MARCH 6, 2019

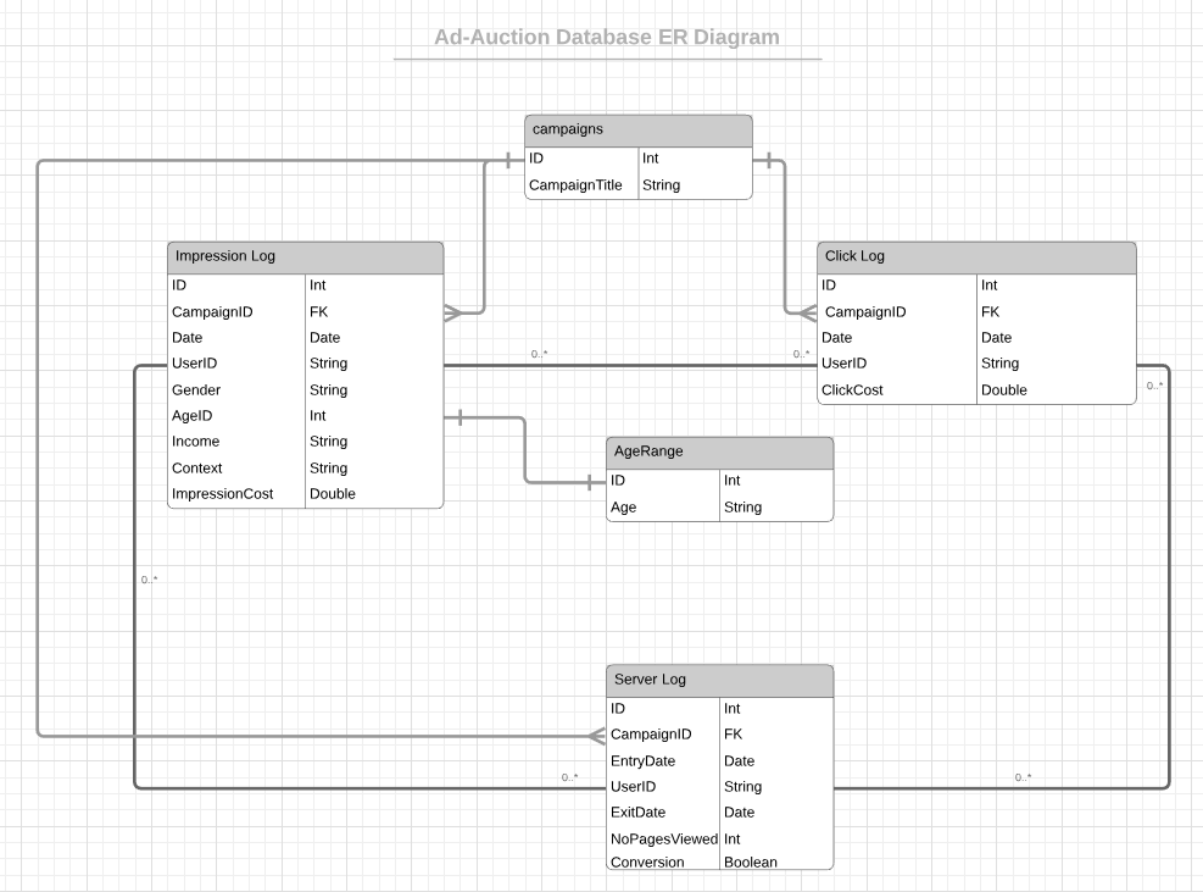
AD-AUCTION BOARD INCREMENT 1

GROUP 13

ISTVAN ZENO HEVESI (IZH1G17)
YAVOR EDIPOV (YE1G17)
WILLIAM KEWELL (WK3G17)
FURQAN HASHMI (MFH1G17)
CHRIS WANG (LW2N17)

KEY DESIGN ARTIFACTS AND CHOICES

To begin with, we began the development of application using Java 8. For the project we have decided to use **Swing and AWT** for the UI of the application as most of the team members are familiar with these technologies and this way everybody can easily make contribution. Furthermore, we took advantage of several Design Patterns which allowed us to easily distribute the work and open the application for future extensions. To name them: **Model View Controller and Observer Pattern.** We have also used functional programming in Java **(Streams)** which help with easily and efficiently working on large volumes of data. As requested by the client, we have planned to connect the application with an external database which would allow one to access their data even remotely. What is more, once loaded the data from files into the database, no further loading would be needed and could be instantly worked on. For this purpose, we have decided to use **MySQL Database,** because the advertisers work with structured data which is well suited for such database. This is the ER Diagram we have produced. I believe this will be the final choice, but we are open to changes in the future as well.

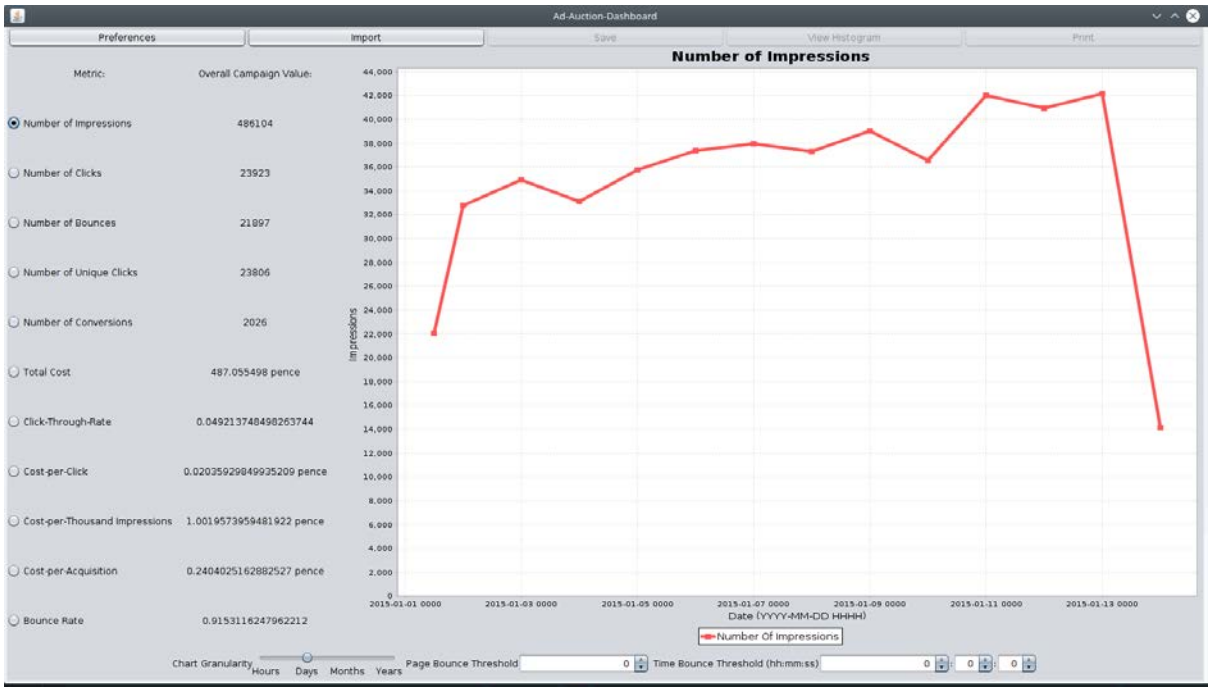


Impression Log, Click Log and Server Log all contain **UserID** and the relationship between all 3 tables is Many-to-Many as we have identified that the UserID is not unique.

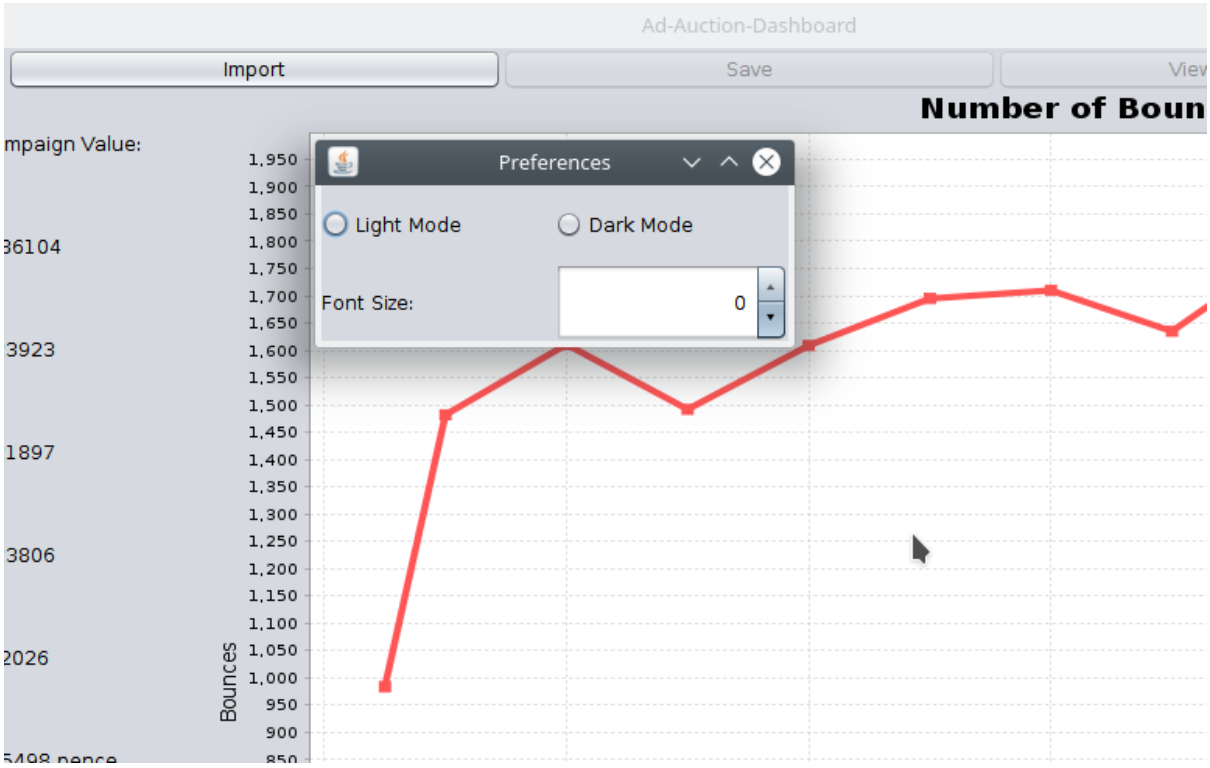
We have decided to separate the **AgeRange** (which is usually given as a string: e.g. <25, 25-34, etc.) into a separate table and take the ID as an indicator of the age range: 1 = "<25', 2 = "25-34" etc. This way manipulating the data becomes easier because we are comparing Integers. This allows quicker response from the application for visualizing data when applying filters. The **Campaigns** table will keep track of different campaigns loaded so that we can offer a dashboard to the user to choose which preloaded campaign they would work on.

For this Increment we have built a simple GUI which would allow us to test our application and deliver a usable product to the client. In the future the representation of the components and the layout might change in order to help with better User Experience. As it can be seen from the picture, there is a button on top **Import** where the user can upload the data files. Once uploaded, the data is parsed and the metrics of a campaign a displayed.

The user might as well choose which metrics they want to be graphed and control the granularity of the chart, analyze it and apply some simple filters. In the future this will be further extended.

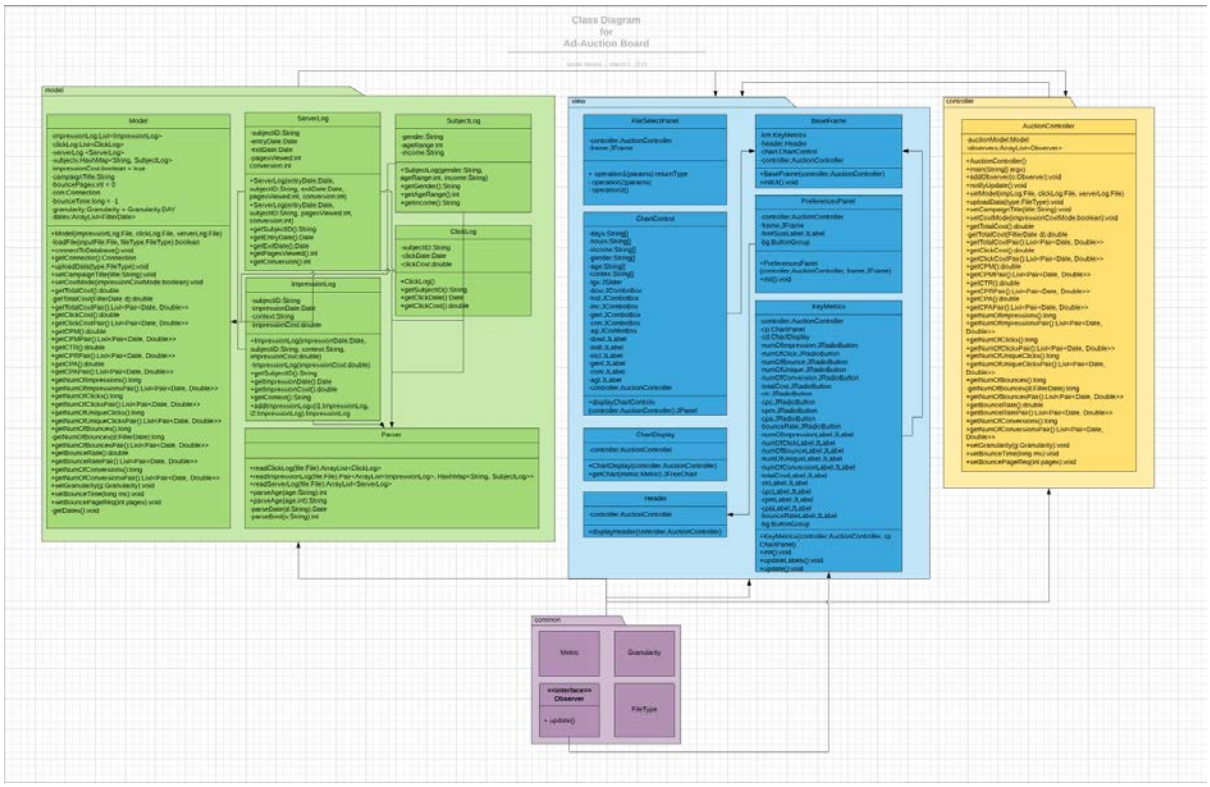


What is more, we have planned in advance to deliver a product that would be accessible to the widest range of people. During development we consider all types of features that would make the product accessible and work on them as well.

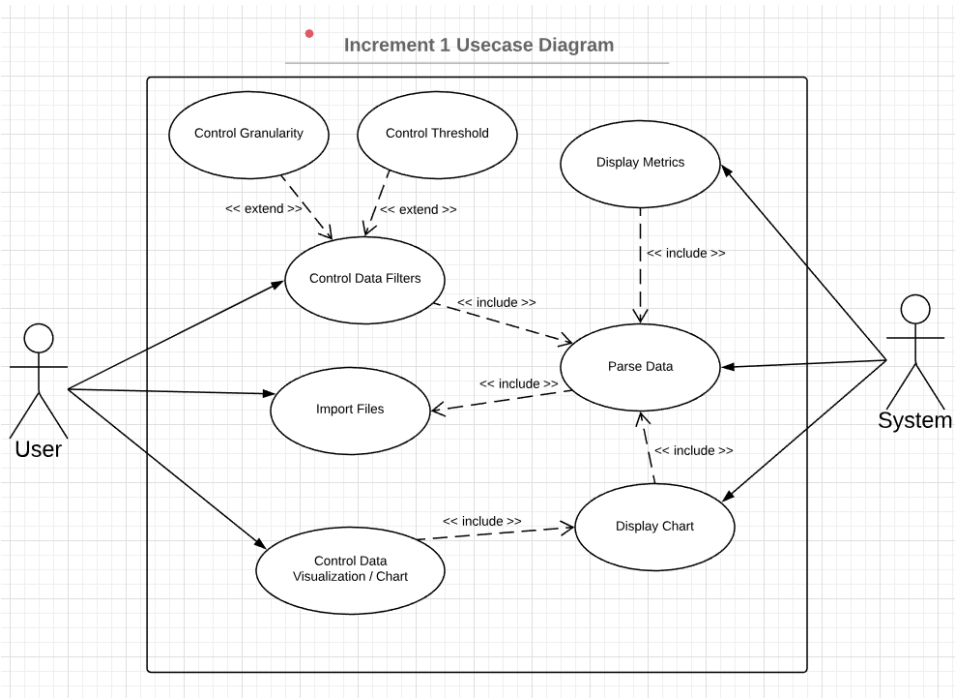


KEY STORYBOARDS

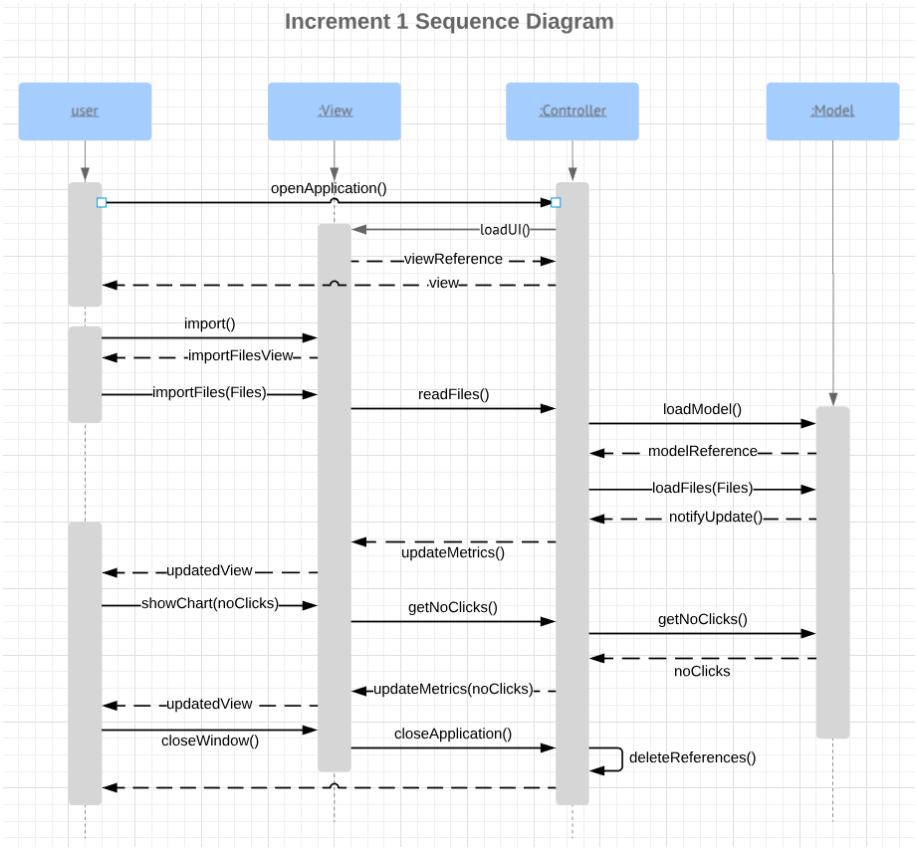
The **UML Class Diagram** describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. We used it for general conceptual modeling of the structure of the application, and for detailed modeling translating the models into programming code.



We have built a **UML Usecase Diagram** in order to get an overview of the system and how it would perform. It assisted us in picturing the interaction between the user and the system. The user can: Import Files, Control Data Filters and Visualization. Some of the actions do depend on others. For instances, in order for the user to be able to work with the data, it needs first to be parsed. However, in order for the system to parse the data in needs to be imported from files.

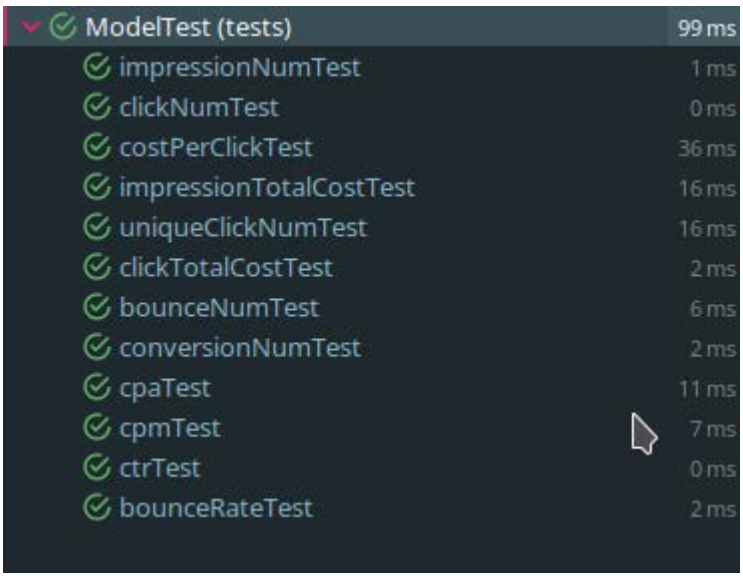


It was useful to build a **UML Sequence Diagram** as well to describe potential actions a user might take. This way we were able to implement needed features with a better understanding of the system and how it is supposed to behave.



KEY TEST OUTPUTS

Writing **Junit Tests** helps us prove the accuracy of the application. We have manually computed the expected results of certain operations using other tools and then tested these against the output of our program. As it can be seen from the picture (refer to the tests.ModelTests.java file for more details), the software passes all the written tests.



RESPONSES TO FEEDBACK

We were happy to receive mostly positive feedback. One thing we had missed was to send the document to the client before the review. It took a note of that and for this Increment and further ones, we are sending the document for review to the client in advance. This would help them prepare with any questions that might arise.

BURNDOWN CHART

INCREMENT 1 BURNDOWN CHART

Work/Tasks				
	14. Control the granularity of the charts	14. Control the granularity of the charts	14. Control the granularity of the charts	14. Control the granularity of the charts
	13. View charts of the key metrics over time	13. View charts of the key metrics over time	13. View charts of the key metrics over time	13. View charts of the key metrics over time
	Test data input/parsing	Test data input/parsing	Test data input/parsing	Test data input/parsing
	12. View the bounce rate	12. View the bounce rate	12. View the bounce rate	12. View the bounce rate
	11. View the clicks per thousand impressions (CPM)	11. View the clicks per thousand impressions (CPM)	11. View the clicks per thousand impressions (CPM)	11. View the clicks per thousand impressions (CPM)
	10. View the cost per click (CPC)	10. View the cost per click (CPC)	10. View the cost per click (CPC)	10. View the cost per click (CPC)
	9. View the cost per acquisition (CPA)	9. View the cost per acquisition (CPA)	9. View the cost per acquisition (CPA)	9. View the cost per acquisition (CPA)
	8. View the click through rate (CTR)	8. View the click through rate (CTR)	8. View the click through rate (CTR)	8. View the click through rate (CTR)
	7. View the total cost of campaign	7. View the total cost of campaign	7. View the total cost of campaign	7. View the total cost of campaign
	6. View number of conversions	6. View number of conversions	6. View number of conversions	6. View number of conversions
	5. View number of bounces	5. View number of bounces	5. View number of bounces	5. View number of bounces
	Test system	Test system	Test system	Test system
	Test granularity functionality	Test granularity functionality	Test granularity functionality	Test granularity functionality
	Test simple GUI	Test simple GUI	Test simple GUI	Test simple GUI
	Code data input and parsing	Code data input and parsing	Code data input and parsing	Code data input and parsing
	Code simple GUI	Code simple GUI	Code simple GUI	Code simple GUI
	Design final GUI	Design final GUI	Design final GUI	Design final GUI
	UML Diagrams	UML Diagrams	UML Diagrams	UML Diagrams
	4. View number of unique clicks	4. View number of unique clicks	4. View number of unique clicks	4. View number of unique clicks
	3. View number of clicks	3. View number of clicks	3. View number of clicks	3. View number of clicks
	2. View number of impressions	2. View number of impressions	2. View number of impressions	2. View number of impressions
	1. Import Data from CSV files	1. Import Data from CSV files	1. Import Data from CSV files	1. Import Data from CSV files
Day	0	1	2	3
		Thursday 21st	22	23

14. Control the granularity of the charts	14. Control the granularity of the charts		
13. View charts of the key metrics over time	13. View charts of the key metrics over time		
Test data input/parsing	Test data input/parsing		
12. View the bounce rate	12. View the bounce rate		
11. View the clicks per thousand impressions (CPM)	11. View the clicks per thousand impressions (CPM)	14. Control the granularity of the charts	
10. View the cost per click (CPC)	10. View the cost per click (CPC)		
9. View the cost per acquisition (CPA)	9. View the cost per acquisition (CPA)	13. View charts of the key metrics over time	
8. View the click through rate (CTR)	8. View the click through rate (CTR)	Test data input/parsing	
7. View the total cost of campaign	7. View the total cost of campaign	12. View the bounce rate	14. Control the granularity of the charts
6. View number of conversions	6. View number of conversions	11. View the clicks per thousand impressions (CPM)	
5. View number of bounces	5. View number of bounces	10. View the cost per click (CPC)	13. View charts of the key metrics over time
Test system	Test system	9. View the cost per acquisition (CPA)	Test data input/parsing
Test granularity functionality	Test granularity functionality	8. View the click through rate (CTR)	12. View the bounce rate
Test simple GUI	Test simple GUI	7. View the total cost of campaign	10. View the cost per click (CPC)
		6. View number of conversions	9. View the cost per acquisition (CPA)
Code data input and parsing	Code data input and parsing	5. View number of bounces	8. View the click through rate (CTR)
Code simple GUI	Code simple GUI	Test system	7. View the total cost of campaign
		Test granularity functionality	6. View number of conversions
		Test simple GUI	5. View number of bounces
Design final GUI	Design final GUI	Code simple GUI	Test system
			Test granularity functionality
UML Diagrams	UML Diagrams		Test simple GUI
4. View number of unique clicks	4. View number of unique clicks	Design final GUI	Code simple GUI
3. View number of clicks	3. View number of clicks		
2. View number of impressions	2. View number of impressions	UML Diagrams	
		4. View number of unique clicks	Design final GUI
		3. View number of clicks	
1. Import Data from CSV files	1. Import Data from CSV files	2. View number of impressions	UML Diagrams
4	5	6	7
24	25	26	27

14. Control the granularity of the charts			
13. View charts of the key metrics over time	14. Control the granularity of the charts	14. Control the granularity of the charts	
Test data input/parsing			
12. View the bounce rate	13. View charts of the key metrics over time	13. View charts of the key metrics over time	
10. View the cost per click (CPC)	Test data input/parsing	Test data input/parsing	
9. View the cost per acquisition (CPA)	12. View the bounce rate	12. View the bounce rate	
8. View the click through rate (CTR)	8. View the click through rate (CTR)	8. View the click through rate (CTR)	14. Control the granularity of the charts
7. View the total cost of campaign	7. View the total cost of campaign	7. View the total cost of campaign	
6. View number of conversions	6. View number of conversions	6. View number of conversions	13. View charts of the key metrics over time
5. View number of bounces	5. View number of bounces	5. View number of bounces	Test data input/parsing
Test system	Test system	Test system	Test system
Test granularity functionality	Test granularity functionality	Test granularity functionality	Test granularity functionality
Test simple GUI	Test simple GUI	Test simple GUI	Test simple GUI
Code simple GUI	Code simple GUI	Code simple GUI	Code simple GUI
Design final GUI	Design final GUI	Design final GUI	Design final GUI
UML Diagrams	UML Diagrams	UML Diagrams	UML Diagrams
8	9	10	11
28	1	2	3

It is important to be noted that we have moved some tasks ***(20. Define how a bounce is registered, Test Bounce Definition),*** which were planned to be done in the Increment 2 to Increment 1. The agile methods allowed for this flexibility and deliver a better product to the customer.

Test bounce definition		
20. Define how a bounce is registered		
Test data input/parsing		
Test system		
Test granularity functionality		
Test simple GUI		
	Test bounce definition	
	Test data input/parsing	
Design final GUI	Test system	
	Test granularity functionality	
UML Diagrams	Test simple GUI	
12	13	14
4	5	6

INCREMENT 2 BURNDOWN CHART

Work/Tasks	
	Test metrics per time of day
	25. View performance metrics per time of day
	Test metrics per day of week
	26. Display performance metrics per day of week
	Test overlapping charts
	19. Compare charts with different filters applied to them by overlapping them
	Test multi-campaign data loading
	24. Load data from multiple campaigns
	Test printing
	27. Printing functionality
	Test histogram creation
	15. View distribution of costs as a histogram
	Test context filter
	16. Filter metrics and charts by context
	Test date range filter
	17. Filter metrics and charts by date range
	Test audience segment filter
	18. Filter metrics and charts by audience segments
	Test final GUI
	Code Final GUI
	UML Diagrams
Day	0