Increment 2 report

Team 38
- Alex - tll1g19
- Carlos - cab1g19
- Jamaal - jm3g19
- Stoyan - sv1u19
- Tom - tdh1g19
- Vlad - vgh1u19

Design choices

BACKEND

Filtering Metrics and Charts

- During the first increment we displayed all the metrics and a very basic chart that displayed all the data from the dataset.
- We have since added functional dropdown menus for the filters and are easily accessible at the top of the relevant pages.
- As highlighted during increment 1, we added multithreading. When filtering, we made a few design choices that indicates to the user what is going on.
- Without multithreading, the program would freeze whilst it worked. With multithreading it doesn't and then the metrics/charts update a second later.
- To show that the program is "working" we disable the filters and change the cursor to a loading icon. Most internet users should be familiar with this as a display of "loading."
- We stopped the compare charts with 2 lines and changed it to be able to compare two charts by showing them side by side on the compare page (see the changes in the burndown chart)

Calculators

- An important part of our backend calculations was to use Java classes for storing, reading and calculating the data. For the first increment, we created a general collection of classes that we may have needed for this increment. During this increment, we properly planned out the backend structure before we began with the filtering.
- The first designed classes were the entries. This was quite important as we needed to make the data as atomic as possible and hence we designed 3 subclasses of Entry for each log file. The superclass has a value for the user id.
- We also designed a campaign class that is responsible for storing all the data and reading in the files.
- For the filtering and calculating we also wanted to use calculators for metrics, charts and histogram. During the development, we made the design choice to cut the histogram as it wasn't needed as there would be no need for filtering so its functions were integrated into the chart calculator.

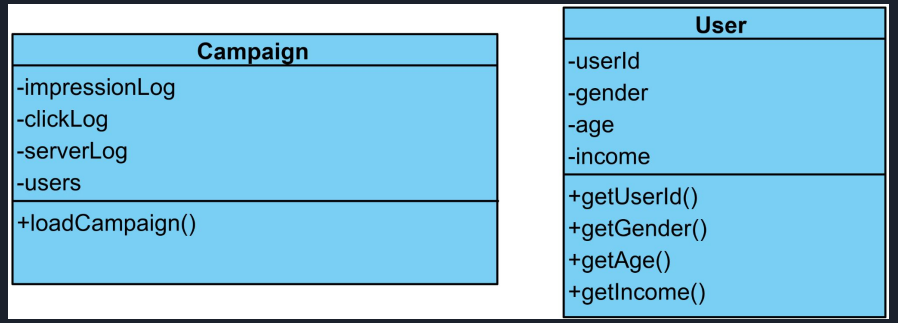
Backend Planning

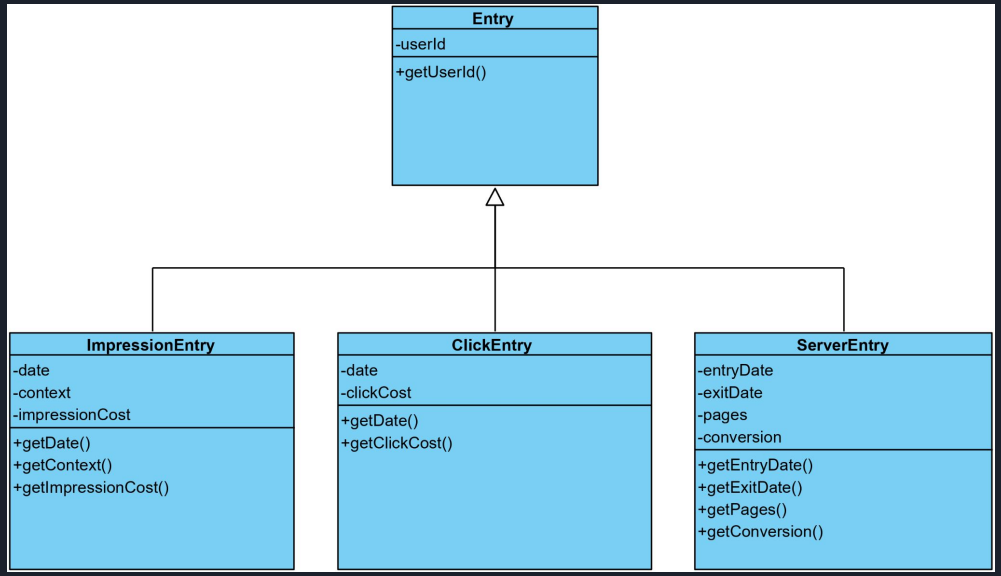
- One of the most important bits of feedback that we acted on was to improve our planning of the programming, particularly on the backend.
- Based on this, we created UML diagrams prior to developing the code to properly convey how we would approach the development.
- This allowed us to split up our development into 3 main sections: entries (for storing / reading), calculators and the general campaign class which would act as the interface between the GUI and the backend.
- As a result, we were able to work on sections separately and together allowing us to work more collaboratively.

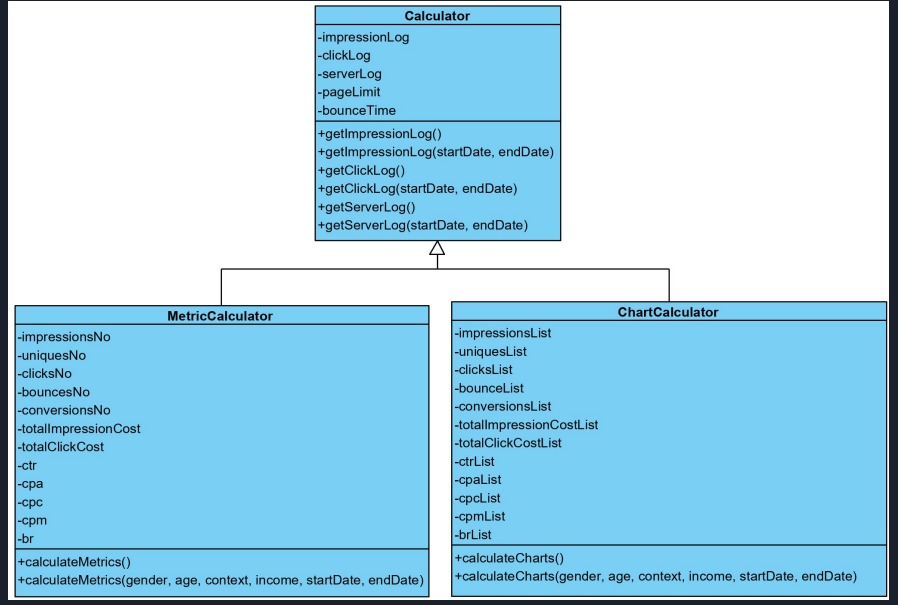
Design Artifacts

Initial UML DIagrams vs Final UML Diagram of the classes

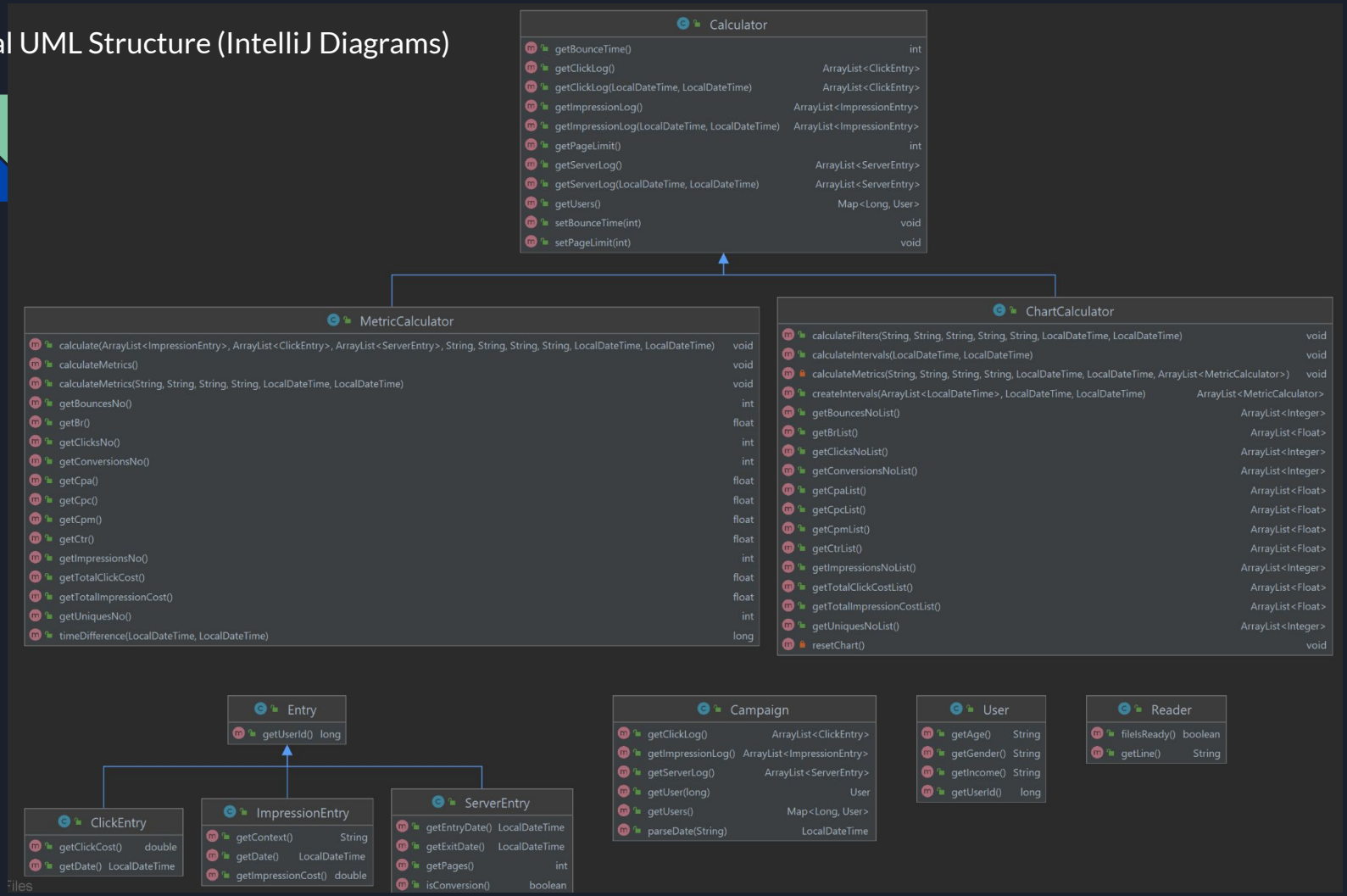
UML Design (using Visual Paradigm)







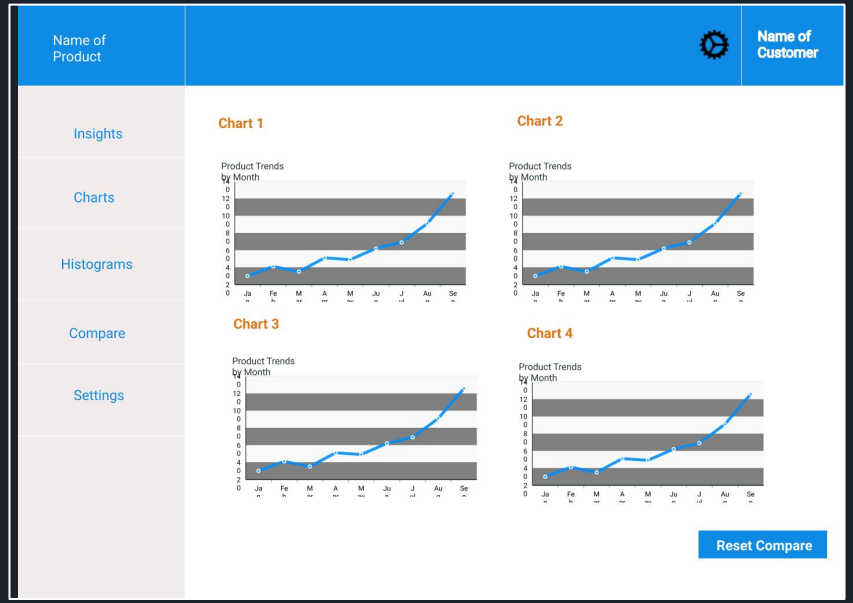
Final UML Structure (IntelliJ Diagrams)



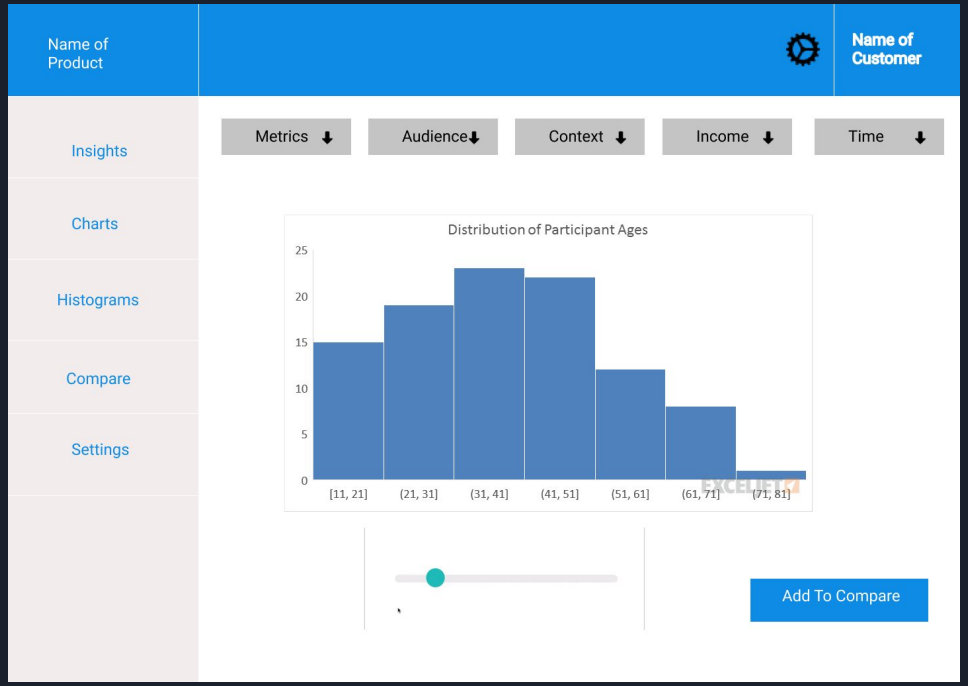
Design Artifacts

Storyboards

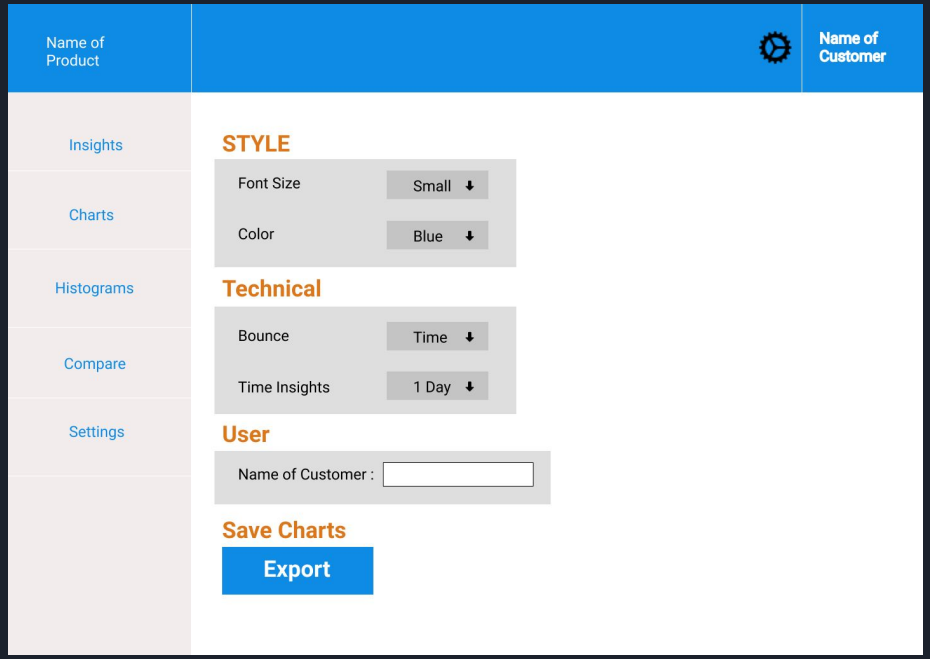
Compare Page



Histogram Page



Settings Page



GUI - FrontEnd

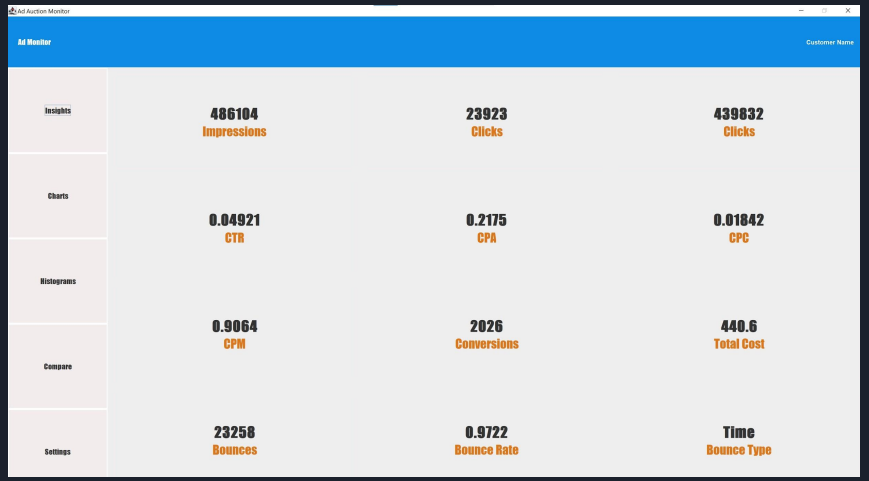
General Changes

- Full screen Application. We decided to make it full screen after talking with the supervisor and presented this idea.Managing and analyzing of all the numbers of the ads is challenging and the user shouldn't be interrupted or distracted by what is happening on the side on the computer. Also it is a choice based on the top bar of our application would look bad in combination with the windows top bar, which is default for every window app. Beside it was a technical choice because we wanted to be able to show all the metrics on one page.
- We added an exit button in a distinct red color that allows the user to close the program, and will also be warned when they are about to do so. This was in response to make the program fullscreen to focus the users attention and of removing the windows top bar border which had the previous close button.

- A load campaign button to bring up a menu for selecting files. Users can then confirm their choices to load the data. There is also a temporary testing button for demonstration purposes (will be removed for the last increment.)

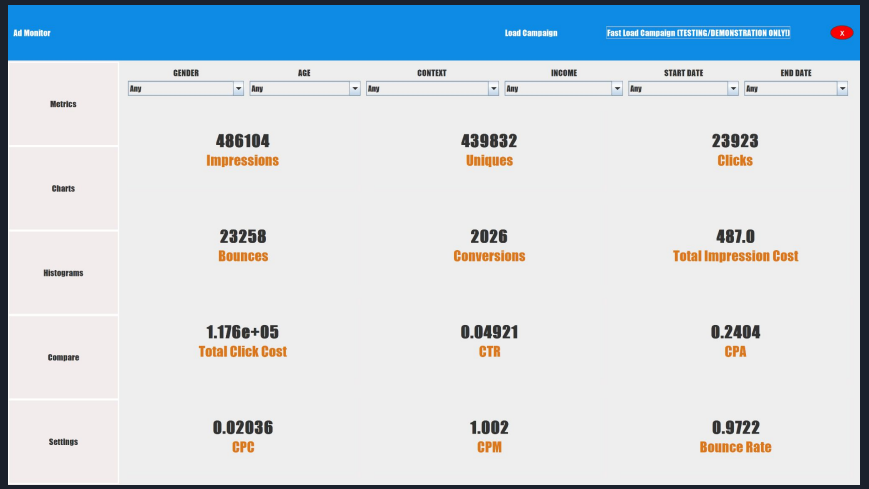
Metrics Page

OLD



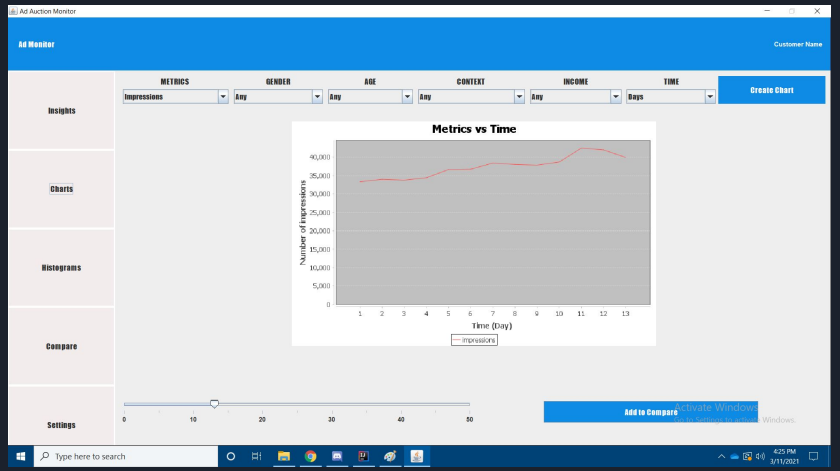
- Added filtering for the metrics page
- Added “Total Click Cost” as a metric and moved bounce type in the settings page

NEW



Charts Page

OLD



- Deleted the “create chart” button and made it load automatically for easier use
- Changed the slider to display measure units for time instead of numerical values
- Added “start date” and “end date” as filters

NEW

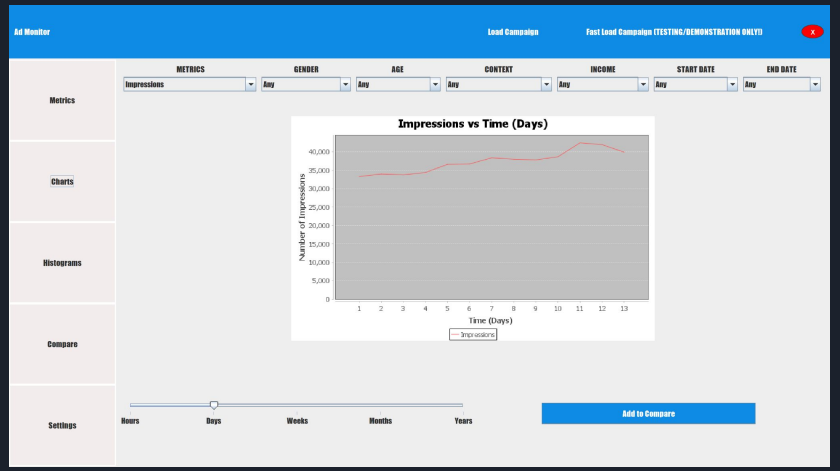
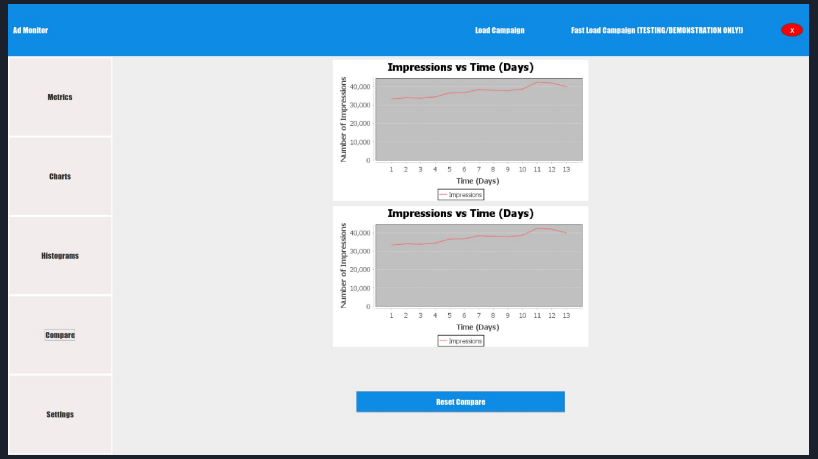


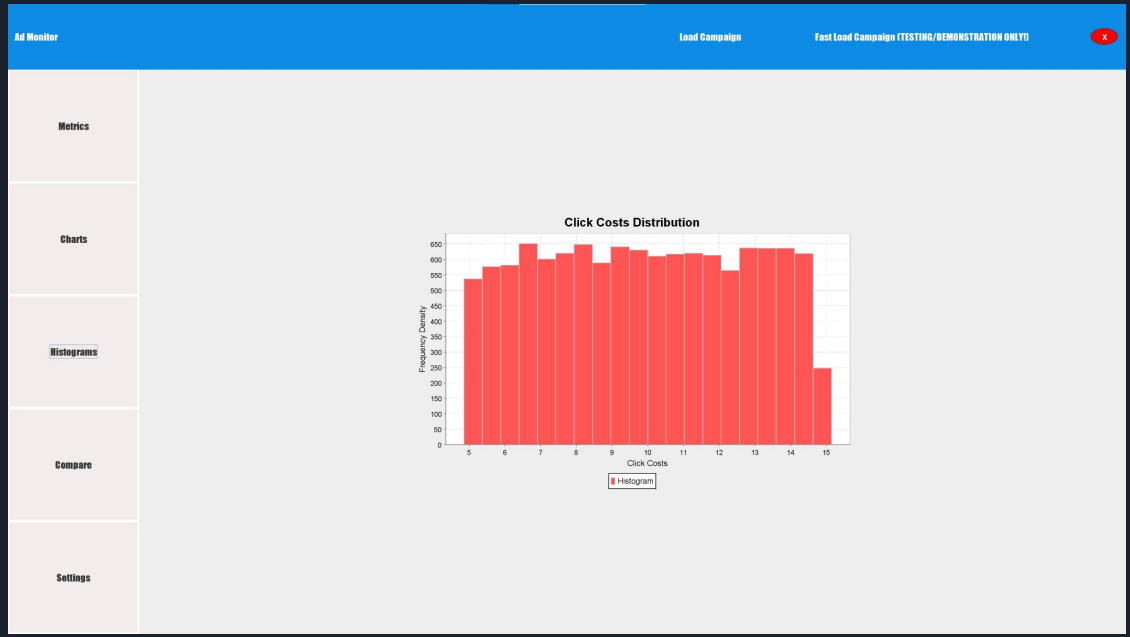
Chart Compare Page

- Charts displayed side by side for easier comparison
- Maximum of 4 charts can be added



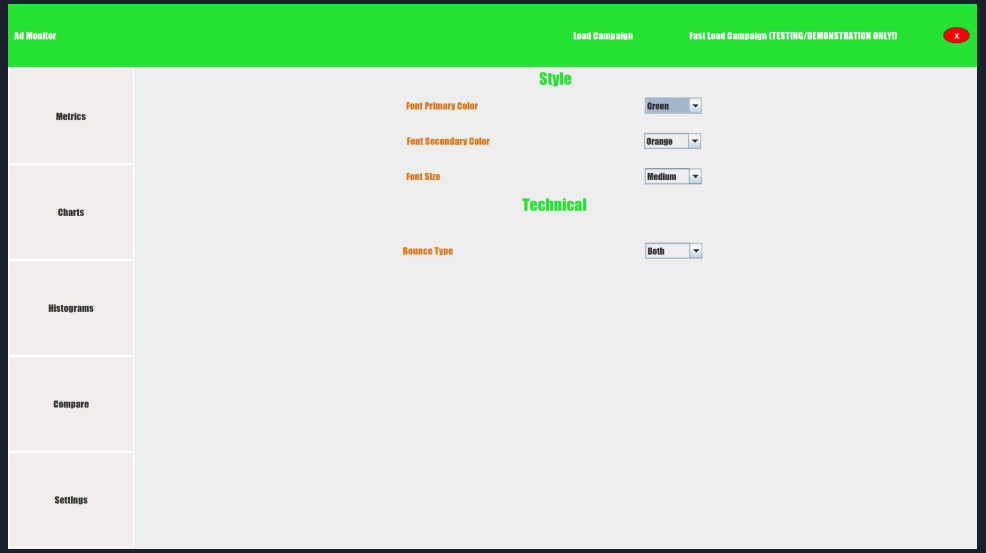
Histograms Page

- Histogram page was added
- The distribution of the click costs are shown
- Got rid of the filtering from the storyboard of the histogram page because the histogram is only for Click Cost Distribution



Settings Page

- Settings page was created
- The functionality of changing the Primary and Secondary Color
- The functionality of changing the font size



General Changes

- We added an exit button in a distinct red color that allows the user to close the program, and will also be warned when they are about to do so. This was in response to make the program fullscreen to focus the users attention.

- A load campaign button to bring up a menu for selecting files. Users can then confirm their choices to load the data. There is also a temporary testing button for demonstration purposes (will be removed for the last increment.)

Testing

Tests

Automated Tests

- Unit Testing

Manual Tests

- User scenarios (based on user stories)
- Manually comparing values displayed on Insights and values in Charts

Automated Tests

Metric calculator unit tests

ID	
Positive Long	No error
Negative Long	Logical mistake w/o error msg
Decimals	java.lang.NumberFormatException
Incorrect format (e.g. ABC)	java.lang.NumberFormatException

Metric calculator unit tests (cont.)

Date / Entry date / Exit date	
<0001-01-01 00:00:00	DateTimeParseException
0001-01-01 00:00:00 -9999-12-30 12:59:59	No error
>10000-01-01 00:00:00	DateTimeParseException
Incorrect Format (e.g. ABC)	DateTimeParseException
Exit date < Entry date	Logical mistake w/o error msg

Metric calculator unit tests (cont.)

Click cost / Impression cost	
0	No errors
Positive doubles	No errors
Negative Doubles	Logical mistake w/o error msg
Incorrect Format (e.g. ABC)	Number format error

Metric calculator unit tests (cont.)

Gender / Conversion / Age / Income / Context	
Correct input (e.g. 25-34, blog etc.)	No error
Incorrect string (e.g. ABC)	Logical mistake w/o error msg
Incorrect format (e.g. 123, !&% etc.)	Logical mistake w/o error msg

Metric calculator unit tests (cont.)

Pages viewed	
0	No error
Positive Ints	No error
Negative Ints	Logical mistake w/o error msg
Invalid format (e.g. ABC)	java.lang.NumberFormatException

Manual Testing

Increment 2 Relevant User Stories

User Stories

Based on our feedback from the previous deliverables, we have created more user stories that are more specific on actions.

(1) As a <fashion business owner>

I want <to be able to identify which gender are buying more of our clothes>

So that <I can invest more money into producing clothes for our current audience>

Filtering

(2) As a <retail business who has very limited stock>

I want <be able to see our sales with more detail>

So that <we can identify and micro market to our most active customers>

Time Granularity (Small)

(3) As a <salesman for an up and coming business>

I want <to be able to see that over a large timescale we are receiving more ad engagement>

So that <we can demonstrate to investors our business is growing>

Time Granularity (Large)

(4) As a <large business owner>

I want <be able to change the look of the product>

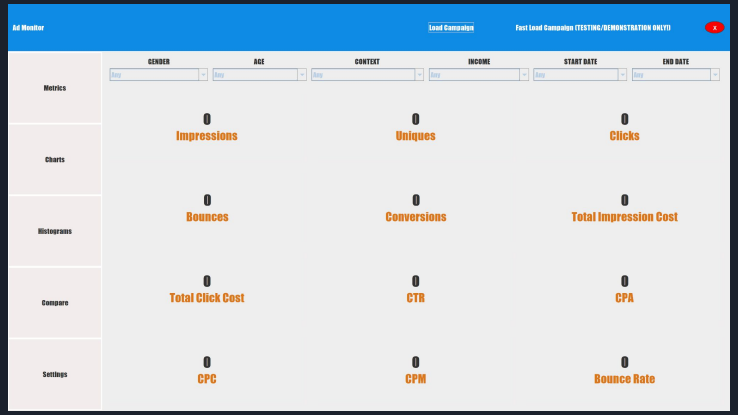
So that <it feels on theme to our business>

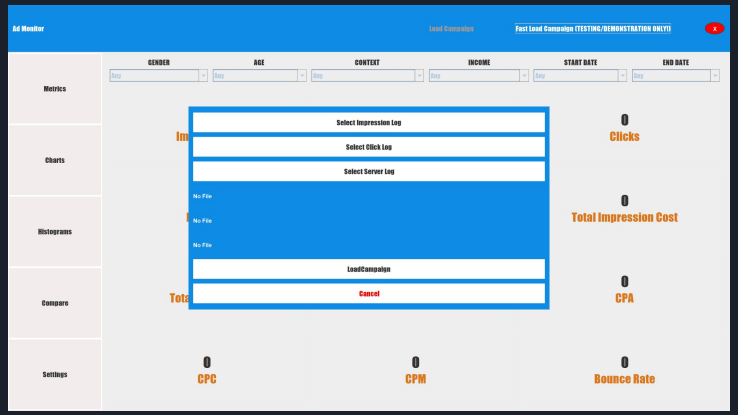
?

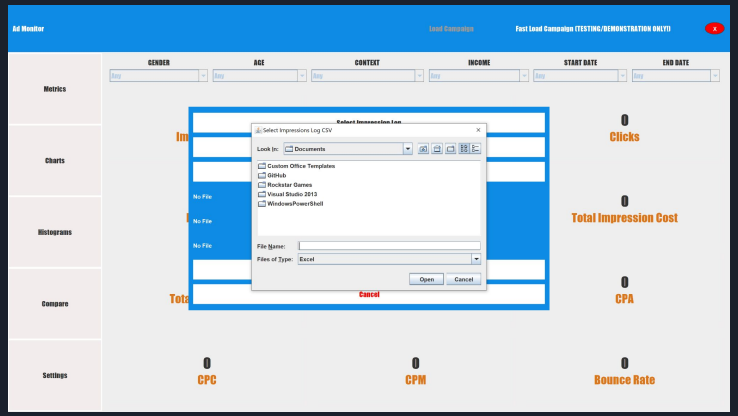
User scenarios and outputs

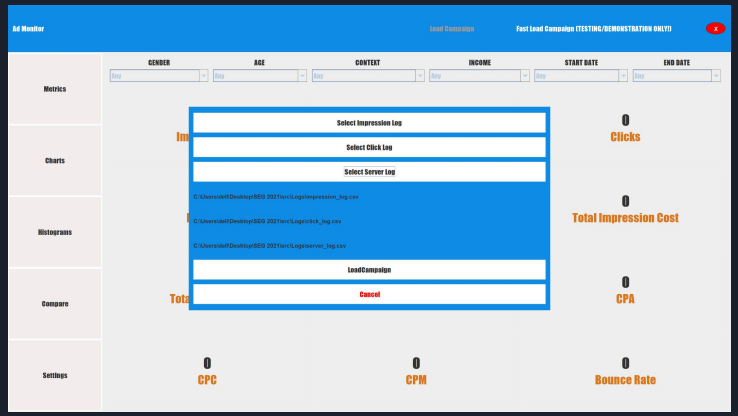
Scenario: User wants to load campaign and see metrics

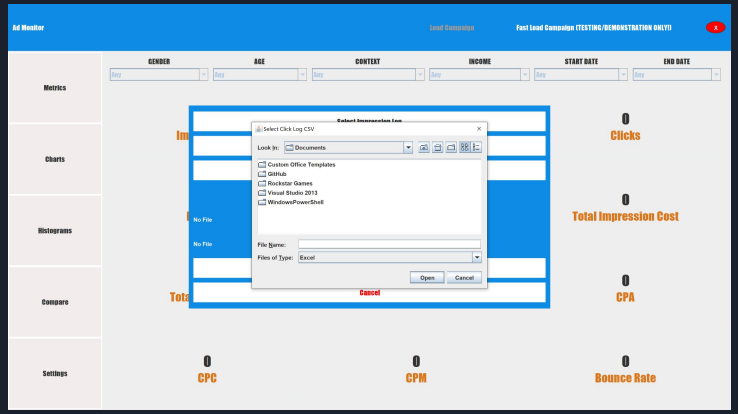
1. User opens application
2. Application loads
3. User presses “load campaign” button
4. Pop up appears where user can input the files
5. User presses “select impressions log”
6. Application opens directory for user to choose file
7. User presses “select clicks log”
8. Application opens directory for user to choose file
9. User presses “select server log”
10. Application opens directory for user to choose file
11. User presses “load campaign” button
12. Pop up disappears and campaign is loaded
13. Metrics are displayed in the metrics page

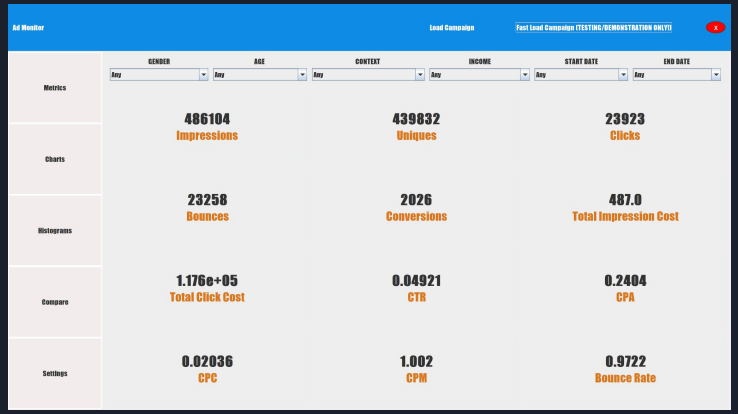





Ad Monitor
Load Campaign
Fast Load Campaign (TESTING/DEMONSTRATION ONLY!)
Metrics
Charts
Histograms
Compare
Settings
GENDER
AGE
CONTEXT
INCOME
START DATE
END DATE
Any
Select Impressions Log CSV
Look In:
Documents
Custom Office Templates
GitHub
Rockstar Games
Visual Studio 2013
WindowsPowerShell
File Name:
Files of Type:
Excel
Open
Cancel
No File
No File
No File
Cancel
0
Clicks
0
Total Impression Cost
0
CPA
0
CPC
0
CPM
0
Bounce Rate


Ad Monitor
Load Campaign
Fast Load Campaign (TESTING/DEMONSTRATION ONLY!)
Metrics
Charts
Histograms
Compare
Settings
GENDER
AGE
CONTEXT
INCOME
START DATE
END DATE
Any
Select Impression Log
Select Click Log
Select Server Log
C:\Users\dell\Desktop\SEO 2021\arc\Logs\impression_log.csv
C:\Users\dell\Desktop\SEO 2021\arc\Logs\click_log.csv
C:\Users\dell\Desktop\SEO 2021\arc\Logs\server_log.csv
LoadCampaign
Cancel
0
Clicks
0
Total Impression Cost
0
CPA
0
CPC
0
CPM
0
Bounce Rate


Ad Monitor
Load Campaign
Fast Load Campaign (TESTING/DEMONSTRATION ONLY!)
Metrics
Charts
Histograms
Compare
Settings
GENDER
AGE
CONTEXT
INCOME
START DATE
END DATE
Any
Select Click Log CSV
Look In:
Documents
Custom Office Templates
GitHub
Rockstar Games
Visual Studio 2013
WindowsPowerShell
File Name:
Files of Type:
Excel
Open
Cancel
No File
No File
Cancel
0
Clicks
0
Total Impression Cost
0
CPA
0
CPC
0
CPM
0
Bounce Rate

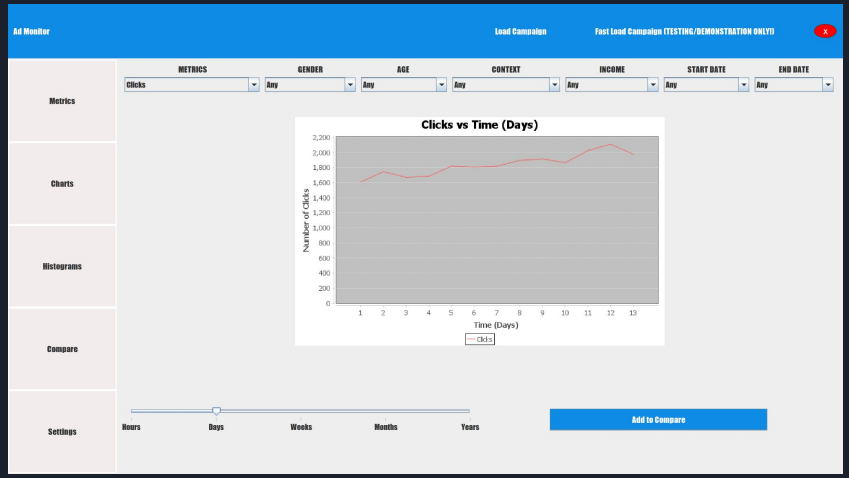

Ad Monitor
Load Campaign
Fast Load Campaign (TESTING/DEMONSTRATION ONLY!)
Metrics
Charts
Histograms
Compare
Settings
GENDER
AGE
CONTEXT
INCOME
START DATE
END DATE
Any
486104
Impressions
439832
Uniques
23923
Clicks
23258
Bounces
2026
Conversions
487.0
Total Impression Cost
1.176e+05
Total Click Cost
0.04921
CTR
0.2404
CPA
0.02036
CPC
1.002
CPM
0.9722
Bounce Rate

Scenario 1: Jeff, Car Shop Owner

- Jeff wants to know how effective his campaign was by looking at a chart of how many people clicked on his ad. Jeff opens the Ad Monitor application, loads a campaign on the main page and opens the charts page using the “Charts” button on the left side vertical menu. He then selects “Clicks” from the drop down menu and a chart is displayed in the middle of the UI.

Scenario: User wants to see a chart of the clicks an ad campaign generated

1. User opens application
2. User loads campaign
3. User presses “Charts” button in the left vertical menu
4. Application displays charts page with “Impressions as default”
5. User changes the metric in the filter to “Clicks”
6. Clicks chart is displayed

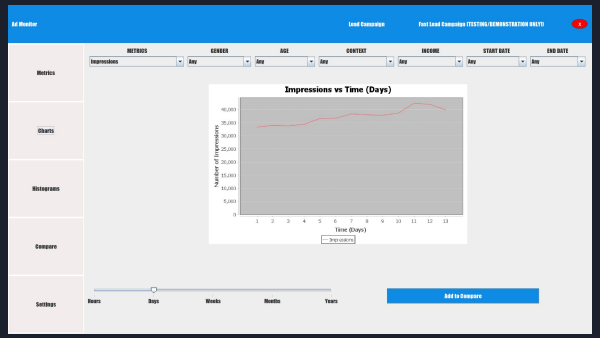


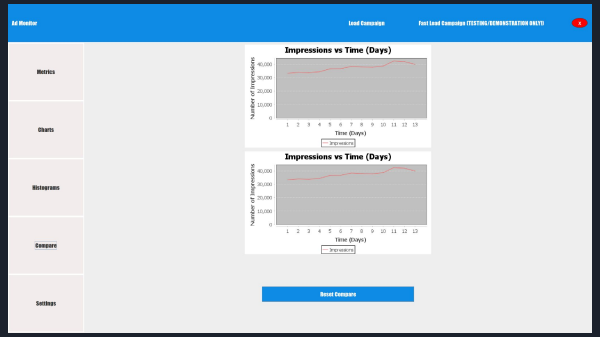
Scenario 2: Jeff, Car Shop Owner

- Jeff wants to compare the impressions he got from the first month of the campaign to the last month of the campaign.
- Jeff opens the “ad monitor” application. He loads the campaign on the main page. He presses the “Charts button” to go to the charts page. He filters the chart to show impressions in the first month of the campaign, then presses the “Add to compare button”. He changes the filters to display the impressions in the last month, Then presses “Add to compare button” again. He presses the “compare” button to compare the two graphs.Afterwards, he wants to get rid of the charts so he presses the “Reset Compare” button at the bottom of the UI.

Scenario 2: User wants to compare 2 “Impressions” charts

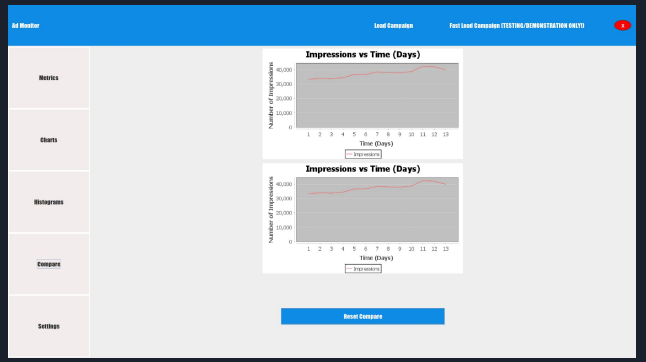
1. User opens application
2. User loads campaign
3. User presses “Charts” button in the left vertical menu
4. Application displays charts page with “Impressions as default”
5. User filters chart
6. User presses “Add to compare button”
7. User changes filters again
8. User presses “Add to compare button”
9. User presses “Compare” button in the left vertical menu
10. Application displays compare page with the 2 charts side by side

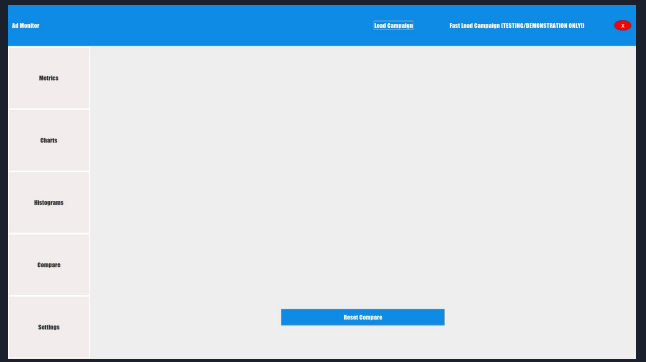




Scenario 2.2: User wants to reset compared charts

1. User opens application
2. User loads campaign
3. User presses “Charts” button in the left vertical menu
4. Application displays charts page with “Impressions as default”
5. User filters chart
6. User presses “Add to compare button”
7. User changes filters again
8. User presses “Add to compare button”
9. User presses “Compare” button in the left vertical menu
10. Application displays compare page with the 2 charts side by side
11. User presses “Reset Compare” button in “Compare” page
12. Application displays blank compare page



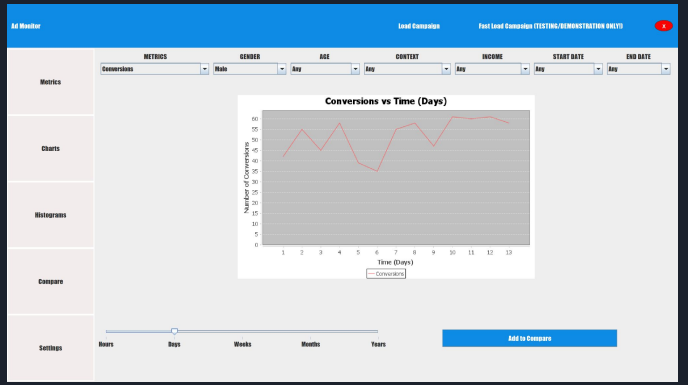


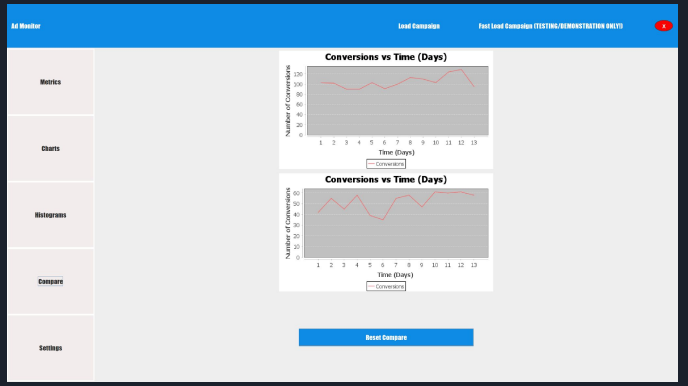
Scenario 3: Bianca, Clothing Brand Owner

- Bianca opens the *Ad Monitor* application aiming to find the differences between conversions of male and female shoppers. While in the main page, she loads a campaign and the goes to the charts page using the left side vertical menu of the UI. She then filters the chart to display male conversions and adds it to the compare page using the button “Add to compare”. She repeats the process for female conversions. Then she goes to the compare page and looks at the 2 graphs side by side.

Scenario 3: User wants to compare “Conversions” charts of “female” and “male” gender

1. User opens application
2. User loads campaign
3. User presses “Charts” button in the left vertical menu
4. Application displays charts page with “Impressions as default”
5. User changes the metric filter to “Conversions” and the gender filter to “female:
6. “Female Conversions” chart is displayed and added to compare
7. User changes the metric filter to “Conversions” and the gender filter to “male”:
8. “Male Conversions” chart is displayed and added to compare
9. User presses “Compare” button in the left vertical menu
10. Application displays the 2 charts side by side



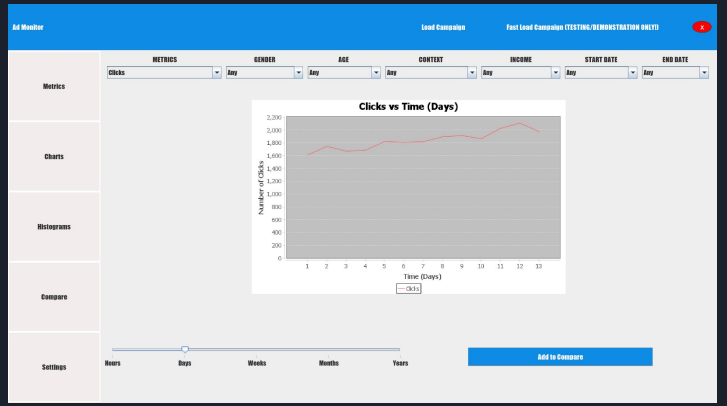


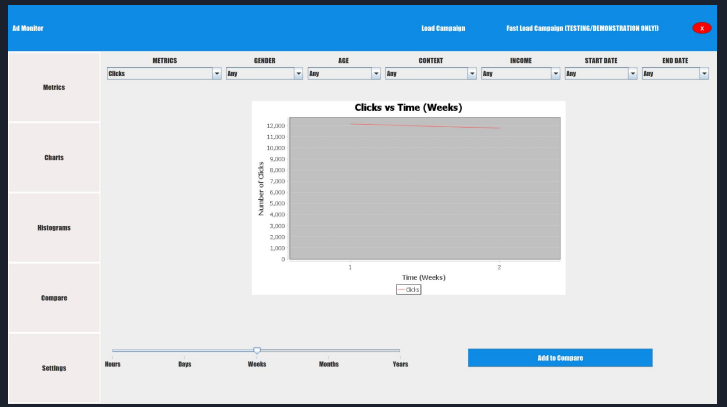
Scenario 4: Stephany, Artist

- Stephany is an up and coming artist who sells her work online .
- Stephany opens the Ad Monitor application and wants to find out how well her advertised work gets clicked on in terms of months instead of days. She goes to the charts page and filters the metrics by number of clicks. Then she adjusts the time granularity slider at the bottom from days to weeks. The chart is now updated displaying total number of clicks in terms of weeks.

Scenario 4: User wants to change time granularity in a chart

1. User opens application
2. User loads campaign
3. User presses “Charts” button in the left vertical menu
4. Application displays charts page with “Impressions as default”
5. User filters chart to display clicks
6. User changes Time granularity to “Weeks” using slider
7. Chart is updated and displayed



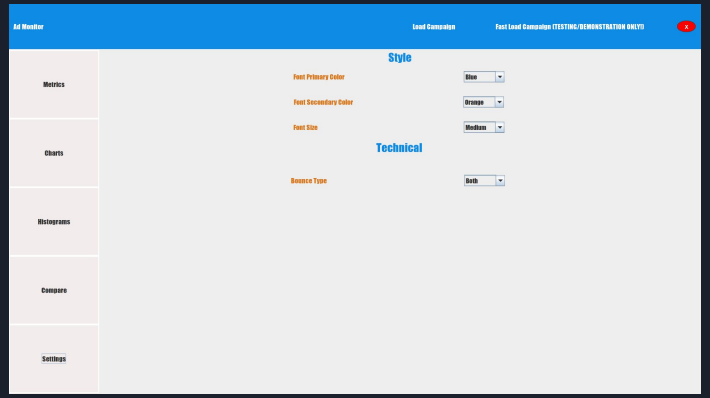


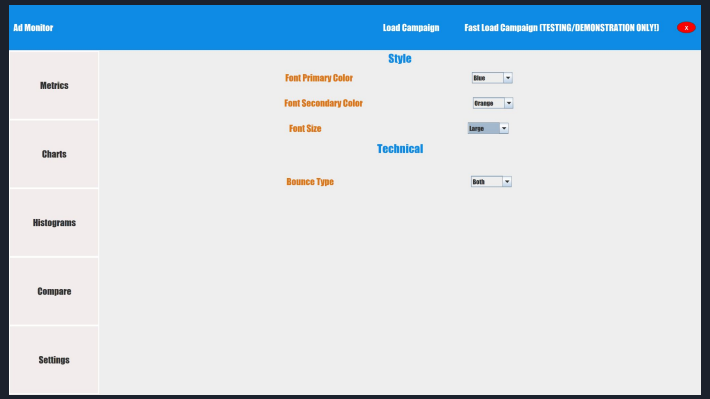
Scenario 5: Stephany, Artist

- Stephany is an up and coming artist who sells her work online
- Stephany opens the Ad Monitor application and finds that the font size and colours are not aesthetically appealing to her artistic vision. She goes to the settings page and chooses her preferred font size. Then she chooses her preferred font primary colour and preferred font secondary colour. The application then reloads with her new customized settings.

Scenario 5.1: User wants to change Font size

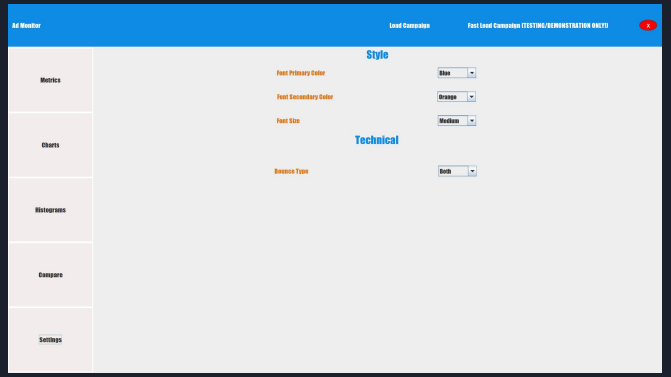
1. User loads application
2. User presses “Settings” button in the left vertical menu
3. Application displays settings page
4. User chooses preferred font size
5. Application reloads with chosen font size

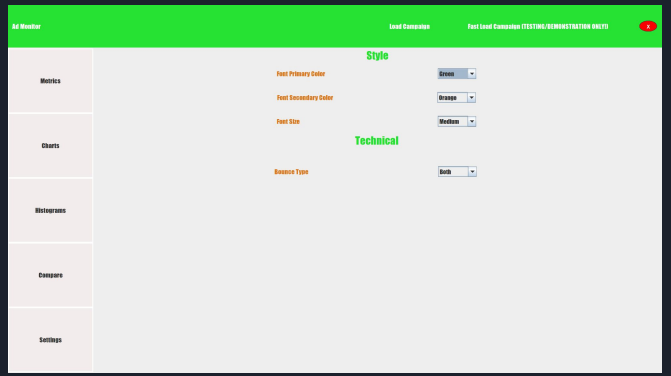




Scenario 5.2: User wants to change Font Primary Color

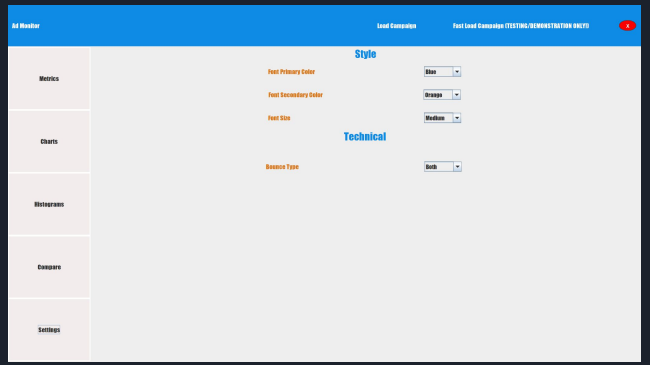
1. User loads application
2. User presses “Settings” button in the left vertical menu
3. Application displays settings page
4. User chooses preferred font primary color
5. Application reloads with chosen font primary color

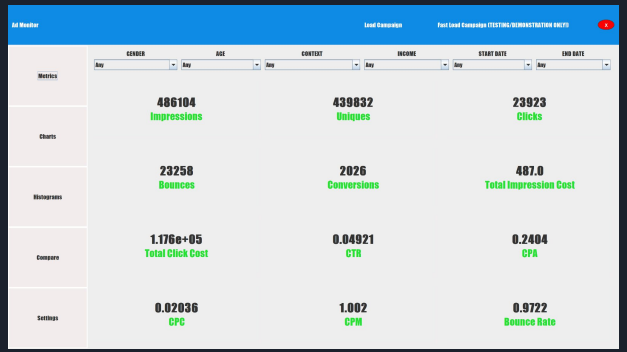


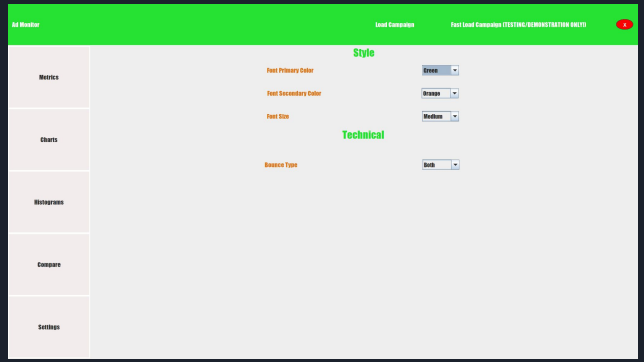


Scenario 5.3: User wants to change Font Secondary Color

1. User loads application
2. User presses “Settings” button in the left vertical menu
3. Application displays settings page
4. User chooses preferred font secondary color
5. Application reloads with chosen font secondary color





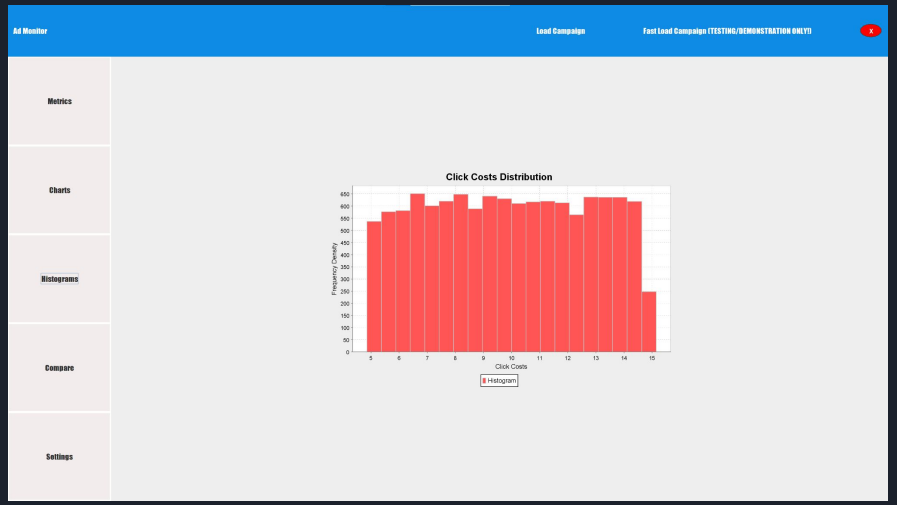


Scenario 6: Bianca, Clothing Brand Owner

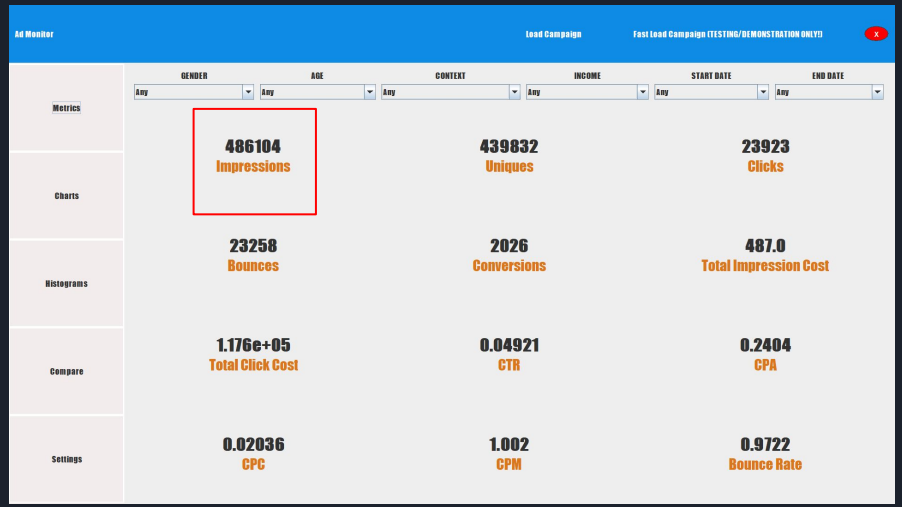
- Bianca opens the *Ad Monitor* application aiming to see how her finances have been used by looking at a histogram of “Click Cost Distribution”. While in the main page, she loads a campaign and the goes to the histogram page using the left side vertical menu of the UI. There, a histogram of “Click Cost Distribution” is displayed in the middle of the UI.

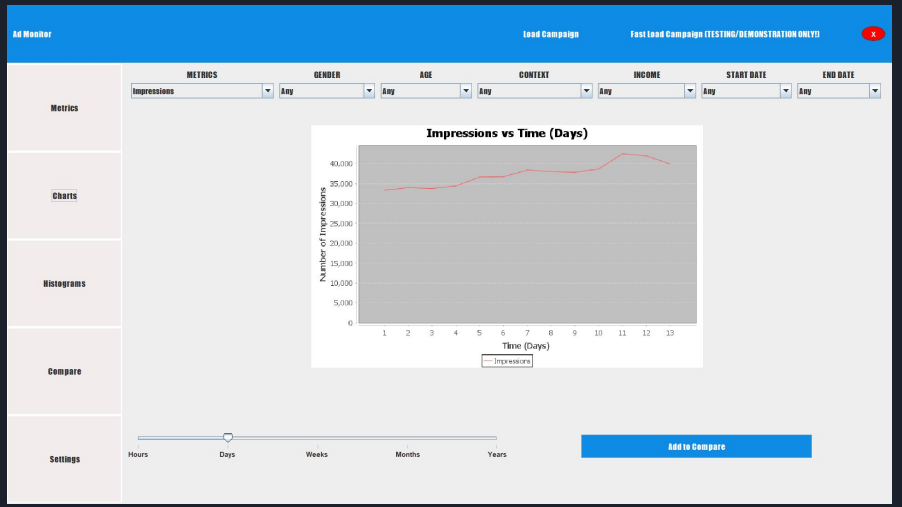
Scenario 6: User wants to see a histogram of Click Cost Distribution

1. User loads campaign
2. User presses “Histogram” button in the left vertical menu
3. Application displays histogram page with loaded “Clicks Cost Distribution” histogram

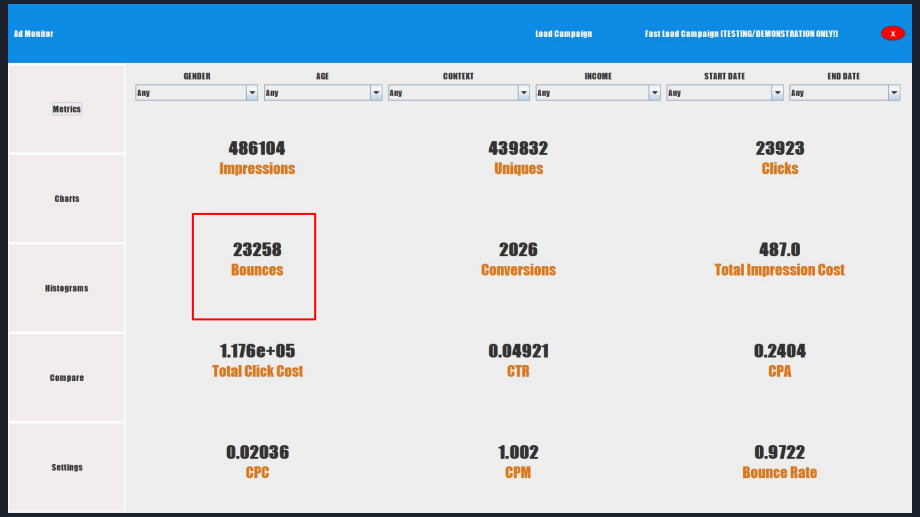


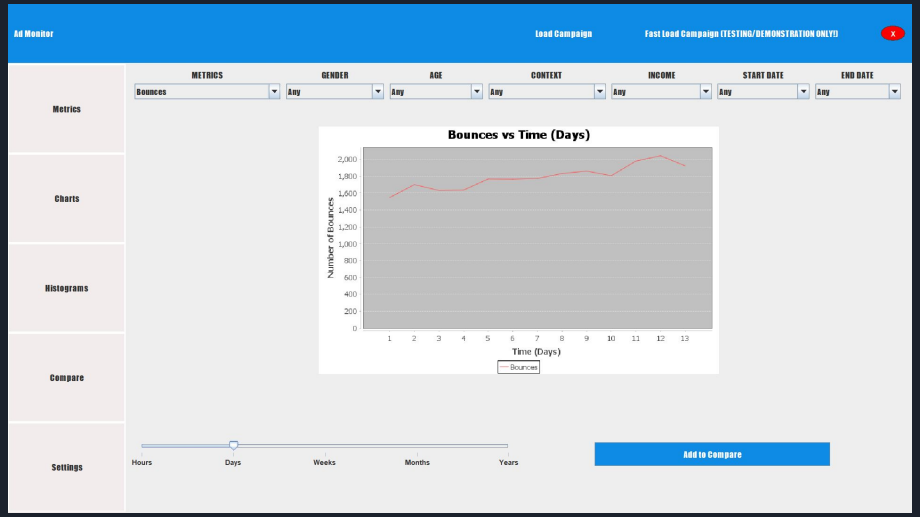
Testing if metrics page values and chart values are the same



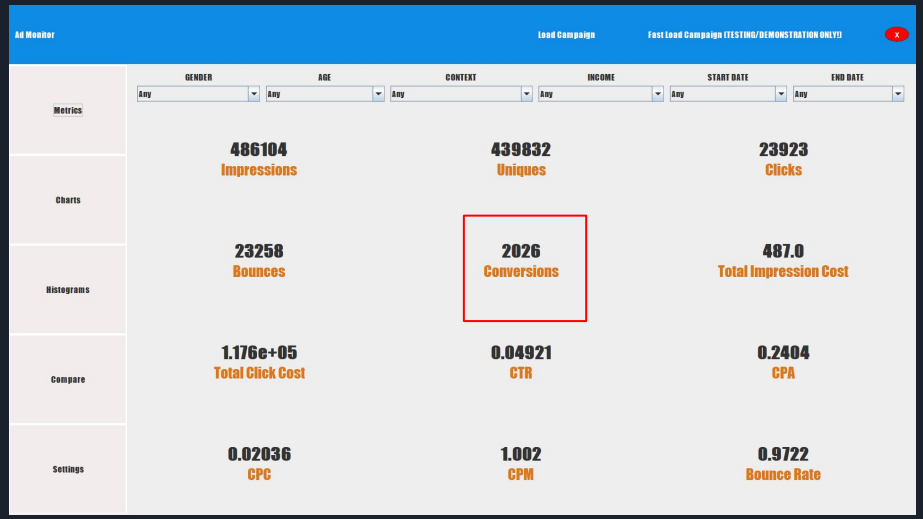


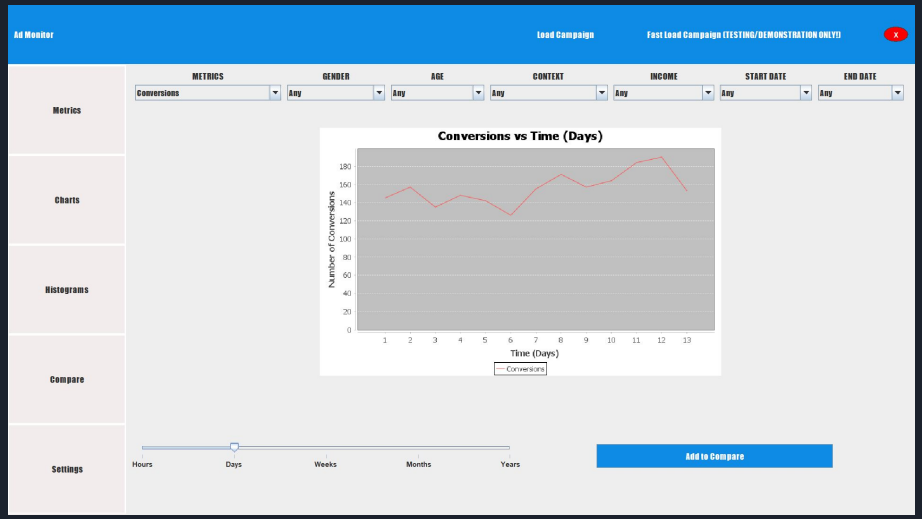
Testing if metrics page values and chart values are the same



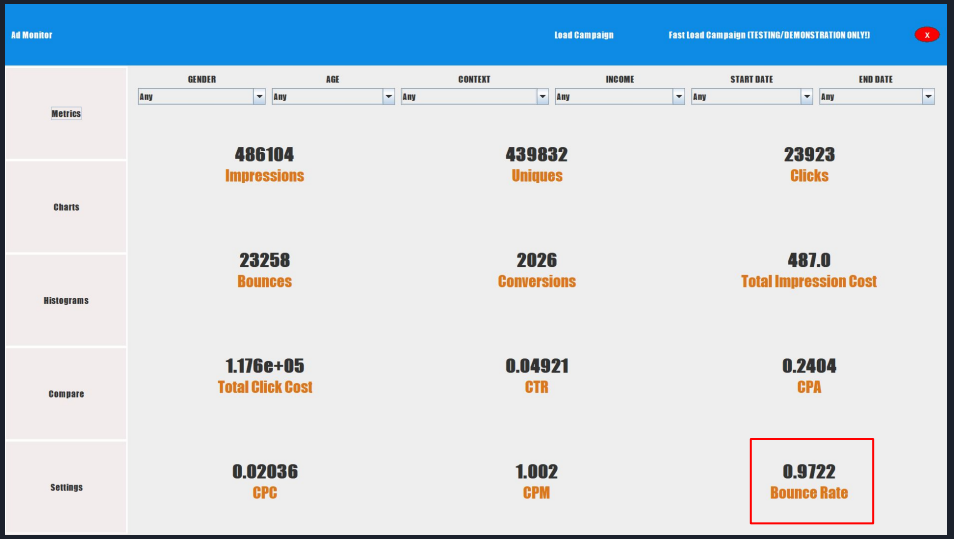


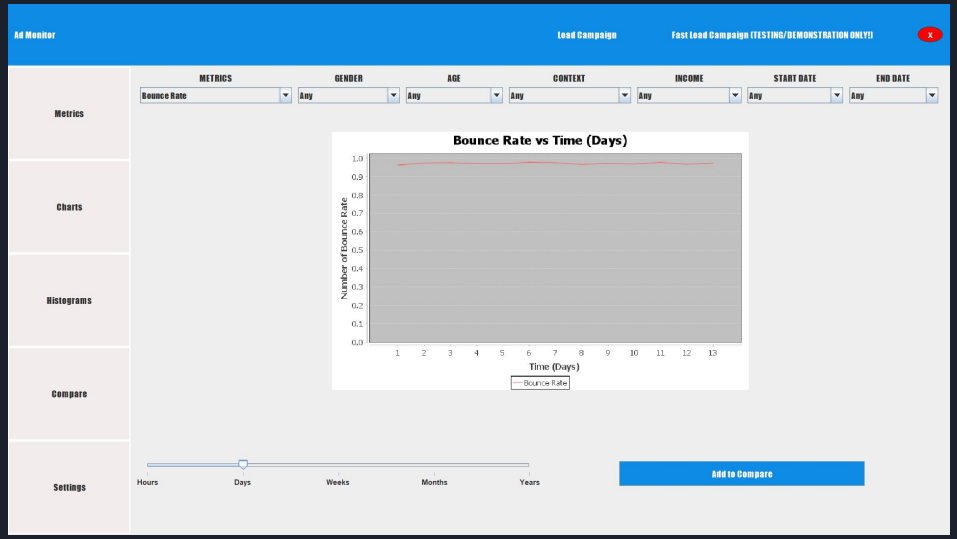
Testing if metrics page values and chart values are the same





Testing if metrics page values and chart values are the same





Burndown Charts

Tasks

Team 1 (Alex + Tom):

- Metrics over time
- Time granularity calculations
- Calculate filters

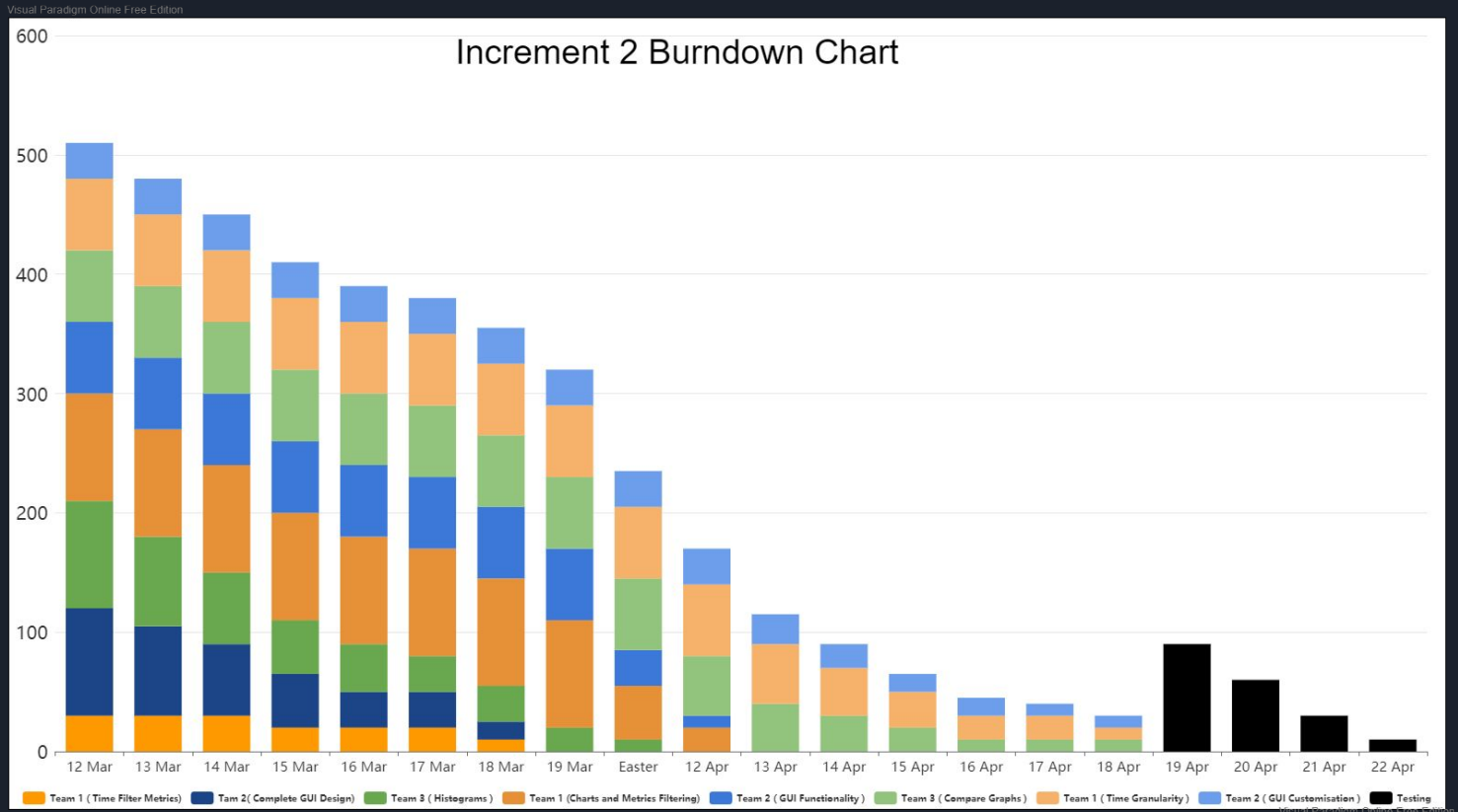
Team 2 (Vlad + Carlos):

- Coding pages of “compare metrics”, “histogram” and “settings”
- Adding functionality to the “compare metrics”, “histogram” and “settings”
- GUI customisation

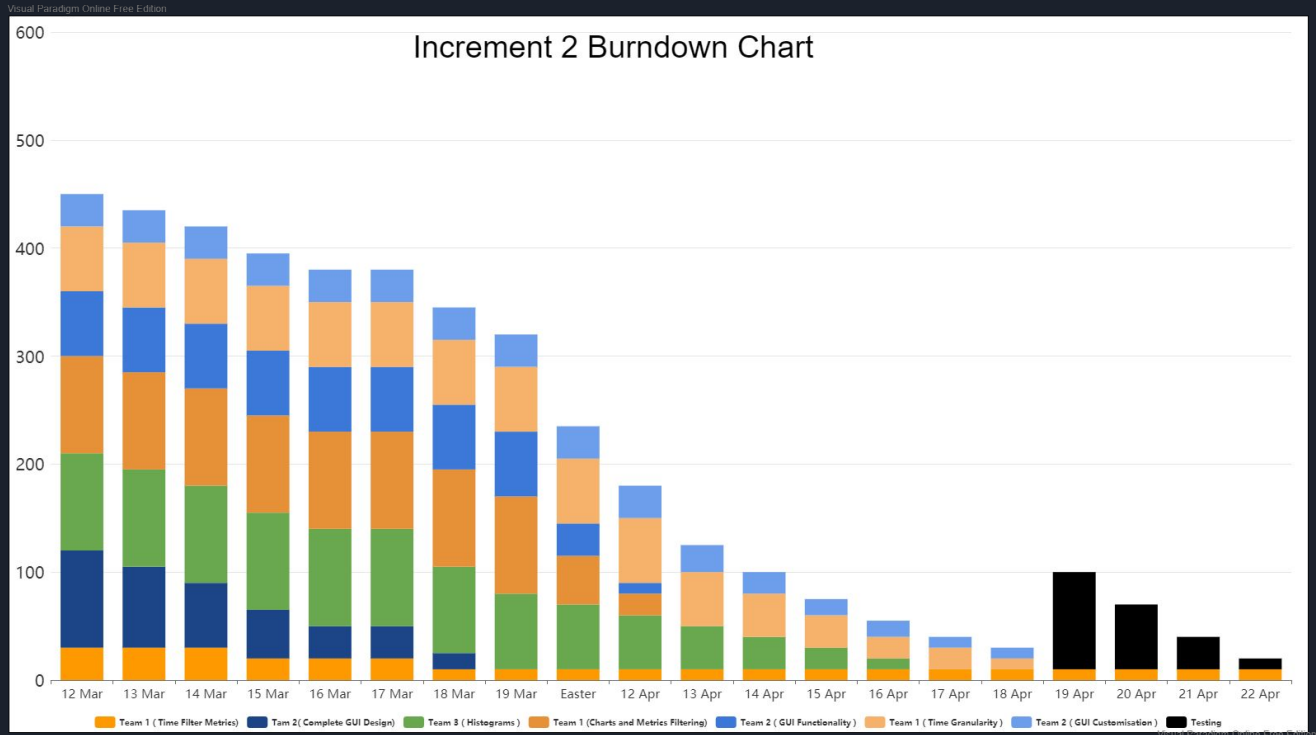
Team 3 (Jamaal + Stoyan):

- Histograms
- Compare graphs (two lines)

Planned Burndown Chart



Actual Burndown Chart



Increment 3 plan

Sprint Plan - (days in red have a Scrum Meeting)

Day	Team 1	Team 2	Team 3
24/04	Define bounce	Create Bounce Type filtering on settings page	Save Charts as FIles
25/04			
26/04	Filter Start and End Date		
27/04	Testing and Bug Fixes	Testing and Bug Fixes	Printing Functionality
28/04			Test and Bug Fixes
29/04			
30/04			

Increment 3 goals

- Allow user to define a bounce (Small)
- Create Bounce Type filtering on settings page (Small)
- Save charts as files (<Medium)
- Filter Start and End Date (Small) (remaining from last increment see burndown)
- Printing functionality (<Small)
- Testing and Bug Fixes

Increment 3 Burndown Chart