

Edinburgh

## System Design Project

### What

As part of a team I was supposed to work on building a robot which is capable of playing football using Lego, Arduino and a camera system in a closed football pitch made for robots.

### Why

This module enabled me to learn how to tackle a complex problem as part of a group touching on not only the software aspect of things, but hardware and design.

### How

We first split the team into minigroups of two people who were responsible for each specific aspect of the robot, the design, the vision system that would be connected to the camera and the software part for the robot. I was responsible for the design and there were two main areas that we had to focus on 1. the robot is fast and 2. The robot is able to deliver a strong kick. We got the holonomic wheels which helped our robots move significantly faster when compared to the other robots in our cohort. The second area of focus was to get the kicker part to work which we implemented using a lego part attached to a strong lego motor, which delivered a powerful kick. Last few weeks we spent making some additions to our robot which was adding a part which is able to grab the ball prior to kicking it in order to be able to direct the ball in the direction of the goal.

It was really good experience being part of a team of really smart individuals trying to design the best robot which is capable of winning against the other teams competing

### Lessons Learned

This helped me enhance my knowledge by not only teaching me a lot of technical aspects involved in designing a robot and writing code on arduino, but the main takeaway was how I to work as a team to solve a complex problem.

## Indoor Localization System

### What

This project was about creating a system which would be able to accurately show the location of a person inside a building, or a structure.

### Why

The GPS system doesn't give the accurate location if you are inside a building, since the signals get reflected and there is a lot of interference when indoors. This system would be able to accurately measure a person's location inside a building.

### How

We were provided with an Arduino board, a Bluetooth LE enabled phone, some sensors and Bluetooth LE enabled beacons placed inside one floor of a building. We were supposed to build an app on the Android phone provided which would be able to connect to a local cloud provided by the Uni process the data that we get from the sensors connected to the Arduino. Using the Signal strengths of different Bluetooth beacons provided at different locations we were supposed to read the signal strengths on the sensor connected to Arduino, send the data over to the Phone which would send the data to the cloud to each endpoint. Once the data was processed on the cloud the predicted location would be sent over to the mobile app which would display this.

### Lessons Learned

Edge Computing, Cloud Computing, Working with Different sensors. Coming up with the right strategy to process the data, this involved testing signal strengths, analyzing different factors using other sensors provided like the gyro and heat sensor.

## Analysis of Different Stocks and Predictions

### What

The Aim of this project was to analyse some of the stocks, predict future prices and create a portfolio optimised for risk using the chosen stocks.

### Why

### How

We were supposed to get data related to a stock using the Bloomberg terminals, cleanse the data and make predictions for the future prices. The price predictions were done through a couple of techniques like regression, Markov chain simulations using the data that was gathered for the stocks provided and training a neural network to perform the predictions. The first two were easy to implement but adapting the data and getting a neural network setup with the libraries in R were slightly tricky. The final part was getting the optimal model which involved a few layers of neural networks along with some regularization to get a prediction model which performed decently against the test data.

The final part of the project involved making an optimal portfolio with the stocks that were chosen in the first part of the project. This involved a lot of analysis in terms of what stocks should be included in the portfolio given the portfolio had to be optimized for risk. I had to run simulations as to which combination would be least risky and at the same time maximizing the return.

### Lessons Learned

Bloomberg terminals, Data Cleansing, Neural Networks, Risk Analysis, Portfolio Optimisation.

Barclays

Config checker

What

We were migrating our API's to a new build pipeline which involved making a lot of changes to our config and having it in a set format. Prior to making a release for a microservice that was migrated the team would review the config manually. I came up with the idea of automating this task which enabled less time to be spent of reviewing config.

Why

Before the config checker was introduced the config was checked manually by a few team members before releasing to production this would reduce the team productivity as this was something that could be automated and reduce the amount of manual work that was required. This would take out the errors that would come up in case of checks performed manually.

How

I implemented a tool in Java that would compare all the properties present in a given file with the file that we consider a source of truth. We had a template of all the properties that were supposed to be present in a given which the inputted files would be evaluated against. This would also show if there were any key differences in the properties from that of the template

Lessons learned

Communicating with the wider team to understand the problem at hand, how to formulate a solution for a given problem and future improvements on it with time.

## Adding Caching to the Barclaycard Microservices

### What

As part of the API team I was responsible for adding Gridgain cache to some of the microservices that were responsible for majority of the requests.

### Why

The microservices used by the upstream services most were relatively slow, our team identified the possible areas we could improve the overall responsiveness of the platform by digging into the API response times of the services taking a longer time to respond. Through the Analysis conducted we identified that improving performance of the API's responsible for Payments, Journaling and Card details would significantly boost the performance of the platform.

### How

The Initial part of the project was planning which involved a lot of analysis of how and where adding Gridgain Cache would help, this involved mapping all the calls to the downstream services and the data we would be getting and visualizing the response times by each downstream service to analyse where adding a cache would be a benefit. Following this was the implementation phase where we added caching using the libraries provided by Gridgain. Once the implementation completed we did several rounds of performance testing for the API's with the Caching mechanism implemented and we benchmarked it against the response times with Gridgain cache disabled.

### Lessons Learned

This gave me the opportunity to enhance my knowledge of the platform and understand where and how we can deliver performance improvements to the platform. I had the opportunity to get involved with implementing caching and performance testing some of our API's as well.

## Service Mesh Migration

### What

A service mesh is a dedicated infrastructure layer for facilitating service-to-service communications between services or microservices, using a proxy. This project was about migrating our API's to eventually be deployed into a service mesh. As part of this project we were also implementing Dynamic Configuration using Spring Cloud.

### Why

A Service Mesh has many benefits such as providing observability into communications, providing secure connections, or automating retries and backoff for failed requests. This would help the overall stability of our platform.

The Dynamic configuration would help us make trivial changes like updating certificates, changing some properties for one of our microservices smoother.

### How

### Lessons Learned

Amex

What

The Modern Accounts Receivable System is the new system that I am working on as part of a team of nearly a 100 engineers . This will be replacing the alternative and legacy Accounts receivable we have in place for processing our transactions.

Why

The Legacy systems for Accounts Receivable are hard to maintain and are not cost effective, We couldn't charge the customer the correct amount of interest because of some limitations with Globestar which is one of the legacy systems we have in place for AR. Furthermore there were issues with reusability of functionality for transaction processing when it came to launching different products in different countries which involved different rules and regulations. MARS is the global system that would be replacing all the AR platforms which will help us increase our revenue, help us scale better and be more cost effective.

How

The MARS platform is a bunch of different event sourced microservices built on top of our own framework for designing Domain Driven microservices. Our microservices are built using Springboot and Spring Webflux and follow the reactive paradigm.

Lessons Learned

Working closely with the product owners to deliver features, understanding the business requirements to implement better solutions. Identifying issues with the platform, Reactive programming using spring webflux, etc.