COMP2211

INCREMENT 1

GROUP 42: Dylan Marodeen, James Hartley, Palak Jain, Vasilena Pencheva, William Kewell

Application

The application compiles, works and gives a usable and valuable graphical user interface that shows a graph of the data over time for the different metrics. We also allow for CSV input containing the files provided, producing all the key metrics we set out to display. All of our tasks for increment 1 were completed.

It must be noted that although the GUI gives you the option to add multiple campaigns, and that the toolbar has a number of buttons, neither of these are functional yet as they are part of later increments. We added aspects of these features to make development easier in the future, but they are not expected to work for this deliverable.

To run the jar file provided, follow the steps in the README.

Design

We used MVC to create our program. By splitting the project into 3 different packages, it allowed us to simultaneously edit and update our program on GitHub. We split into groups to be able to do pair programming to optimise our efficiency on both ends of the project. We made our layout simple and easy to understand, so that the user could quickly grasp what they could do in the program for this increment.

Design Decisions

Throughout our first SCRUM meeting for increment 1 we decided to complete our project by utilising a number of design methodologies.

JavaFX was our definite choice of GUI library due to its use of FXML files to decouple the code from the GUI. We used SceneBuilder to modify the GUI as it allowed us to clearly see the changes we were making as we were making them.

On the backend side of things, we used SQL to store the campaign data. We decided that while SQL was slightly slower that streams on the initial loading process, the greater memory efficiency and querying of SQL made it a superior choice.

Before we began programming, we designed a storyboard for our GUI, a class diagram and a use case diagram to help us on our project. We did diverge from these at times, however they were useful throughout the process as something to look back to. These are discussed more later on in the report.

To successfully develop our program, we used the agile development process, with SCRUM meetings every few days. We collaborated using Git when we were not meeting up and pair programming during our general meetings.

Architecture

We use a variety of features of JavaFX to allow for greater maintainability. For one, we use JavaFX FXML file and css style sheets for the front-end of our program. This not only decouples most of the GUI from the programming, but also means that we can simply add new GUI elements to our FXML file using SceneBuilder. We can then add relevant event code to the controller to facilitate changes to the model and view upon the user interacting with the interface.

We also used SQL when working with the data sets. This benefit here is that we can get relevant information using just queries. This means for features in future increments such as the filters, we can simply reuse the SQL class code with new queries and receive new data. The reason we chose SQL over an array or java stream data type is because of the efficiency and time we will save on inputting and outputting data. Initially, loading the files using SQL took far too long, however after some optimisation we were able to significantly cut down on loading time.

We also used GitHub in order to allow all of us to make modifications on different devices and keep a centralized master that can be maintained. This syncs very well with IntelliJ and allows us to directly commit and pull changes straight from the IntelliJ interface. GitHub also gives us maintainability as any mistakes people make can be rolled back so that our code should always be working.

Design Artifacts

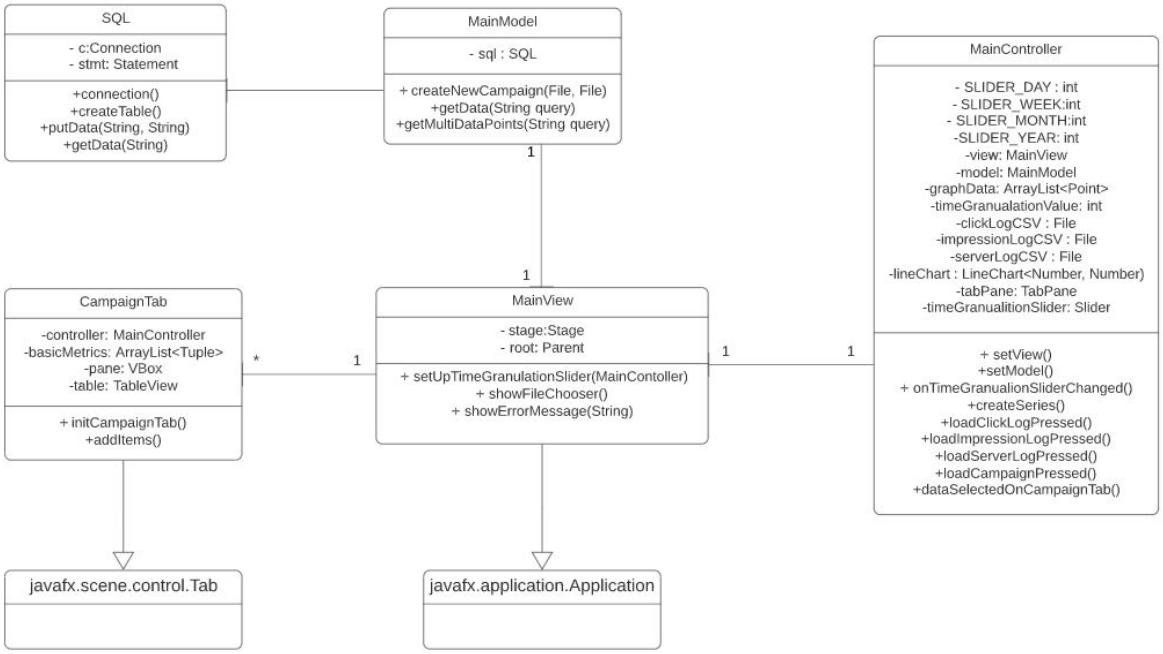
The design artifacts chosen for this project are the UML Use Case Diagram and the UML Class Diagram(both created using LucidChart), both of which were created prior to starting programming, and give an indication of what our application must do and how it must do so. These design artifacts were key to understanding our audience.

Through the class diagram, we identified the key classes, attributes, methods, and the relationships between them. This helped us plan and develop the structure of the application, and with a rough class diagram allowed the back-end and front-end teams to work exclusively without relying on each other which sped up the development process. The current class diagram shows the classes that would be needed for the first increment. A one-to-one relationship was identified between MainView, MainContoller and MainModel classes. The CampaignTab class handles all the data related to a specific campaign.There is a many-to -one relationship between this class and the MainView class, as data for several campaigns can be loaded and compared (several campaign tabs can be opened). We updated the class diagram as we went along with the programming, using variables like SLIDER in the Main Controller class order to keep track of each of the 4 ticks on the time granularity slider.

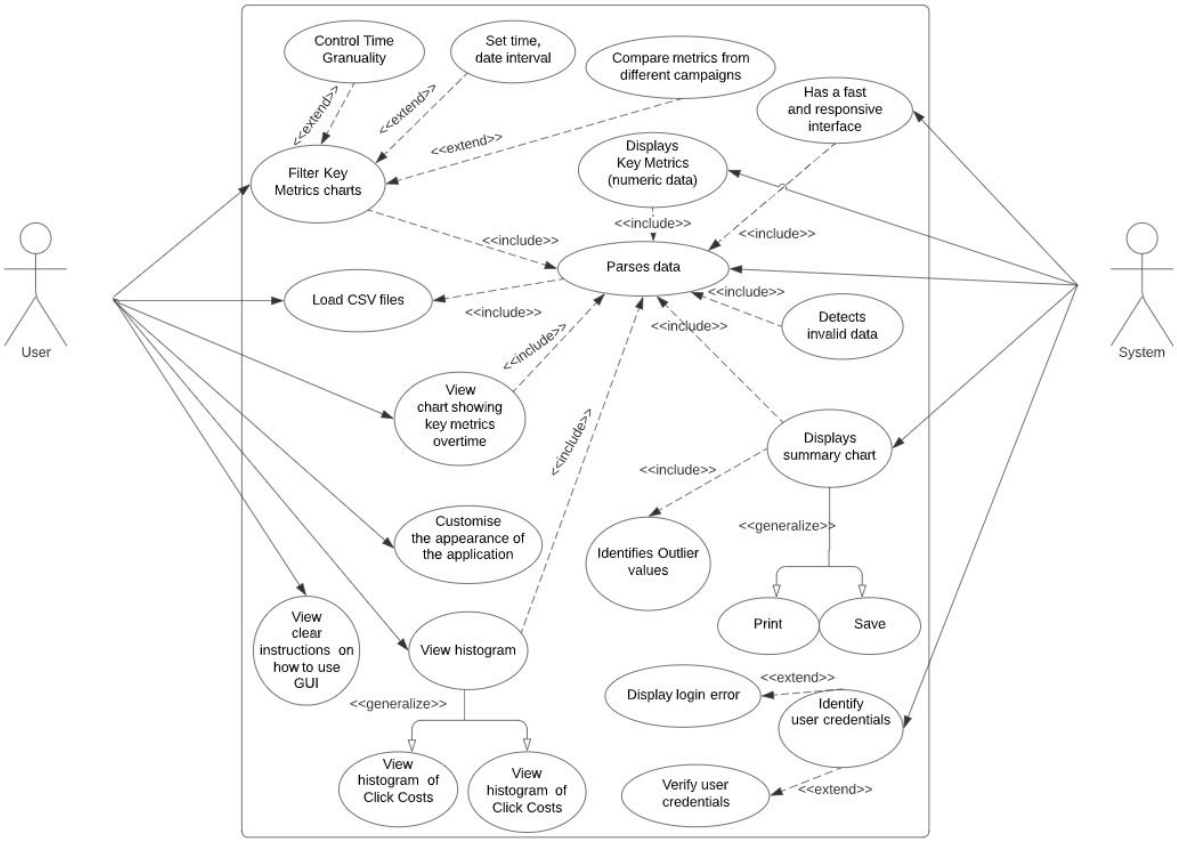
We used the Use Case diagram to clearly understand the relationship between the client of the ad agency and his/her interaction with the system(the two actors). The diagram identifies all the main use cases in the system(as of the final deliverable), and identifies the relationship between the actors and the use cases. Parsing of the data has been identified as a core use case as it is essential to any other functionality of the system or however the user uses the system. This solidified our initial thoughts that parsing data and displaying key metrics should be the main focus of deliverable one.

The Front-End Team drew out a series of story-boards on a whiteboard to display, discuss and integrate their ideas. One such rough drawing can be found below. The final storyboards, developed at the end of the planning phase with Moqups software, allowed the front-end team to visualize how they would lay the GUI and start programming it. This meant that their time was spent efficiently trying to achieve what they had already planned. Additionally, the labelling of the components of the design allowed us to develop methods and objects to implement functionality to each of them..

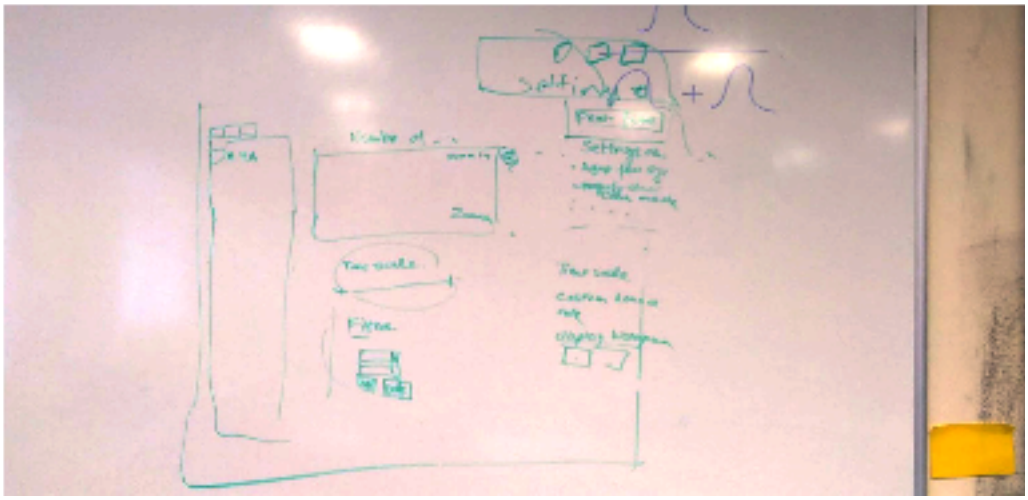
UML Class Diagram



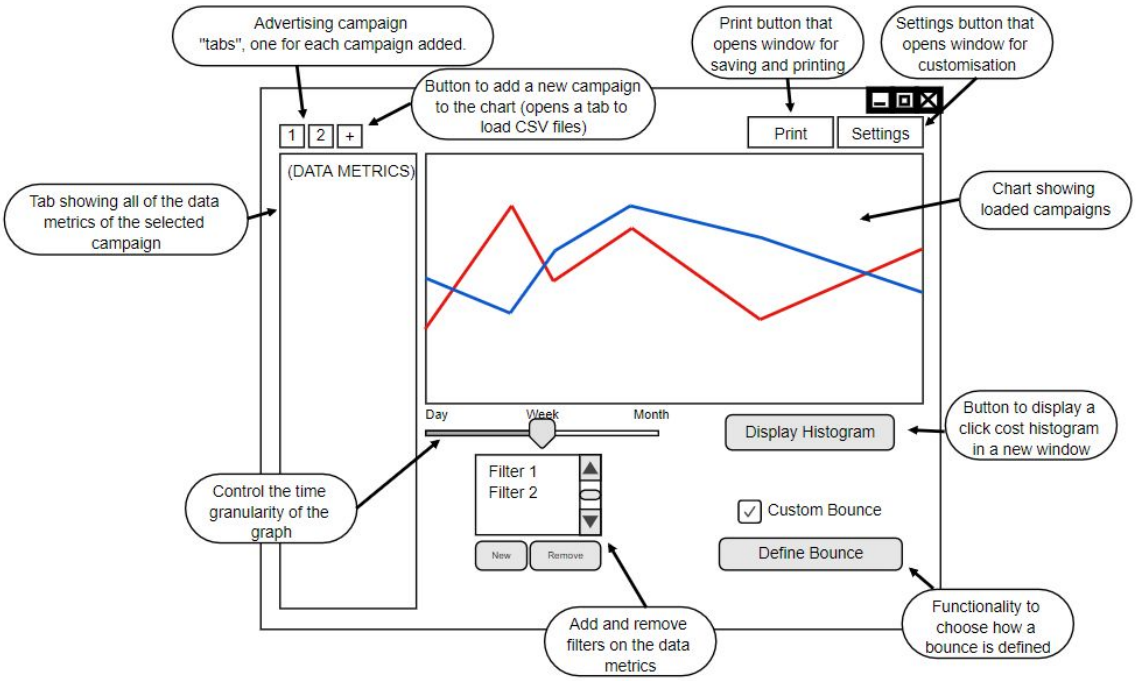
UML Use Case Diagram



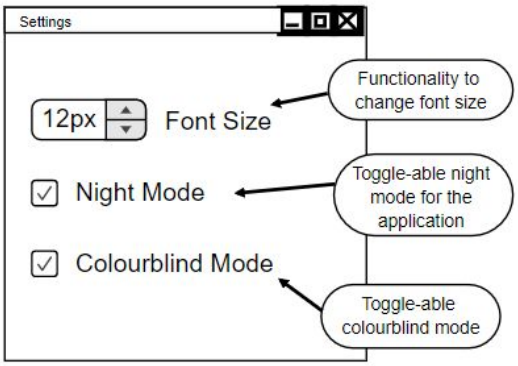
Storyboards



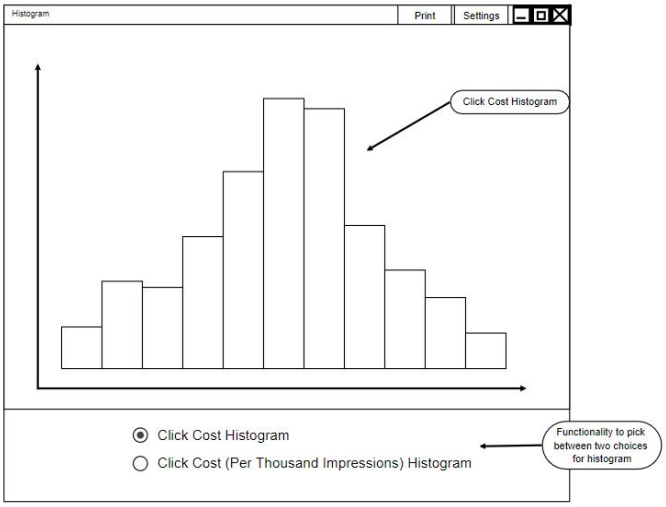
Main Window



Settings Window

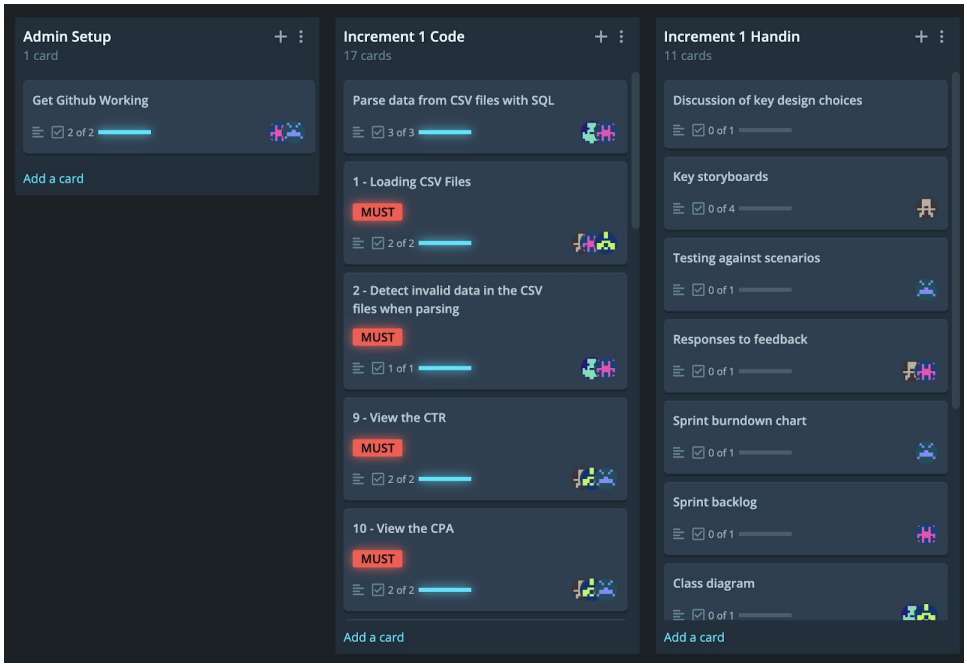


Histogram Window



How we responded to feedback

Feedback	How we responded
In your backlog, you should list all the "user stories" rather than requirements	We changed the format of our backlog so that each task in the backlog related to a user story by reference instead of a requirement. The user stories are copied below.
Incorrect format of the burndown chart. Where is the x axis? and y axis? scale and unit? You have numbered all the user stories so just specify the number.	We have added an x and y axis to the burndown chart to make it clearer what it is showing. We have now also changed the burndown chart to have the user story numbers instead of the text so that it is more readable.
You may want to consider Trello	During the envisioning deliverable and increment 1 we have been using gitKraken glo which is very similar to Trello and some of our group members already had experience with it. This was specified in the envisioning section. See below some example output from the software.



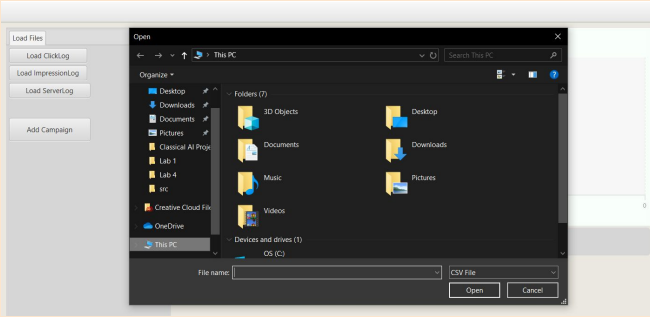
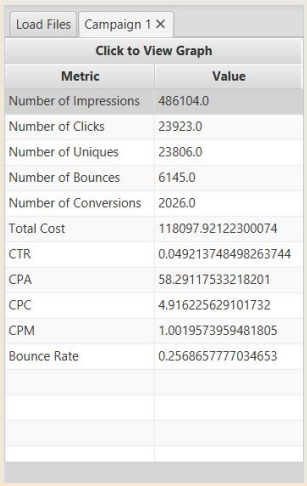
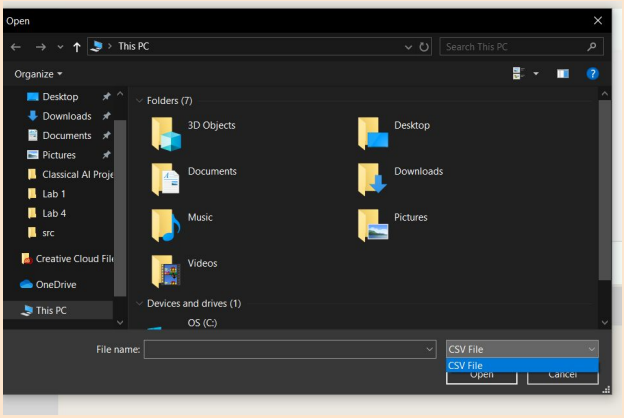
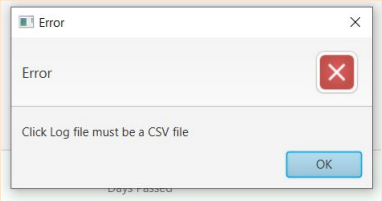
Testing

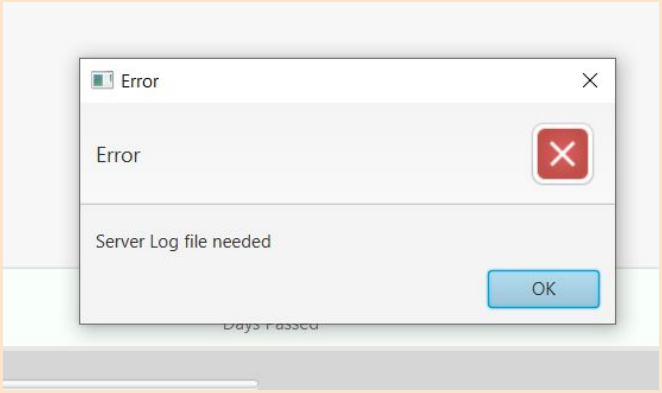
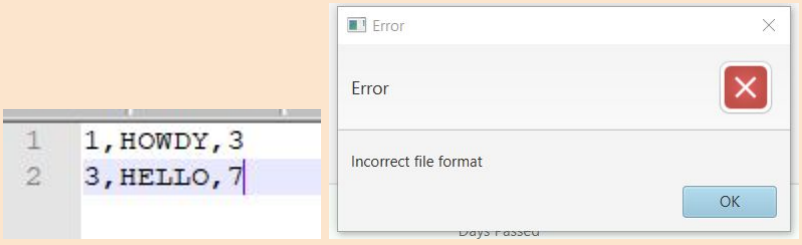
For this deliverable, we decided on using manual tests for our software. This is because the requirements are all based on information the user can see, IE its all visual. It was also tricky to determine any boundary tests since the only user inputs are the CSV files.

We determined a number of user story acceptance tests based on our sprint plan and sprint backlog for this deliverable. The table below describes each test. Further below that is a set of screenshots displaying the results of each test.

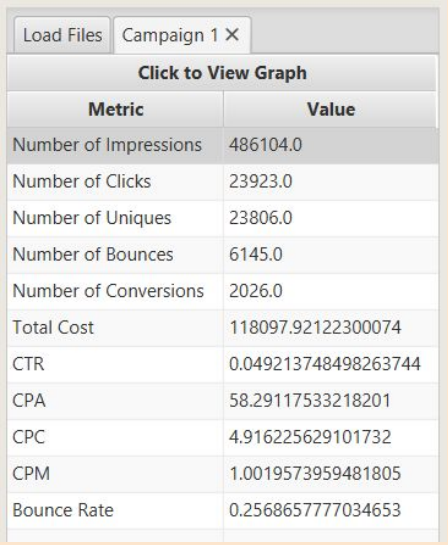
Test Number	Related User Stories	Test Criteria	Test type	Result
1.1	1	-**Campaign Manager** can select one of the “load file” buttons to open a file chooser screen -On click, it must be possible to select a CSV file -Once the **Campaign Manager** has added 3 valid files to there respective inputs, they can start a new campaign tab by pressing “Add Campaign” -A new campaign tab will be created will all the relevant information on it	Erroneous	Success
2.1	2	-**Campaign Manager** can select one of the “load file” buttons to open a file chooser screen -**Campaign Manager** will be unable to select any file that is not of the CSV format	Erroneous	Success
2.2	2	-**Campaign Manager** can choose any number of CSV files to be selected -**Campaign Manager** can select “Add Campaign” -An error message will be returned stating that a certain file was not added if the **Campaign Manager** chose not to add it	Erroneous	Success
2.3	2	-**Campaign Manager** can select the 3 relevant CSV files, where the formatting is correct but some of the data is wrong -**Campaign Manager** can create a new campaign tab -Any incorrect data in the CSV file will be ignored	Erroneous	Success
2.4	2	-**Campaign Manager** can select one of the “load file” buttons to open a file chooser screen -**Campaign Manager** can select any CSV file to be added -**Campaign Manager** can select “Add Campaign” -**Campaign Manager** will receive an error message stating that a file was of the wrong format	Valid	Success but could be improved with more custom error messages
3.1	3,4,5,6,7,8,9,10,11,12,13	-**Campaign Manager or Sales Manager** can create a new campaign tab -**Campaign Manager or Sales Manager** can select the new tab on the top tab bar -The different metrics are visible on the new tab	Valid	Success
4.1	14	-**Campaign Manager** can create a new campaign tab -**Campaign Manager** can click on a metric to see that metric over time on the graph	Valid	Success
5.1	15	-**Campaign Design Team Member** can create a new campaign tab -User can click on a metric to see that metric over time on the graph -User can modify the time granularity of the graph to see the trends and success of the campaign over days/weeks/months/years	Valid	Success

6.1	23	-**Campaign Manager** can select the relevant CSV files and click “Add Campaign” -The campaign will be loaded in a decent time -The different metrics over time can be selected and will change the graph view quickly	Valid	Success however uncertainty under different circumstances

Test Number	Screenshots/Results	Description
1.1	  Load Files | Campaign 1 × **Click to View Graph** **Metric** / **Value** Number of Impressions: 486104.0 Number of Clicks: 23923.0 Number of Uniques: 23806.0 Number of Bounces: 6145.0 Number of Conversions: 2026.0 Total Cost: 118097.92122300074 CTR: 0.049213748498263744 CPA: 58.29117533218201 CPC: 4.916225629101732 CPM: 1.0019573959481805 Bounce Rate: 0.2568657777034653	Upon clicking each of the load file buttons, a system specific file chooser appeared and allowed me to select the relevant valid files. Upon selecting all the files and pressing “Add Campaign”, a loading alert box appeared. After a short time, the loading box was hidden and a new campaign tab with all the required metrics appeared.
2.1	  Error Click Log file must be a CSV file OK	Upon selecting one of the load file buttons, a file chooser will pop up. As you can see in the screenshot, the only file format available to select is the CSV file format. Any other file types will simply not appear. It is possible to bypass this safeguard by directly searching for a hidden file, however our program will detect this and return an error.

2.2		For this test, I loaded the valid CSV files for the click log and the impression log, but not the server log. After clicking “Add Campaign” I got an error alerting me to that fact that I did not select the server log file.
2.3	Load Files | Campaign 1 × **Click to View Graph** **Metric** / **Value** Number of Impressions: 486104.0 Number of Clicks: 23923.0 Number of Uniques: 23806.0 Number of Bounces: 6145.0 Number of Conversions: 2026.0 Total Cost: 118097.92122300074 CTR: 0.049213748498263744 CPA: 58.29117533218201 CPC: 4.916225629101732 CPM: 1.0019573959481805 Bounce Rate: 0.2568657777034653 Load Files | Campaign 1 × **Click to View Graph** **Metric** / **Value** Number of Impressions: 486104.0 Number of Clicks: 23923.0 Number of Uniques: 23806.0 Number of Bounces: 6145.0 Number of Conversions: 2026.0 Total Cost: 118088.09376700074 CTR: 0.049213748498263744 CPA: 58.28632466288289 CPC: 4.915814833800139 CPM: 1.0019573959481805 Bounce Rate: 0.2568657777034653	For this test, I first loaded the campaign with unaltered data. The metrics for this campaign are shown on the leftmost screenshot. I then changed one of the click costs to “HELLO”. After loading the campaign again, you can see that any metrics that take click cost into account have gone down. This is because the system simply ignored the invalid value and reverted it to 0.
2.4		For this test, I gave correct inputs for impression log and server log, but I gave a file with invalid information for the click log. An error was returned, alerting me of an invalid file.

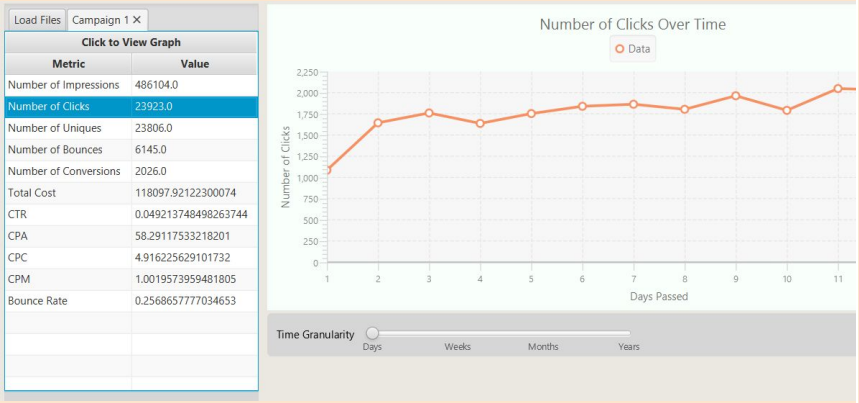
3.1



Metric	Value
Number of Impressions	486104.0
Number of Clicks	23923.0
Number of Uniques	23806.0
Number of Bounces	6145.0
Number of Conversions	2026.0
Total Cost	118097.92122300074
CTR	0.049213748498263744
CPA	58.29117533218201
CPC	4.916225629101732
CPM	1.0019573959481805
Bounce Rate	0.2568657777034653

Upon creation of a new campaign tab, the metrics will be visible on the tab. The format is clean and easy to interpret. Based on what we can gather from reading the data ourselves, the values appear to be correct based on our interpretation of the given CSV files.

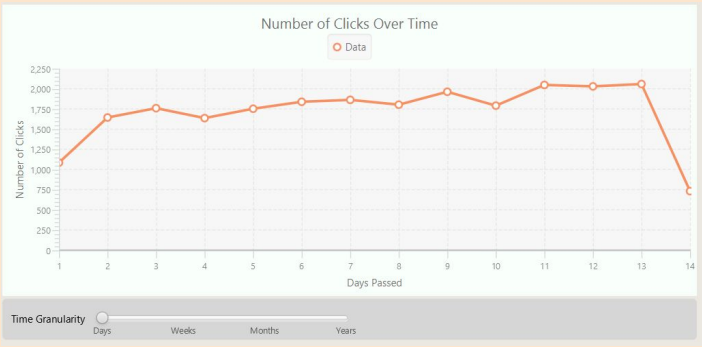
4.1



Metric	Value
Number of Impressions	486104.0
Number of Clicks	23923.0
Number of Uniques	23806.0
Number of Bounces	6145.0
Number of Conversions	2026.0
Total Cost	118097.92122300074
CTR	0.049213748498263744
CPA	58.29117533218201
CPC	4.916225629101732
CPM	1.0019573959481805
Bounce Rate	0.2568657777034653

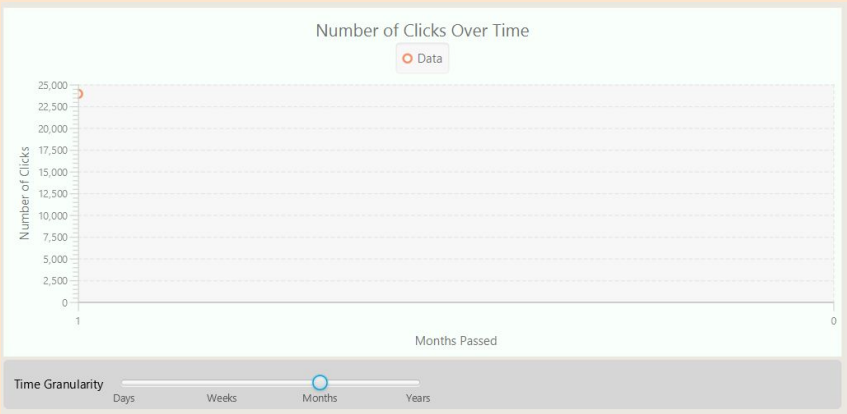
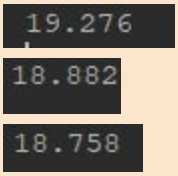
Upon selection of a metric for a campaign tab, its data over time will be shown on the graph.

5.1



Altering the time granularity of the graph will cause the number of ticks to change depending on the data. As shown here, by changing the time granularity of this 14 day campaign from days to weeks results in the number of ticks changing from 14 to 2. This is because there are 2 weeks in 14 days.

Changing the time granularity to months will result in a graph with 1 node since in a 2 week campaign, all the data will obviously come from the first month and the first year.

		
6.1	 	We programmed in a simple timer to count how long it took to load the files for testing purposes. It takes on average 20 seconds for the 2 week campaign to load. This is good however many factors could change this value. We also cannot test on larger data as of this time since the 2 month data set is currently unavailable.

Planning

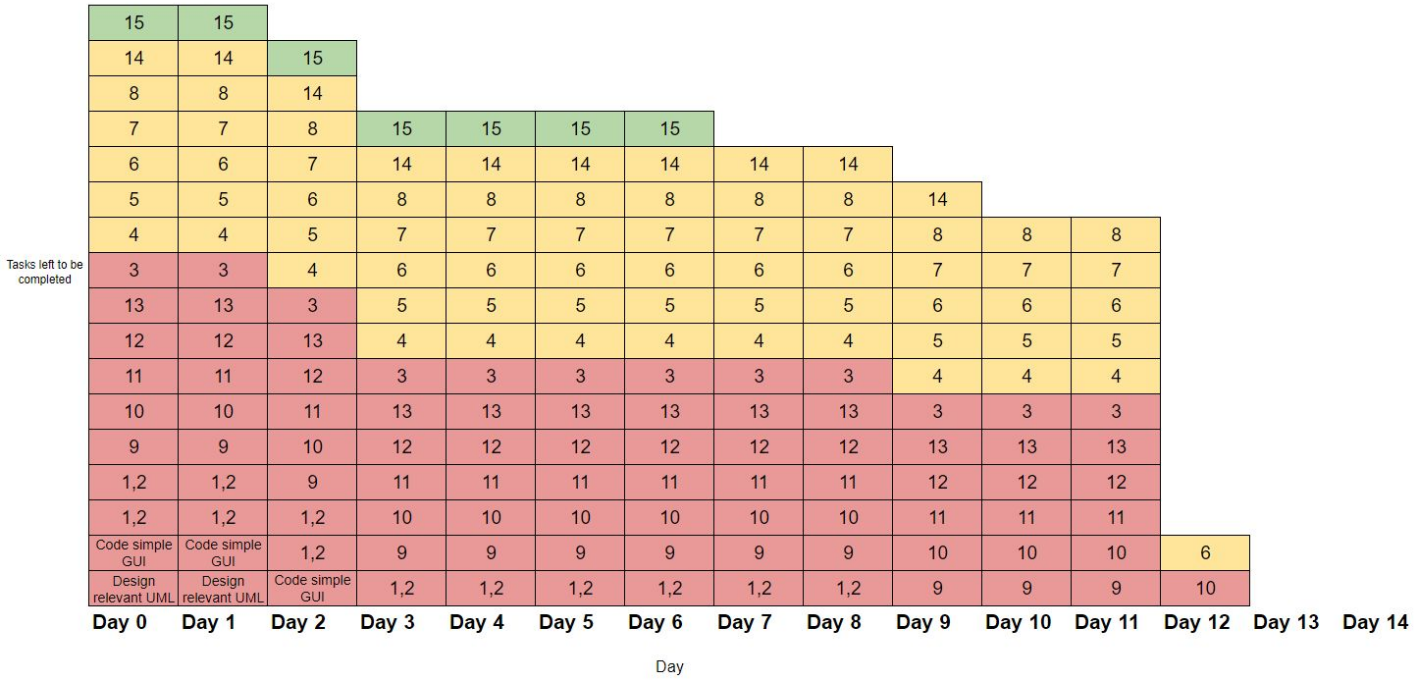
The process of developing increment 1 was slower than expected. After the second SCRUM meeting we all agreed that we have to rearrange our priorities and time management if we want to satisfy the customer`s wish (that the application can show at slightest one chart based on the input information) to begin with and then the user stories. This can be reflected in the burndown chart below showing our progress. As we explained in the previous increment (and should have thought about) quite a few of our tasks were very small and therefore on day 11 when we got round to them it didn't take long to finish them. This can all be observed in the chart below.

USER STORIES RELATED TO THIS INCREMENT (For reference purposes only)

1 - Loading CSV Files	As a Campaign Manager of a small business, I want to be able to load in the CSV files so that I can analyse the data from a clear dashboard.
2 - Detect invalid data in the CSV files when parsing	As a Campaign Manager of the Client I want to be able to avoid invalid data early on so that the performance metrics illustrated by graphs and charts are accurate.
3 - View number of impressions	As a Campaign Manager of a small business, I want to be able to view the number of impressions so that I am able to see the outreach of the advertising campaign.
4 - View number of clicks	As a Campaign Manager of a small business, I want to be able to view the number of clicks so that I can see how effective the advertising campaign is at drawing attention from the users.
5 - View number of uniques	As a Campaign Manager of a small business, I want to be able to view the number of unique clicks so that I am able to get a better understanding of how many people are interacting with the campaign.
6 - View number of bounces	As a Campaign Manager of a small business, I want to be able to view the number of bounces so that I can see how effective the advertising campaign is at keeping the attention of the users.
7 - View number of conversions	As a Campaign Manager of a small business, I want to be able to view the number of conversions so that I measure the success of the advertising campaign.
8 - View the total cost	As a Sales Manager of a small business, I want to be able to view the total cost of a campaign so that I can see how much was spent on the campaign.
9 - View the click through rate (CTR)	As a Campaign Manager of a small business I want to be able to view the click through rate of the campaign so that I can tell how effective the advert was at captivating users.
10 - View the CPA (Cost per acquisition)	As a Sales Manager of a small business I want to be able to view the cost per acquisition of the campaign so that I can determine the average profit of the campaign.
11 - View the CPC (Cost per click)	As a Sales Manager of a small business I want to be able to view the cost per click of the campaign so that I can see the average cost it took to gain the attention of a single user.
12 - View the CPM (Cost per thousand impressions)	As a Sales Manager of a small business I want to be able to view the cost per thousand impressions so that I can view the average required cost taken to gain the attention of a large number of people.

13 - View the Bounce Rate	As a Campaign Manager of a small business I want to be able to view the bounce rate of the campaign so that I can see the average number of people who were initially interested but lost interest soon after.
14 - The system should be able to display key metrics over time	As a Campaign Manager of a small business I want to be able to graph the data over time so that I can see trends and evaluate which time intervals are most effective for our campaign.
15 Ability of the user to control the time granularity of the time charts	As a Campaign Designer of a small business I want to be able to control the time granularity of the graphs so that I can evaluate the data and see trends over different periods of time to improve future campaigns.
23 - The software should be fast and responsive	As a Campaign Manager of a small business I want the software to be fast and responsive so that I can get my work done efficiently and become less frustrated due to waiting for results.

Increment 1 Complete Burndown chart



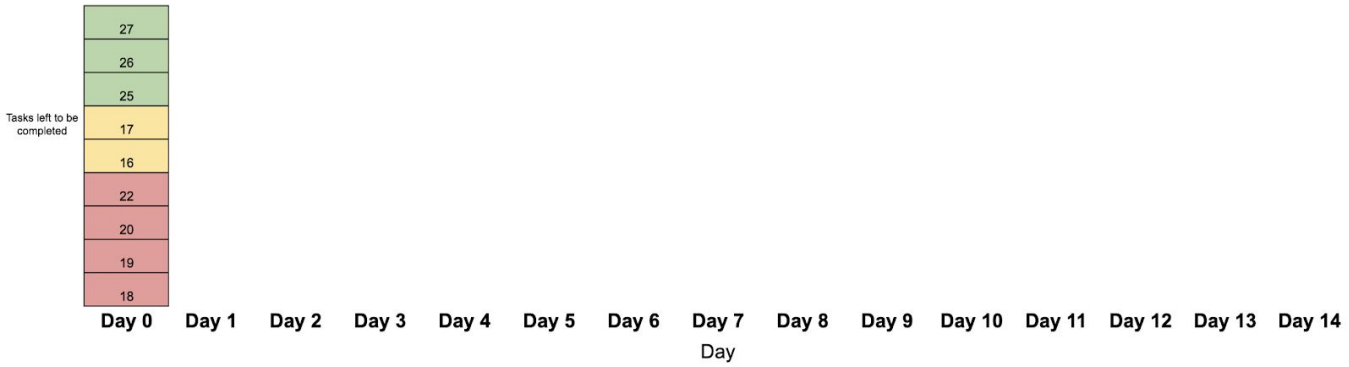
COPY OF USER STORIES RELATED TO NEXT INCREMENT

16 - Display a histogram of the click costs	As a Sales Manager of a small business I want to be able to view some data in a histogram format so that I can more easily evaluate the data using a visual representation.
17 - Display a histogram of the click costs (showing the distribution of costs) per thousand impressions	As a Sales Manager of a small business I want to be able to view click costs per thousand impressions so that I can more easily evaluate whether the cost incurred through a campaign is suitable compared with its number of impressions.
18 - Filter the metrics and charts by date range	As a Campaign Manager of a small business I want to be able to filter the metrics and charts by date range so that I can easily locate data about different periods of time.
19 - Filter the metrics and charts by audience segments	As a Campaign Manager of a small business I want to be able to filter the metrics and charts by audience segments so that I can see how different groups of people reacted to our advertisement campaign.
20 - Filter the metrics and charts by context	As a Campaign Manager of a small business I want to be able to filter the metrics and charts by context so that I can get specific information about the success of an advertising campaign.
22 - Define how a bounce is registered	As a Campaign Manager I want to be able to define how a bounce is registered so that I can determine the number of bounces based on what I deem to be a bounce.
25 - Display performance metrics per time of day or per day of week	As a Campaign Manager of the Client I want to be able to display performance metrics on a chart per time of day or per day of week to visualise the effectiveness of a campaign at different times.
26 - Save summary charts to an image or pdf file	As a Head of Communication of the Marketing Agency I want to be able to save summary charts to a file so that I can keep it for records and distribute them to interested parties.
27 - Print summary charts	As a Head of Communication of the Marketing Agency I want to be able to print summary charts so that I can keep hard copy records and for presentation purposes.

SPRINT BACKLOG FOR DELIVERABLE 2

Related User Story	Summary of Task	Estimated Task Size
User Story 18	Code functionality to filter the metrics and charts by date range	Medium
User Story 19	Code functionality to filter metrics and charts by audience segments	Large
User Story 20	Code functionality to filter metrics and charts by context	Large
User Story 22	Add functionality to the GUI to allow to user to define how a bounce is registered	Medium
User Story 16	Add code to the GUI in order to display a histogram showing click costs	Medium
User Story 17	Add functionality to the histogram in or-der to view the distribution of costs per 1000 impressions.	Small
User Story 25	Display performance metrics per time of day or per day of the week	Medium
User Story 26	Add functionality to the GUI in order to save the charts as an image or pdf file.	Small
User Story 27	Add functionality so the user can print the charts.	Small

Day zero burndown chart for Increment 2



Increment 2 initial task breakdown

We have begun planning our distribution of tasks for the next increment. While the first increment focused on base features like viewing the data, this increment will focus more on less fundamental but equally complex features.

The front-end team will focus on adding interface features for the users to filter the metrics, display the histograms and save/print the charts.

The front-end team consists of Palak and Dylan who will use pair programming to develop the program.

The back-end team will focus on the functionality of these features, given relevant data.

The back-end team consists of James and Vasi who will also use pair programming.

Finally William acts as a connection between the 2 teams, being able to work and help on both when needed.

Increment 1 focused on the base functionality of the application, where pair programming was more the focus. Now that is completed, tasks are easier to individually distribute because it is now more about adding smaller and more independent functionalities that do not rely as much on each other.