Election Simulator

Client: **eCivix**

Team: **Cerebero**

Team Members:
Frederick Ehlers 11061112
Jacobus Marais 15188397
Rikard Schouwstra 15012299
Victor Twigge 10376802

Demo 2

Table of Contents

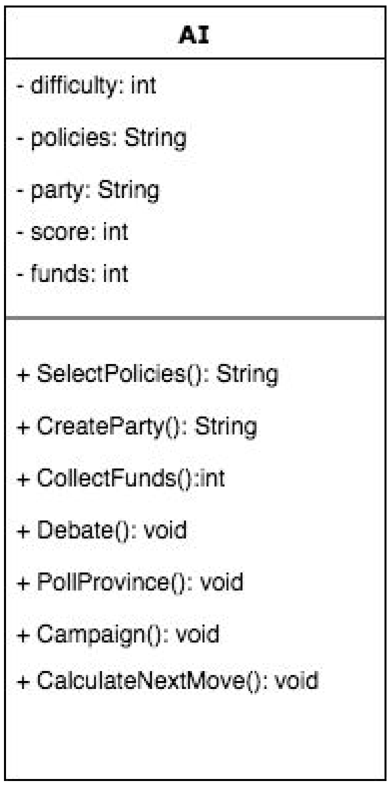
Diagrams

Technologies used

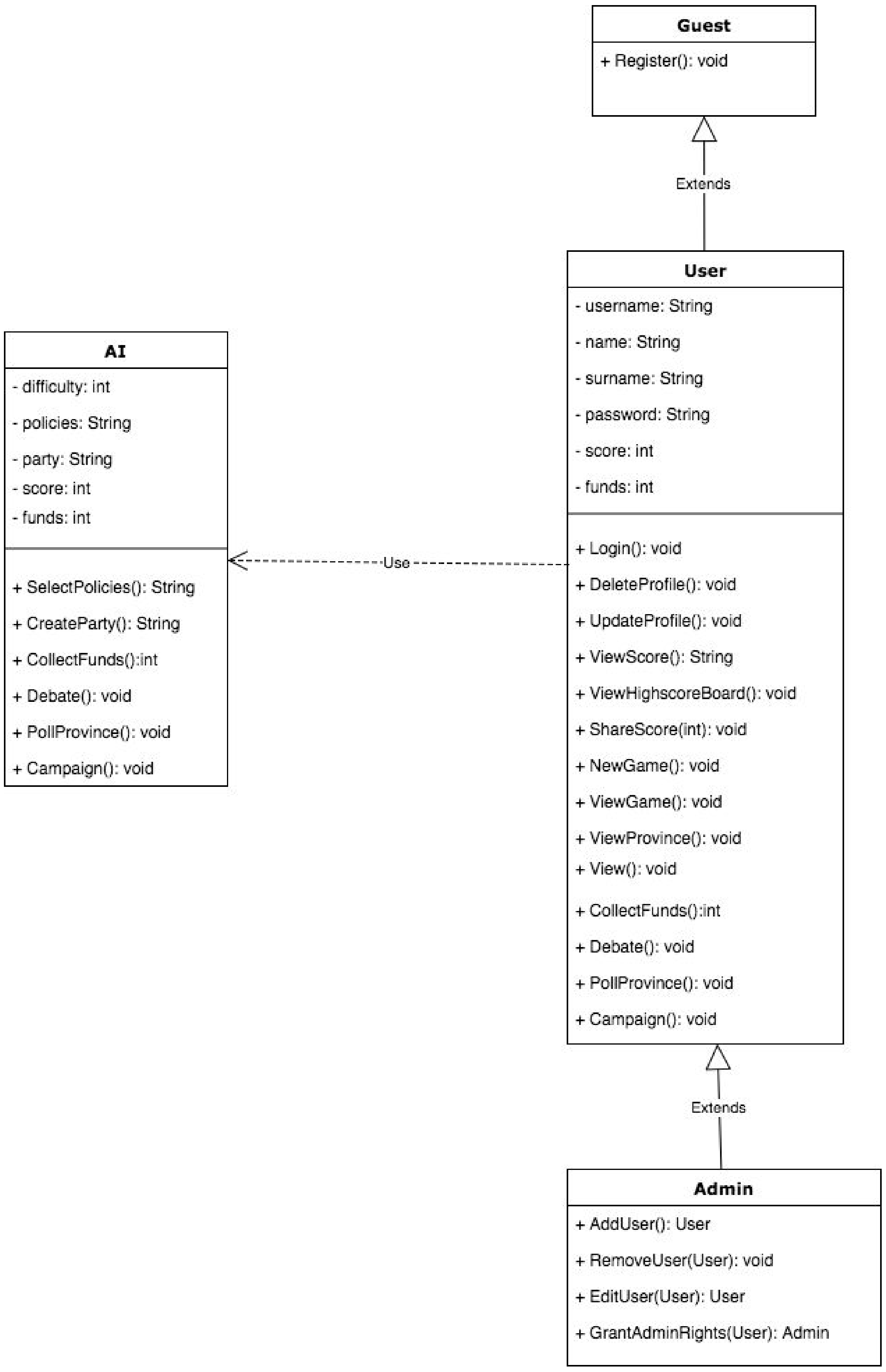
- Server:
- Client:
- Database:
- Artificial Intelligence:
- Design:
- Additional technologies used:

Diagrams

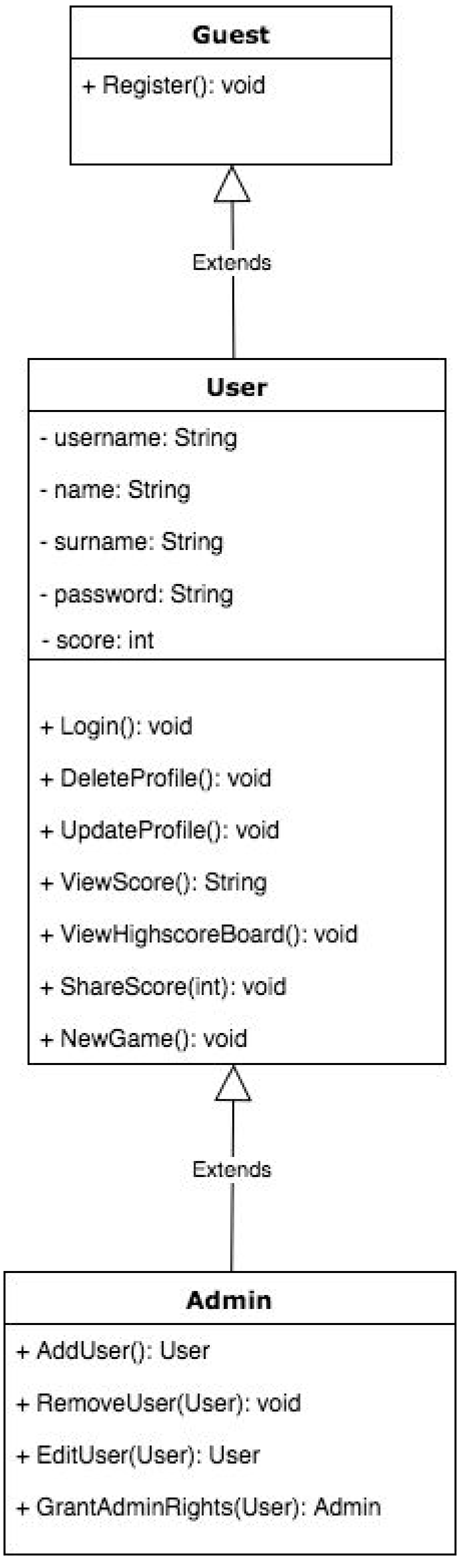
AI Domain Model



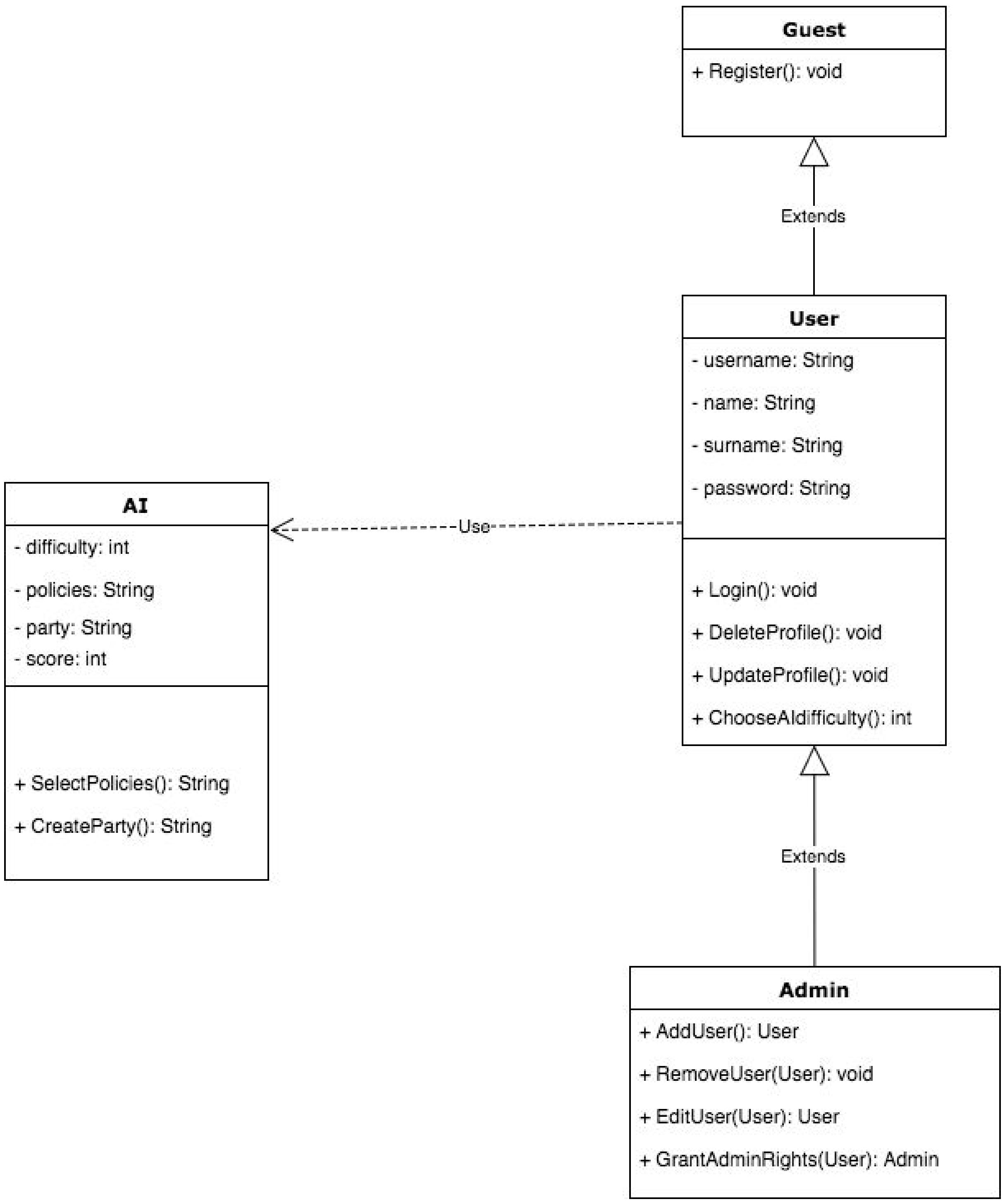
Gameplay Domain Model



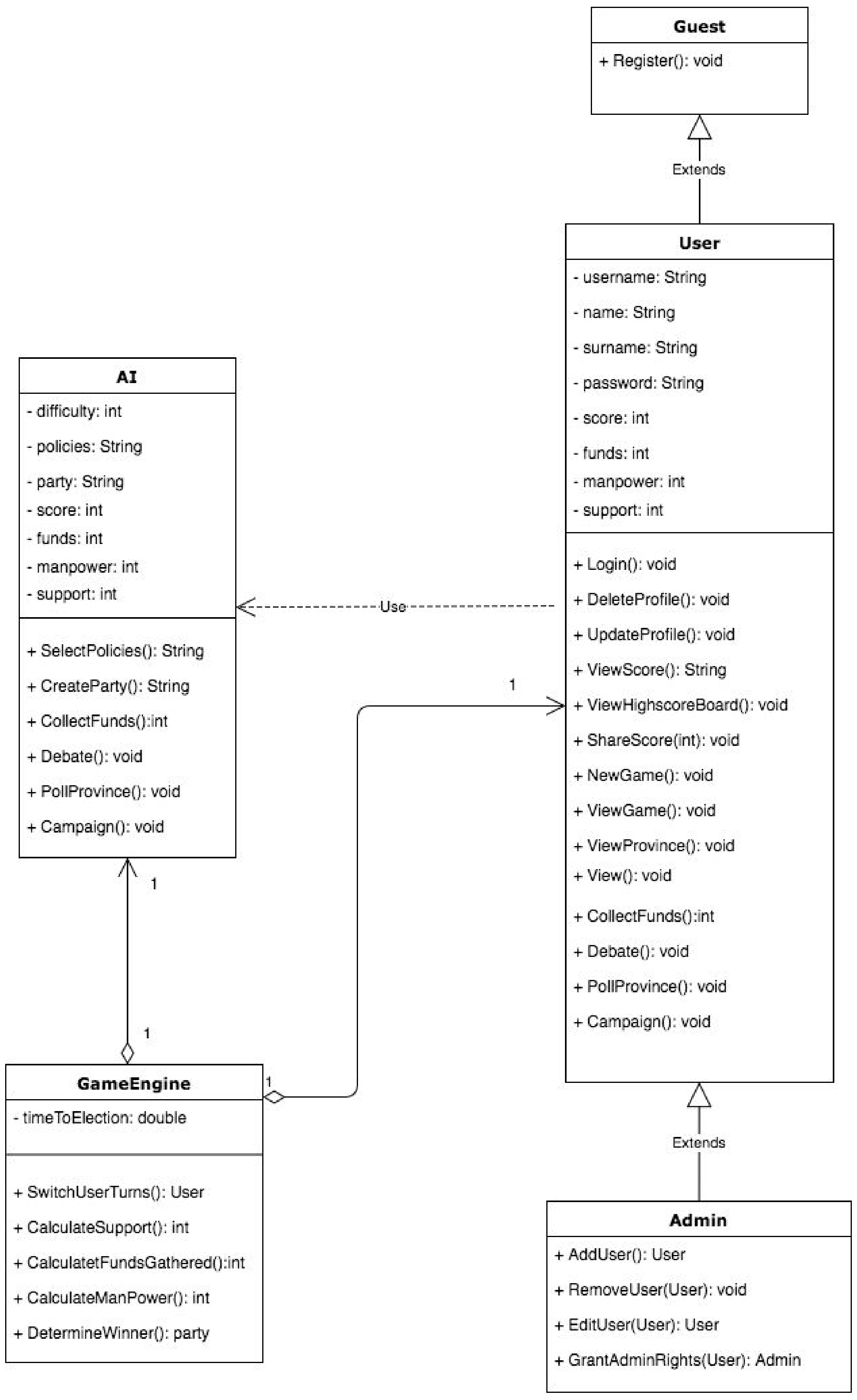
General UI Domain Model



User, AI domain model



Overall System Class Diagram



Technologies used

Server:

Node.js

We have chosen Node.js as the technology to use for our server as it is very fast and there are a lot of free libraries that we can use for it, there are a lot of people who use it so there is a lot of support for it. Because it uses javascript, all of our members can help with coding the server as we have all worked in javascript before.

Client:

Unity

Saves time
Unity allows us to focus our time on making the game and not to worry about spending our development time on creating tools. It enables us to focus more on gameplay and the user experience, rather than the technology itself.

Greater creative freedom
It really helped us focus on creating compelling experiences instead of spending critical time building engines.

An efficient engine
Unity is very suitable for quick prototyping. The technology allows us to work at speed on our ideas.

Modularity
It is also ideal for modularity because the Unity project can easily be worked on separate of the other technologies.

Cross platform
The technology helps keep your options open when identifying target platforms, and can make developing for new hardware relatively cheap and easy to convert, particularly compared to the same process on custom technology.

Lots of documentation
Unity also provides a wealth of resources explaining just how the technology works, while a support team is always on hand to answer questions, however complex.

Reuse code and assets from different scenes
Unity also provides us the opportunity to reuse certain design elements in all of the scenes with minimal effort, which also enables us to keep the design constant.

Minimize workload
The Unity Asset Store also offers cost and time savings. Assets and tools that enables us to develop quicker and allows us to focus more on key areas.

Affordable licensing
The Personal Edition of Unity that we use, is completely free.

Angular

DataBinding

The fact it can provide data binding allows for dynamic data feedback to the user anything from error reporting to dynamic data from a database.

DOM Manipulation

Clean and lightweight it is clean and lightweight due to its structured MVC approach of implementation

Simple manipulation makes it attractive as it is a versatile technology that is very power that will allow DOM manipulation happen easily

MVC

Angular allows us to use the architecture MVC and it does it smoothly

Simpler and less code

With the MVC approach and how Angular is implemented it promotes a very structured approach. The ease of DOM manipulation and data binding that angular promotes it will allow for simpler, more readable code.

Unit testing is possible

Database:

PostgreSql

Reliable and stable

Easy to use, reliable, stable and fast with the use of JSON objects

Integration with our other technologies

It integrates seamlessly with Angular thus allowing us to easily access and change the data as required.

Scalable

It is scalable, so when we have a lot of data requests and large datasets being sent around it will do so quickly , reliably and efficiently.

Artificial Intelligence:

Python

We have chosen to use Python to code our AI for the game in. We have done this because it is free to use and with all of its calculation libraries it is one of the best languages to code an AI in. We have started doing tutorials in it and have seen that it really does simplify the whole AI process. We thought of using Keras.io to implement a Neural Network for our game. We are still considering some game trees as our game playing agents and won't decide completely until we start coding on the actual game rules, then we will be able to fully grasp what is required and thus make the best decision. Until then we continue doing research into current technologies used for AI implementation.

Design:

Photoshop
It’s a powerful image editing program with advanced editing features and an easy to understand interface. We chose to use Photoshop for it’s great support and powerful features as well as the fact that we have experience with it. The goal was to create most of the graphics on photoshop, export it as images and import it to Unity. This adds to the performance of the game as Unity doesn’t need to render the graphics every time the game is played.

Testing:

Travis.CI
We have chosen to use Travis.CI to help with our testing because everytime that we merge a branch into our development branch, then it runs all the tests that we wrote and lets us know if any of our tests have failed before we merge it in, thus we do not have to run the tests separately. It is quick to setup and easy to sync with our Github project so it helps us test and deploy our code in minutes.
Postman
We use Postman to test all of our API calls to see what it would respond with and to see if our API calls are correct. It is a free technology that gives you all the info you need on all the API calls such as the headers and type of communication.

Additional technologies used:

Heroku
We have chosen Heroku as the technology to host our server on. It has brilliant documentation, lots of support, quick responses and supports a lot of technologies which makes it ideal for what we require. It is also free for our purposes and if the client wants to upgrade to a better plan, then it won’t cost them that much.
Swagger.io
We have chosen to use Swagger.IO to help with the development of our API as it helps with all the steps in the API development lifecycle, from planning to documentation.
https://app.swaggerhub.com/apis/KobusMarais/eCivixAPI/1.0.0
Trello
It helps us to manage our project easily. It allows us to visualize when tasks needs to be completed so that we complete all necessary tasks in time.
Slack
It’s a great tool that brings all communication to one place. What we also found very helpful was the linkage to Trello. Getting updates on tasks and their status helps a lot

with our workflow. All content is also searchable from one search box which saves time. We are also able to share important files quickly. The last thing to mention is its great accessibility across multiple devices as this helps us to always be able to stay up to date with our project.