Software Engineering Group Project

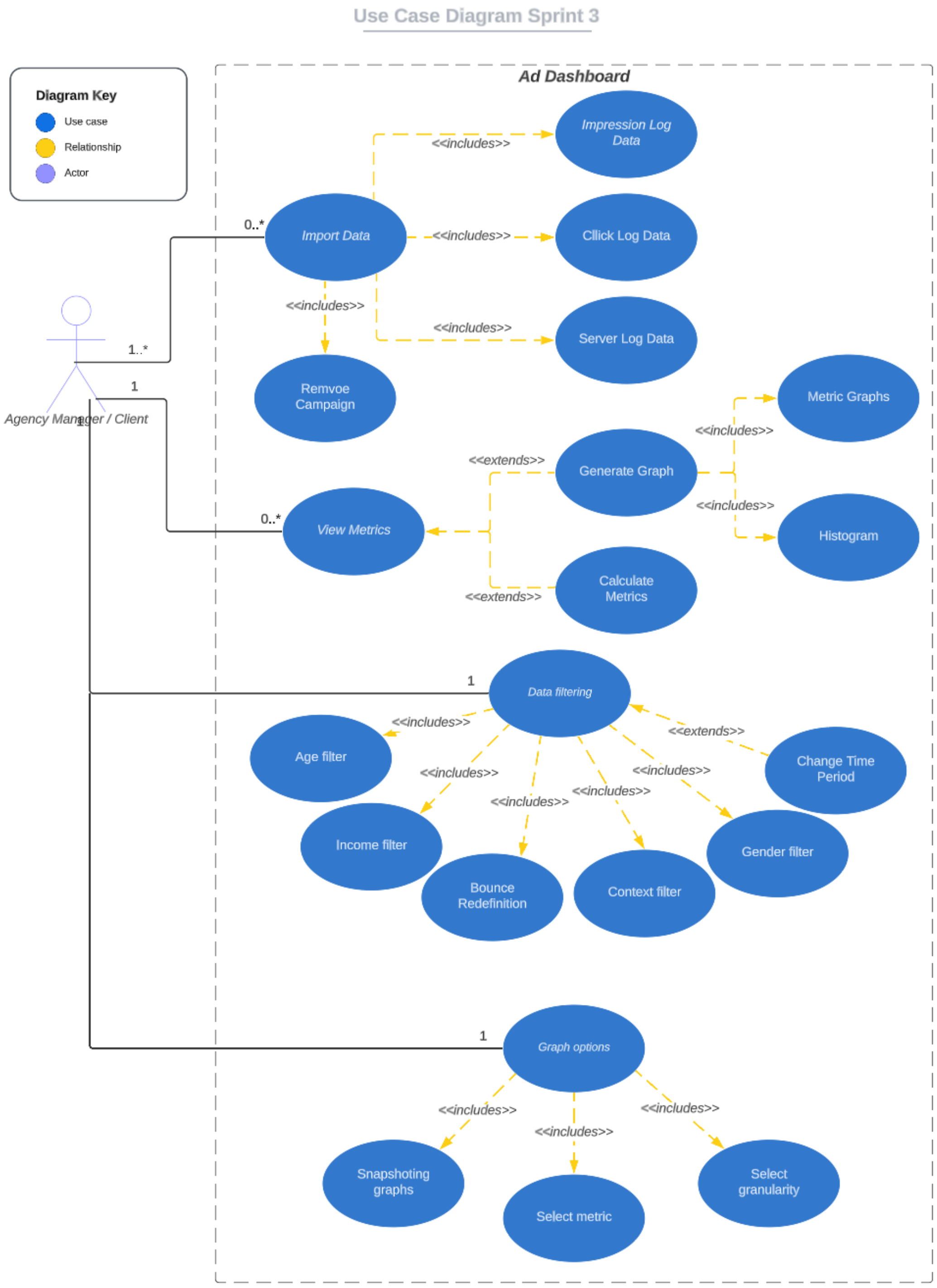
Group 36: Project Increment 2

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

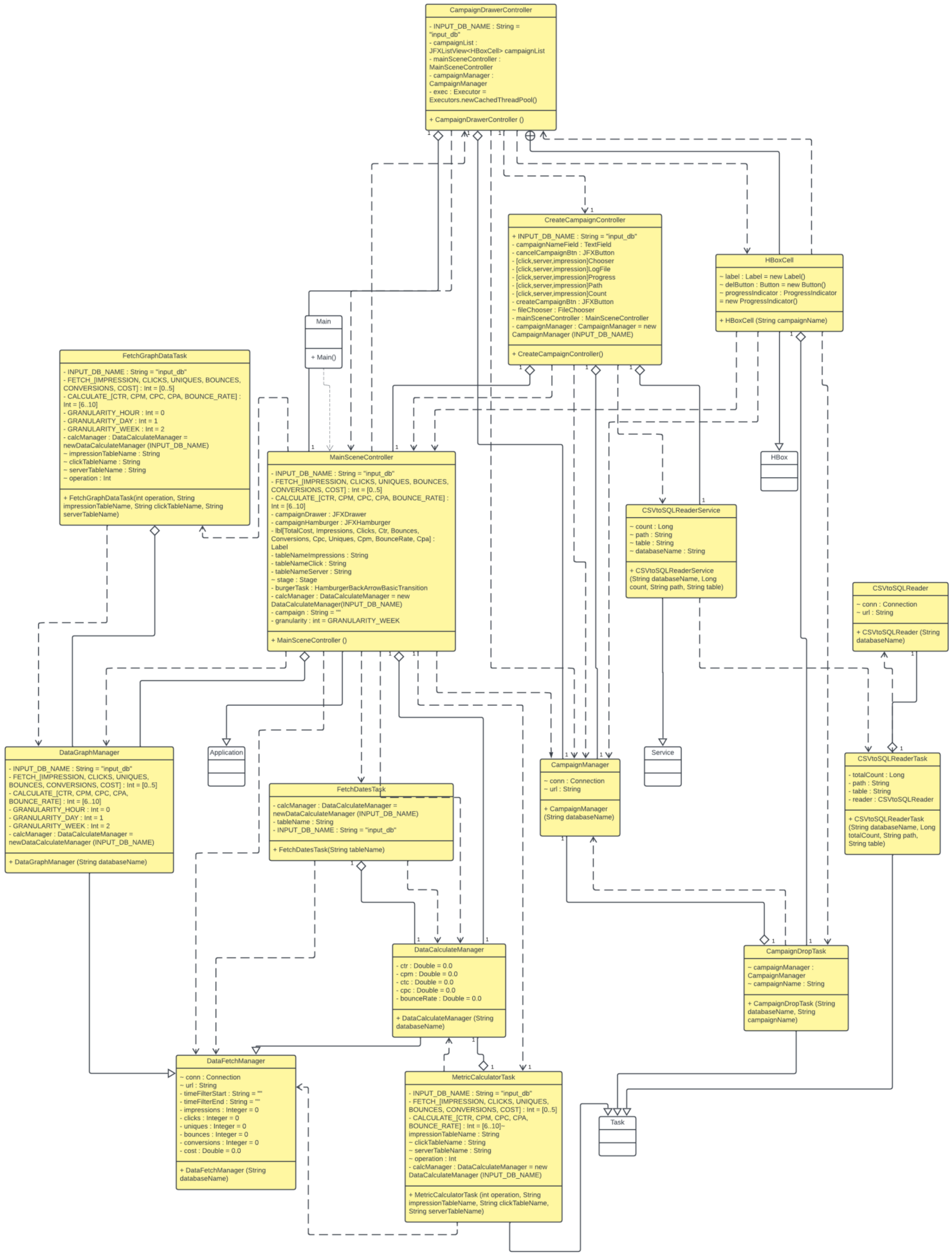
1. DESIGN ARTIFACTS

1.1 Use Case Diagram

The use case diagram below is based on the Sprint 3 artifacts implemented.



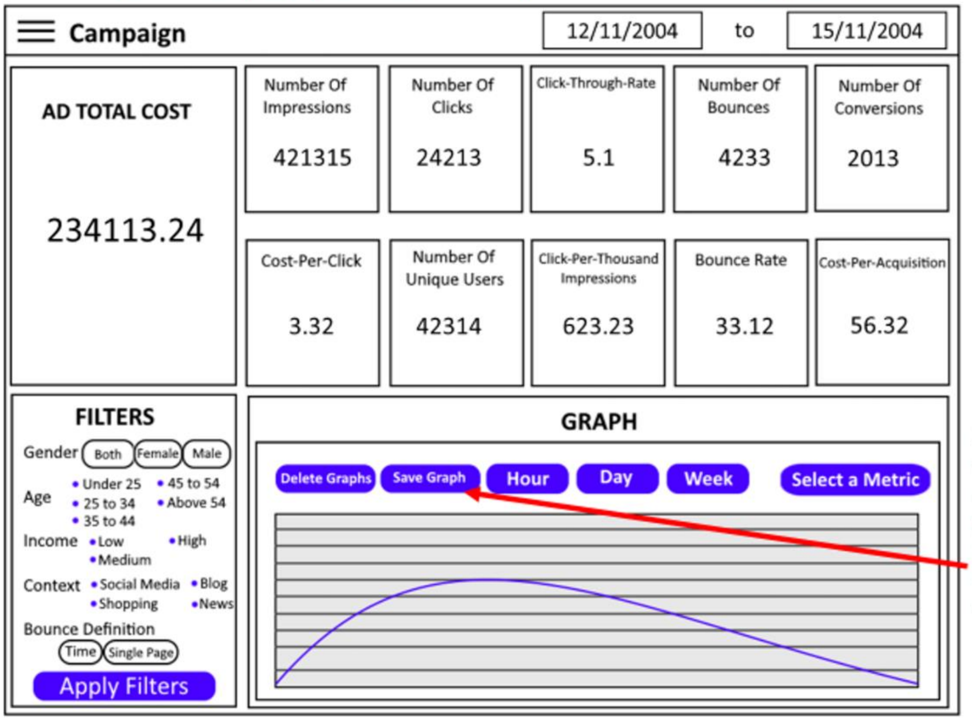
1.2 UML Class Diagram


CampaignDrawerController
- INPUT_DB_NAME : String = "input_db"
- campaignList : JFXListView<HBoxCell> campaignList
- mainSceneController : MainSceneController
- campaignManager : CampaignManager
- exec : Executor = Executors.newCachedThreadPool()
+ CampaignDrawerController ()
CreateCampaignController
+ INPUT_DB_NAME : String = "input_db"
- campaignNameField : TextField
- cancelCampaignBtn : JFXButton
- [click,server,impression]Chooser
- [click,server,impression]LogFile
- [click,server,impression]Progress
- [click,server,impression]Path
- [click,server,impression]Count
- createCampaignBtn : JFXButton
~ fileChooser : FileChooser
- mainSceneController : MainSceneController
- campaignManager : CampaignManager = new CampaignManager (INPUT_DB_NAME)
+ CreateCampaignController()
HBoxCell
~ label : Label = new Label()
~ delButton : Button = new Button()
~ progressIndicator : ProgressIndicator = new ProgressIndicator()
+ HBoxCell (String campaignName)
Main
+ Main()
FetchGraphDataTask
- INPUT_DB_NAME : String = "input_db"
- FETCH_[IMPRESSION, CLICKS, UNIQUES, BOUNCE, 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)
~ impressionTableName : String
~ clickTableName : String
~ serverTableName : String
~ operation : Int
+ FetchGraphDataTask(int operation, String impressionTableName, String clickTableName, String serverTableName)
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
+ MainSceneController ()
HBox
CSVtoSQLReaderService
~ count : Long
~ path : String
~ table : String
~ databaseName : String
+ CSVtoSQLReaderService (String databaseName, Long count, String path, String table)
CSVtoSQLReader
~ conn : Connection
~ url : String
+ CSVtoSQLReader (String databaseName)
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)
+ DataGraphManager (String databaseName)
Application
CampaignManager
~ conn : Connection
~ url : String
+ CampaignManager (String databaseName)
Service
CSVtoSQLReaderTask
- totalCount : Long
- path : String
- table : String
- reader : CSVtoSQLReader
+ CSVtoSQLReaderTask (String databaseName, Long totalCount, String path, String table)
FetchDatesTask
- calcManager : DataCalculateManager = newDataCalculateManager (INPUT_DB_NAME)
- tableName : String
- INPUT_DB_NAME : String = "input_db"
+ FetchDatesTask(String tableName)
DataCalculateManager
- ctr : Double = 0.0
- cpm : Double = 0.0
- ctc : Double = 0.0
- cpc : Double = 0.0
- bounceRate : Double = 0.0
+ DataCalculateManager (String databaseName)
CampaignDropTask
~ campaignManager : CampaignManager
~ campaignName : String
+ CampaignDropTask (String databaseName, String campaignName)
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)
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)
Task
1

1.3 Storyboarding

1. After a user has uploaded the initial campaign and the data has been processed, you can save the graph and then edit the filters to see different sets of data on the graphs.

1. Upload a campaign (same as previous sprint so omitted from storyboard)



12/11/2004 to 15/11/2004

AD TOTAL COST 234113.24

Number Of Impressions 421315

Number Of Clicks 24213

Click-Through-Rate 5.1

Number Of Bounces 4233

Number Of Conversions 2013

Cost-Per-Click 3.32

Number Of Unique Users 42314

Click-Per-Thousand Impressions 623.23

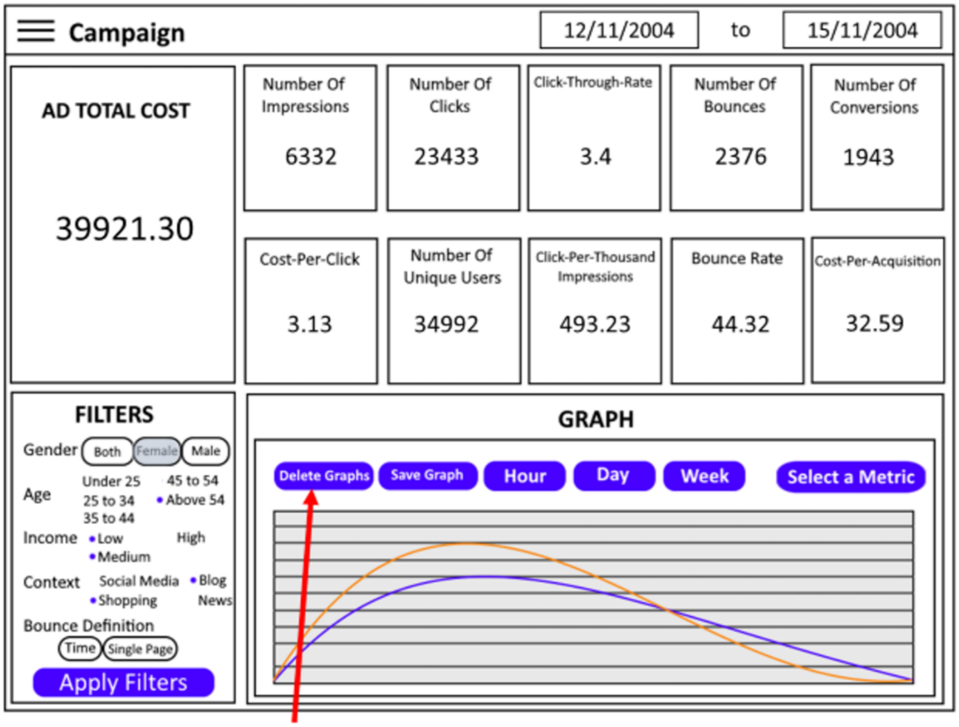
Bounce Rate 33.12

Cost-Per-Acquisition 56.32

2. If you wish to compare multiple graphs, click save graph to save the current graph

3. Set the filters to as you wish and then click apply filters

2. Once you have changed the filter, you can see a new graph on top of the previous one helping you to compare data. You can repeat the process to compare more graphs. Once complete, click Delete Graphs to refresh.



12/11/2004 to 15/11/2004

AD TOTAL COST 39921.30

Number Of Impressions 6332

Number Of Clicks 23433

Click-Through-Rate 3.4

Number Of Bounces 2376

Number Of Conversions 1943

Cost-Per-Click 3.13

Number Of Unique Users 34992

Click-Per-Thousand Impressions 493.23

Bounce Rate 44.32

Cost-Per-Acquisition 32.59

4. The new graph will appear on top of the previous one. Axis will automatically adjust

5. Either repeat the steps 2-4, or once you have finished comparing different graphs click "Delete Graphs" to restart

2. SCENARIOS AND TESTING

To keep the application stable and bug-free throughout the development cycle, we used the following testing procedure:

When a new feature is added:

- Unit tests are performed to ensure feature is stable in isolation.
- Regression testing is performed to ensure old features were unaffected.
- Boundary tests are preformed to detect unusual behaviour.
- If a bug is identified, bug is removed, and procedure is repeated.

When old functionality is modified:

- Validation tests are performed to ensure changes haven't introduces new bugs.
- Unit tests are performed to ensure feature is still stable in isolation.
- If a bug is identified, bug is removed, and procedure is repeated.

2.1 Scenarios testing

Alice, 48 - A Senior Finance Director

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

Total campaign costs and metrics over time
Scenario 1

As a **<Senior Finance Director>** I want **<to know the distribution of costs per click>** so that **<I can budget appropriately campaigns in the future>**

As a **<Senior Finance Director>** I want **<to know the total campaign cost>** so that **<I can asses campaign performance relative to investment>**

Test against the scenario

Type of test: Manual

Preconditions:

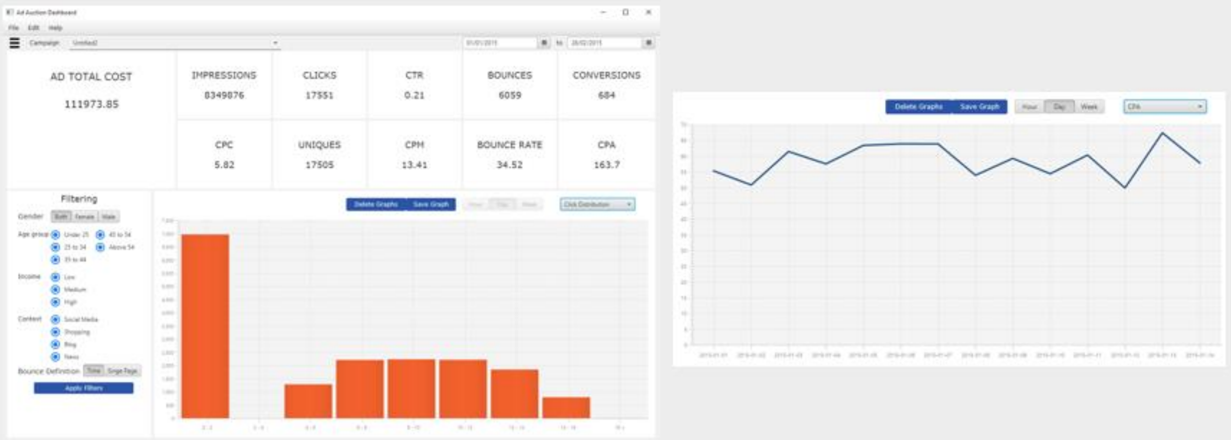
- Campaign is loaded in the application

Actions:

- Loaded application
- Selected campaign from dropdown
- Selected desired time granularity
- Selected desired metric to view
- Selected 'click distribution' metric

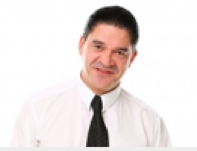
Application opens successfully
Dropdown correctly displays all available campaigns
Selected campaign loads correctly
Graph is loaded correctly
Histogram is displayed correctly
Appropriate buttons are disabled when selecting the histogram
No major delays in display time
Buttons are correctly made available once the data is loaded
Data is in correct format

Test screenshot



AD TOTAL COST	IMPRESSIONS	CLICKS	CTR	BOUNCES	CONVERSIONS
111973.85	8349876	17551	0.21	6059	684
	CPC	UNIQUES	CPM	BOUNCE RATE	CPA
	5.82	17505	13.41	34.52	163.7

Philip, 41 - Client Company CEO



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

Bounce redefinition option
Scenario 2

As a **<Client Company CEO>** I want **<to be able to define a bounce in reports generated>** so that **<reporting stays consistent with our company standards>**

Test against the scenario

Type of test: Manual

Preconditions:

- Campaign is loaded in the application
- Campaign is selected as the current

Actions:

- Loaded application
- Selected campaign from dropdown
- Switched to the "Bounces metric"
- Selected desired bounce definition
- Pressed the "Recompute Data" button

Application opens successfully

Dropdown correctly displays all available campaigns

Selected campaign loads correctly

Bounces total number is computed correctly

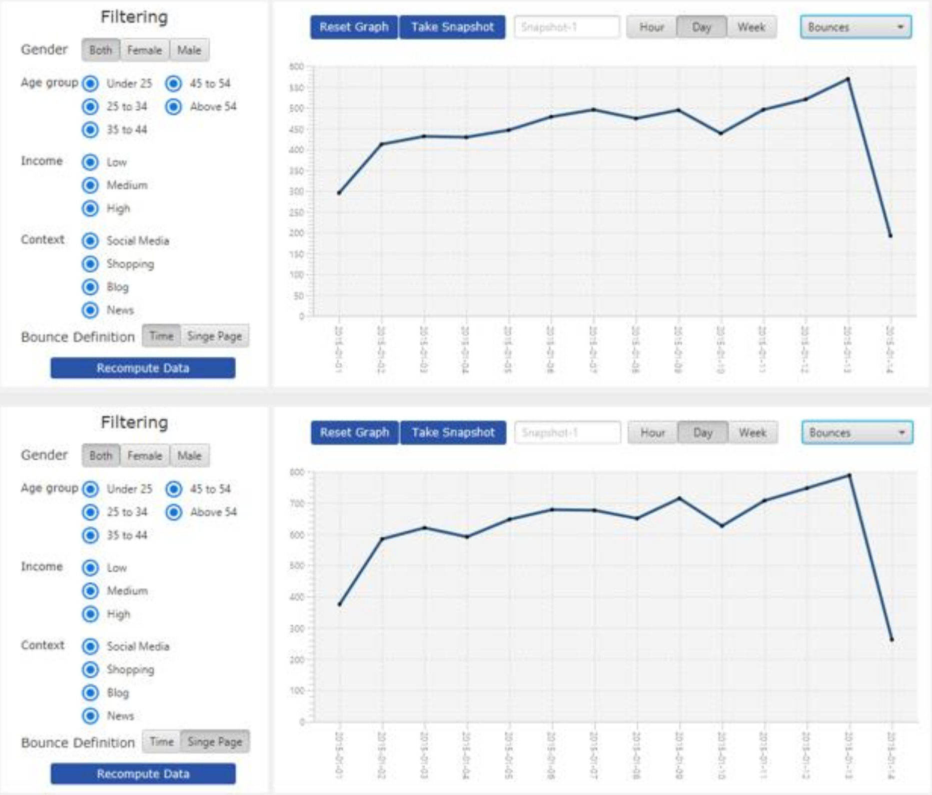
Bounces graph is loaded correctly

Bounce definition is applied correctly in isolation

Bounce definition is applied correctly with other filters

No major delays in display time

Test screenshot



Robert, 23 - A Marketing Data Analyst

Goals	Frustrations
Finding particular audiences	Generating inaccurate data
Analyzing data effectively	Performing analysis manualy
Quickly learn the new tool	Working with advanced marketing dashboards

Time period data visualization
Scenario 3

As a **<Marketing Data Analyst>** I want **<charts over diverse time periods for metrics>** so that **<I can visualize the campaign performance over time>**

Test against the scenario

Type of test: Manual

Preconditions:

- Campaign is loaded in the application
- Queries support time periods

Actions:

- Loaded application
- Selected campaign from dropdown
- Selected start/end date

Application opens successfully

Dropdown correctly displays all available campaigns

Selected campaign loads correctly

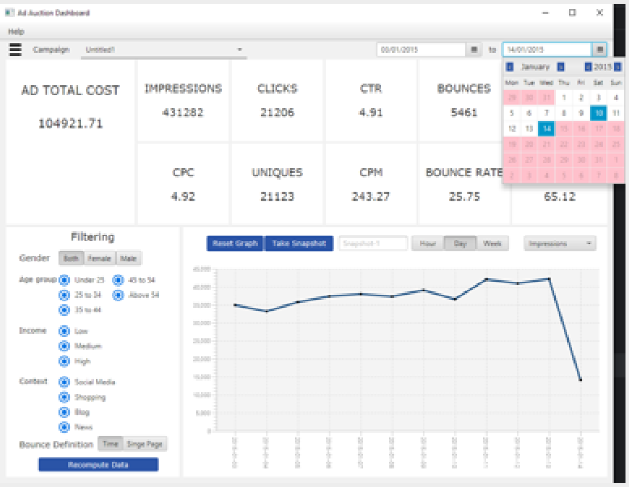
Start and end date of campaign is computed correctly and displayed in the selectors

Selectors correctly don't allow selection of a date outside the campaign available dates

Error message is correctly displayed if start > end date

Time period is correctly applied and data is automatically recalculated

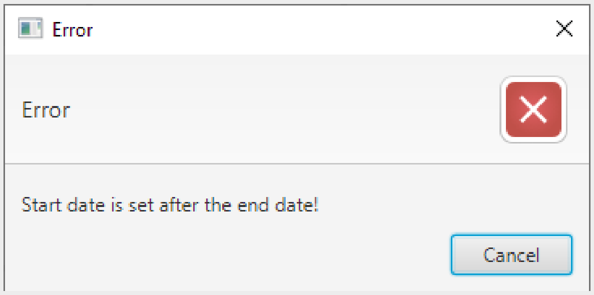
Test screenshot



AD TOTAL COST	IMPRESSIONS	CLICKS	CTR	BOUNCES
104921.71	431282	21206	4.91	5461
	CPC	UNIQUES	CPM	BOUNCE RATE
	4.92	21123	243.27	25.75

65.12

14/01/2015



Error

Error

Start date is set after the end date!

Cancel

Robert, 23 - A Marketing Data Analyst

Goals	Frustrations
Finding particular audiences	Generating inaccurate data
Analyzing data effectively	Performing analysis manualy
Quickly learn the new tool	Working with advanced marketing dashboards

Time period data visualization
Scenario 4

As a **<Marketing Data Analyst>** I want **<to be able to compare charts on the same metrics>** so that **<I can report on performance for different time frames>**

As a **<Marketing Data Analyst>** I want **<a diverse range of metrics>** so that **<I can measure the campaign performance>**

Test against the scenario

Type of test: Manual

Preconditions:

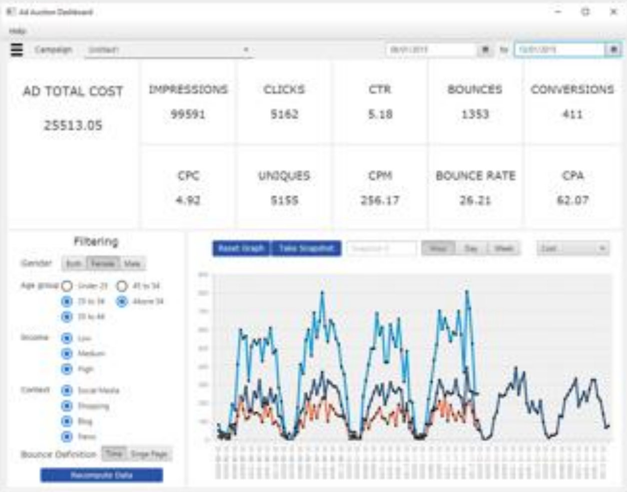
- Campaigns loaded in the application
- Graph supports snapshotting

Actions:

- Loaded application
- Selected campaign from dropdown
- Selected metric and filters
- Selected time period and granularity
- Put desired name for snapshot
- Pressed the "Take Snapshot" button
- Repeated process for multiple graphs
- Reset Graph after test done

Application opens successfully
Dropdown correctly displays all available campaigns
Selected campaign loads correctly
Filters and options applied correctly
Snapshot name is applied correctly
Graphs are overlayed correctly
Graph not overlayed unless snapshot taken
Switching metrics works correctly
Graph resetting removes previous snapshots and leaves current graph
Correct behaviour when switching metric or time period
Correct point values and names when hovering over data points

Test screenshot



AD TOTAL COST	IMPRESSIONS	CLICKS	CTR	BOUNCES	CONVERSIONS
25513.05	99591	5162	5.18	1353	411
	CPC	UNIQUES	CPM	BOUNCE RATE	CPA
	4.92	5155	256.17	26.21	62.07

Deborah, 39 - DIY Online Shop Owner

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

Reading campaign log files
Scenario 5

As a **<Agency Client>** I want **<data to be read from files in a specific format>** so that **<the application can be used with search engine data tables>**

Test against the scenario

Type of test: Manual

Actions:

- Loaded application
- Opened side benu with burger button
- Pressed "Add Campaign" button
- Input campaign name in field
- Selected input files with an invalid one
- Selected input files with an invalid one

Application opens successfully

Side menu opens correctly

Add campaign promt disables main screen

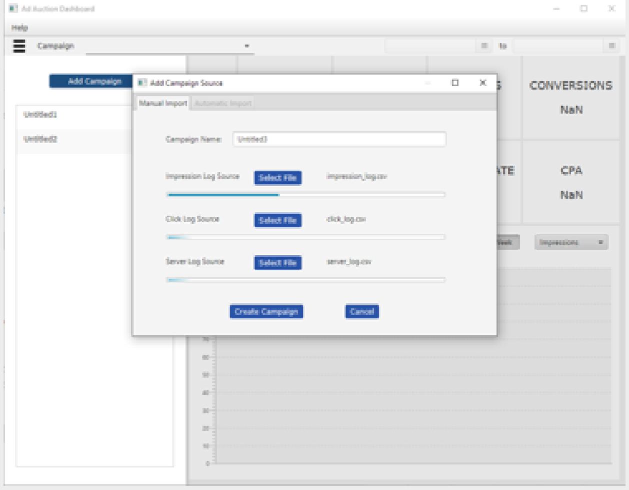
Error message is displayed for invalid files

Valid files are processed with appropriate visual feedback

Window closes and campaign is loaded into current view

Canceling at any time in the process has expected behaviour

Test screenshot

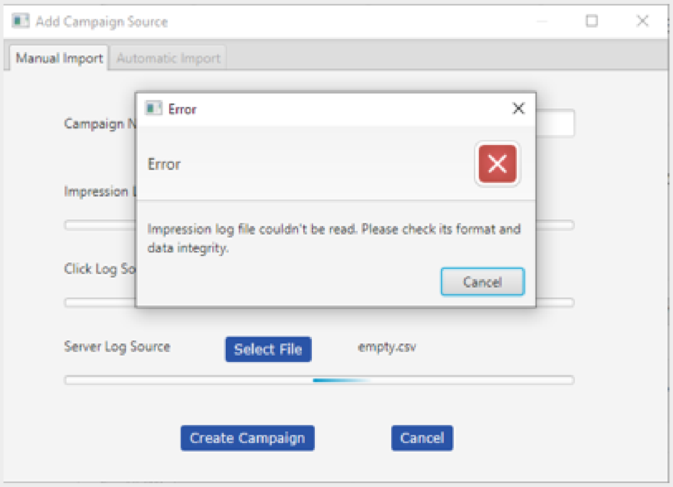


Campaign Name: Untitled3

Impression Log Source: impression_log.csv

Click Log Source: click_log.csv

Server Log Source: server_log.csv



Error

Impression log file couldn't be read. Please check its format and data integrity.

Server Log Source: empty.csv

Deborah, 39 - DIY Online Shop Owner

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

Campaign filtering
Scenario 6

As a **<Agency Client>** I want **<filtering by gender, age, time frame and context>** so that
<I can better understand the ad's audience and adapt the marketing strategy accordingly>

Test against the scenario

Type of test: Manual

Preconditions:

- Campaign is loaded in the application
- Queries support filtering

Actions:

- Loaded application
- Selected campaign from dropdown
- Selected desired filters
- Selected desired time frame
- Pressed the "Recompute Data" button
- Repeated process for different filter combinations

Application opens successfully

Dropdown correctly displays all available campaigns

Selected campaign loads correctly

Toggling between age filters disables the other two

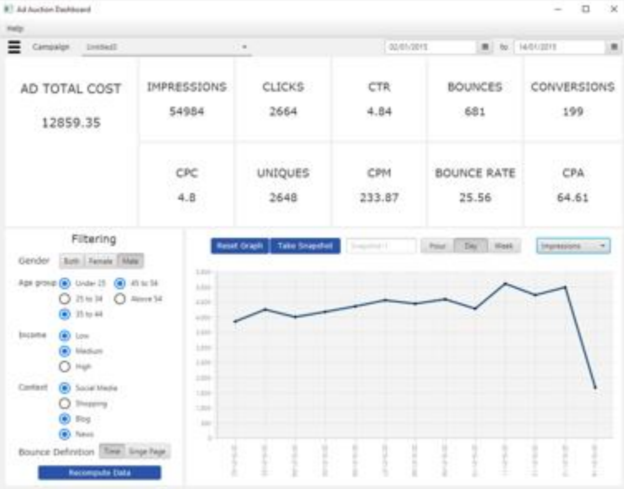
All filters correctly require at least one option to be selected

UI elements are disabled correctly during recalculation

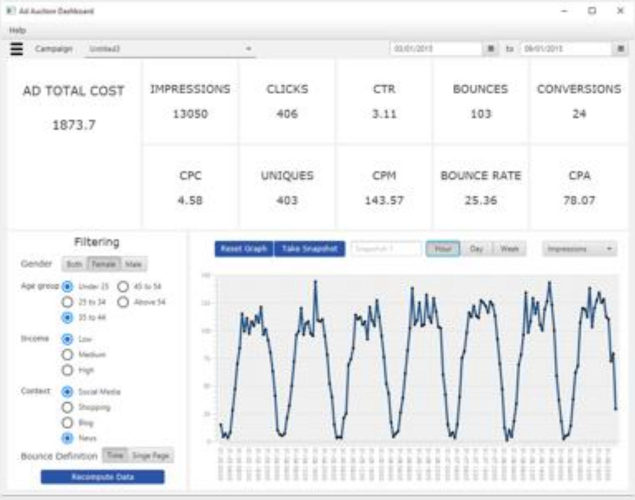
Recalculation takes an appropriate amount of time

No major delays in display time

Test screenshot



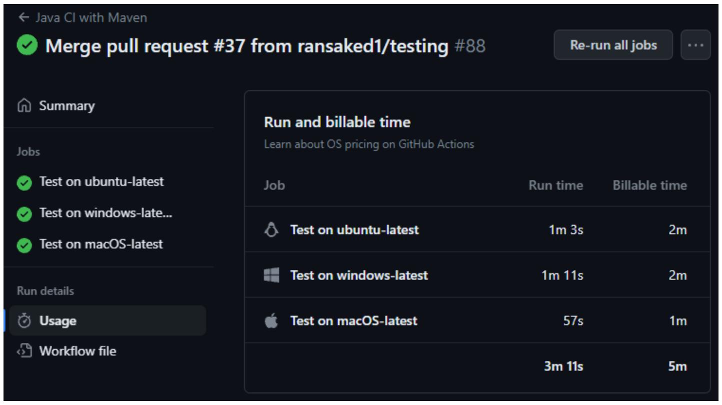
AD TOTAL COST	IMPRESSIONS	CLICKS	CTR	BOUNCES	CONVERSIONS
12859.35	54984	2664	4.84	681	199
	CPC	UNIQUES	CPM	BOUNCE RATE	CPA
	4.8	2648	233.87	25.56	64.61



AD TOTAL COST	IMPRESSIONS	CLICKS	CTR	BOUNCES	CONVERSIONS
1873.7	13050	406	3.11	103	24
	CPC	UNIQUES	CPM	BOUNCE RATE	CPA
	4.58	403	143.57	25.36	78.07

2.2 Unit testing

All unit tests are executed when code is pushed to the repository to detect issues as soon as possible using GitHub Actions. The testing script compiles and runs the Surefire tests on Windows, Ubuntu and MacOS operating systems. Below is a breakdown of the final unit test results:



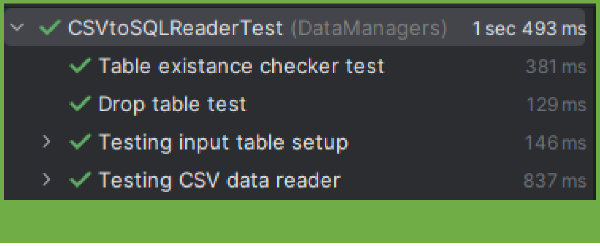
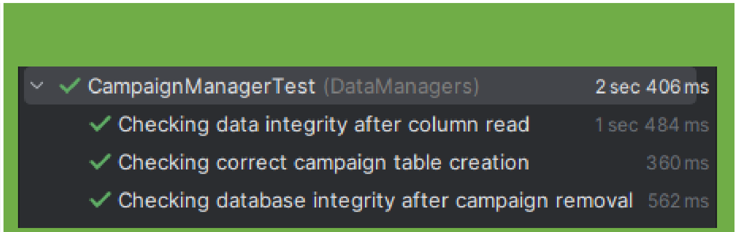
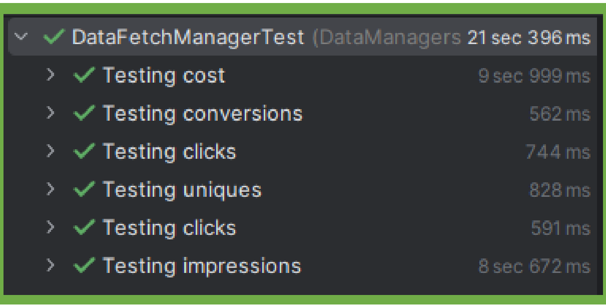
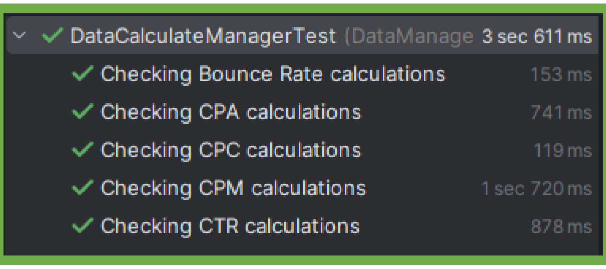
Merge pull request #37 from ransaked1/testing #88

- Test on ubuntu-latest
- Test on windows-late...
- Test on macOS-latest

Run and billable time

Learn about OS pricing on GitHub Actions

Job	Run time	Billable time
Test on ubuntu-latest	1m 3s	2m
Test on windows-latest	1m 11s	2m
Test on macOS-latest	57s	1m
	3m 11s	5m

Unit Test	Pass/Fail
CSV to SQL Reader - Before and after each test all tables are dropped - Testing table existance method - Testing table setup and removal - Checking data is read correctly or failure is reported for different CSV files in valid and wrong formats	 ✓ CSVtoSQLReaderTest (DataManagers) 1 sec 493 ms ✓ Table existance checker test 381 ms ✓ Drop table test 129 ms ✓ Testing input table setup 146 ms ✓ Testing CSV data reader 837 ms
Campaign Manager - All tables are dropped after the test is finished - Testing correct campaign creation with appropriate tables being populated - Checking tables are correctly populated - Testing campaign is correctly dropped and all related tables are removed	 ✓ CampaignManagerTest (DataManagers) 2 sec 406 ms ✓ Checking data integrity after column read 1 sec 484 ms ✓ Checking correct campaign table creation 360 ms ✓ Checking database integrity after campaign removal 562 ms
Fetch Manager - Before testing starts, csv files are read and tables are populated including padding columns used for filtering - Before each test all filtering is removed - After all tests, all tables are dropped - For each metric testing the output with individual and mixed filters	 ✓ DataFetchManagerTest (DataManagers 21 sec 396 ms ✓ Testing cost 9 sec 999 ms ✓ Testing conversions 562 ms ✓ Testing clicks 744 ms ✓ Testing uniques 828 ms ✓ Testing clicks 591 ms ✓ Testing impressions 8 sec 672 ms
Calculate Manager - Before testing starts, csv files are read and tables are populated including padding columns used for filtering - After all tests, all tables are dropped - Basic tests are run as this class is extending the fetch manager which has more comprehensive testing	 ✓ DataCalculateManagerTest (DataManage 3 sec 611 ms ✓ Checking Bounce Rate calculations 153 ms ✓ Checking CPA calculations 741 ms ✓ Checking CPC calculations 119 ms ✓ Checking CPM calculations 1 sec 720 ms ✓ Checking CTR calculations 878 ms

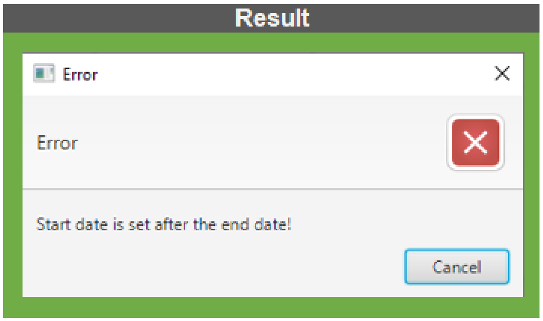
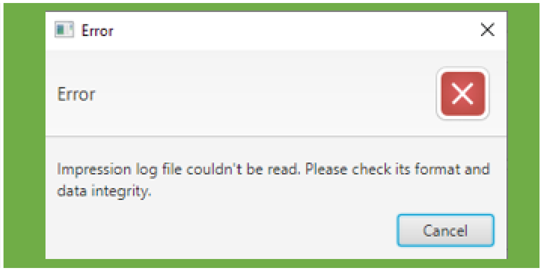
2.3 Regression testing

Regression Test	Test Description & Results	Pass/Fail
Sprint 1 Features		
Uploading a campaign	Campaigns can still be uploaded correctly, by uploading the three source files separately. It takes a reasonable amount of time to be uploaded, with added functionality of progress bars to show the current progress.	Pass
Fetch and calculate metrics	Metrics are generated across the top of the main window allowing the user to see the key metrics.	Pass
Sprint 2 Features		
Sidebar Functionality	The sidebar opens and closes easily through clicking the hamburger icon.	Pass
Changing the time period	Changing the selected date region still functions, allowing you to customise what period of data you wish to look at.	Pass
Switching between campaigns	The drop down to switch between campaigns hasn't changed and still functions as it should. You can combine this with the graph to allow comparing between two graphs.	Pass
Deleting campaigns	The sidebar still functions with deletion icons for each campaign. These icons work quickly allowing you to remove multiple campaigns in succession.	Pass

2.4 Integration testing

Integration Test	Test Description & Results	Pass/Fail
Importing file data into database and removing it	**Manual and unit testing** performed to check database is correctly populated and correct table naming convention is followed and then the data is correctly cleared **Actions:** uploaded log files and checked database tables **Results:** database is populated correctly and unit tests passed	Pass
Integration between GUI, graphs and database	**Manual and unit testing** performed to check correct integeration between GUI, graphs and database **Actions:** Check filters and time periods affect all data elements of the campaign, campaign removal affects database and UI **Results:** Minor visual bugs detected in random instances	**Pass**
Integration between snapshotting and graph drawing	**Manual testing** performed to check correct overlaying of compared graphs using the graph snapshot feature **Actions:** overlaying diverse graph with different filters and time periods applied, and with different names **Results:** Graphs are overlayed correctly and corerct points shown when hovering the graph	Pass

2.6 Boundary and partition testing

Test Requirement	Test Description	Result
Start date has to be before end date in the date pickers	Error message is displayed and the metrics are still computed but NaN values and 0s are returned	 Start date is set after the end date!
Log file has to be in a valid format when importing new campaign	Error message is displayed and the add campaign window is reset	 Impression log file couldn't be read. Please check its format and data integrity.

3. RESPONSE TO FEEDBACK

After the review meeting for the third deliverable/second increment, we took on all the feedback to make sure to meet the requirements of the customer/supervisor.

Application
From both the discussion in the meeting and our written feedback afterwards, it was clear that our front-end and actual functionality of the program was our key focus. We prioritized the graph and completing all the "musts" and even managed to finish some "shoulds".

Design
We continued our previous design artifacts from the last sprint, but after a discussion with our supervisor we decreased the detail within the UML class diagram as it became too complex to read. As mentioned, we improved our prioritization focusing on the key features and requirements such as the graphs.

Testing
We have continued our previous testing from the last sprint but have expanded it further especially since this is the final sprint. We have completed scenario testing for every persona and have used unit testing to test specific methods and classes.

Planning
We followed our sprint plan from last time but edited it slightly to match our reprioritization focusing more on the front end. Our goal was to make sure that we delivered value to the customer. We supported this by organising more meeting with our supervisor to give us more opportunities for feedback.

4. SPRINT RETROSPECTIVE

4.1 Sprint 3 Retrospective

The main goal of this sprint was finishing the UI of the application before trying to implement the additional features requested by the client. All "musts" related to 01/02 (**data graphing**) and 05 (**data filtering**) have been completed. Our team then went on to develop features related to 14/19 (**comparing multiple campaign graphs**). Additionally, the UI was populated with many more elements to support the changes made to the backend the last sprint.

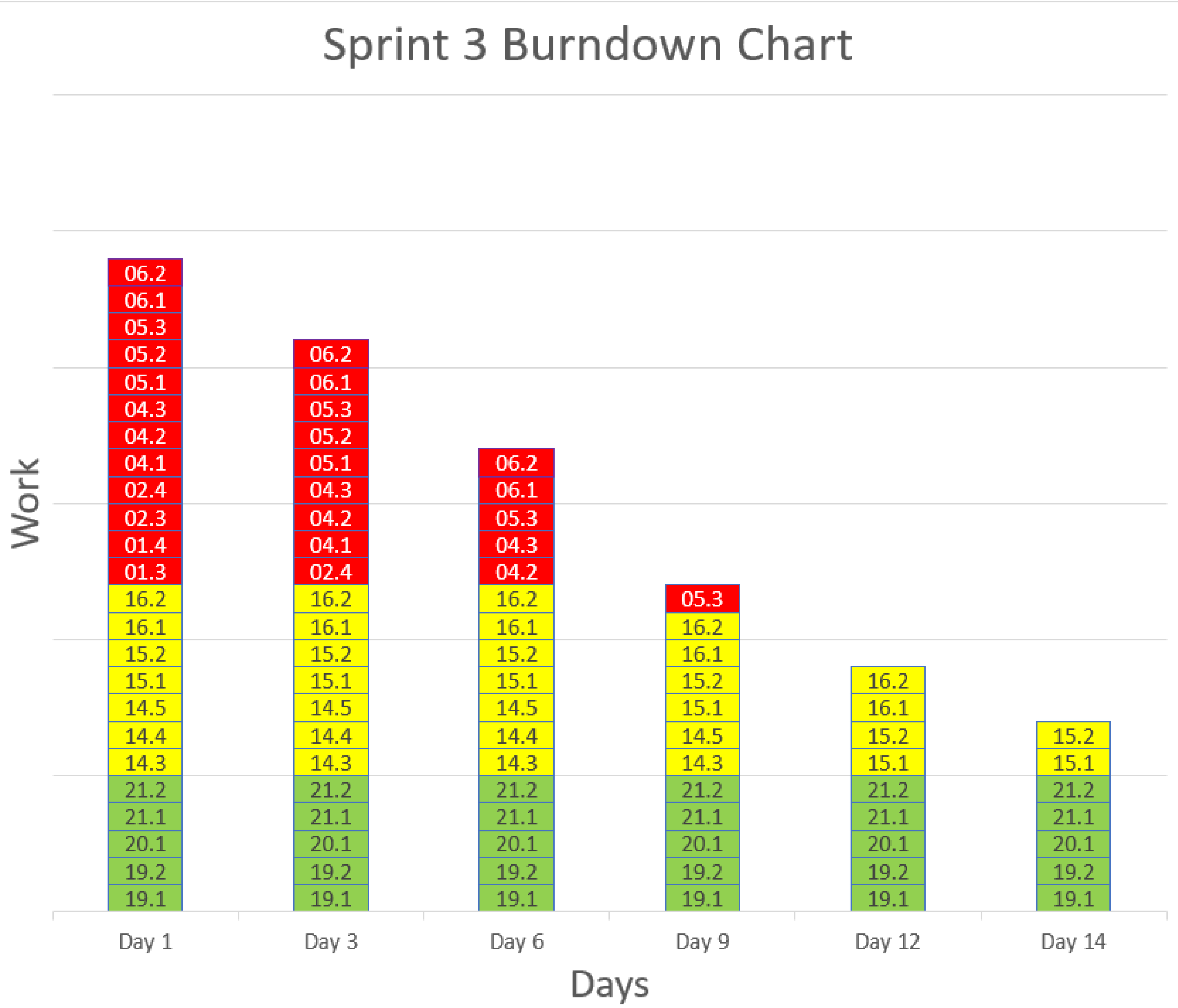
The remaining time was spent on rigorous testing and bug fixing to make sure the application does not crash even if an unexpected run-time error is encountered, and errors are reported to the user.

By now team cohesion has significantly improved and we were able to keep productivity at a high level without meeting as frequently as in the previous sprints. Each team member was aware of their role and responsibilities and communication was maintained mainly through messages and status updates with less group meetings. We believe this contributed significantly to the success of this sprint.

We are very satisfied with the progress made this sprint after implementing the feedback received the previous sprint.

4.2 Sprint 2 Burndown Chart

ID	Assignee Name	Time (h)
01.3	Pingding He	5
01.4	Pingding He	3
02.3	Connor Calkin	5
02.4	Connor Calkin	3
04.1	Lucas Sayers	3
04.2	Lucas Sayers	3
04.3	Benjamin Lewis	2
05.1	Daniel Braghis	3
05.2	Daniel Braghis	3
05.3	Connor Calkin	2
06.1	Benjamin Lewis	4
06.2	Daniel Braghis	2
14.3	Daniel Braghis	9
14.4	Daniel Braghis	4
14.5	Lucas Sayers	6
16.1	Benjamin Lewis	3
16.2	Benjamin Lewis	4


Sprint 3 Burndown Chart
Work
Days
Day 1
06.2
06.1
05.3
05.2
05.1
04.3
04.2
04.1
02.4
02.3
01.4
01.3
16.2
16.1
15.2
15.1
14.5
14.4
14.3
21.2
21.1
20.1
19.2
19.1
Day 3
06.2
06.1
05.3
05.2
05.1
04.3
04.2
04.1
02.4
16.2
16.1
15.2
15.1
14.5
14.4
14.3
21.2
21.1
20.1
19.2
19.1
Day 6
06.2
06.1
05.3
04.3
04.2
16.2
16.1
15.2
15.1
14.5
14.4
14.3
21.2
21.1
20.1
19.2
19.1
Day 9
05.3
16.2
16.1
15.2
15.1
14.5
14.3
21.2
21.1
20.1
19.2
19.1
Day 12
16.2
16.1
15.2
15.1
21.2
21.1
20.1
19.2
19.1
Day 14
15.2
15.1
21.2
21.1
20.1
19.2
19.1