

Real-Time ScoreBoard Project Assessment

Feasibility Study



1st August

R&D Students

AUT

**Team Members**

Hayley-Belle Cleverdon   
Vinicius Alves Ferreira  
Karanjit Gahunia   
Seung-Kyu Jin   
Alex Lu

**Client**

Dr. Robin Hankin

**Supervisor(s)**

Nikola Kasabov

Akshay Raj Gollahalli

**Moderator:**

Dr. Stephen Thorpe

Version

Feasibility Study Version 1.0

Contents

[Technical Assessment 3](#_Toc488939615)

[Venue Infrastructure 3](#_Toc488939616)

[Venue Information 3](#_Toc488939617)

[Networks 4](#_Toc488939618)

[Existing Equipment 6](#_Toc488939619)

[Hardware requirements 7](#_Toc488939620)

[Hardware specifications 9](#_Toc488939621)

[Other Elements of an in-house Implementation 12](#_Toc488939622)

[Cloud Solutions 14](#_Toc488939623)

[Web Hosting 14](#_Toc488939624)

[Amazon 15](#_Toc488939625)

[Cloudflare 15](#_Toc488939626)

[Existing Solutions 16](#_Toc488939627)

[Estimated Costs 16](#_Toc488939628)

[Solution Summary 17](#_Toc488939629)

[System Implementation 17](#_Toc488939630)

[Application Solutions 19](#_Toc488939631)

[Legal feasibility 21](#_Toc488939632)

[Operational feasibility 22](#_Toc488939633)

[Resource and Schedule study 23](#_Toc488939634)

[Conclusion 24](#_Toc488939635)

# 

# **Technical Assessment**

The purpose of this assessment is to inform and advise the client (and future teams undertaking this project, if this is the case) of the technical needs and solutions for the project under discussion.

This assessment consists of an analysis of 3 main areas:

1. The venue’s infrastructure and existing hardware.
2. The hardware solution evaluation.
3. The cloud based solution evaluation.

At last a recommendation is given based on the information presented in each section. It aims to address the best resolution to the problem domain.

The proposed solution for a software is a web-based application which is within our believes the best match for the client requirements. However, our assessment is not limited to one approach.

**Venue Infrastructure**

Note that most of the information in this report was obtained on a visit to the ASB Stadium by Karanjit Gahunia on the 30th of June. Brian Tomlinson, the general manager of ASB Stadium, was interviewed during this visit. The findings are based on the interview as well as observations made on this visit. Photos were taken during this visit and Mr Tomlinson has given permission to use those photos in this report.

### **Venue Information**

**Venue Size**

The Auckland MATHEX events are held at the ASB Stadium in Kohimarama. The main gym is used for this event which spans over 2 days, with 2 sessions each day. The main gym floor is 45m × 30m (1350 sq/m) and has a floor above with seating for approximately 3500 people. In previous MATHEX events, there have been approximately 5000 people in attendance each day (between the 2 sessions).

**Unused Seating Section**

The layout of MATHEX competitions at ASB Stadium can be seen in figure 1. In previous years, the seating section behind the judges has been completely closed off (figure 2). This is to prevent spectators from cheating by seeing the answer sheets that the judges have. However, in recent years, higher rows of seating have been opened due to high attendance at these events.

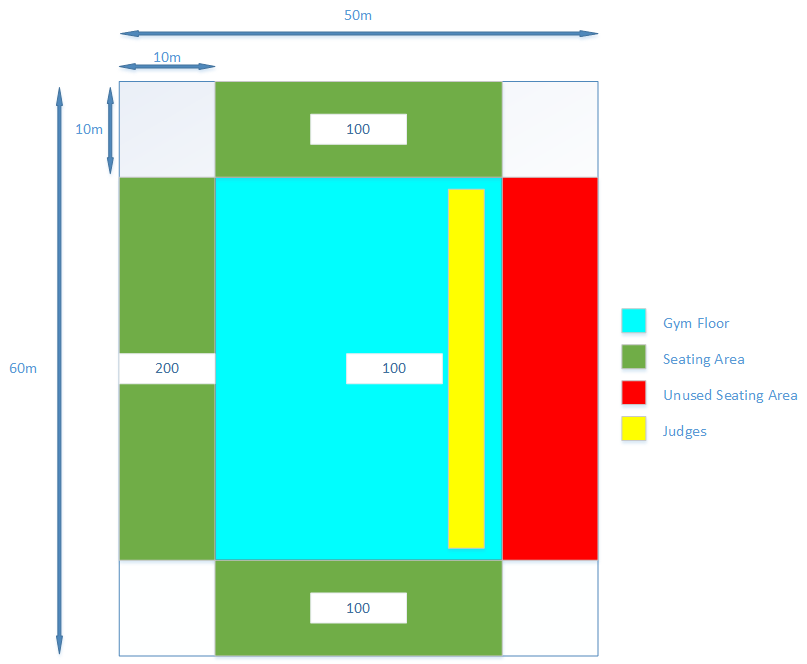


Figure 1. MATHEX competition layout at ASB Stadium



Figure 2. Unused stand during MATHEX

### **Networks**

**Overview**

There are 2 Wi-Fi networks installed in the main gym area of ASB Stadium. The first network belongs to the stadium itself whereas the second network belongs to Selwyn College. Both networks have one access point each installed in the main gym (figure 3). According to Mr Tomlinson, both networks can potentially cover the entire main gym area.

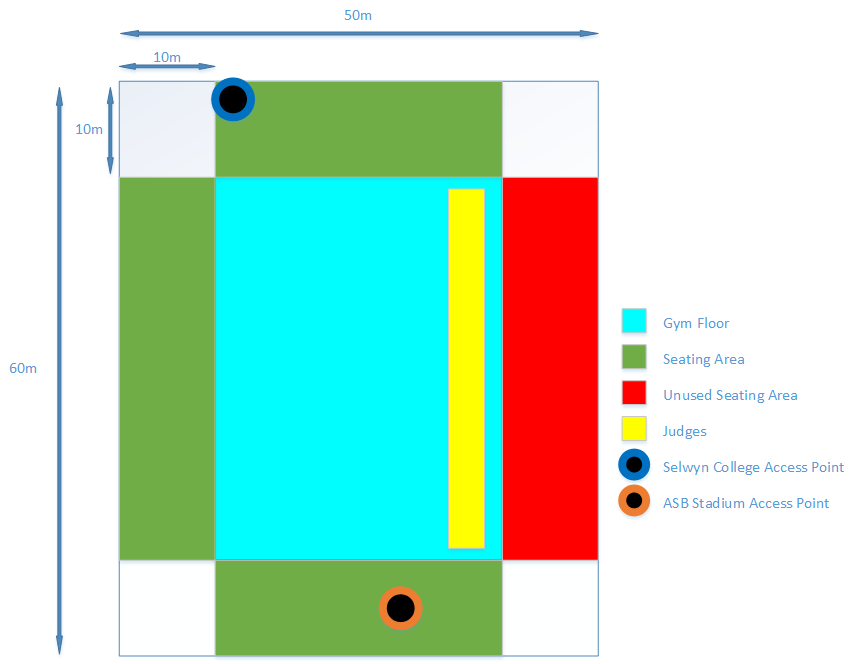


Figure 3. Location of Access Points in the main gym

**ASB Stadium network**

The ASB Stadium network is a broadband internet network. The access point is located under the seating area. There are other access points located in the stadium that are also connected to this network (e.g. reception area). A speed test of the network showed that the connection had 17 ping, 15 Mb/s download, and 12 Mb/s upload. Mr Tomlinson says that adding repeaters would extend the range to cover the main gym area. The stadium is open to us bringing our own equipment to install and utilise along with their network.

**Selwyn College network**

The Selwyn College network is a fibre internet network. There is one access point installed in the main gym area (figure 4). Permission to utilise the network would have to be requested from the Maths department of Selwyn College. There is no speed test data for this network.



Figure 4. Access Point for the Selwyn College Wi-Fi network

### **Existing Equipment**

**Scoreboard Display**

There is no projector screen or other type of display installed in the main gym area. However, there is plenty of space to install a temporary screen and projector for a big scoreboard display. Mr Tomlinson recommends using a space in the unused seating section (figure 5) as it has been used in the past for similar uses. This location would be visible from most spots for spectators as well as competitors in the MATHEX competitions.  
  
**Network and Server Equipment**  
There is no existing equipment such as access points, wireless repeaters, or servers that could be used for the Real-Time Scoreboard System implementation at the stadium.

Further Research Areas

Some of the areas that require further research include:

* The capacity and range of access points installed in the main gym.
* The type of equipment (if required) for extending the existing networks.
* Permission regarding the use of the Selwyn College network.
* Speed of the Selwyn College network.

## **Hardware requirements**

This section aims to evaluate the hardware necessary in order to host the application in a local environment. As there are many options available, our goal is to recommend adequate hardware to satisfy the client's needs.

It is crucial to understand that this research was undertaken as an individual component of the project, and it disregards any information about the venue’s current infrastructure such as networking or non-physical solutions, like a cloud-based system.

Any pricing provided was acquired through the research process on (23/05/17), all prices are subject to change.

The research was done based on the requirements of the system, see below:

* Project goal is to create a **real-time** scoreboard application or website.
* Can handle 100 users(judges) updating the database simultaneously.
* Can handle at least 400 users(audience members) or requests simultaneously.
* OS was not specified as client required an open source system. AUT could also provide assistance with software.

*Please see* appendix A for *hardware specifications and quotation and appendix C for definitions.*

Solution 1

In order to host a back-end system for either a web-based-application or for a mobile phone application We recommend the system to be divided in three servers: Reverse Proxy, Application Server and Database server, please see figure 1 for a high-level network draft. For reliability and performance, it is recommended that you separate them. (*See distributed system reference for more info*).

The use of a reverse proxy server would, among many other benefits, distribute the load from incoming requests. It will also protect against common web-based attacks, like DoS or DDoS and malware. Additionally, it would reduce the load on its origin servers by caching static content.

The application server would be responsible for handling requests from the user or proxy server. It would inquire the database for and serve it back to users, as well as handle all processing required by each request.

The database is responsible for managing and storing all data of the application, and serving it back to the application server when requested.

Machine generated alternative text:
RESTful request 
Internet/ 
Response with ISON 
Application 
Figure 1 : 
Senaer Side 
Reverse Proxy server (NGINX Server) 
Application server 
(Tomcat server) 
Database server (Postgresql) 
Discussed framework 

**Solution 2**

A more compact solution that would also impact on the cost of the system is to take use of virtualization (*See virtualization in reference for more info*). The use of this strategy would allow us to co-populate 2 servers on one host, that is to say that one server would have role of two. In this case the application server and the reverse proxy server will be located in one server. This strategy would eliminate the need of one of the servers. However, some enhancements could be required, such as a higher RAM capacity.

Beyond the cost benefits, using virtualization is a more efficient use of the processing power of a server. Taking into consideration that most of the requests will be **I/O bound** rather than **CPU bound**, the hardware recommended would not have problem responding to the demand. ( *See Request-per-second in reference* )

However, there is consequential reliability on the server in discussion, creating greater risks. For the same reason, we recommend that the database server is a separate system, in simple words, this is where all your data will be stored.

**Solution 3**

A simpler approach is to have one sever that works as a database and application server, to eliminate the need for a proxy server. This is common practice for small applications or businesses.

This set up could provide a working environment to the application, however, it faces potential issues such as slowness, no response at all, security issues, and a much higher risk of the entire system crashing.

Therefore, we discourage this solution until the criticality of the system is fully assessed. System failure could lead to a major disruption in the competition.

### **Hardware specifications**

There is no simple or exact way to predict performance and scalability of a system. Each application and environment is different. Programming languages, programs, APIs , functionalities all requires their fair share of resources and use different amount of memory, processing power, internet bandwidth and others as it also behaves differently.

The most accurate way to determine the correct specifications to a server or servers is to measure performance which will require the application itself or a prototype application to create a testing environment. Tests can be performed using virtual servers, load testing applications like "[JMeter](http://jmeter.apache.org/)" and application performance management tools, for instance [AppDynamics](https://www.appdynamics.com/) or [DynaTrace](http://www.dynatrace.com/en/index.html), to measure performance and identify bottlenecks. Only then, we can estimate what hardware specifications will suffice, without building an overzealous and unnecessary system.

For now, we have estimated what we believe would be required for the application to run and satisfy the requirements. However, this are only assumptions.

Attention: prices are subject to change based on the parts' brand, type, quantity. There are several ways to achieve the same or similar specifications.

See **Appendix C** for definitions.

**See below recommendation provided to us :**

Database Server

HDD - 2 TB or more SSD enterprise grade in RAID set-up.

Processor - Intel Xenon processor that support Error Correct Code (ECC).

128 Gb (minimum) with ECC.

Reverse Proxy Server

HDD - 1 TB or more SSD enterprise grade in RAID set-up.

Two or more processors - Intel Xenon processor that support Error Correct Code (ECC).

128 Gb (minimum) with ECC.

Application Server

HDD - 1 TB or more SSD enterprise grade in RAID set-up.

Two or more processors - Intel Xenon processor that support Error Correct Code (ECC).

128 Gb (minimum) with ECC.

*Note: these estimations above were provided by "Akshay Raj Gollahalli".*

**What we have got a quote for (See appendix A for full description):**

Database Server

2x Intel Xeon E5-2620 v4 2.1GHz Processor, 8Core/16Thread ( Supports ECC)

8x 16GB (128GB)of Memory RAM ( ECC Registered)

Hardware RAID controller with 2GB flash backed write cache

2x 150GB SSD (RAID1 – OS/Boot only – 150GB Raw usable)

6x 960GB SSD (RAID10 – Database – 2.8TB Raw usable)

Write workload max: 3.6TB per day

Reverse Proxy Server

2x Intel Xeon E5-2620 v4 2.1GHz Processor, 8Core/16Thread ( Supports ECC)

8x 16GB (128GB)of Memory RAM ( ECC Registered)

Hardware RAID controller with 2GB flash backed write cache

2x 150GB SSD (RAID1 – OS/Boot only – 150GB Raw usable)

6x 480GB SSD (RAID10 – Hot Data – 1.4TB Raw usable)

Write workload max: 1.8TB per day

Application Server

2x Intel Xeon E5-2620 v4 2.1GHz Processor, 8Core/16Thread ( Supports ECC)

8x 16GB (128GB)of Memory RAM ( ECC Registered)

Hardware RAID controller with 2GB flash backed write cache

2x 150GB SSD (RAID1 – OS/Boot only – 150GB Raw usable)

6x 480GB SSD (RAID10 – Hot Data – 1.4TB Raw usable)

Write workload max: 1.8TB per day

We believe that the specifications of the servers quoted for are more than sufficient to satisfy the requirements, whilst also providing scalability in the future. However, due to the frequency of the MATHEX competitions, we recommend to purchase only hardware that will be used.

*Why are the specifications so high?* The same requirements listed at "Hardware requirements" section and the recommendation provided above were used to inquiry for a quote. Our retail contact assumed that this is an enterprise-scale application, inferring that the servers will be running continuously with a constant 500 users or more. Therefore, the amount of storage memory is significantly high, it uses 2 processors and the memory matches the recommendations.

**What we think will be necessary:**

When assessing the current requirements, clearly there is not a significant amount of data that requires storage. For this reason, the database queries will be fast and simple. Also, the application should not perform complicated tasks that require heavy usage of the CPU. The challenge at hand is the hundreds of users using the application concurrently. Due to the powerful hardware available in the market, we believe that one server (Solution 3, see Hardware Requirements) should suffice. **See specifications below (See Appendix B for full description):**

Application Server \ Database Server

1x Intel Xeon E5-2620 v4 2.1GHz Processor, 8Core/16Thread (Supports ECC)

4x 16GB (64GB)of Memory RAM ( ECC Registered)

1x 150GB SSD (OS/Boot only – No RAID solution provided)

2x 960GB SSD (RAID1 – 960GB Raw usable)

(excluded) ~~Hardware RAID controller with 2GB flash backed write cache~~

(Exchanged) ~~Supermicro 1028R-WC1RT Barebone~~

In this estimation we have replaced the barebone as only 1x Cpu is needed, note there are 4x extra slots in case more memory is needed. Also, excluded CacheVault Supercapacitor as its extra protection is not necessary at this point.

Using this estimation, extra servers could be purchased to comply with the other solutions highlighted previously at "Hardware requirements". If this is the case, the specification above can be manipulated to reduce costs and better adjust to needs.

**Additional costs:**

Some other equipment may be necessary such as:

- Racks or cabinets to hold the servers

- UPS (Uninterruptible Power Supply)

**Other Elements of an in-house Implementation**

Along with acquiring the adequate hardware to satisfy the current requirements, there are other components which require attention. In many cases these are considered the drawbacks of purchasing your own hardware.

**Failure management - Disaster management**

It is important to plan for any events that may occur and affect the system performance. These events include; hardware faults, power outage, system overheating, and any natural disaster. Therefore, a disaster or recovery management plan would be necessary. However, such plans are usually the responsibility of the client to develop, this depends on whether the client determines that the information is vital, and that creating the plan is a necessity. The plan must also comply to the venue’s health and safety policies.

Common solutions are to have; the servers in a temperature monitored room, a fire suppression system in the event of a fire, a generator that kicks in automatically to supply power to the servers during an outage, back-up servers to take place when one stops working and daily back-ups of data, the list goes on, but these are the most significant regulations.

In the case of MATHEX competition, the current paper-pen system would not be discarded as last resort back-up plan.

**Maintenance**

It includes regular hardware and software checks, to ensure the system is up to date and working to specifications. The solution may also undergo updates, for bug fixes and adding features, these will require testing.

Maintenance may also be extended to other parts of the system, such as the options mentioned in the failure management section above.

Further information can be found at: <http://sebokwiki.org/wiki/System_Maintenance>.

**Cost to run the equipment.**

The system will require resources since energy power to labour. We assume that the servers and its peripherical devices will not be located at the venue and will need to be moved from place to place, which will also infer set-up costs and tests.

Maintenance of the system will also generate costs as well as possible hardware faults.

**Durability**

A computer also has a lifespan, which is usually estimated between 3 to 5 years, but it is subjective. It will depend on how it is used, how often it is used, and how it is maintained.

There are many systems in existence that have been running for several years and will likely still live for many to come with the appropriate maintenance.

However, replacing parts is to be expected, which will incur further costs.

**Cloud Solutions**

We have reached the conclusion that there are two different approaches we can take for a cloud-based solution for this project.

The first is a series of static web pages for all necessary components of the system. This includes; a small database and two or more webpages.

The small database, consisting of one or two tables will store data. A user will access a web page which will send a request to the server, which pulls data from the database and displays it on the webpage. The judges will also be using a web page, but instead they will be sending requests for the server to transfer data into the database. To add security to the system, there is the option to include a login page for judges, otherwise they can be given a generated URL to access the pages privately.

This solution requires that data about the competition is collected, such as what each question is worth in points, the teams and students who are participating. It may also be necessary to generate login credentials for judges to keep the database secure.

Technologies used for this solution are expected to be as follows:

* Amazon S3: Web hosting.
* Cloudflare: Web security and optimisation.
* Amazon RDS: Database Engine
* Languages Used: PostgreSQL, HTML and JavaScript OR Java (run on a Tomcat servlet).

The second solution would be to develop an application that displays on the web, this application would need to have the same capabilities of the first, but it would also include ways for administrators to set up competitions. This application will take up significantly more data on the cloud server, but it will also be a dynamic way to set up competitions, as the front end will provide controls to create custom leader boards.

Technologies used for this solution are expected to be as follows:

* Amazon EC2(T2): Web hosting
* Cloudflare: Web security and optimisation.
* Amazon RDS: Database Engine
* Languages Used: PostgreSQL, HTML and Java (run on a Tomcat servlet).

### **Web Hosting**

Web hosting can be thought of as having a high-tech computer, with a huge amount of storage space and processing power, all available through an internet browser. In reality, the web host already has all the necessary (and very powerful) hardware, they’re just letting you use it for a small price, and providing an interface for you to control it.

A web application is stored with the web host, and set up with a domain so that users can access the system. From there the web host handles all the data, requests and runs the software as it was designed. This solution is reliable, easily accessible and low-maintenance.

### **Amazon**

Amazon is a powerful and cost effective solution to web-hosting. It provides more computing power than any other online service and is completely free to join.

Amazon only charges for what you use, so setting up a small website will only cost cents per year.

S3 stands for Simple Storage Service. It provides an object storage to host cloud-based applications, websites, repositories and much more. It is designed to be fully scalable, boasts durability, a 99.99% uptime, easy to manage and fully integrated with a wide range of Amazon’s other web services.

EC2 provides a very similar service, but acts as a remote computer that can run software. EC2 is geared towards running applications, so the processing happens on Amazon’s side, rather than the user’s internet connection.

RDS is Amazon’s leading relational database system, it is a simple way to implement a database into an existing Amazon service and utilise it. A small database is free!

Amazon’s web services are unique because they are distributed worldwide, so users around the globe are able to access sites with improved latency. They are also unique due to their dynamic scaling – as soon as more storage or processing power is used, that is what you are charged for. Rather than most web-hosting services, which require you purchase the amount you need, and if you need more you must buy a set amount, and if you don’t use it all, well that’s just too bad because you’ve already paid for it.

All this information and more can be located at <https://aws.amazon.com/>

### **Cloudflare**

Cloudflare is like a virtual router for websites. It is a proxy server that filters malicious visitors, saves bandwidth and accelerates user connection to the website. Best of all, for a small website with low needs, Cloudflare is totally free!

Cloudflare increases the speed of a user’s connection by routing based on the user’s location, connecting them to the nearest datacentre in their location. Cloudflare also caches data on your website (temporary storage) so that when a user is loading up a page with the same images and code scripts, the data is all waiting in Cloudflare, rather than needing to be requested from the web host once again.

Cloudflare also reads a visitor’s IP to determine whether they are a threat to your website. Any detect threats are screened from the site and have no access to the site’s bandwidth.

All this information and more can be located at <https://www.cloudflare.com/>

### **Existing Solutions**

There has been a demand for custom leader boards, though not significant one. After some research we found two suitable pre-made solutions.

AirScoreboard: <http://www.airscoreboard.com/>

An iOS exclusive app, for an affordable $0.99US. This solution allows users to create and manage leaderboards as well as share them to various social media sites. It has a few additional features like locations, languages, posters, and it has a way for ‘athletes’ to register themselves. A lot of these features are more than what is needed for the MATHEX scoreboard, and the iOS-only limitation is a large barrier.

Rise: <https://www.rise.global/pages/simple>

This site offers a large variety of leaderboards, at what appears to be a commercial standard. It portrays itself as a visually-polished application and is available on web, mobile or on-screen (presumably one with ‘smart’ capabilities). While this is a great solution, there are no metrics to measure its reliability or security, it is not clear on the site whether they are using another web hosting service or have their own. Should this solution be explored, further information should be gather from a representative of Rise for this information before making a decision. There is an undeniably larger cost involved, at 14.99GPB (up to 100 teams) or 29.99GBP (up to 300 teams) per month. Though, it would only need to be up and running one month out of the year.

### **Estimated Costs**

Assumptions:

* 400 audience users, each making about 50 requests from the database over the competition duration.
* 100 judges, each making 100 requests to the database over the competition duration.
* Total storage of 500Mb or less.
* Data transfer in/out at 100MB/1GB per month at most.

|  |  |  |
| --- | --- | --- |
| Solution | Monthly Cost\* | Yearly/Total Cost\* |
| Solution 1 | $0.21 | $0.37 (per year) |
| Solution 2 | $3.33 | $12.41 (per year) |
| AirScoreboard | $0.00 | $140.89 (total – for 100 devices) |
| Rise | $27.65 | $27.65 (yearly, by cancelling after 1 month) |

\*Currency converted to NZD 25/05/17 – this does not include conversion charges.

### **Solution Summary**

The most feasible choice here is Solution 1 – a simple set of static web pages hosted on Amazon S3. Not only is this a cost-effective solution, that allows AUT to be branded alongside the AMA to encourage learning in New Zealand, it is highly achievable.

A simple site will not take an extended period to get up and running and ready for testing. This is ideal to ensure that the client gets to see an early prototype and have input on design decisions. The prototype will have plenty of time to be user-tested and presented to stakeholders part of the AMA.

It will also be very easy to build upon a simple site, to a full application in future, as the design decisions will have already been made.

Next up is between Rise and Solution 2. Both are supplying the same kind of functionality. However, Rise robs our group of the opportunity to develop anything, which we’re very keen to do. Solution 2, the Java application, will take some time to build, and may not have much time for testing. It is likely the prototype will be very basic and lacking a few features.

Solution 2 will be ideal to explore after the prototype of Solution 1 has been fully tested, it is unlikely that this will be undertaken by our group, unless outside of the Research and Development paper – the source code will always be available for anybody to build upon should they wish to.

Lastly, AirScoreboard is barely worth mentioning. It has a lot of ‘fun’ features, that are not necessary for the commissioned system, and it being a paid app exclusive to iOS puts massive limitations on it. The judges would each need to have an iOS device to log scores, which is not at all feasible.

System Implementation

**The Infrastructure**

Based on the current state of the venue, it is expected that additional equipment will be required to provide access to the wireless local area network for all MATHEX attendees. Both existing networks at the stadium have just one access point in the main gym area. More access points and repeaters would be needed to extend the range and capacity of the networks to accommodate for the MATHEX competition attendees. Moreover, depending on the choice of application solution and system implementation, the existing internet connections may not have sufficient bandwidth to handle the amount of traffic. In this case, solutions with the internet service provider will need to be explored. Note that this will likely be at the expense of the venue for any potential upgrades.

On the other hand, there is also the option of internet access through the user's own data plan, which will be at their own cost.

As a reminder, because the MATHEX event could take place in different venues in the future, the full assessment of the venue is not our main focus.

**Purchasing the Hardware**

The benefit of having all the servers at the same location where the Mathex event occurs is that there is no need for internet access. The presence of Wireless access network (WAN) would satisfy the needs to run the application and connect all the users present at the event.

However, the equipment purchase costs can be very high. It should also be highlighted future costs, such as; the set-up of the system, transferring the server location, maintainability and labour. Additionally, a disaster management plan may be required to comply with venue's policies and general health and safety regulations, as well as to keep the data backed up and stored, and keep the competition running.

Mathex events are infrequent. As we understand, those servers and any other equipment would not be used for the rest of the year which would could lead to waste of resources. The equipment will also need to be stored at someone’s expense, and be insured in the event that it is damaged, misappropriated or lost for any reason.

The Mathex events seem to be increasing in popularity each year, this could lead to an issue. To scale the system to satisfy the demand may not always be feasible, it could require new hardware parts or servers as time goes on to meet up with consumer needs, replace old and worn parts and keep up with advancements in technology. To conclude, adding more capabilities to the system comes reapplies all the risks of implementing the system in the first place, costs will need to be calculated each time and will most definitely increase.

Therefore, we believe that purchasing and maintaining your own servers for this particular solution is not cost effective and is a misuse of resources.

**Using Cloud Solutions**

Using cloud solutions will require internet access, which means the internet bandwidth must support the traffic of data otherwise users will experience slowness when using the application.

There are different cloud services available, but many of them allow you to pay as you go, meaning you are only charged for what you use. With this feature, there is a significant reduction in costs, as the application and the resources necessary can be reserved whenever they are needed and paid accordingly. Furthermore, it eliminates any need or concern for servers. Their location, status, storage, maintenance and all other concerns listed above are of no consequence to the developers, client, venue or any other stakeholders. The companies that provide these services are responsible for their hardware and software.

In case the number of users increases, more resources can be purchased on click of buttons. For some cloud providers it adapts automatically and then it is charged accordingly.

It is evident that using a cloud service will be significantly cheaper than purchasing your own hardware. It will eliminate many headaches, and can be used at any place and time as long access to internet is available.

Note: It could be of the client’s wishes to purchase the hardware, if they are not readily available in AUT. The same system could be maintained at AUT's grounds and be provided over the internet. This approach would work over the internet similarly to the Cloud approach.

**The Technology - Conclusion**

After our assessment, we have concluded that the best option for the implementation of an application that satisfies the client's requirement and reduces short and long term costs is the use of cloud technologies.

It does not mean that Cloud services is better than having your own hardware but for this specific purpose the benefits of using cloud services instead surpasses the benefits of buying hardware.

As students plus the contribution of our lack of experience, we believe that the set-up of the servers will be complex and therefore, we cannot commit to the reliability of the system. On the contrary, using system already set up for us provides a higher change of success.

**Application Solutions**

In this section, we would like to emphasize potential applications which aim to fulfil the project scope. We concentrate our discussion on what they are, and why or why not proceed with them. We have identified three ways the application can be built; web-based application, mobile device application and a computer application. We list them below.

Web-Based application:

**What:** It is an application which runs in a web browser.

**Why:** Because any computer or mobile device, using any operation system such as windows, Linux, Android or IOS, can access and use the application as long as there is access to the internet (or the network where the app is hosted) and a web-browser capable to surf the internet. This is the most universal approach compatible to all devices.

Mobile phone application:

**What**: it is an application that runs on mobile devices, smart-phones or tablets.

**Why:** The great majority of people have at least one type of mobile devices in New Zealand. A mobile phone application are usually more user friendly. It is easier to access and the user interface is easier to interact with.

**Why Not:** There are several types of mobile phones and tablets which increases the complexity of creating an application but it gets more complicated when we look at their operational systems(OS). The most common ones are Android, IOS and Windows and therefore supporting all the OS may not be feasible.

Furthermore, in order to use and application in a mobile device it must be downloaded and installed first. Usually an application is considerably large. To download the application, users must have access to the internet or downloaded it before coming to the Mathex competition, otherwise having hundreds of people downloading at the same time using the same network will generate the network to be very slow generating delays on the scoreboard system. Lastly, because uploading the app to the device’s respective app store would incur additional fees, it will need to be downloaded from an unauthorised location, which many users may not trust.

Computer application:

**What:** A desktop application that runs in a computer. This approach would not be available to the spectators but only to the judges so the answers can be entered. The score board would be mirrored to a screen(s) through a projector positioned such that spectators can visualize the scoreboard .

**Why:** The aim of this solution is simplicity. It would reduce significantly the amount of resources needed such as powerful servers, internet access and Wi-Fi connection for spectators. Due to the low traffic of data one computer could take over all the work that needs to be done ( See Solution 3 of hardware requirements section). The most important part of this solution is setting up a project\screen at the venue. However, the simplicity of this solution may not satisfy all client's requirements. Further costs are expected for implementing a projector and screen.

**Why Not:**  There is a high risk that implementing a scoreboard using a projector will not solve the problem. It is possible the spectators will have problems seeing the screen's content . Also, only the top teams will be shown on the screen unless a different solution is provided. At this stage, this approach does not add any considerable value to the Mathex competition instead more complications.

Note: The application used by the judges to enter the answers is not defined here. It could be a mobile app, an web-page or another computer application.

# **Legal feasibility**

# **Operational feasibility**

# **Resource and Schedule study**

Our schedule indicates there is a period of approximately three months to develop an application that meets the client's request. Throughout this period a prototype could be developed and tested along with client's approval. By the time the application is ready to be tested live, there will be no Mathex competition scheduled and there is still the possibility the implementation system of choice is not available yet. Therefore, We have not looked at the opportunity to test the application during the Mathex competition as it happens once a year and the application as well as the system to host it will not be ready.

The reason to develop an prototype is to showcase all the functionalities that will be used during the competition and ensure that it will be well accepted by the stakeholders. This application prototype will then be opened for improvement. There will be no need for the client to acquire any of the implementation solutions. The development will happen independently and can be demonstrated using minimum hardware.

This project schedule can be claimed feasible if the client follows and agrees with our recommendations. See our provisional Milestone table below:

|  |  |
| --- | --- |
| Provisional Milestones | Finish |
| Stage 0 | 28/07/17 |
| **Plan - design of project model complete** | 28/07/17 |
| Stage 1 | 11/08/17 |
| **Framework and Database setup complete** | 11/08/17 |
| Stage 2 | 25/08/17 |
| **Website development complete** | 25/08/17 |
| Stage 3 | 22/09/17 |
| **Website integration with database completed** | 22/09/17 |
| Stage 4 | 06/10/17 |
| **Online Real Time Scoreboard implementation complete** | 06/10/17 |
| Stage 5 | 20/10/17 |
| **Website UX\UI improvement complete** | 20/10/17 |
| **Handover Complete** | 27/10/17 |
| Client feedback | 29/09/17 |
| Supervisor feedback | 20/10/17 |
| Poster | 3/11/17 |
| Reflective Report | 3/11/17 |
| Portfolio and Final Product | 3/11/17 |

Legend: Red text or Bold(if printed black and white) indicates Project Deliverable

**Conclusion**