# Progress Document

## Changes Since previous design

**Data** – I Realised the previous plan to create my own data from recording a traffic simulation through JMeter would not be feasible. JMeter does not have the ability to generate load on a website in a way that would accurately simulate a real-world website traffic.

As such I am instead using the WorldCup98 data set. This dataset tracks the Requests per minute for the WOrldCup98 website over 3 months. This provides more than enough information to train a model and can provide a fair representation of traffic fluctuation over a period of time.

**Model** – The idea behind the model remains largely the same however its original goal is now to be split between two models. One will have the sole responsibility of predicting future load based off previous data. This predicted load will now be fed into a new model to determine the optimum pods to be running for this load. This is done by running iteratively testing the cluster with different number of pods at different load. The optimum will be the number of pods that consumes the least energy whilst maintaining a minimum level of lost requests.

This allows the framework to be web app specific. By understanding the specific resource needs of an application at different load levels, the system can make more informed scaling decisions. This leads to better resource utilization and minimizes the risk of SLA violations or over-provisioning. (adaptative ai-based for Kubernetes paper)

**Testing** – As previously discussed, I have found JMeter limited capabilities in accurate load simulation makes it unsuitable for \_\_ from this project. I will instead be utilising Locust. This will allow me to program my own load test in python and utilise the WorldCup98 database as a “validation set” for the system as a whole as well as the time-series model.

## A diagram of a cluster Description automatically generatedArchitecture

## Updated Plan

### Semester 2

* Week 6 (This week)
  + Get Chapter 1 reviewed
  + Create first instance of time series model
  + Create first instance of workload profiling model
  + Integrate the models into the auto-scaler and have it inform the autoscaler
* Week 7
  + Develop ML model
  + Create an accurate workload profile for the web app
  + Start chapter 2
* Week 8
  + Evaluate and finalise both models
  + Run tests on the system as a whole against benchmark to gather performance data
* Over Easter
  + Complete Chapter 2
  + Complete Chapter 3
  + Start and Complete Chapter 4
* Week 9-10
  + Add the finishing touches to the report.
  + Submit the report and all supporting documents and repositories.