

Evaluation

Customer Relationship Management System (CRM)

Kieran Williams

Evaluation of The Technologies Used

The programming technologies used: JavaScript, EJS, HTML, CSS, and NodeJS.

The libraries used: Bootstrap and Express

The package manager: NPM

I benefitted greatly from using JavaScript as it allowed me to use the same programming language for both server side and client-side programming. Initially, I intended to do the server programming with PHP but that would have required me learning a separate language to the JavaScript knowledge needed to do the client-side programming. This saved me time as I did not need to learn multiple programming languages to achieve the same outcome. I was able to do this through the use of NodeJS. NodeJS is a run time engine that equips the language with web server functionality. I utilised NodeJS' file system module to add edit and delete files. However, the main use of the NodeJS runtime engine, was to set up a server on both my local machine and on an external server. It was very easy to start developing with NodeJS as it has very good documentation. Although NodeJS is relatively new, it has been widely adopted by many developers and so there was an abundance of help on the internet for the difficulties I incurred.

I enjoyed programming in JavaScript as the syntax is not cluttered and allowed for rapid development. By having such an easy to use language, I was able to develop algorithms in far less time than if I was to do it in other languages such as Java. However, due to JavaScript being predominantly a scripting language, many of its features are implicit. This caused me some difficulty as I would sometimes be unable to uncover the root of a problem effectively, due to me not understanding the underlying processes the language was undertaking.

JavaScript is a dynamically typed language and because of that, I was unable to specify the type of data a variable should store. Bugs were able to get into my code without me noticing as a result of the language's dynamic nature. An example of a bug that incurred, was one where I had a variable set to a String type piece of data. I then had the output of a function being saved to that variable, that was meant to be a string but due to me forgetting to cast the integer output into a string, the variable was storing an integer after the function had been executed. This bug laid dormant until that variable got accessed again. Due to the data held being an integer type, the other functions using that variable as an argument would result in errors. Tracking down the bug would prove very difficult in some instances where the function returning the wrong data type would not always get executed.

One of the biggest downsides to the JavaScript language for server-side programming is that it does not have the ability to make methods private and so when a class is instantiated all of its methods can be accessed. I thought up a way of utilising JavaScript's Symbol syntax to make private and public methods differentiable however it cluttered the syntax majorly. It would be nice if in the next ECMAScript version, there is a way to make methods private so that they are only accessible within the class. However, there is a better solution currently available. The TypeScript language can be used wherever JavaScript can be used and allows for both static typing as well as allowing methods to be made private.

By making use of the ECMAScript6's class syntax to take care of the prototype property in the object model. I did this to imitate an object-oriented approach to programming. This allowed me to increase productivity as my code became more organised and allowed me to reuse code. This syntax made it far quicker to develop.

One of my favourite features of the JavaScript language is that functions are a data type. This allowed me to store a function to a variable. The main benefit I gained from this was that I could pass a function as an argument into another function/ method. By doing that, I could use an anonymous function as a call back function within another function. I used this technique to call a custom function defined at a higher level of the code (passed as an argument), to get executed after all the other code within the function had executed. What made this such a useful feature was that I used it extensively when dealing with asynchronous code. It allowed me to define a function that would only get executed after the other code within the function had finished. An example benefit of this, is that it circumvents the potentiality of a variable being accessed before a result has been stored to it, this would cause errors within the program.

I utilised the EJS templating language to render html with dynamic data server side and serve that html document to the user on their browser. It was very easy to understand and use as it is almost identical to HTML with an extra few tags to allow for embedded JavaScript within the mark-up. The downside of using the templating language however was that it made the mark-up less readable and it would take extensive effort to modify in future versions of the system, impairing the ease for maintainability of the system.

The bootstrap CSS library made making the user interface of the web application a much quicker process than writing completely custom CSS files. I enjoyed using bootstrap as it provided my system very presentable and consistent styles amongst all webpages with just a content delivery network link at the top of every EJS file. It was also very easy to use as all CSS classes within the library are well documented and provided examples on how to use them.

Writing custom CSS on top of the bootstrap library was tedious and I struggled to keep styles consistent amongst all webpages. This was due to the inability to write reusable code in CSS files. Resultantly, this makes changing the styles in the future far more difficult as I would need to change every instance of a style amongst the site for consistency. As a result of the non-reusable code, the maintainability of the system's styling is now a much more time-consuming process. If I was to continue development of the system, I would want to refactor my CSS files to a CSS pre-processor such as SASS, so that I could write reusable code and properties.

Throughout the development of the system, I decided to use different code editors for different purposes. For the majority of the system's development, I used WebStorm. I really enjoyed using WebStorm as it had typescript plugins for all the newest technologies (things such as relevant: syntax highlighting, syntax suggestions, keyboard mappings, etc...) this increased productivity substantially. WebStorm lacked in the ability to mass modify documents quickly (things such as editing the CSV files storing test data) which would have slowed me down. To overcome this, I used the text editor Vim. Vim allowed me to modify documents very efficiently, I was able to enter a recording of an edit I wanted to do and assign it to a key mapping. From the recorded mapping, I could then execute that mapping on every line. This made editing a lot of repetitive data extremely quickly.

Evaluating My System in Contrast to Commercially Available Systems

I will be comparing my system to ZOHO as it is a widely used, currently existing online customer relationship management system. However, ZOHO is aimed towards enterprise organisations not towards small businesses/ freelancers. With this in mind, I will be comparing its essential features in comparison to mine as the target market for both systems is quite different. My desk-based research on the ZOHO system from my Investigation document will be used to compare and evaluate my system in contrast to it.

ZOHO has a way to create contacts in its system. The form for adding contacts in ZOHO is far more comprehensive than the form in my system. I feel as though my system captured all of the most important details of any contact and so it fits in with the lightweight nature of the system, I set out to create. One thing that I think my system lacks in contrast to ZOHO's data capture, is optional fields. This is due to the presence check validation occurring on all text fields. I overlooked this feature in the investigation on other systems. An example benefit of having optional fields would be that, if the user of the system didn't have for example the contact's phone number, they would need to enter a fake number as 'N/a' would not be a valid telephone number and neither would leaving it blank as it would not pass the presence check. To improve this in a future version, I would need to

disable presence check on some of the not so necessary fields and only run other validation methods on the field when data is present.

Another feature that ZOHO has in its data capture forms, is validation that occurs before submission of the data capture. I am assuming that this functionality has been obtained through the use of AJAX requests. In my system, data from a data capture is sent to the server after submission, where the validation processes are performed. After the data has been validated, if the data is valid, the user will get redirected to the next relevant page. However, if the data is not valid, the user is redirected to the same page, with the inputted data filled back into the fields, and a relevant validation message gets displayed. My solution to the validation works, but is less elegant, validation messages only appear after the user has submitted the form. This makes entering data more tedious and multiple validation errors could be thrown at the user all at once. By utilising AJAX, I could have had live validation messages appearing after the user has entered invalid data into one field, and then moved the focus to another field.

In ZOHO, it is possible to both import from and export to a CSV file. There is also a sample file so that the user can see the format in which contact details need to be entered in as. In my system there is only export functionality due to the target audience of my system only wanting to be able to export contacts. The problem with my system in this aspect is that it only exports the contact's details and not their notes and extra notes text area. In ZOHO, notes relating to a contact also get exported in the CSV file. In a future version I will need to add this functionality. To achieve this, I would need to link the contacts notes, extra notes and the file that contains the contact's details, all into one CSV file. This would save the user time and loss of data if the purpose of the export is to back up the data.

Both my system and ZOHO have a way to schedule tasks and appointments. However, in ZOHO there is a way to link a scheduled task/appointment to a contact, this is a feature that my system lacks. I feel as though this would add complexity to the system, and because my aim was to make the system as simple to use as possible, I think that the lack of this functionality is actually advantageous.

In ZOHO scheduled tasks are presented far nicer than in my system. ZOHO puts all scheduled tasks into a calendar page that displays all scheduled tasks in a very visual manner. In my system,

scheduled tasks are displayed in chronological order with the date and time of the task in the item along with the subject and message. I also have the tasks relevant to the current date put into a separate list to those that are for future dates. Although my system displays the scheduled data very differently to the way in which ZOHO does, it is still very organised and intuitive to use. Due to my lack of experience with CSS, creating a calendar user interface would have been a bit too much of a challenge on top of all the other functionality my system includes. I also like the way that both the input of a scheduled task and the display of the tasks are all on a single page. By having everything on a single page, it makes the system very intuitive to use by reducing the number of pages in the overall system.

The analytics page of my system is vastly different to the Analytics tools that ZOHO provides. I created my analytics tools from the bottom up with the user's input from the interviews in mind. I did this for two main reasons. Firstly, to analyse ZOHO's analytics tools, I would need to spend time using their system to generate enough data to properly see how they work. Secondly, ZOHO's target market is different to that of mine. The main features that the user's wanted was to be able to see how their conversions for the month were doing and to be able to visually see in a graph how the number of clients gained, and clients lost fluxgate over time. Through the use of only 3 input buttons and a time scale drop down menu, I was able to achieve this. I feel as though my system's analytics page is far easier to use than ZOHO's tools due to the simplistic inputs and outputs that it has. However, a downside of this is the lack of tools available for deep analysis in how the business is doing. I think that for the purposes of my system, the tools I provide are sufficient and by adding more tools, it'd add additional complexity which would diminish its ease of use. For that reason, my choice of limited yet useful tools, is aligned with the system I set out to create.

A feature that my system provides that ZOHO does not is the ability to encrypt a block of text. I think this separates my system from ZOHO because if a ZOHO account was to get compromised, all the user's data on their contacts could also get leaked and would consequently be very bad for their business' reputation.

One of the biggest difference between my system and ZOHO is that my system is open source and the user will need to set it up on a server themselves, in contrast to ZOHO's system that is a proprietary website in which a user pays a monthly fee for an account to use the system. I think this makes my system a great alternative to proprietary software as my target market is freelancers who

might have very low income when they are first starting. However, a disadvantage to this is that it requires the user to have some technical ability to get it working for them. This is a big downside to my application as most small business owners will not want to set it up themselves. To combat this, in the future I could sell a service to set up the system for the users.

Evaluation of My Final System

The aim of my system was to be simplistic and intuitive to use. I feel as though I achieved this very well by sticking to the design which was made with that in mind. The design was made to have limited links, good amounts of whitespace and for every page to have a title. Extending on this point, I feel as though I made very good use of space by grouping all relevant inputs and outputs onto a single page. Examples of this include: the scheduled tasks page where you can both input and view scheduled tasks on the same page. The individual contacts page is similar in that it has all features relevant to the contact (details, input and display of notes related to the contact, and the contact's extra notes). The analytics page does the same thing by having the inputs for client gained, client lost, and lead lost on the same page as the only analytics outputs (the graphs and the conversion rate display label). This benefits the user as it makes the system very organised and easy for the user to navigate to the functionality of the system they want to use.

In my opinion the HCI of the system is very good. One of the main reasons I feel this way is due to how I made the system give a relevant response to any action the user could perform in the form of a message at the top of the screen. Examples of this include when a value is searched for, the contacts list will return the matches in it and at the top of the screen it will say "Searched for: *the search value*" with a flash message in green reading " *no. of matches* matches ". When there are no matches found for a search value then a red flash message is outputted reading " No Matches Found ". This sort of communication with the user is carried throughout the system. The main benefit of this is that the user knows at all times what the system has done and what is currently happening. I think this is very important in all systems as a lack of it can make the user feel as though they don't know what is going on. For example, if the user searched for a contact that doesn't exist and nothing changed with no message to let the user know what happened. The user could think that the system does not work, when in fact it does work, the user just spelled the contact's name wrong and there were no matches, so all the contacts got returned to the list. So, I am very

glad I was able to include this into my system as it will make it a much more enjoyable experience for the users of the system.

I think the encrypted notes section in the individual contact page is a great addition to the system and I am very happy with how it works. I had the objective of not storing any passwords in any files and through using XOR encryption, I was able to meet that objective. The XOR encryption works by encrypting the input from a text area and displaying that data in the same text area. It also passed all the testing done on it in the testing documentation, and if I am happy with how it turned out.

I really like the solution I developed to get the conversion rate ratio output to work. This is because I came up with an elegant solution to getting the ratio to get displayed in its lowest form. I achieved this by creating a small recursive algorithm that finds the greatest common divisor and then both sides of the ratio get divided by the greatest common divisor. I like it because the recursive method is very contained and that I was able to develop it based off of a mathematical algorithm developed by the mathematician Euclid.

Another feature of my system I like is the search algorithm. It is a binary search with the addition of finding multiple matches by looking either side of the initial match until it finds no more matches. This works because the array needs to be sorted before the array can be searched with the binary search and a result of this is that all duplicates are next to one another. I think my solution of getting the duplicate matches is very efficient as it has a worst case of $O(\log(n))$. Using an efficient search algorithm was a must as searching is used a lot. For example, every time a user clicks a link to a contact on the contact's list page, that contact's data needs to get retrieved by searching the data store for the contact's ID as a search value to retrieve the record.

I also feel that my choice of storing data in text files was a good choice as it allowed me to showcase my ability to develop algorithms to manipulate data without the need to rely on a structured query language and a database. My system also carries very well as all data is stored within the project file system. As the project is going to be made open source, it needs to be as easy for the user to set up themselves as possible and by having the data stored in csv files, the user will not need to set up a database on a server on top of also having to host their site.

There are many improvements I would make to my system if I was to make another version. Firstly, my system has validation messages that appear after a data capture has been submitted to notify the user on why their data input was not valid for the system to fully process it. However, the message only gets displayed to the user after submission and if there were multiple things that got flagged as invalid data input, the validation shows multiple messages in one string. This validation message can sometimes be quite long and hard to follow (can be seen in the Testing documentation). To overcome this, I could have validation messages appearing in real time after the user has entered invalid data into one field, and then moved the focus to another field. This could be implemented using AJAX requests to send a post request to the server with the inputted data and the field name that has just lost focus. The server could validate that field and then send a validation message back to the client side if anything was invalid with that field. By doing this, data could be changed before the final submission and would avoid a big validation message after submission of the data capture form.

One of my objectives I neglected to meet, was to set up an SSL on my website. The issue with this is that it leaves users of the site susceptible to man in the middle attacks where all data transferred to and from the website could be easily seen by the perpetrator. One of the main reasons for not setting this up was because the intentions of my system changed. Initially, I intended to have users that would have accounts to one website. However, my system is now an open source project that users will have to set up and a consequence of that is that it'd be very difficult to set up an auto-configuring SSL for the system. To work around this, in the documentation for setting up the system, I will show how the user can set it up themselves if they wish for their system to have it.

Another big downside to my system is the sorting algorithm I use. The algorithm is a bubble sort which has a worst case of $O(n^2)$. My sorting algorithm is used substantially in my system for both sorting and searching as the binary search requires data to be sorted first. In my Testing documentation, under the scalability testing section, it can be seen that my system still works even with 1000 records however it took around 2 seconds for the page to load with that many records. This test was done with a good internet connection with 75mb/s download speed, but if the user was to have bad internet speed, this additional 2 second wait for ever page to load would begin to become noticeable to the user and deter them away from wanting to use the system. A solution to this problem would be to use a sorting algorithm with a faster average and worst-case time

complexity. A good suggestion would be to use the Quicksort algorithm as it would be easy to implement and it has a worst case of $O(n \log(n))$ which is faster than a bubble sort.

My system's source code and documentation has not yet been uploaded to GitHub. This has not been done yet to avoid questioning of my coursework being plagiarised. After my coursework has been marked, the coursework will then be submitted to my GitHub profile so that the users can easily clone the repository and have access to the system's documentation. By doing this, other developers will be able to also work on the system and submit pull requests for their changes. This will also allow for developers to use my general-purpose system as a baseline for a more niche CRM they could develop it into.

In the next version of my system, I will need to work on the search bar functionality that is in the contact's list page. The search bar works well but there it isn't as natural to use as other real-world search bars. In most professional real-world systems, minor spelling mistakes, and search values that contain data that is spread into multiple columns would usually all get returned without a problem. My system is able to search for all values in the contacts list, but it has to be spelled correctly (the search value being upper case, lowercase or capitalised does not impair the search at all), but it can only search through individual columns. An example of when it would not work is if there was a search value of 'James Donald' and that name existed in the file being searched through, no match would be found because the name in the data store file is split up into separate columns for first name and last name. In most systems that use an SQL database, they can utilise the query syntax as well as wildcard characters to get the search to be less demanding on accuracy in a very efficient way. However, because my system uses CSV files to store the data, I would need to implement a smarter searching algorithm that is able to split the search value into separate search values and then perform multiple searches with them separate search terms and all records that have either of the search terms in them should get returned.

The biggest drawback of my system is that I have written almost all of the algorithms used to process data myself. I have relied on minimal logic related libraries so that I can showcase my ability to design and develop algorithms to solve problems and carry out tasks. The issue with this is that not every method I have written has been put under rigorous testing unlike how popular libraries would and so, there is a higher chance of remaining bugs and efficiency issues in my

system. To overcome this, in future versions of the system, I will refactor my code so that it relies on libraries and standard functions of the language instead of my own implementations.

Evaluation of My Own Performance

If I was able to go back to the start of the project and do it all again, I would take a different approach to the coursework in particular the development of the system. One of the biggest downfalls to the development of my system was that I hadn't researched and experimented with the technologies I intended to use enough before embarking on the design and development of the system. There were a few problems with the approach of learning the technologies as I went along and develop the system. Firstly, I anticipated on doing a lot more than I was eventually able to do. For example, I initially intended to develop the ability for users to be able to create accounts, log in, and for there to be the ability of multiple users simultaneously using the system with their separate accounts. This proved to be quite hard to develop in JavaScript as it is a single threaded language. Although it was possible it overwhelmed me at the time of developing as I also had many other aspects of the system to think about and how they would all interact with one another. To make it easier for me to be productive and get work done instead of just trying to work out how I'd get my system to work together, I decided to neglect the user's aspect of the system as there was no real need when all the other objectives not related to the users could be met. It's worth noting that I could have easily achieved this functionality by using a library to take care of it all for me but as that would have gained me very few marks if any at all, I decided to not go down that route. Secondly, it was very hard to plan ahead and design the system when I was unsure on if I'd be able to do it. For example, I initially thought that writing to a CSV file would be impossible because there was no built-in support for it in the language or even in NodeJS. There was also very little help on the internet around the matter. However, after a bit of experimentation, I realised that I could read and write to a .txt file and then change the file extension after doing so. The issue was that I was under a lot of stress at this point because I had lots of developing to do and had no clue on how to do even basic stuff in the language or in NodeJS. These problems could have been easily circumvented if I had just experimented with the technologies before delving into the project with very little knowledge.

I don't think that I made the most out of the time I had to do the coursework project. There was plenty of coursework lectures and workshop sessions offered through the year to which I only started attending in the last few months before the deadline date. This partially due to me preferring to do work on my own without anyone else around to distract me but I would rarely utilise that time to get work done and as a result, in the last few months, I have had to really work hard to try and complete the coursework within the remaining time and so the stress was entirely self-inflicted. My biggest regret this year surrounding this project is not starting it and working hard on it sooner. during the workshops I would have had the opportunity to ask for advice on any problems I was facing but by not attending them I was forced to work through problems myself which I feel benefitted me greatly as it gave me the mindset that I don't need to rely on other people to solve a problem for me if I just spend enough time thinking about it and trying things out. This is a skill that I feel many people are likely to lack and so I have actually benefitted from not asking for help and advice frequently.

Looking back, I could have utilised my time a lot more effectively if I had started the project earlier on. I started properly working on my coursework at around January, meaning that I had done relatively nothing for four months which had an impact on the stress I felt when it came to completing the documents in time for the deadline. The worst stress I experienced was when it came to actually developing the system; I had only a few months left before the deadline and had very limited experience in the technologies I intended to use to develop the system. However, this stress forced me learn very quickly and apply that knowledge straight after learning it. I found the most efficient way to do this was to go directly to the documentation for any of the technologies and learn from there. I noticed that if I was to try and find solutions to problems on Stack Overflow or YouTube, they'd rarely be helpful as the problem they'd be addressing would often be different to my problem. When trying to scour the internet for a solution, I found that I would waste more time than when I would just read through the documentation and experiment with the code to come up with my own solution. If I was not under as much stress to get my system done with enough time left to do the remaining documentation pieces of the coursework, I think I might have just tried looking for tutorials for every solution which wouldn't have developed my skill to think through a problem logically. And as a result of my stress to get things done in the most efficient way possible, I think I actually benefited at least when it came to programming.

In the limited amount of time I spent developing the system, I learned a lot about programming, the internet and other networking related topics that helped me to develop the system. I already understood the basics of Java and object-oriented programming and so when it came to programming with JavaScript, it wasn't too much of an issue because JavaScript's syntax is similar to C as is Java. I learned how to develop with a dynamically typed language as JavaScript is one, it meant that I would have to be constantly cautious of the type of a variable as it would be implicitly declared. However, there was a great benefit to using the dynamic nature of the language when it came to returning data of different data types from a method dependent on the conditions met in the method. I learned about and used mark-up languages such as HTML, LaTeX, and XML for defining the structure of documents and data. I also learned how to write and use JSON as a means of transferring data to and from an API. I also learned how to style documents with the declarative language CSS. To save time, I did use bootstrap which is a CSS library and I feel as though this saved me a lot of time in making the system look presentable and modern. I had to learn about how API's worked so that I'd be able to transfer data from the server to the client's machine. Learning about API's was quite easy but when it came to implementing them into the system it became a much bigger challenge and as a result, I decided to look into easier ways of transferring data to the user. I then came across the Express framework to take care of all HTTP requests to the server, this made it far easier to generate a webpage on the server and send it to the user. I really enjoyed learning about security best practices and how to make a website as secure by researching the OWASP top 10 which is a the top 10 ways websites are usually susceptible to attacks and it gives all the precautions that should be taken as well as the testing that can be performed to check that it is secure. I also enjoyed learning about different ways a browser can store data such as cookies and sessions, it was interesting to learn about when to use one over the other and the security issues surrounding them.

I had a lot to learn to be able to produce my web application. However, I am happy that I did so because if I was to have done a project in Java, I would not have learned anywhere near the amount I was forced into learning by creating a web application. There is also a movement from desktop to the web which will only get more and more prominent over the years and so I think it was a good use of my time to get familiar within the web space and using web technologies seems as I was already fairly comfortable programming in Java. There is still so much for me to learn about web development and because I enjoyed creating the system and using the web technologies, I am

looking forward to learning more about web development such as front-end frameworks such as React to produce the user interface of a web application with greater efficacy and consistency.

I benefited greatly from being able to produce a prototype to which the potential users were able to use and give me feedback on. It was particularly useful as it allowed me to then redesign the system to a more realistic final system that I knew could be completed. The issue before the Post Prototype Refinement and Design was that I had designed a system without even understanding how the technologies worked together or what functionality the different technologies had to offer. By having the opportunity to further consult with the potential users of the system, I was then able to redesign the objectives to what the users needed but with the consideration this time with what is actually possible in the timescale. The biggest change I decided to make to the system was to remove the users as it made developing the system substantially harder and it did not interfere with any of the system functionality objectives so it was a good choice made possible by having the opportunity to showcase the system and have the opinions from potential users on my intentions to change the system as well.

If I was to do a similar project in the future, I would definitely ensure that I was more organised in understanding all components of the project. I would also try to spend more time experimenting with the technologies before designing the system; it is very hard to predict how long something will take to do and the difficulty of it when you have no idea on how it will be done. Next time, I would also utilise the time more effectively by working on the project during free time, coursework lectures and during workshops, even when the deadline is a while away; to avoid stress close to the deadline of the project.