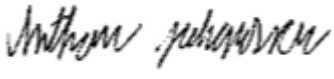


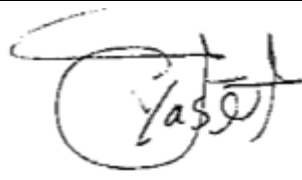


Design Documentation

Group name:	New Shadow Corp	
Member Name	Signature	Area of Contribution
Antho Johansson 6770097		Added the currency chart and backend C# code in the web service to create chart datapoints , research how to implement and use web service, project management
Mohsein Abd Elkadir 6099231		Original idea ,creating web service and referenced it to our project, made function in web service to calculate the result by using live currency rates from external web API, validation of fields
Nduka Ernest Odum 6717966		Added Combo boxes for currencies, added library for currency codes(e.g. USD,EUR etc), , idea of adding a currency chart
Yaser Alsehiman 6110709		Wireframe mockups, startpage and buttons/textboxes, idea of connecting our web service to other existing web services for live currency rates

Start:

My Drive - Google Drive x Namnlös presentation - G x https://cumoodle.coventry x Currency converter - Moq x Convert page x

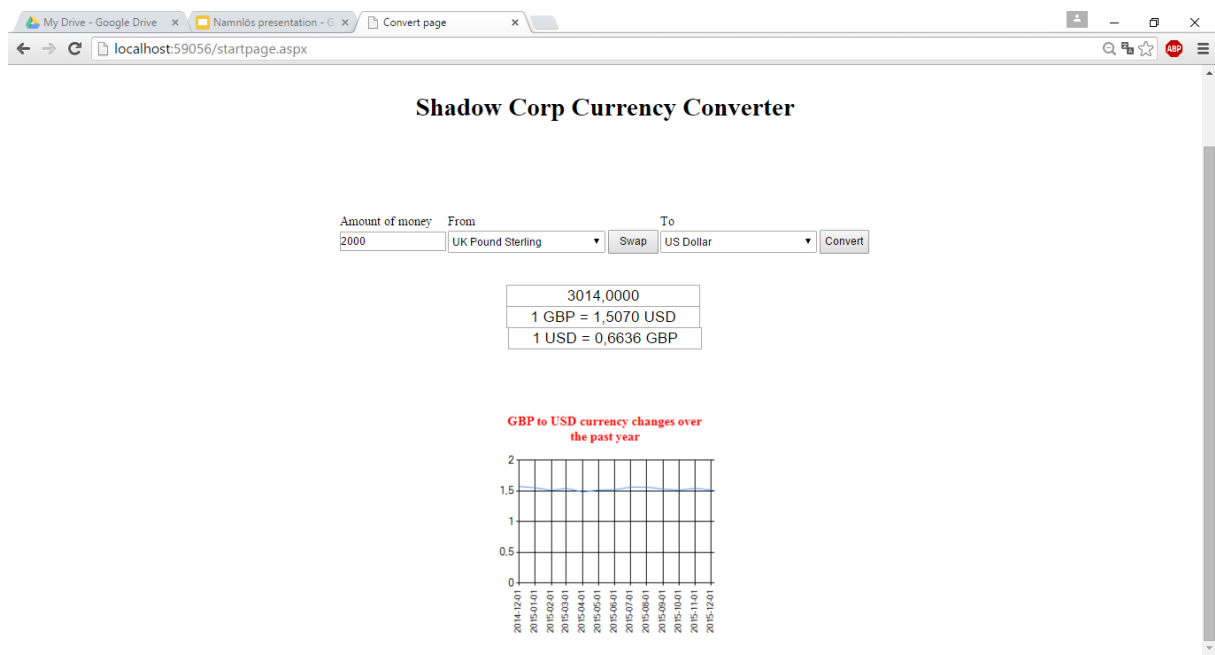
localhost:59056/startpage.aspx

Shadow Corp Currency Converter

Amount of money From To

1854 UK Pound Sterling Swap Euro Convert

Result:



We have made a currency converter which can convert one currency into another and display the historical growth or loss of the currency value in a line graph. Most currency converters are very easy to use and show the converted amount in a simple way, therefore we chose to design our currency converter this way as well.

Instead of filling the entire page with redundant objects we decided to put as few objects as possible in the design, only the necessary textboxes and buttons have been added to the design. The entire design area has been centralized to make it easier for the user as well.

When the web page was implemented the main focus was to keep the .aspx file as clear as possible. In most cases it was possible to add design objects to the .aspx file generically through the code in the c# file, i.e. populating dropdown boxes or charts.

We have implemented the following objects:

- A textbox for entering the amount to convert into the other currency
- Dropdown boxes to pick the from and to currencies
- A button to swap the “to” and “from” currencies
- A button to start the conversion between the currencies
- Three textboxes displaying the conversion results
- A line graph displaying how the “from” currency has increased/decreased in value compared to the “to” currency

The tests we performed show the expected results which means that our web application works as expected. It isn't as fast as the existing currency converters but it is most likely due to the lower processing speed of our laptops.

Link to wireframe design:

<https://moqups.com/anthon92.johansson@gmail.com/ktpFyxR8/p:af3708a1c>

(Also found as pictures below).

Development challenges

In the beginning the biggest challenge was to determine the size of the project and what kind of application to create. Most of the group members were inexperienced when it comes to web development and therefore we chose to make a currency converter.

In the start the group had problems understanding the difference between a web application and a web service, this affected the work speed the first few weeks since we did not know what we were creating.

After the first weeks when all the labs had been performed and all the definitions were clear we spent a lot of time reading about how to create and consume a web service. This was very challenging and took longer than expected. Many of the texts and examples that were found always seemed to use java script or php to consume the service, something none of us knew anything about.

When the web service was finished and all of us knew how to use it, the next problem was to replace all the static html fields. Our first thought was to replace them with a database source but this was also sort of advanced so we decided to use an xml file first. This xml file was replaced by an Asp .Net library instead since it was easier to use and offered the functionality we requested.

To add more functionality to the web application we added a currency chart the last few weeks. This was not really challenging since the web service was working already and most of the coding was done in c# code. The difficult part was to find a web API that was able to deliver historical currency data, the one that was found and used is very limited in functionality but it is enough for our needs.

Conclusion

This project has taught us a lot and given us a good understanding of how modern websites work. The term web service was previously unknown to us before this but now we fully understand its purpose and how it works. Learning how to actually build a web service was quite challenging and figuring out how to incorporate it into our web application took some time. It would had been nice to learn more about java script and the technologies similar to it (JSON, jquery) since it is so useful when you develop things for the web.

We found that working on a project of this small size with a group this large was quite difficult, the main problem was including everyone in the work which was difficult because of the project size.

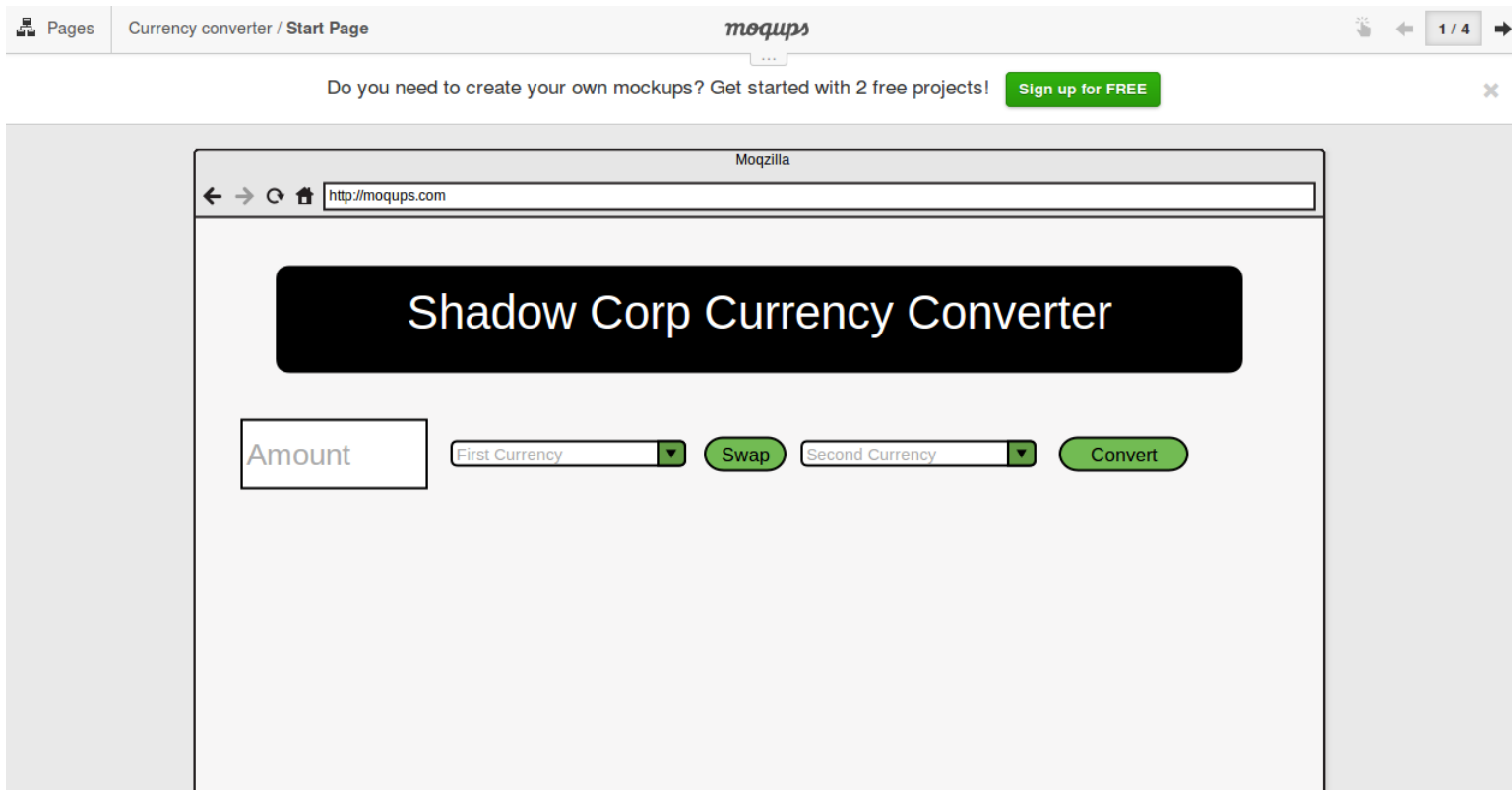


Fig 1. Landing Page

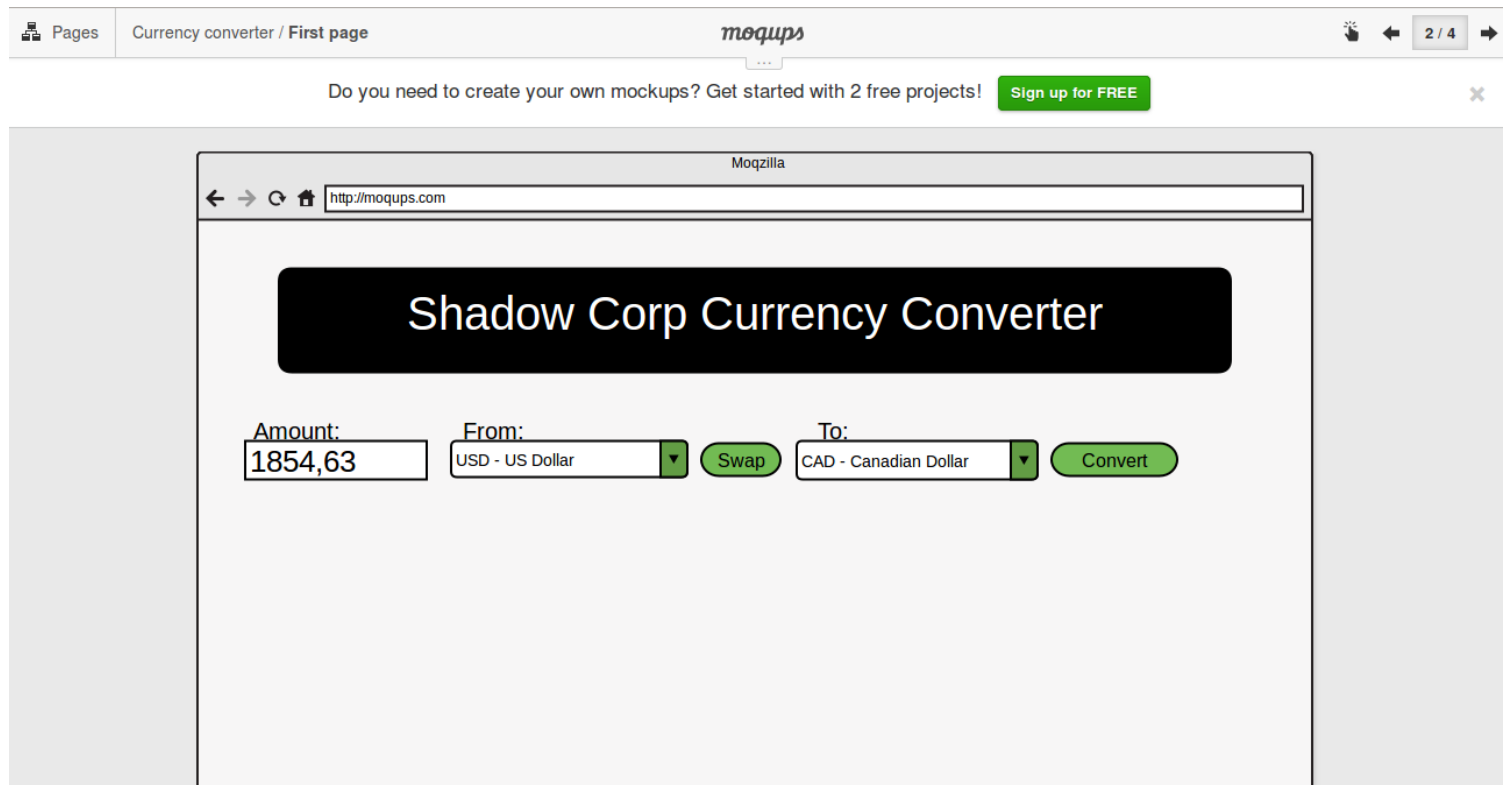


Fig 2.Amount and currencies entered.

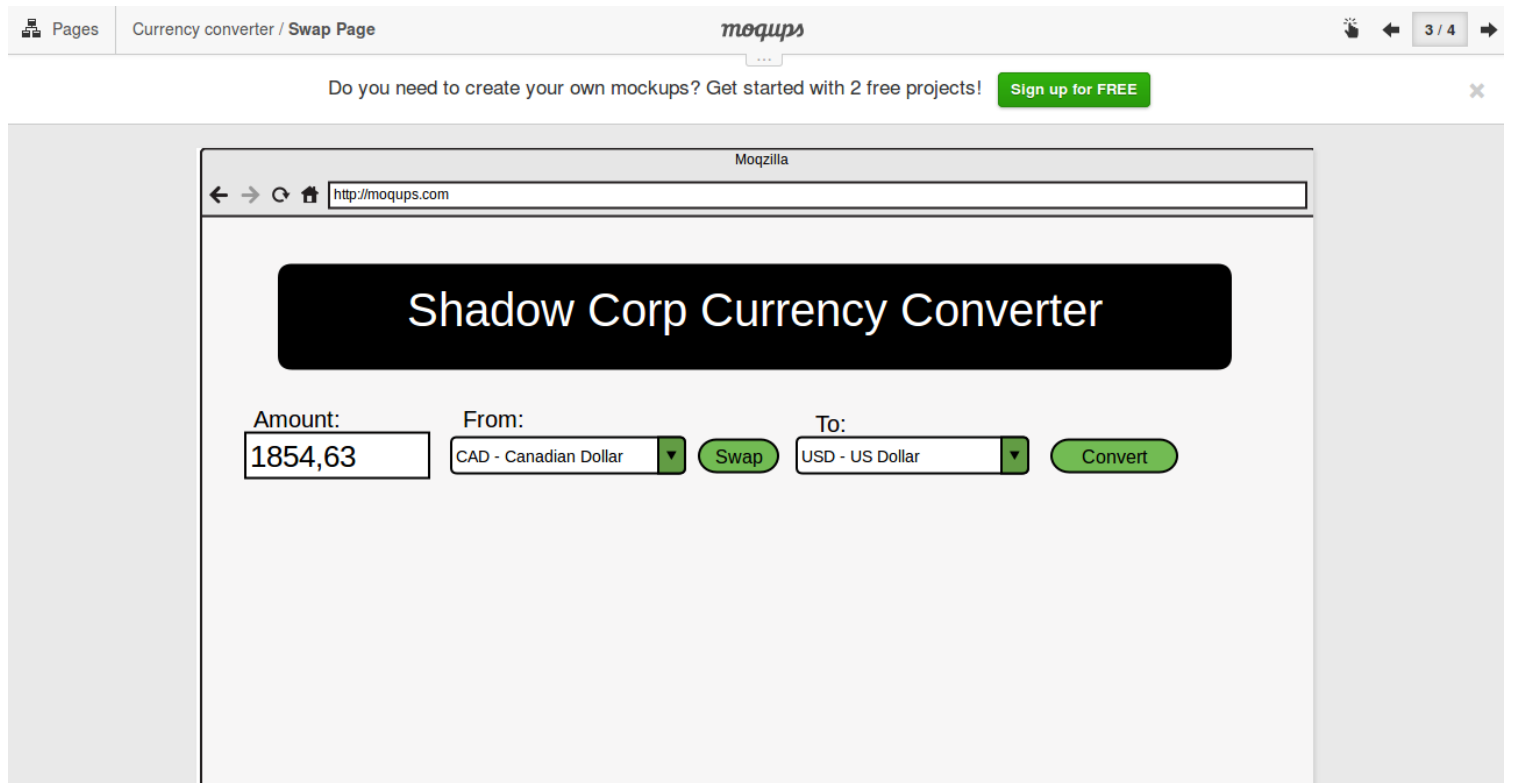


Fig 3. Currencies swapped.

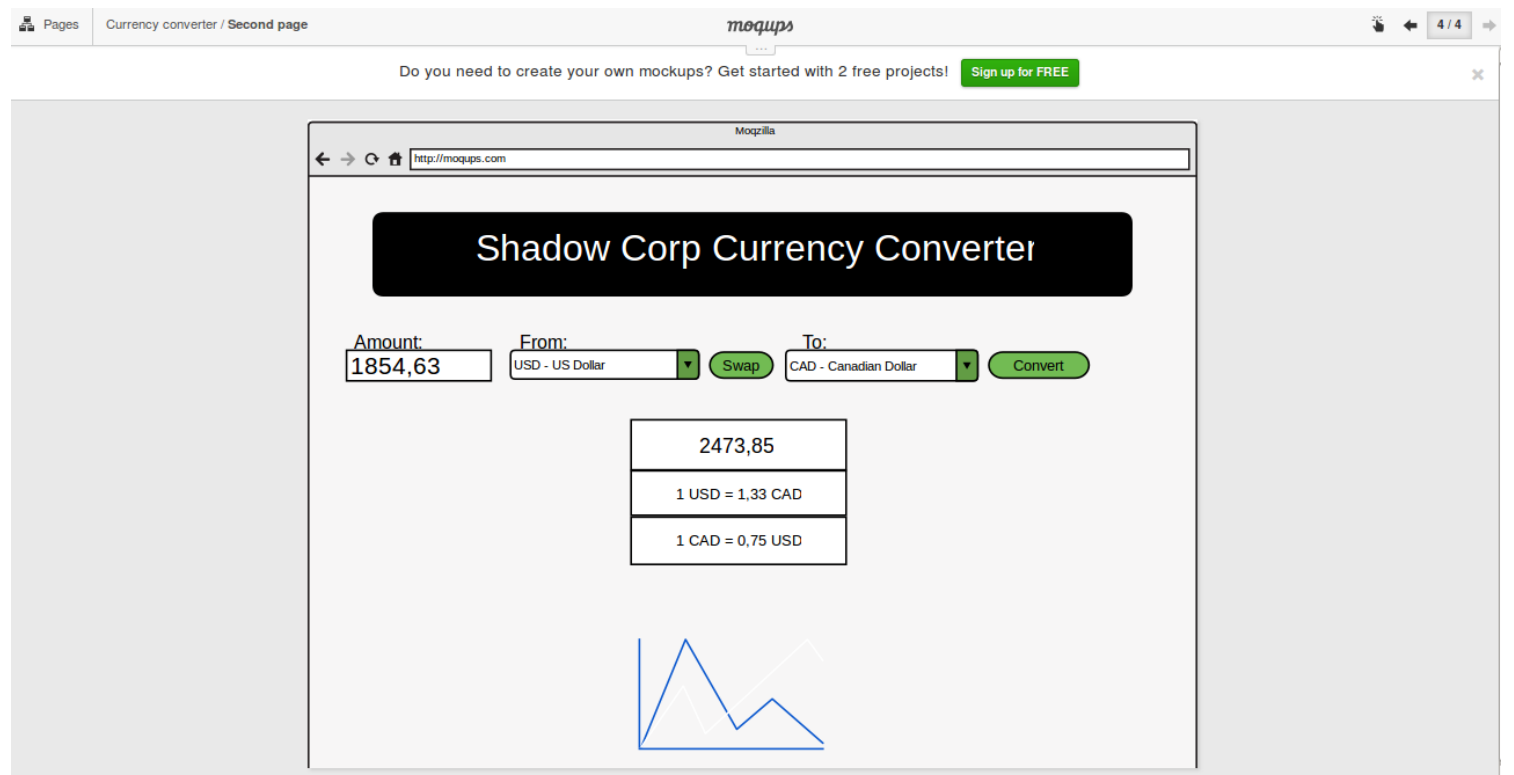


Fig 4. The conversion result.