Software Engineering Group Project

Group 36: Project Increment 2

Connor Calkin, Daniel Braghis, Benjamin Lewis, Lucas Sayers, Pingding He

1. DESIGN ARTIFACTS

1.1 Use Case Diagram

The use case diagram below is based on the Sprint 2 artifacts implemented.

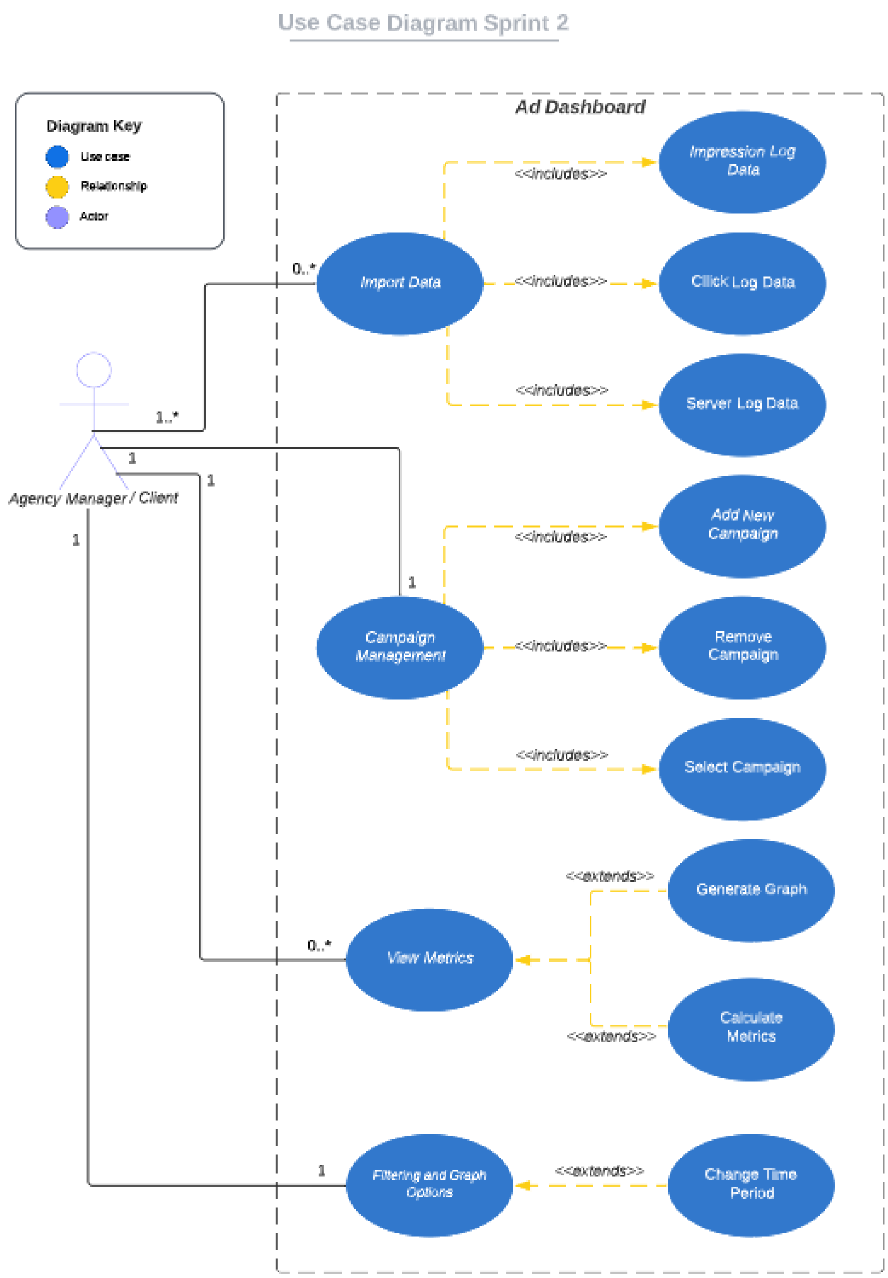


Figure 1.1

1.2 UML Class Diagram

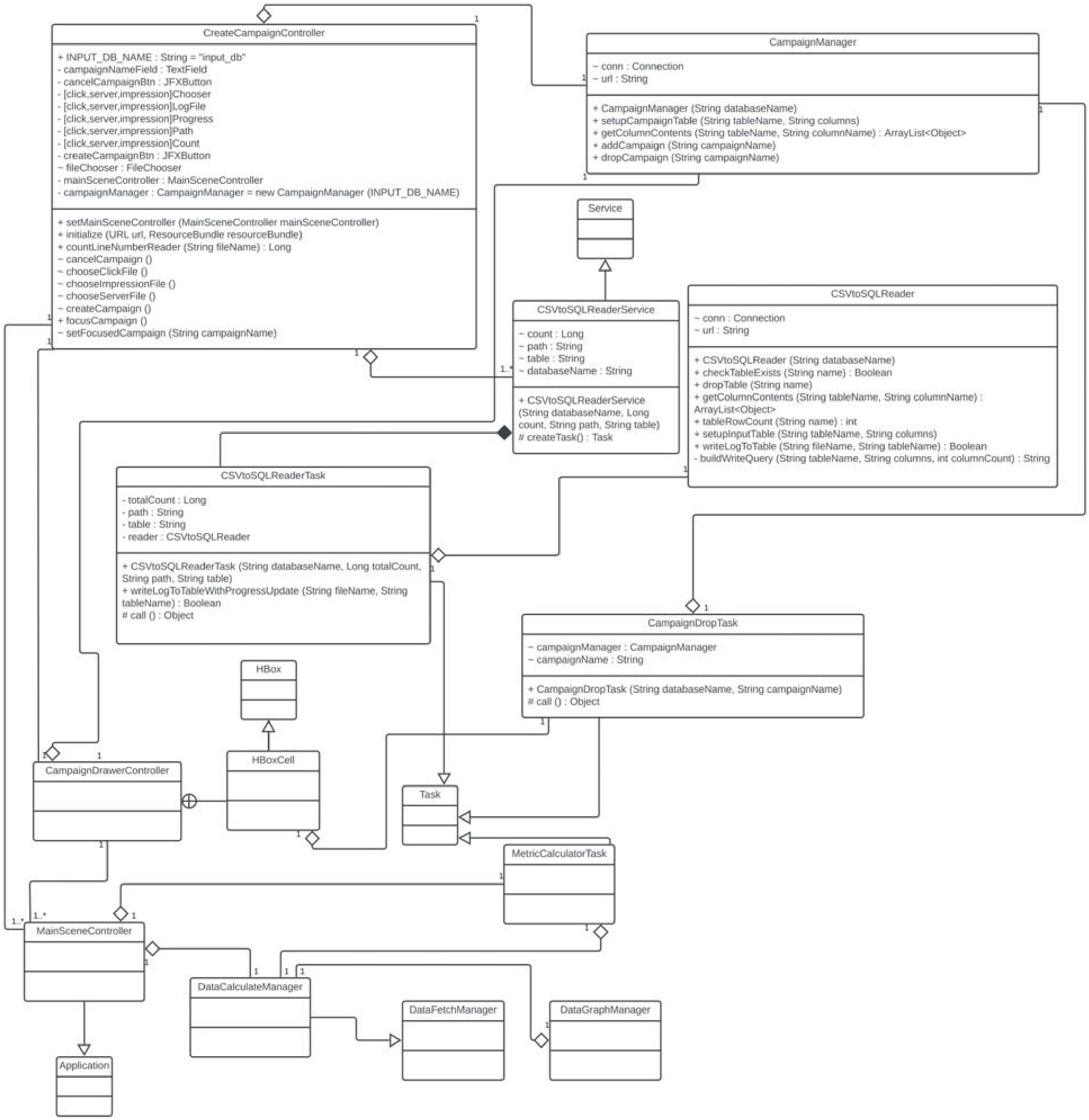
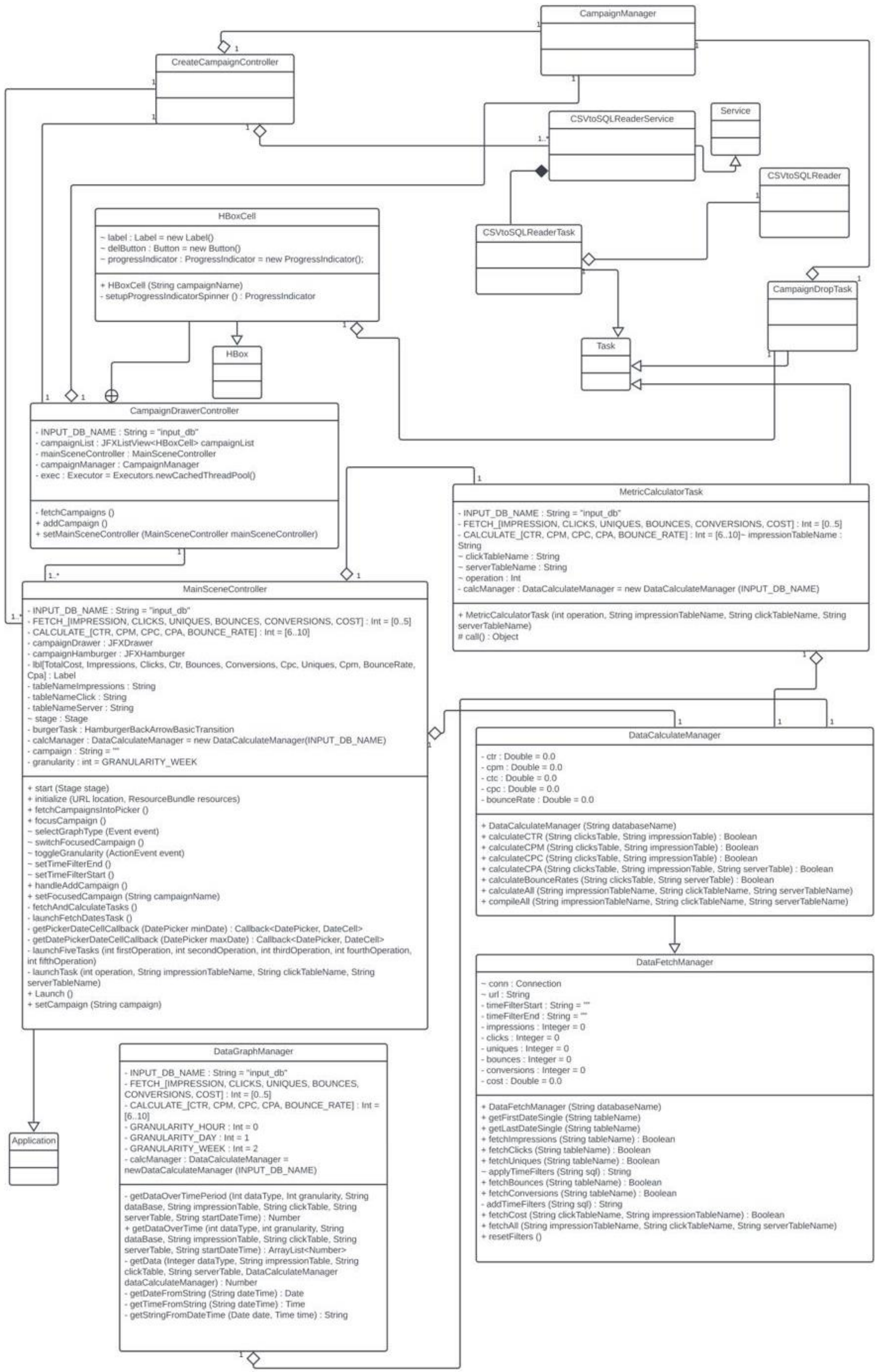


Figure 1.2 Campaign Manager

Figure 1.3 Data Manipulation (below)


CampaignManager
CreateCampaignController
CSVtoSQLReaderService
Service
CSVtoSQLReader
HBoxCell
~ label : Label = new Label()
~ delButton : Button = new Button()
~ progressIndicator : ProgressIndicator = new ProgressIndicator();
+ HBoxCell (String campaignName)
- setupProgressIndicatorSpinner () : ProgressIndicator
CSVtoSQLReaderTask
CampaignDropTask
HBox
Task
CampaignDrawerController
- INPUT_DB_NAME : String = "input_db"
- campaignList : JFXListView<HBoxCell> campaignList
- mainSceneController : MainSceneController
- campaignManager : CampaignManager
- exec : Executor = Executors.newCachedThreadPool()
- fetchCampaigns ()
+ addCampaign ()
+ setMainSceneController (MainSceneController mainSceneController)
MetricCalculatorTask
- INPUT_DB_NAME : String = "input_db"
- FETCH_[IMPRESSION, CLICKS, UNIQUES, BOUNCES, CONVERSIONS, COST] : Int = [0..5]
- CALCULATE_[CTR, CPM, CPC, CPA, BOUNCE_RATE] : Int = [6..10]~ impressionTableName : String
~ clickTableName : String
~ serverTableName : String
~ operation : Int
- calcManager : DataCalculateManager = new DataCalculateManager (INPUT_DB_NAME)
+ MetricCalculatorTask (int operation, String impressionTableName, String clickTableName, String serverTableName)
# call() : Object
MainSceneController
- INPUT_DB_NAME : String = "input_db"
- FETCH_[IMPRESSION, CLICKS, UNIQUES, BOUNCES, CONVERSIONS, COST] : Int = [0..5]
- CALCULATE_[CTR, CPM, CPC, CPA, BOUNCE_RATE] : Int = [6..10]
- campaignDrawer : JFXDrawer
- campaignHamburger : JFXHamburger
- lbl[TotalCost, Impressions, Clicks, Ctr, Bounces, Conversions, Cpc, Uniques, Cpm, BounceRate, Cpa] : Label
- tableNameImpressions : String
- tableNameClick : String
- tableNameServer : String
~ stage : Stage
- burgerTask : HamburgerBackArrowBasicTransition
- calcManager : DataCalculateManager = new DataCalculateManager(INPUT_DB_NAME)
- campaign : String = ""
- granularity : int = GRANULARITY_WEEK
+ start (Stage stage)
+ initialize (URL location, ResourceBundle resources)
+ fetchCampaignsIntoPicker ()
+ focusCampaign ()
~ selectGraphType (Event event)
~ switchFocusedCampaign ()
~ toggleGranularity (ActionEvent event)
~ setTimeFilterEnd ()
~ setTimeFilterStart ()
+ handleAddCampaign ()
+ setFocusedCampaign (String campaignName)
- fetchAndCalculateTasks ()
- launchFetchDatesTask ()
- getPickerDateCellCallback (DatePicker minDate) : Callback<DatePicker, DateCell>
- getDatePickerDateCellCallback (DatePicker maxDate) : Callback<DatePicker, DateCell>
- launchFiveTasks (int firstOperation, int secondOperation, int thirdOperation, int fourthOperation, int fifthOperation)
- launchTask (int operation, String impressionTableName, String clickTableName, String serverTableName)
+ Launch ()
+ setCampaign (String campaign)
DataCalculateManager
- ctr : Double = 0.0
- cpm : Double = 0.0
- ctc : Double = 0.0
- cpc : Double = 0.0
- bounceRate : Double = 0.0
+ DataCalculateManager (String databaseName)
+ calculateCTR (String clicksTable, String impressionTable) : Boolean
+ calculateCPM (String clicksTable, String impressionTable) : Boolean
+ calculateCPC (String clicksTable, String impressionTable) : Boolean
+ calculateCPA (String clicksTable, String impressionTable, String serverTable) : Boolean
+ calculateBounceRates (String clicksTable, String serverTable) : Boolean
+ calculateAll (String impressionTableName, String clickTableName, String serverTableName)
+ compileAll (String impressionTableName, String clickTableName, String serverTableName)
DataFetchManager
~ conn : Connection
~ url : String
- timeFilterStart : String = ""
- timeFilterEnd : String = ""
- impressions : Integer = 0
- clicks : Integer = 0
- uniques : Integer = 0
- bounces : Integer = 0
- conversions : Integer = 0
- cost : Double = 0.0
+ DataFetchManager (String databaseName)
+ getFirstDateSingle (String tableName)
+ getLastDateSingle (String tableName)
+ fetchImpressions (String tableName) : Boolean
+ fetchClicks (String tableName) : Boolean
+ fetchUniques (String tableName) : Boolean
~ applyTimeFilters (String sql) : String
+ fetchBounces (String tableName) : Boolean
+ fetchConversions (String tableName) : Boolean
- addTimeFilters (String sql) : String
+ fetchCost (String clickTableName, String impressionTableName) : Boolean
+ fetchAll (String impressionTableName, String clickTableName, String serverTableName)
+ resetFilters ()
DataGraphManager
- INPUT_DB_NAME : String = "input_db"
- FETCH_[IMPRESSION, CLICKS, UNIQUES, BOUNCES, CONVERSIONS, COST] : Int = [0..5]
- CALCULATE_[CTR, CPM, CPC, CPA, BOUNCE_RATE] : Int = [6..10]
- GRANULARITY_HOUR : Int = 0
- GRANULARITY_DAY : Int = 1
- GRANULARITY_WEEK : Int = 2
- calcManager : DataCalculateManager = newDataCalculateManager (INPUT_DB_NAME)
- getDataOverTimePeriod (Int dataType, Int granularity, String dataBase, String impressionTable, String clickTable, String serverTable, String startDateTime) : Number
+ getDataOverTime (int dataType, int granularity, String dataBase, String impressionTable, String clickTable, String serverTable, String startDateTime) : ArrayList<Number>
- getData (Integer dataType, String impressionTable, String clickTable, String serverTable, DataCalculateManager dataCalculateManager) : Number
- getDateFromString (String dateTime) : Date
- getTimeFromString (String dateTime) : Time
- getStringFromDateTime (Date date, Time time) : String
Application

1.3 Entity Relationship Diagram

This sprint has seen the addition of the campaigns table to the schema that stores the user given campaign name and the tables in the database associated with it. This is done as a secondary way of finding the data in addition to a simple naming scheme that lets us retrieve the data by only knowing the campaign name. This is done by lower casing the input and replacing the spaces with the "_" symbol, then appending the corresponding postfix "_impression_log", "_click_log", "_server_log". Such structure makes creating, finding, and renaming the campaigns a trivial matter.

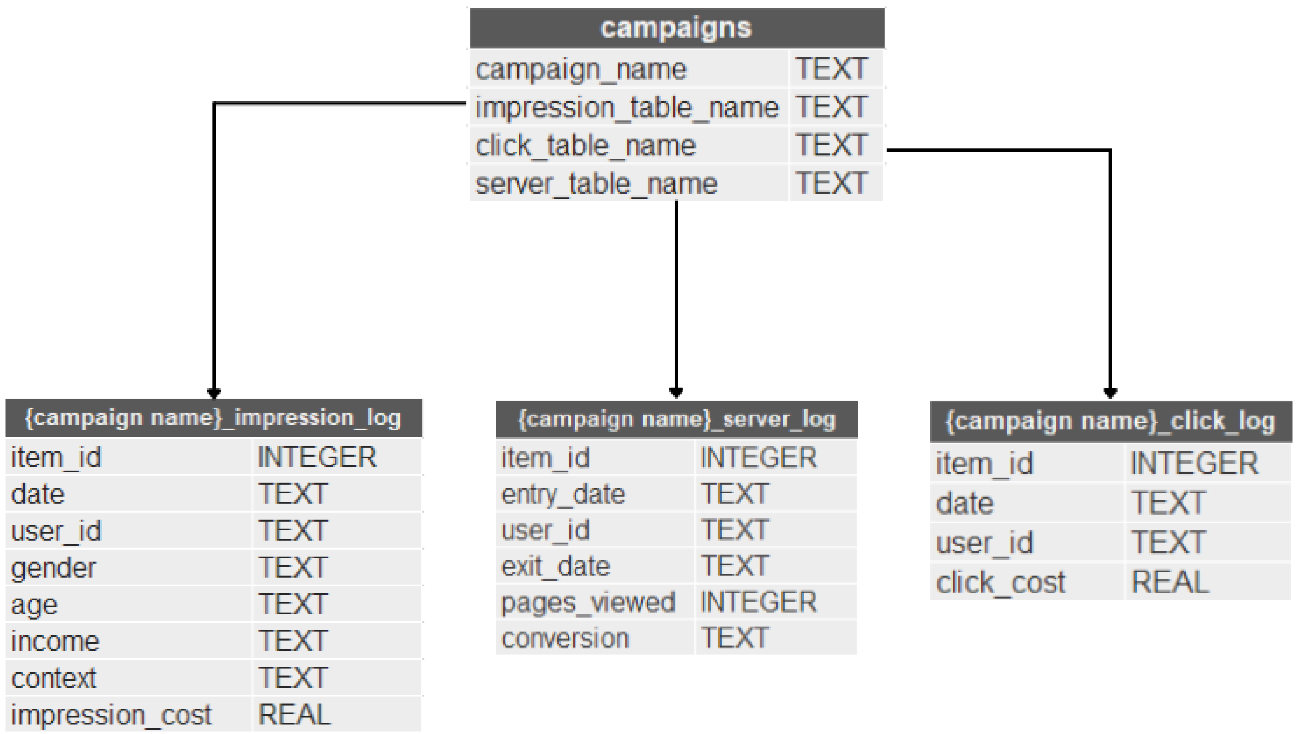


Figure 1.4

1.4 Storyboarding

1. When users first open the application, the first task they will wish to complete is creating a new campaign. They click the hamburger icon in the top right corner which opens a side menu.

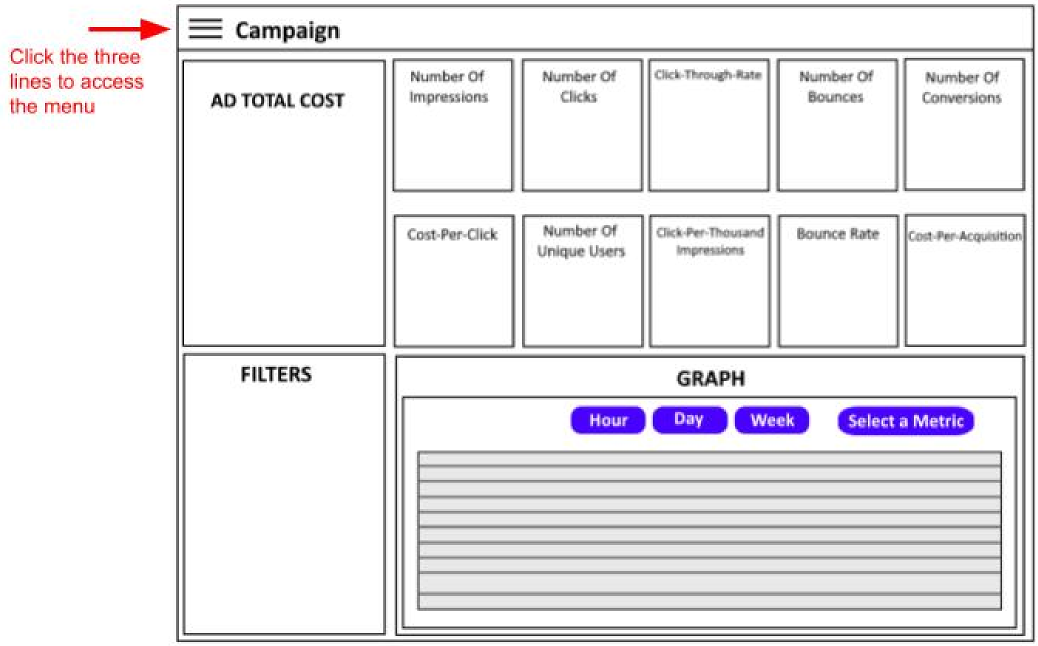


Figure 1.5

2. The side menu will be blank initially other than an "add campaign" button. Click this button to open a new window to add a campaign.

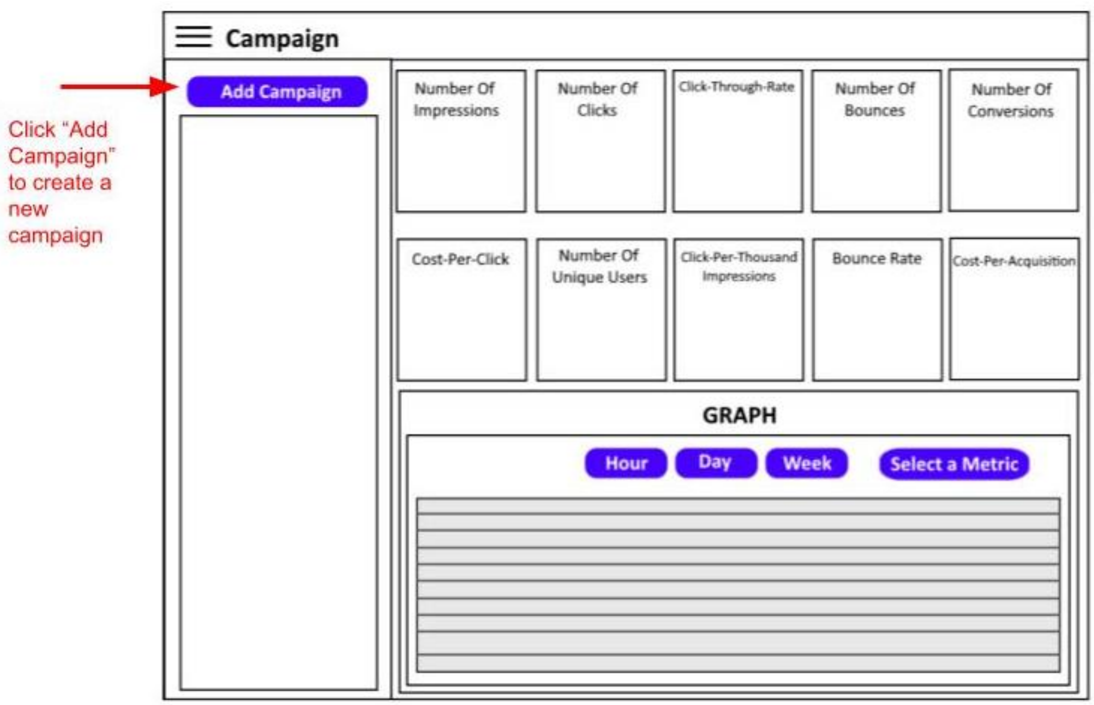


Figure 1.6

3. A new window will open, allowing the user to create a new campaign. In the top bar they type whatever they would like the campaign name to be. Then for each log source they click "Select File" and upload the

relevant file. Once all files are uploaded, they can click create campaign. The progress bars will start loading one by one until all three are completed and the window and side menu will automatically close.

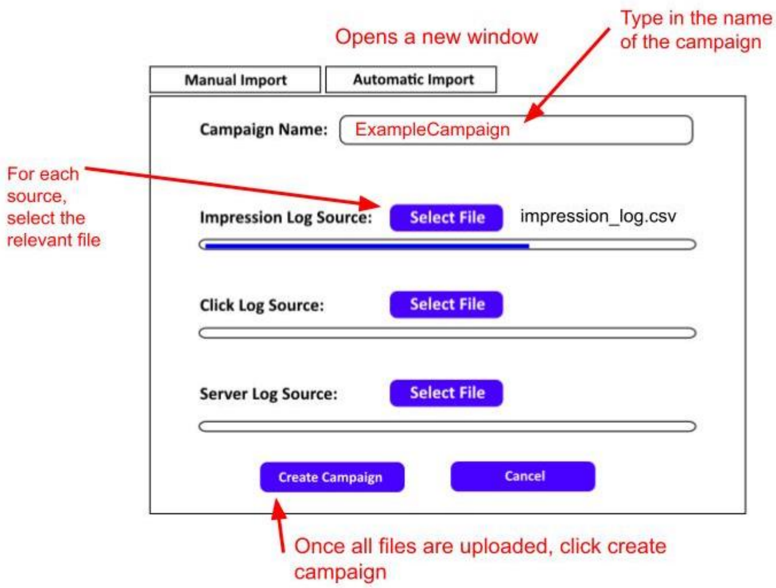


Figure 1.7

4. The metrics will then be displayed along with a graph. The graph can be changed to show different metrics over different time periods. Repeat this process to add more campaigns. If you decide you no longer need a certain campaign, click the red button next to the campaign name to delete it. Close the left menu by clicking the icon in the top left corner.

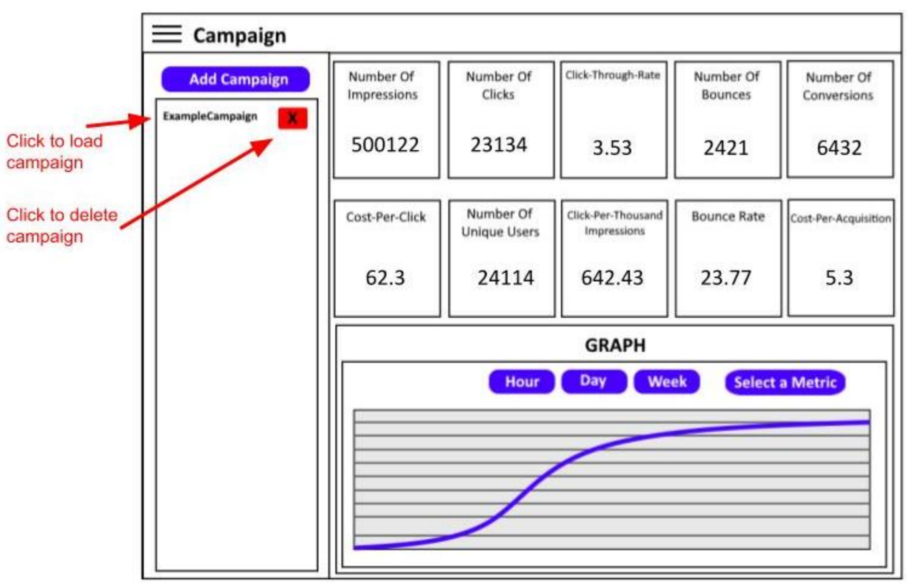


Figure 1.8

2. SCENARIOS AND TESTING

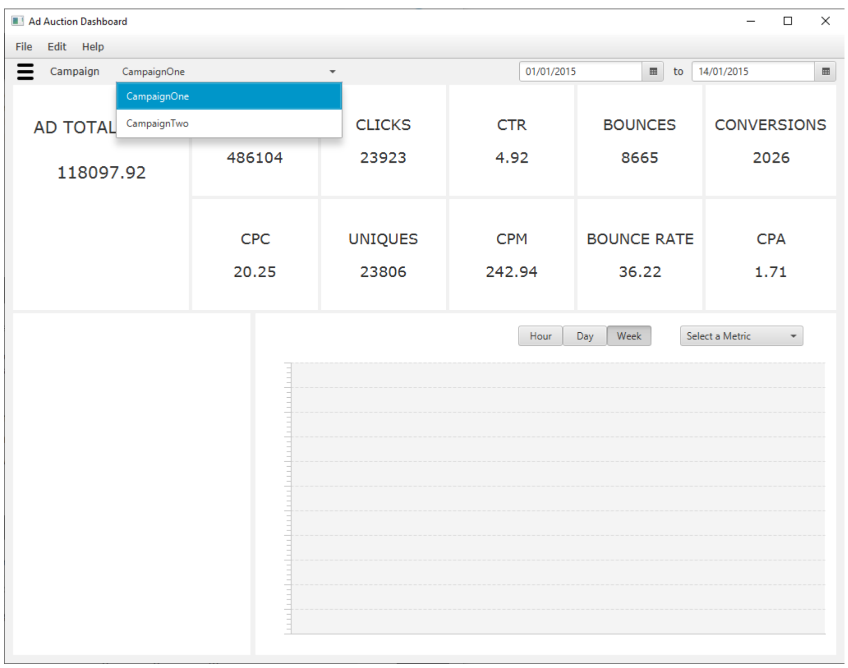
2.1 Scenarios and test results

Alice, 48 - A Senior Finance Director

Alice struggles with adequately budgeting for the current and planned ad campaigns. She wants to be able to get live data on the cost of the campaigns so she can report to the ad campaign clients and recommend ajustments to budget and target audiece background. There are various metrics needed to make an appropriate asessment on campaign performance that should be included in the analysis

Goals	Frustrations
Visualize campaign costs	Seeing performance for all client campaigns
Report on campaign costs	Metrics can vary day to day widely
Get key metrics like CPA, CPC etc.	Spotting under-performing campaign before it ends

Alice opens the software. She clicks the icon in the top-left which opens the side menu. From there Alice uploads one campaign's data. The metrics are displayed on screen. Once that campaign has been created, she repeats the process with a second set of campaign data. Alice can click the drop down at the top of the software to select between the two campaigns and compare. All the metrics are displayed however the graph does not function.



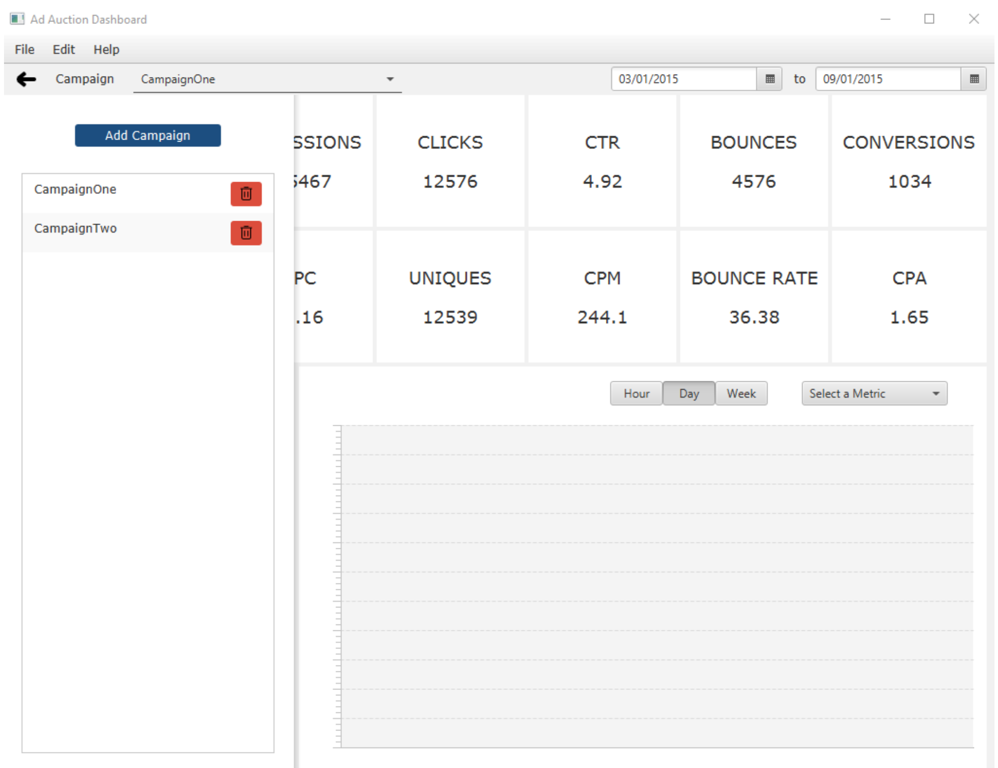
Alice appreciates being able to quickly analyse the data to see the different metrics. She values being able to compare the different campaigns by selecting each one. Alice appreciates being able to change the time period at the top of the screen as a simple way to look at how metrics change over time. Alice would like to be able to see the data graphically and would prefer a more automatic method to see how the metrics change over time, so she can fix underperforming campaigns at the earliest opportunity.

Philip, 41 - Client Company CEO

Philip has been in the marketing industy for 15 years in which he worked in various positions. He wants to see the data gathered by a campaign. He also wishes to be able to compare it to other campaigns that could and have been run. He wants the functionalities to be understandable even to people who don't have a strong marketing background, like his clients. This way he hopes to build trust with the clients.

Goals	Frustrations
See data on key metrics	Understanding marketing data
Determine successful campaigns	Comparing campaign performance
Improve client relations	Explaining to clients underperforming campaigns

Philip opens the software. He clicks the icon in the top-left which opens the side menu. From there Philip uploads one campaign's data. All the metrics for that campaign load. Once that campaign has been created, he repeats the process with a second set of campaign data. Philip can click the drop down at the top of the software to select between the two campaigns and compare. All the metrics are displayed however the graph does not function.



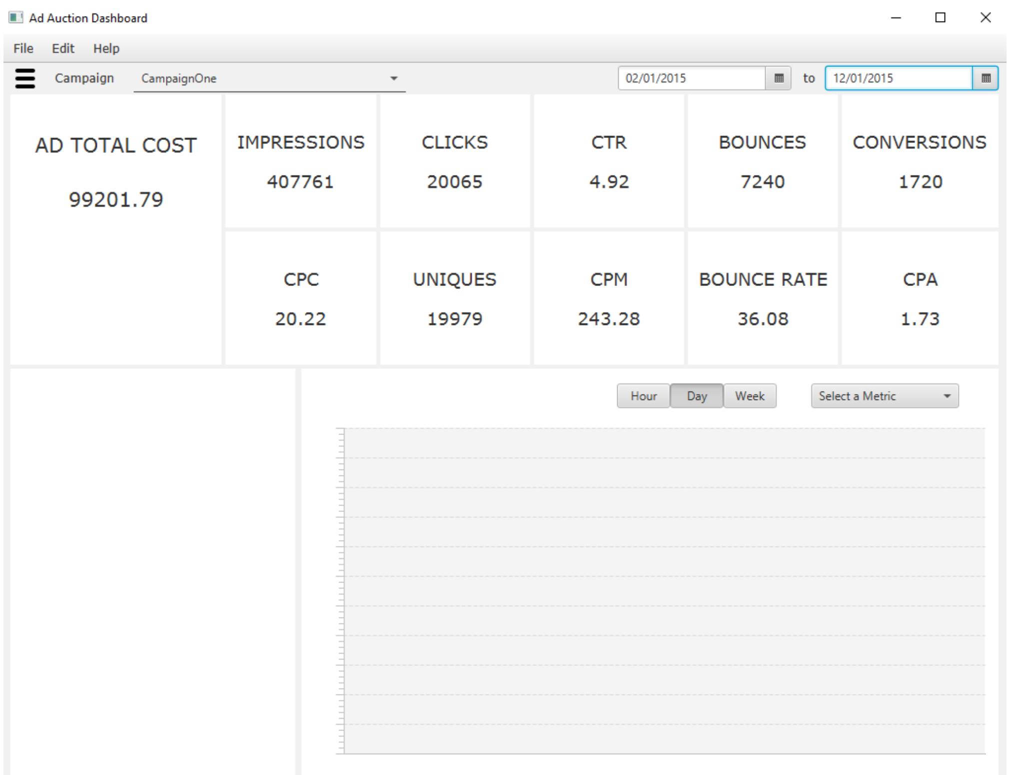
Philip values being able to upload multiple campaigns at once so that he can compare them by switching in between them. Philip appreciates the simplicity of the UI with clear labels, meaning he can show it to clients and they will understand what it means. The software includes the key metrics he is interested in. Philip would prefer if the software had a clearer way to show if a campaign was successful or not, perhaps through the graph.

Deborah, 39 - DIY Online Shop Owner

Deborah is owner of an online shop selling DIY producs. She wants to run ad campaigns for her shop but doesn't have the expertise for it so she contracted the client agency to help her. She would like to better understand the performance of the ad campaigns on different planforms. That way she could find out which platforms to prioritize posting to. She doesn't understand marketing jargon so tool tips explaining terminology would be great.

Goals	Frustrations
Find the channel to grow	Understanding marketing terminology
Identifying potetial new clients	Identifying channels driving business growth
Making campaigns more cost effective	Working with advanced marketing dashboards

Deborah opens the software. She clicks the icon in the top-left which opens the side menu. From there Deborah uploads one campaign's data. Deborah can then read the metrics.



Deborah values the clear language and labels so that she can understand which values match to each label. She would like there to be some documentation to explain what each value means in the context of marketing. Deborah appreciates the simplicity of the dashboard and that it is very intuitive. She wishes that the dashboard showed her more data about who is seeing the campaigns and who interacts so she can identify new clients.

Real Scenarios:

Import CSVs using the 2-Month Campaign Data provided.

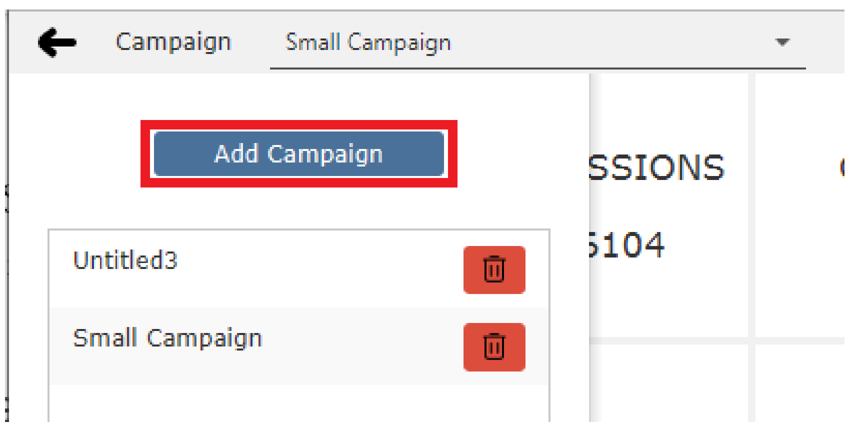
I open the Ad Auction dashboard and click the button with three lines. A sidebar opens and I click “Add Campaign”. This button is self-explanatory and easy to understand.

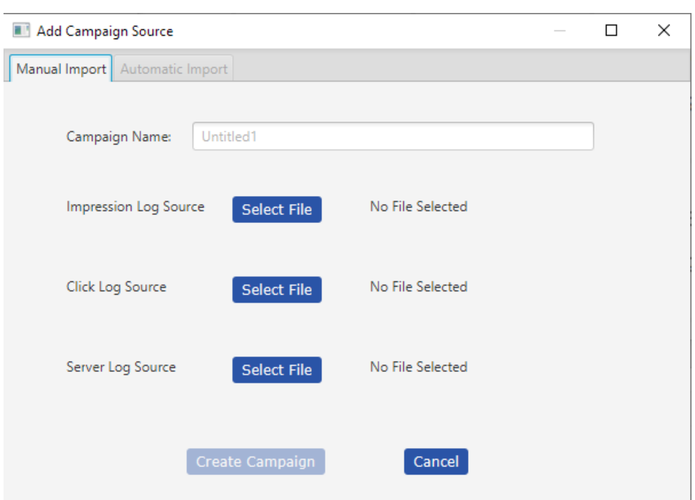
A new window opens that allows me to input the three files as part of the campaign, each one clearly labelled including telling me if I haven’t yet selected a file.

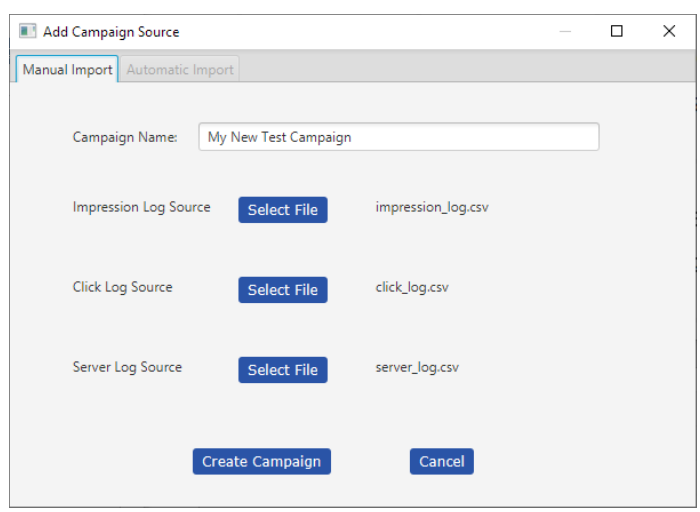
I can also add a name to my Campaign that allows me to easily identify it.

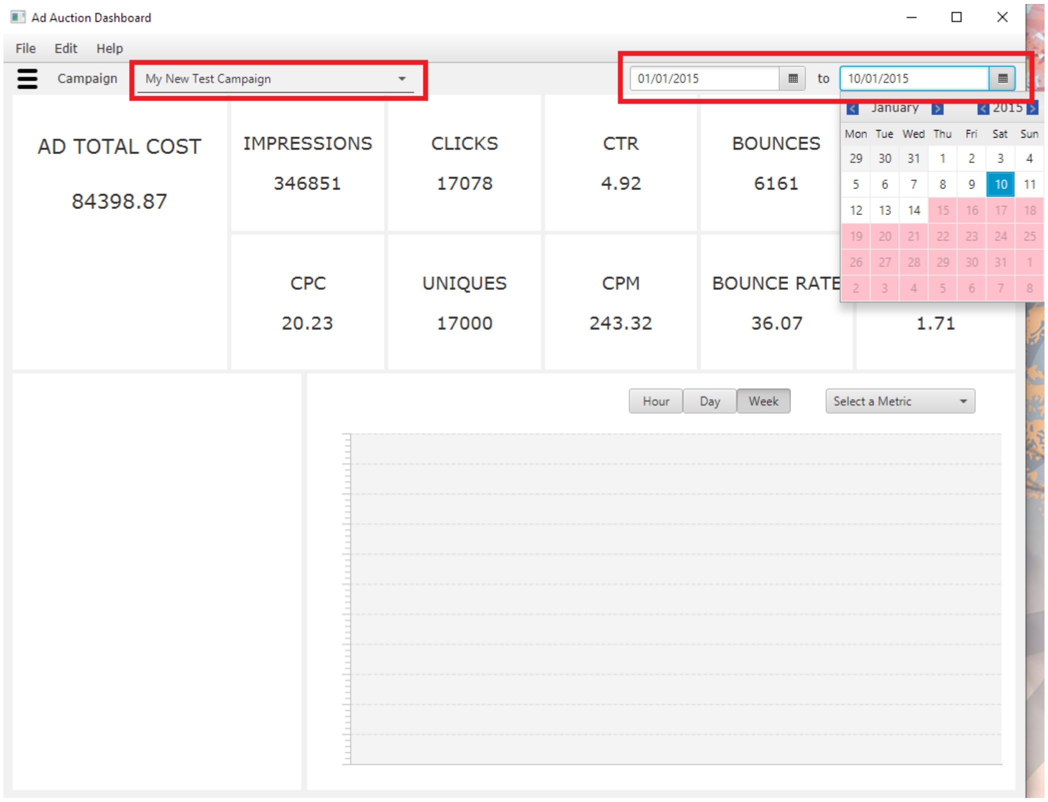
Once I have imported the files I can click “Create Campaign”. After the data is loaded, shown by the progress bars under each campaign. My campaign is now the active one and all the metrics appear at the top of the screen correctly.

At the top of the screen the date pickers are now active, and I can change the period the data is displayed for.









Switching between campaigns using the 2-Week Campaign Data provided:

I open the Ad Auction dashboard and click the button with three lines. A sidebar opens and I click "Add Campaign". This button is self-explanatory and easy to understand.

A new window opens that allows me to input the three files as part of the campaign, each one clearly labelled including telling me if I haven't yet selected a file.

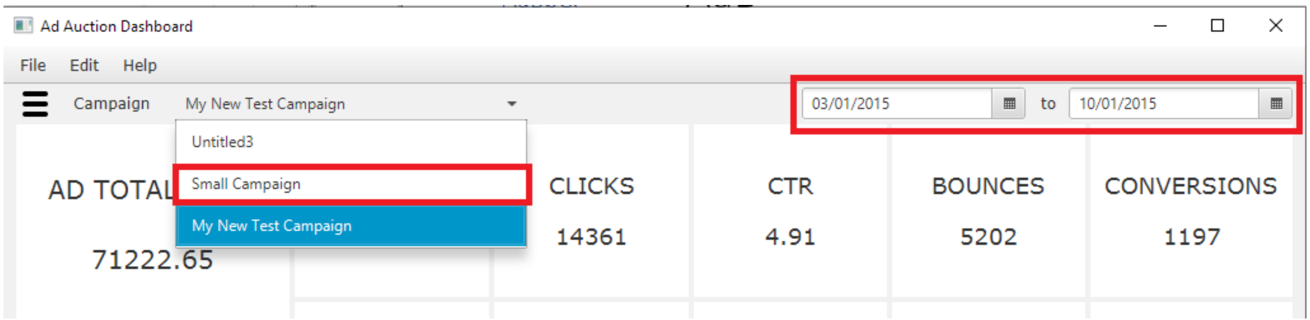
I can also add a name to my Campaign that allows me to easily identify it.

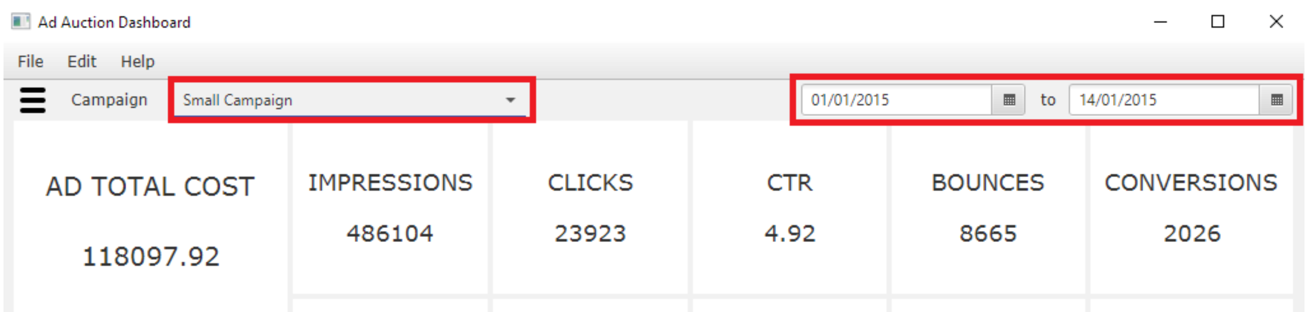
Once I have imported the files I can click "Create Campaign". After the data is loaded, shown by the progress bars under each campaign. My campaign is now the active one and all the metrics appear at the top of the screen correctly.

I repeat this process again with a new Campaign.

Once the second Campaign is created, I click on the dropdown that displays the campaign names. I click on the first campaign and after a few seconds of buffering, all the metrics appear at the top of the screen correctly.

The date pickers at the top have reset to the switched campaign's period.

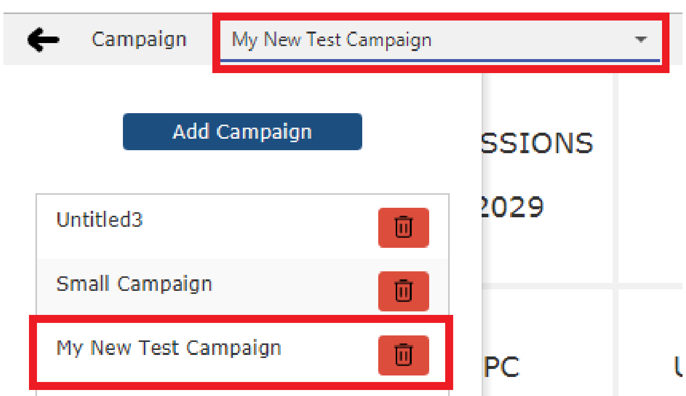


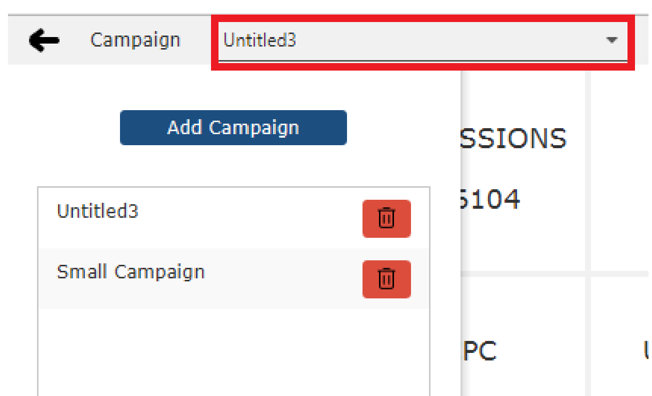


Deleting campaigns:

I open the Ad Auction dashboard and click the hamburger button. A sidebar opens displaying my campaigns. I click on the red delete button next to my campaign and after a few seconds my campaign is deleted.

If the campaign deleted is the active one, the dashboard loads the first campaign from the list of campaigns.





3. SPRINT RETROSPECTIVE AND SPRINT 2 PLANNING

3.1 Sprint 2 Retrospective

The goal of this sprint was to mainly improve the UX of the application. While there are no big changes to the backend of the application besides 14.1 and 2 (**Managing multiple campaigns**) added to the sprint as we found it a very important feature even if not specifically pointed out by the stakeholders, we made significant progress in the overall usability of the application. We now reliably import and manage large datasets with real time feedback on progress 10 (**Large dataset handling**). We also made significant progress on the UI 18 (**Intuitive UI prototyping**) and now have a clear idea of how the dashboard will look in the final product. Graphing the data was almost completed with a lot of prerequisites being completed this sprint, but the feature didn't pass testing and wasn't offering a good quality user experience, so we decided to postpone its release to the next sprint. The remaining secondary tasks were moved to the next sprint.

We are satisfied with the overall performance this sprint. The remaining "musts" in this sprint are almost completed and are coming in the next release.

3.2 Sprint 2 Burndown Chart

ID	Assignee Name	Time (h)
01.2	Pingding He	2
01.5	Lucas Sayers	3
02.2	Connor Calkin	3
02.5	Pingding He	3
07.2	Lucas Sayers	6
10.1	Lucas Sayers	3
10.2	Benjamin Lewis	3
10.3	Daniel Braghis	9
14.1	Lucas Sayers	5
14.2	Daniel Braghis	7
18.2	Connor Calkin	6
18.3	Daniel Braghis	6
18.4	Connor Calkin	3
23.1	Lucas Sayers	3

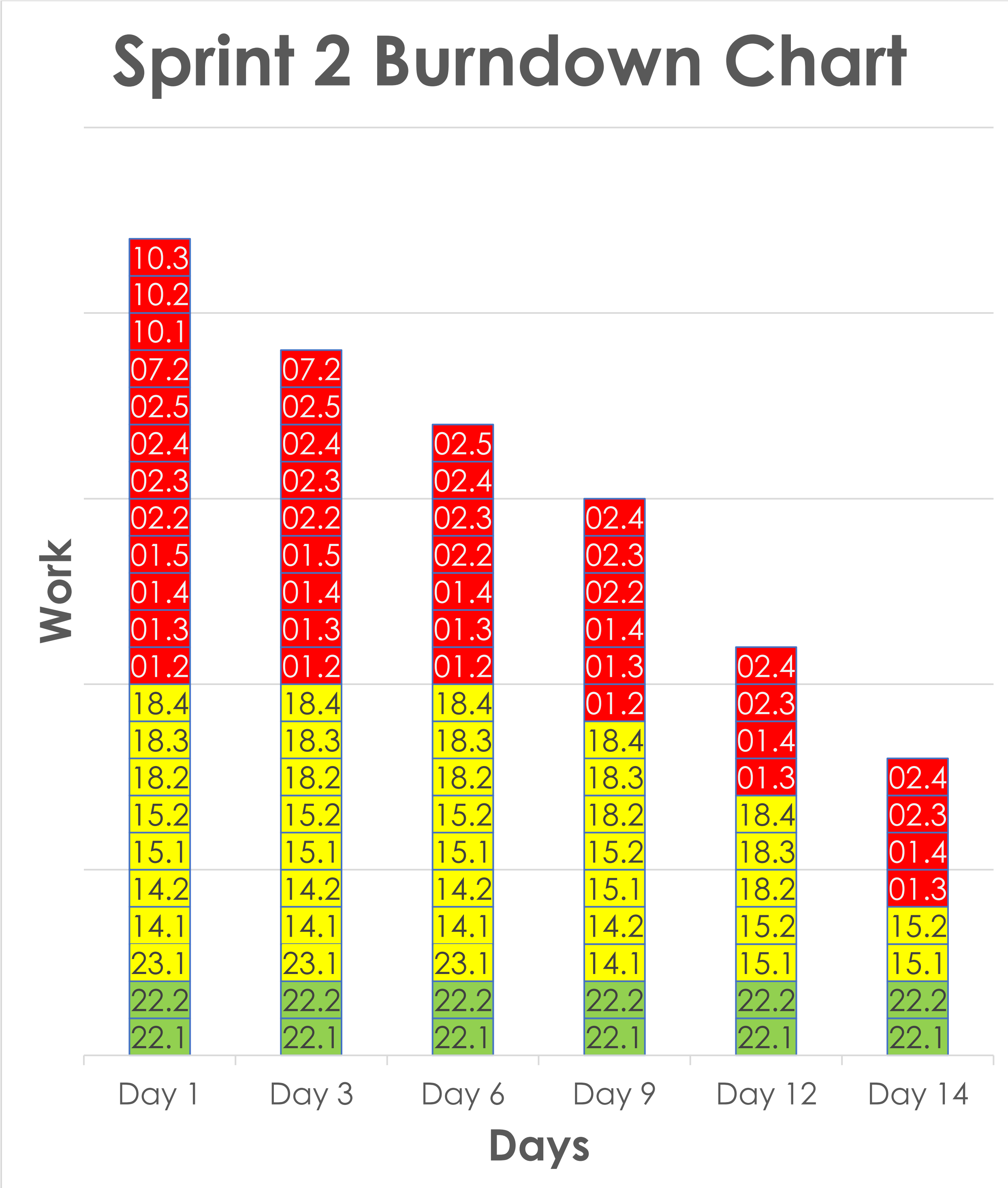

Sprint 2 Burndown Chart
Work
Days
Day 1
Day 3
Day 6
Day 9
Day 12
Day 14
10.3
10.2
10.1
07.2
02.5
02.4
02.3
02.2
01.5
01.4
01.3
01.2
18.4
18.3
18.2
15.2
15.1
14.2
14.1
23.1
22.2
22.1

Figure 3.1

3.3 Sprint 3 Backlog

Value	Meaning (or equivalent)
0	No value producing work involved, only research or planning
1	Only a few lines of code needed or minimal work (about 1h)
2	There are significant changes to a single file (up to 3h)
3	There are significant changes to several files (up to 3h)
5	A new independent component is added with minimal changes (up to 5h)
8	A new independent component is added with significant changes to other components (up to 10h)
13	A new component that needs significant changes/redesigns to other components (up to 20h)
21	A significant system redesign involving core components being rewritten (should be avoided)

SPRINT 3				
Backlog ID & Description	**ID**	**Task Description**	**Size**	**Dependencies**
01 - Number of Impressions, Clicks, Uniques, Bounces	01.3	Graph data	3	-
	01.4	Graphing options	3	01.3
02 - CTR, CPA, CPC, CPM, Bounce Rate	01.3	Graph data	3	-
	01.4	Graphing options	3	01.3
04 - Cost per click distribution	04.1	Calculate data points	2	-
	04.2	Graph data	3	04.1
	04.3	Testing	2	04.1
05 - Gender, age, time frame and context filtering	05.1	Modify data fetching and calculations	3	
	05.2	Add filters to the UI	2	05.1
	05.3	Testing	1	05.1
06 - Bounce redefinition option	06.1	Modify data fetching	3	
	06.2	Testing	2	06.1
14 - Multiple campaign analysis	14.3	Creating "snapshots" of data with modifiers	8	01.3/02.3/04.2
	14.4	Design GUI component	5	14.3
	14.5	Testing	2	14.3
15 - Saving report to various file formats	15.1	Data preprocessing	5	-
	15.2	Generate report in PDF format	3	15.1/01.3/02.3/04.2
16 - Application user manual	16.1	Write manual contents	5	-

	16.2	Integrate manual into application	5	**16.1**
19 - Multiple campaign data comparison	19.1	Integrate with "snapshots"	3	**14.3**
	19.2	Integrate with GUI component	3	**14.4**
20 - Multiple campaign data comparison	20.1	Create style sheet and link in GUI	3	-
21 - Metric and functionality hints	21.1	**Design hint icon and tooltip**	2	-
	21.2	Integrate with the rest of UI	3	**21.1**
		SIZE ESTIMATE TOTAL	**77**	

3.3 Next Sprint

In the next sprint our priority is to make the graphs work as we hoped they would be fully functional this sprint. We also want to add the filters to the UI so that you can filter the metrics based on certain characteristics.

We also aim to create all the user documentation during the next sprint. While we made the UI as intuitive as possible, this documentation will help the user better understand how to use the product and to highlight its different features.

As this is the last sprint and we are on schedule to finish the “musts” of our stakeholders, we can now start looking at implementing more of the “should do” tasks on the increment plan and some of the “could do” task. This may include a dark mode design option and adding the ability to compare campaigns. We will however firstly prioritize making sure all the “must do” tasks are completed first and have been fully tested.