WeCasa

Charts DAR Report

Team HAGS JP

Team Lead:

Allison Austin

Team Members:

Githel Lynn Suico

Haley Nguyen

Joshua Quibin

Judy Li

Matthew Chung

Date Submitted: December 13, 2022

Github Repository:

https://github.com/githelsui/WeCasa

Charts DAR Version Table

Version	Description	Date
1.0	Initial DAR - Use Cases - Business Needs, Metrics, Technology Comparison, References	12/11/2022
1.1	Content Improvements - Added Data Volume metric	12/13/2022

Table of Contents

Business Needs

We plan to use a charting library for WeCasa's features that require specific data visualization techniques. Examples include the Budget Bar, the Circular Progress bar, and the Analytics Dashboard. The Budget Bar feature requires chart types that can depict percentages, and the Circular Progress Bar requires a circular chart to display one percentage (progress chart, ring chart, etc). For WeCasa Analytics, we plan to use a charting library for displaying our metrics (DAU, SPU, Most Used Feature, Error Rate, Retention Rate, Customer Service Ratings, Customer Review Ratings).

Metrics

Cost: WeCasa wants to reduce our technology costs to $0. So any chart library that requires an upfront cost is not ideal, and will be scored as a 0.
Chart Types: Charts should be able to appropriately display time series (graph, table, line chart, scatter plot, etc) and percentages (pie charts, bar charts, etc) to satisfy visualization requirements for WeCasa features.
Chart Views: The library should allow users to alternate between hour, day, week, and month views for all charts.
Cross-Compatibility: The dashboard library must work seamlessly with WeCasa's frontend framework, ReactJS and WeCasa's backend framework, ASP.NET Core. It should also have minimal issues running on WeCasa's primary browser, Google Chrome.
Volume of Data: Since we require charts to be able to display 3 months of data, we want our charting library to be able to render 10,000 data points at 40 FPS, which is the recommended frame rate for charting libraries (*LightningChart*, 2022).
Rendering Efficiency: Charts should update within 3 seconds. We will be comparing performance of different JS chart libraries for scoring in this category using rendering tests setup by other developers.

Technology Comparison

Scale: 1-1.75 with intervals of 0.25, based on how influential that metric is in our decision making. Higher numbers indicate more importance.
Scores: 0-1 with intervals of 0.2, based on how well they match our desired use case.
Total: Scores for each technology will be summed and multiplied by the metric scale.

Metrics	CanvasJS	Chart.js	D3.js

Cost - **1.5**	Free with student license, have to apply - **0.8**	Free, simple install using npm, open source - **1**	Free to download, open source - **1**
Chart types - **1.75**	Includes time series charts (bar charts, line charts) and percentage charts (doughnut charts, pie charts, polar charts, and radar charts) (*CanvasJS*, 2022) - **1**	Includes time series charts (bar charts, line charts) and percentage charts (doughnut charts, pie charts, polar charts, and radar charts) (*Chart.js*, 2022) - **1**	Includes time series charts (bar charts, line charts) and percentage charts (doughnut charts, pie charts, radar charts) (*Holtz*, 2022) - **1**
Chart views - **1.25**	Time series is customizable, charts can be updated dynamically using chart.render() - **1**	Time series is customizable, charts can be updated dynamically using chart.update() - **1**	Time series is customizable, charts can be updated by removing SVG element and redrawing - **1**
Cross-compatibility - **1**	React, ASP.NET, Chrome (*CanvasJS*, 2021) - **1**	React, Chrome, ASP.NET (*Microsoft.com*, 2018) - **1**	React, Chrome, ASP.NET - **1**
Volume of data - **1**	Can render around 10,000 data points while maintaining smooth 60fps (*Eberhardt*, 2020) - **1**	No data - **0**	D3 Charts can handle around 1,000 data points (*Eberhardt*, 2020) - **0.6**

Rendering efficiency - **1**	Rendered on canvas elements, 10x faster than SVG & Flash-based JS charts, takes 16ms to render 100,000 data points (*CanvasJS*, *2021*) - **1**	Rendered on canvas elements, 10x faster than SVG & Flash-based JS charts, takes 1000 ms to render 100,000 data points (*Chart.js Speed Test*, 2019), but the library contains 6 charts and is 11Kb zipped, which makes loading time and page impact low (*Charts*, 2022) - **1**	SVG-based, better performance with smaller number of objects, takes ~3ms to render 12,800 data points (*D3.Js Performance Test*, 2013) - **1**
Total	**7.2**	**6.5**	**7.1**

Recommendation

Although CanvasJS and D3.js have very similar scores, we will choose D3.js as our charting library recommendation as CanvasJS has a steeper learning curve and requires an application for free use. D3.js was also recommended by the client.

References

Beautiful JavaScript Chart Library with 30+ Chart Types. (2022, December 11). CanvasJS. https://canvasjs.com/javascript-charts/

CanvasJS Charts. (2022). Slant - CanvasJS Charts vs Chart.js detailed comparison as of 2022. Slant. https://www.slant.co/versus/4118/10578/~canvasjs-charts_vs_chart-js

Chart.js | Chart.js. (2013). Chartjs.org. https://www.chartjs.org/docs/latest/

Chart.js speed test. (2019). CodePen. https://codepen.io/createwithdata/pen/mvdQjg

D3.js - Performance Test. (2013). Tommykrueger.com. https://tommykrueger.com/projects/d3tests/performance-test.php

Eberhardt, C. (2020). Rendering One Million Datapoints with D3 and WebGL. Scott Logic. https://blog.scottlogic.com/2020/05/01/rendering-one-million-points-with-d3.html#:~:text=With%20Canvas%20you%20can%20expect,a%20subset%20of%20the%20data.

High Performance JavaScript Charts | CanvasJS. (2021, March 5). CanvasJS. https://canvasjs.com/docs/charts/intro/high-performance-javascript-charts/

Holtz, Y. (2022). The D3 Graph Gallery – Simple charts made in d3.js. D3-Graph-Gallery.com. https://d3-graph-gallery.com/

Javascript Graphs and Charts libraries | Comparison tables - SocialCompare. (2016). Socialcompare.com. https://socialcompare.com/it/comparison/javascript-graphs-and-charts-libraries

Performance in Data Visualization: why it matters? (2022, December 5). LightningChart. https://lightningchart.com/blog/high-performance-in-data-visualization/

Render Charts in ASP.NET MVC using Entity Framework & Database | CanvasJS. (2021, August 20). CanvasJS. https://canvasjs.com/docs/charts/integration/asp-net-mvc-charts/how-to/asp-net-mvc-charts-data-entity-database

Using Chart JS in asp.net. (2018). Microsoft.com. https://social.msdn.microsoft.com/Forums/en-US/719fcb35-b6d7-44d5-8310-18e9cb64468e/using-chart-js-in-aspnet