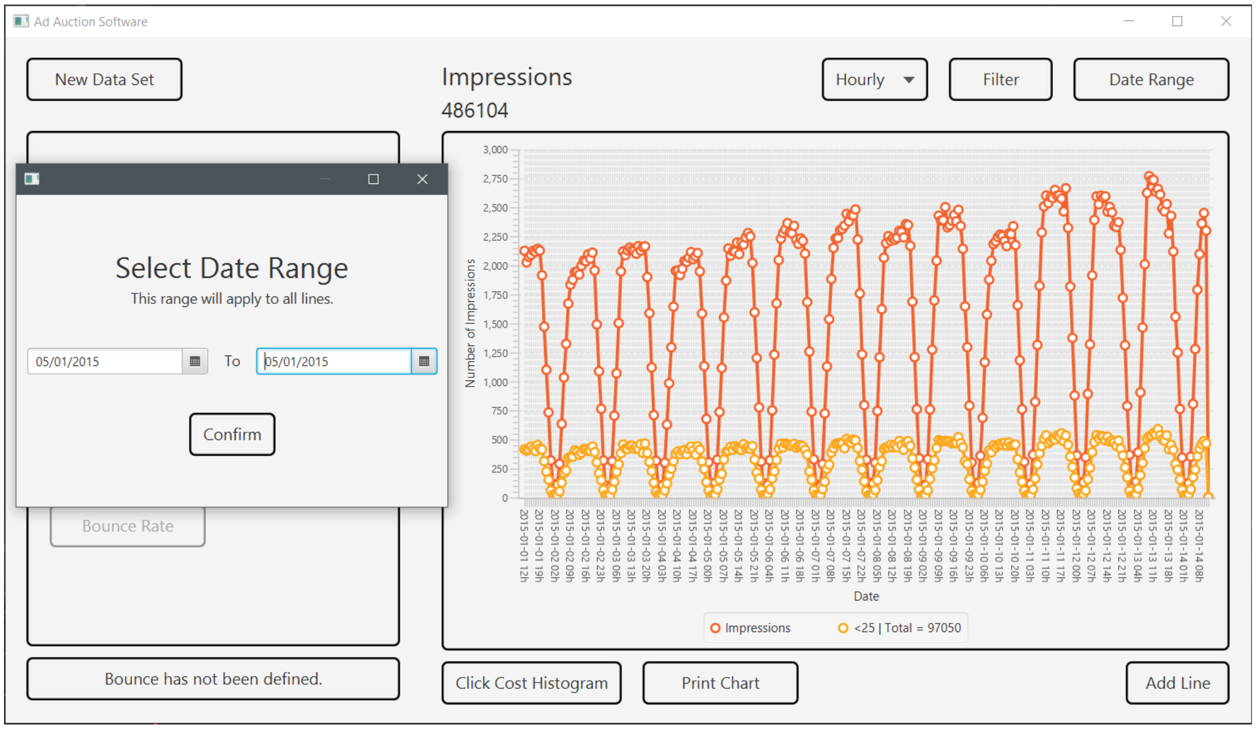


Impressions
486104

Select Filters For New Line

Age: <25

Date Range for Line
01/01/2015 To 14/01/2015

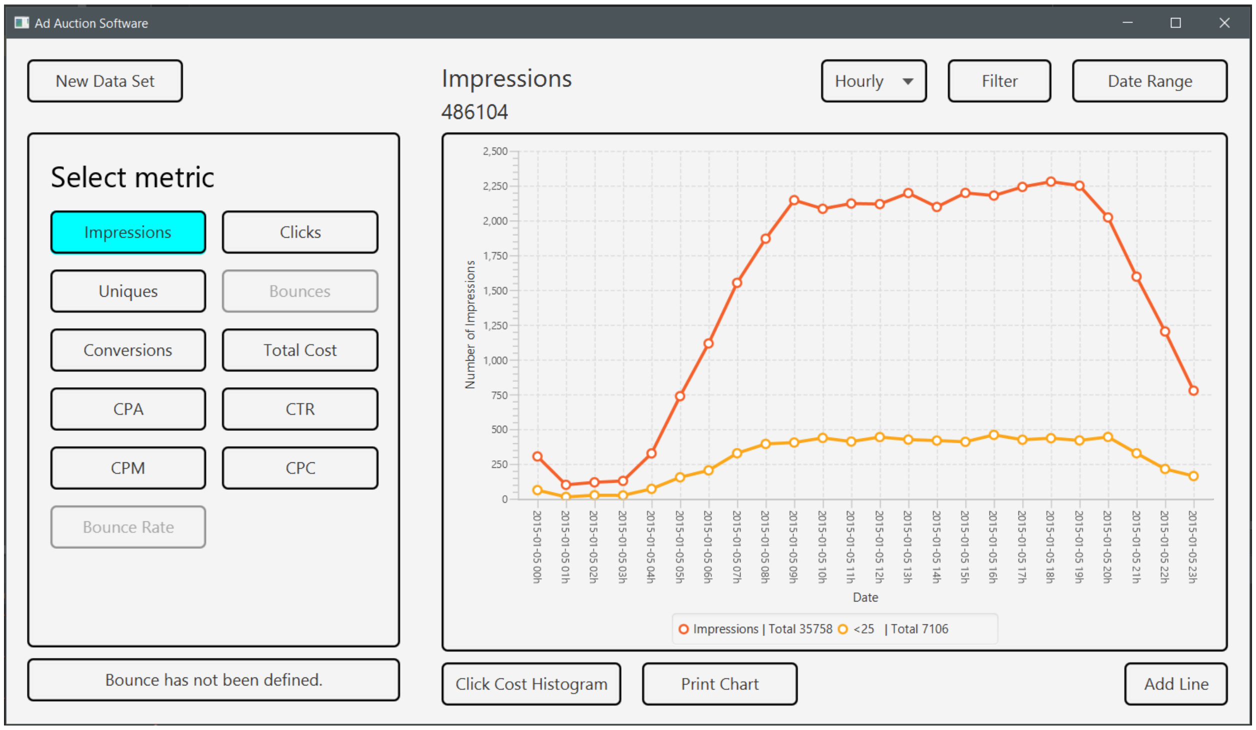


Select Date Range

This range will apply to all lines.

05/01/2015 To 05/01/2015

Bounce has not been defined.



Select metric

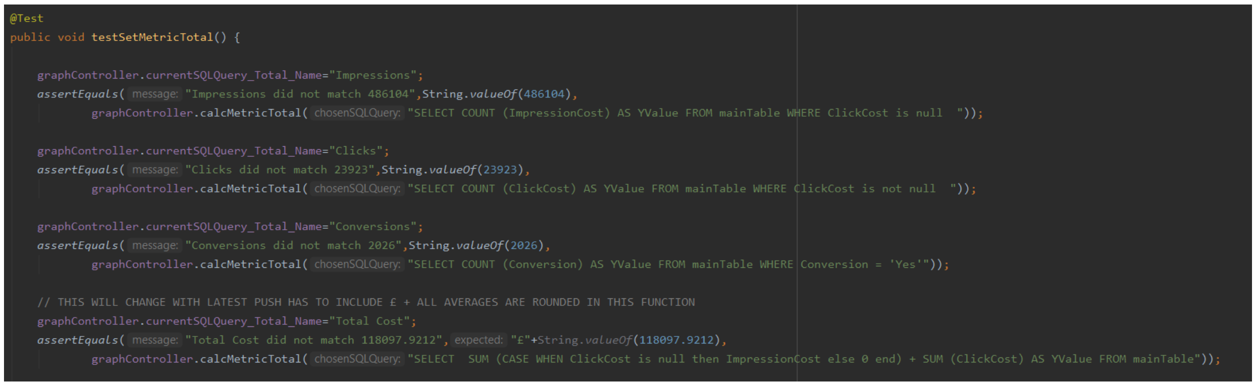
Bounce has not been defined.

Bounce Rate Verification:

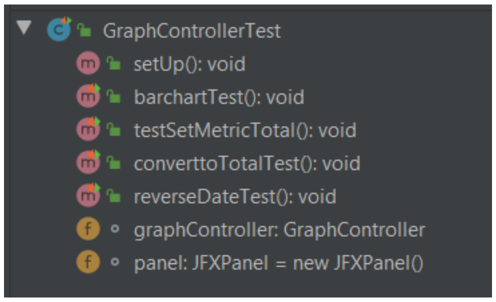
Verified against dummy DB of smaller size. Tested boundary values such as -
Exit time 1;
Exit time 0 (Just checks for ‘n/a’ entries);
Exit time 10000 - Correctly shows that 100% of the entries are registered as bounces.
Pages Viewed 1;
Pages Viewed 0; (Just checks for ‘n/a’ entries);
Pages Viewed 1000 - Correctly shows that 100% of the entries are registered as bounces.

JUnit Testing:

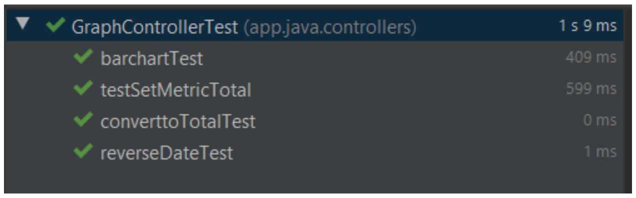
We have implemented tests for various parts of our program. We have Unit tests for 3 controllers (GraphController, DateRangePopupController and DefineBouncePopupController) to test SQL queries and boundaries.



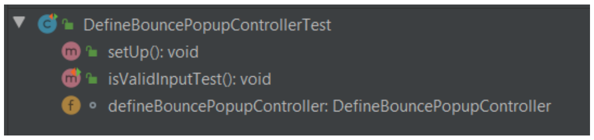
```java
@Test
public void testSetMetricTotal() {

    graphController.currentSQLQuery_Total_Name="Impressions";
    assertEquals( message: "Impressions did not match 486104",String.valueOf(486104),
            graphController.calcMetricTotal( chosenSQLQuery: "SELECT COUNT (ImpressionCost) AS YValue FROM mainTable WHERE ClickCost is null  "));

    graphController.currentSQLQuery_Total_Name="Clicks";
    assertEquals( message: "Clicks did not match 23923",String.valueOf(23923),
            graphController.calcMetricTotal( chosenSQLQuery: "SELECT COUNT (ClickCost) AS YValue FROM mainTable WHERE ClickCost is not null  "));

    graphController.currentSQLQuery_Total_Name="Conversions";
    assertEquals( message: "Conversions did not match 2026",String.valueOf(2026),
            graphController.calcMetricTotal( chosenSQLQuery: "SELECT COUNT (Conversion) AS YValue FROM mainTable WHERE Conversion = 'Yes'"));

    // THIS WILL CHANGE WITH LATEST PUSH HAS TO INCLUDE £ + ALL AVERAGES ARE ROUNDED IN THIS FUNCTION
    graphController.currentSQLQuery_Total_Name="Total Cost";
    assertEquals( message: "Total Cost did not match 118097.9212", expected: "£"+String.valueOf(118097.9212),
            graphController.calcMetricTotal( chosenSQLQuery: "SELECT  SUM (CASE WHEN ClickCost is null then ImpressionCost else 0 end) + SUM (ClickCost) AS YValue FROM mainTable"));
```



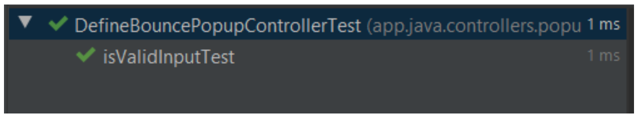
```
▼ GraphControllerTest
    setUp(): void
    barchartTest(): void
    testSetMetricTotal(): void
    converttoTotalTest(): void
    reverseDateTest(): void
    graphController: GraphController
    panel: JFXPanel = new JFXPanel()
```



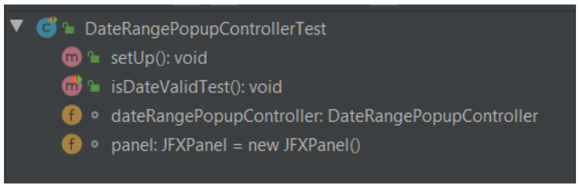
```
▼ ✓ GraphControllerTest (app.java.controllers)   1 s 9 ms
    ✓ barchartTest                               409 ms
    ✓ testSetMetricTotal                         599 ms
    ✓ converttoTotalTest                           0 ms
    ✓ reverseDateTest                              1 ms
```



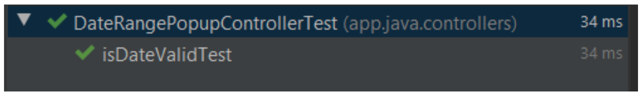
```
▼ DefineBouncePopupControllerTest
    setUp(): void
    isValidInputTest(): void
    defineBouncePopupController: DefineBouncePopupController
```



```
▼ ✓ DefineBouncePopupControllerTest (app.java.controllers.popu 1 ms
    ✓ isValidInputTest                                         1 ms
```



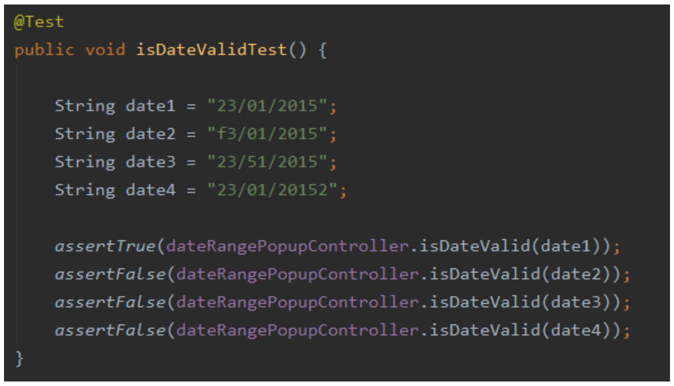
```
▼ DateRangePopupControllerTest
    setUp(): void
    isDateValidTest(): void
    dateRangePopupController: DateRangePopupController
    panel: JFXPanel = new JFXPanel()
```



```
▼ ✓ DateRangePopupControllerTest (app.java.controllers)   34 ms
    ✓ isDateValidTest                                     34 ms
```

Boundary Testing:

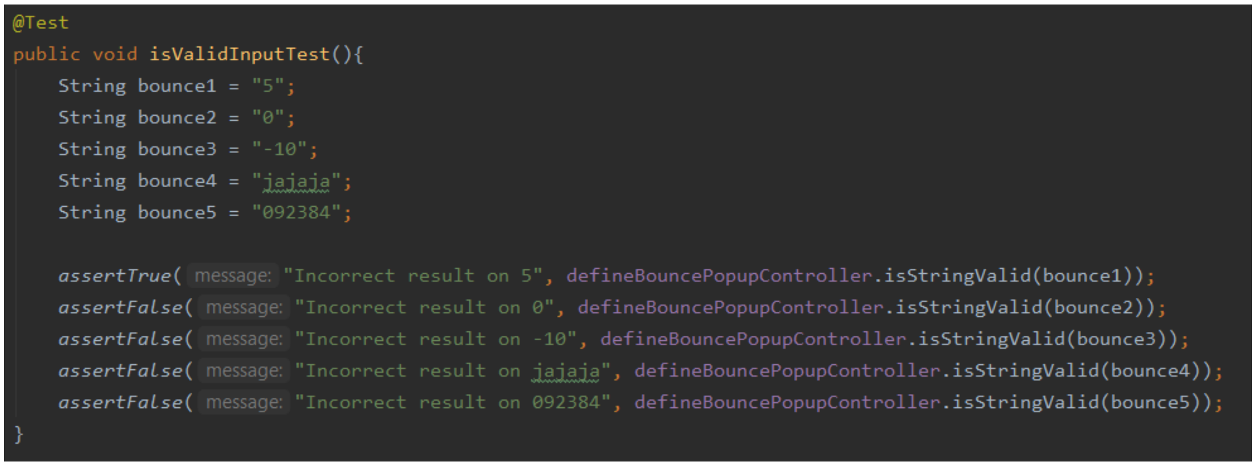
Date Range:



```
@Test
public void isDateValidTest() {

    String date1 = "23/01/2015";
    String date2 = "f3/01/2015";
    String date3 = "23/51/2015";
    String date4 = "23/01/20152";

    assertTrue(dateRangePopupController.isDateValid(date1));
    assertFalse(dateRangePopupController.isDateValid(date2));
    assertFalse(dateRangePopupController.isDateValid(date3));
    assertFalse(dateRangePopupController.isDateValid(date4));
}
```

We used unit tests to test the boundary cases further for the date range textfield. The text field rejects any dates that do not have a length of 10 characters or contains any letters.

Bounce Definition

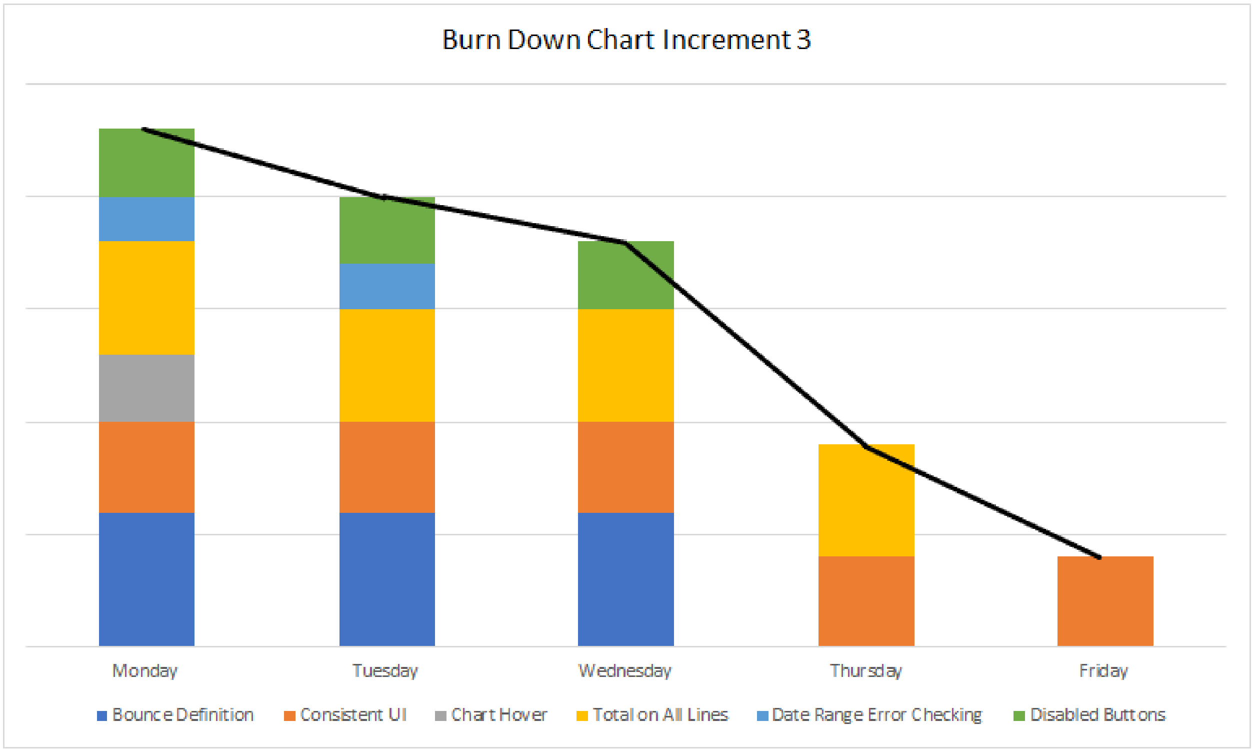


```
@Test
public void isValidInputTest(){
    String bounce1 = "5";
    String bounce2 = "0";
    String bounce3 = "-10";
    String bounce4 = "jajaja";
    String bounce5 = "092384";

    assertTrue( message: "Incorrect result on 5", defineBouncePopupController.isStringValid(bounce1));
    assertFalse( message: "Incorrect result on 0", defineBouncePopupController.isStringValid(bounce2));
    assertFalse( message: "Incorrect result on -10", defineBouncePopupController.isStringValid(bounce3));
    assertFalse( message: "Incorrect result on jajaja", defineBouncePopupController.isStringValid(bounce4));
    assertFalse( message: "Incorrect result on 092384", defineBouncePopupController.isStringValid(bounce5));
}
```

We used unit testing to ensure that all entered values into the bounce definition text fields are handled appropriately. The text field only accepts positive integers and its correctness is determined by the isStringValid() function which runs against the values of the text field after the confirm button has been pressed. We tested how the function responds to inputs which contain strings and negative integers.

Planning:

We were able to get through all of the tasks in the sprint.


Burn Down Chart Increment 3
Monday
Tuesday
Wednesday
Thursday
Friday
Bounce Definition
Consistent UI
Chart Hover
Total on All Lines
Date Range Error Checking
Disabled Buttons