Technical REport Draft

Real-Time Score Board Mid-Semester interview assessment

Hardware Solutions

Abstract

To inform and advise the client and future teams undertaking this project, if this is the case, we have addressed the technical needs and recommend solutions for the project under discussion.

There are 3 main factors to be presented which involves analyses of the venue Infrastructure and current Hardware, hardware solution evaluation, cloud based solution evaluation. At last a recommendation is given based on the information presented on each section, it aims to address the best resolution to the problem domain.

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

Venue Infrastructure

Hardware requirements

This section aims to evaluate the hardware necessary to host the application in a local environment. As many options are available to be chosen from our goal is to recommend the adequate hardware to satisfy the client's need.

It is important to highlight that this research was undertaken as an individual component of the project and it disregards any information about the venue current infrastructure such as networking and or possible 3rd part solutions such as cloud based. A final recommendation will be provided in the recommendation section.

Any pricing provided was acquired through the research process, all prices a subject to changes.

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

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

**1st recommendation**

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 together. (*See distributed system reference for more info*).

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

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 

**2nd solution**

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

to co-populate 2 servers on one host, that is one server would have role of two, in this case the application server and the reverse proxy server will be in one server. This strategy would eliminate the need of one server. However, some enhancements will be need such as more memory RAM capacity.

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

However, there is a higher reliability on the server in discussion and therefore the risks are also higher. For the same reason, we recommend to have the Database server as a separate system, in simple words, there is where all your data will be stored.

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

**3rd Solution**

To have an application that supports images and stores data, a database and a server should be enough and many times a proxy is not used. For light traffic, it could even be a machine that provides both services.

The traffic expected for this application is consider, in our opinion, medium to high traffic allowing space for scalability. So, for reliability purposes we would recommend two independent serves to decrease the risk of failure and data loss.

This set up could provide a working environment to the application but it would still face potential issues such as slowness or no response, security issues, and possible crash of the whole server\system.

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

Therefore, we discourage this solution as there many components which could lead to project failure.

**Disaster management**

What happens if there is a sudden power outage?

Overheating of the servers CAN cause fire, what would you do if anything as such happens?

**Maintenance**

Cost to run the equipment’s.

**Durability**

Appendix A

**Requirements:**

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

**Assumptions:**

Database

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

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

3. 128 Gb (minimum) with ECC.

4. Fail-over power for your server.

Reverse Proxy Server

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

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

3. 128 Gb (minimum) with ECC.

4. Fail-over power for your server.

Application Server

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

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

3. 128 Gb (minimum) with ECC.

4. Fail-over power for your server.

**Quotation**

Note: The computer specifications below are not only a guide. We believe that these are more than sufficient to comport the requirements above. Please see recommendation section.

Database Server

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

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

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

*\*\*\*Quotation has been attached separately.*

**References**

**Images**

Figure 1 - Retrieved from: Feedback 1. Author: Akshay Raj Gollahalli. 10th April,2017.

**Links**

Distributed Systems - <https://en.wikipedia.org/wiki/Distributed_computing> and <https://en.wikipedia.org/wiki/Multitier_architecture#Three-tier_architecture>

Virtualization - <https://en.wikipedia.org/wiki/Virtualization>

Request per second - <https://wrongsideofmemphis.wordpress.com/2013/10/21/requests-per-second-a-reference/>

Cloud-based 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. A 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.

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).

**What is 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.

**Why 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/

**Why 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/>

**Is there already a solution out there?**

Absolutely, yes. There has been a demand for custom leaderboards, though not a very high 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. Allows users to create leaderboards, manage and 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, and what appears to be a commercial standard. It appears to be visually polished 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. There is almost a significantly 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.

**What are the estimated costs?**

Assumptions:

* 400 audience users, each making about 50 GET requests from the database over the competition duration.
* 100 judges, each making 100 PUT/POST 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.

**So, what is the best choice?**

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 commissioned system, and it being a paid app exclusive to iOS really puts limitations on it. The judges would each need to have an iOS device to log scores, which is a bit ridiculous.

Wireless Local Area Network

A Wireless Local Area Network (WLAN) is needed at ASB Stadium to implement the proposed system. The research conducted is based on the information on the official ASB Stadium website (http://www.asbstadium.co.nz/) and past experiences at MATHEX events. A visit to the venue will be conducted soon to gather specifics about the venue and its infrastructure.

**Venue Layout**

The main stadium, where the MATHEX event is held, has 2 separate floors (refer to figure 1). The lower floor, where the competition is held, will have the participants and judges. This area is 30 metres × 40 metres (1350 square metres). The second floor, with the seating arrangements, is where the spectators will be present. Figure 2 shows a diagram of the main gym with estimates of measurements of the second floor. Note that the section of seating area directly behind the judges isn’t utilised for MATHEX events.



Figure 1. Main gym at ASB Stadium

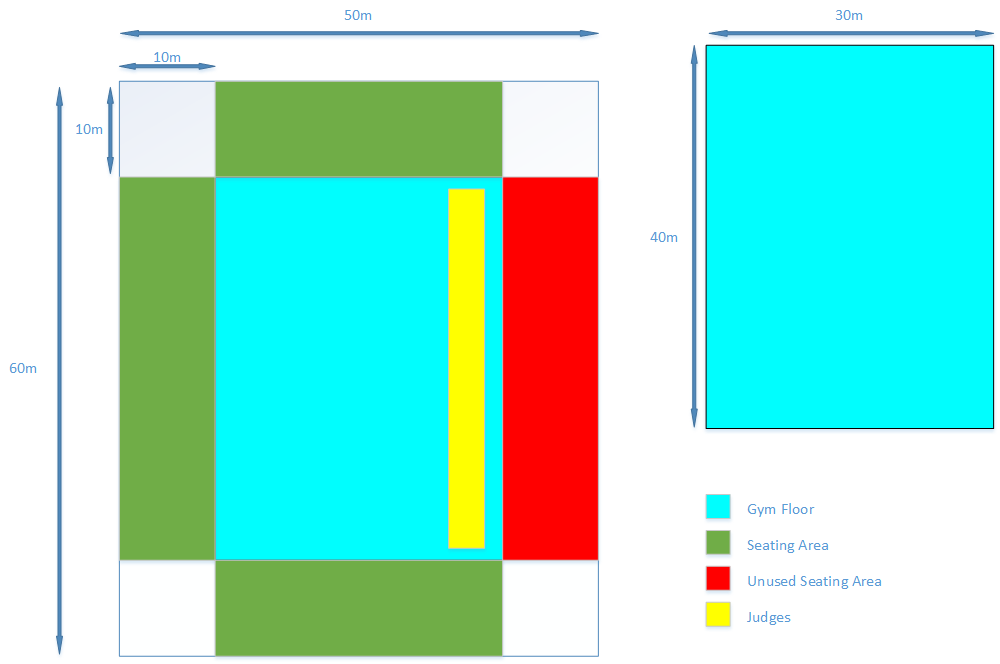


Figure 2. Diagram of main gym

# **Seating Arrangement**

# The WLAN will need to support approximately 500 users, with 100 of those users being judges. We will assume that the smaller seating sections will hold 100 spectators each and the large section will hold 200 spectators. Judges are seated on tables near the wall under the unused seating section (refer to figure 3). This is important for the distribution of access points. Access points shouldn’t be overloaded with users and there should be no access points that aren’t needed (e.g. an access point in the unused seating section).

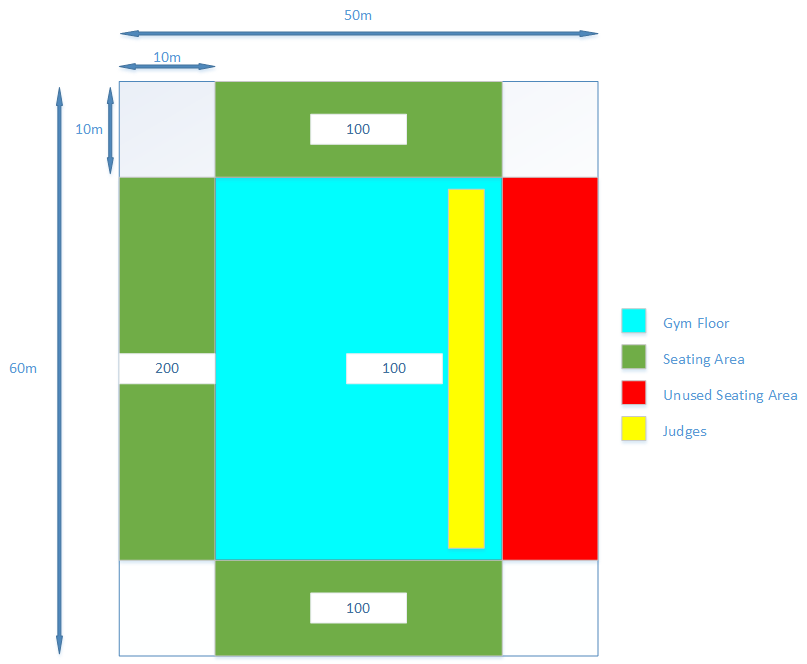


Figure 3. Diagram showing distribution of users

**Network Traffic**

To find the right network devices, the traffic in the network must be estimated. The client has requested a leader board view for the proposed system in which each team’s name and score is shown to spectators. We will assume that all 400 spectators will be using this leader board view and that the leader board is updated every second. It is difficult to calculate overall traffic because it depends on the implementation of the proposed system. However, we can estimate the traffic based on the size of database queries to the PostgreSQL database.

A 4-character version of the team’s name would be used to represent them in the table. The team’s score will also be needed. Note that there are 20 questions in a competition and 5 marks are awarded for each correct question so the maximum score a team can have is 100. Considering this, a SmallInt type would be the most efficient way to store this data. The time that a team finishes the competition should also be recorded in the database. A team finishes when it runs out of questions. The time would be used to order teams so if there are 2 teams with the same score, the team that finished earlier would be ranked higher. PostgreSQL has a Time data type that can be used for this.

|  |  |  |
| --- | --- | --- |
| TEAM PostgreSQL database table | | |
| Column | Type | Size |
| Code | Varchar (4) | 5 Bytes |
| Score | SmallInt | 2 Bytes |
| Time\_Finished | Time | 8 Bytes |

Figure 4. Structure of TEAM table

Considering the table structure in Figure 4, we can see the size of the result of each database query would be 15 bytes. There are approximately 400 spectators at the MATHEX event who request this data every second so there is 6 Kilobytes per second (KB/s) of traffic between the spectators and the servers.

These estimations don’t account for the network traffic that the judges are responsible for because there are various operations that judges conduct when using the proposed system (e.g. answering correctly, answering incorrectly, passing a question). Also, the judges do not make transactions very frequently. Therefore, the traffic that they generate can be ignored.

**Access Points**

For the WLAN, Access Points (APs) will be needed for wireless transmission of data. The Cisco Aironet 1850 series APs are suitable for the venue. They have features such as:

* 802.11ac Wave 2 wireless standards.
* 4x4 MIMO with 4 spatial streams with Single-User MIMO and Multi-User MIMO support.
* Throughput of 1.7 Gigabit per second.
* Power over Ethernet.
* External antenna model available.

MIMO, short for Multiple-Input Multiple-Output, is a wireless technology that uses multiple transmitters and receivers to transfer more data at the same time. There are 4 antennas for transmitting data and 4 antennas for receiving data. This will provide quick transfer of data between users connected to the WLAN.

The throughput of 1.7 Gigabit per second is meets our requirements and leave’s room for growth of the proposed system (if the attendance of MATHEX events increases in the future).

Power over Ethernet (PoE) allows APs to be powered through Ethernet cables. This could be particularly useful considering the venue’s infrastructure is unknown at this stage. By utilising the PoE capabilities, we would be able to place APs per ourselves without being limited by the location of power sources.

External antennas for APs allow the wireless signals to be transmitted in particular directions. This could be useful for the WLAN if APs were to be positioned at the front of each seating section (refer to figure 5). If internal antennas are used, wireless signals would be transmitted towards the participants. This isn’t an ideal use of the AP as participants aren’t allowed phones and won’t be utilising the proposed system. External antennas provide more efficient use of the AP but must be purchased separately.

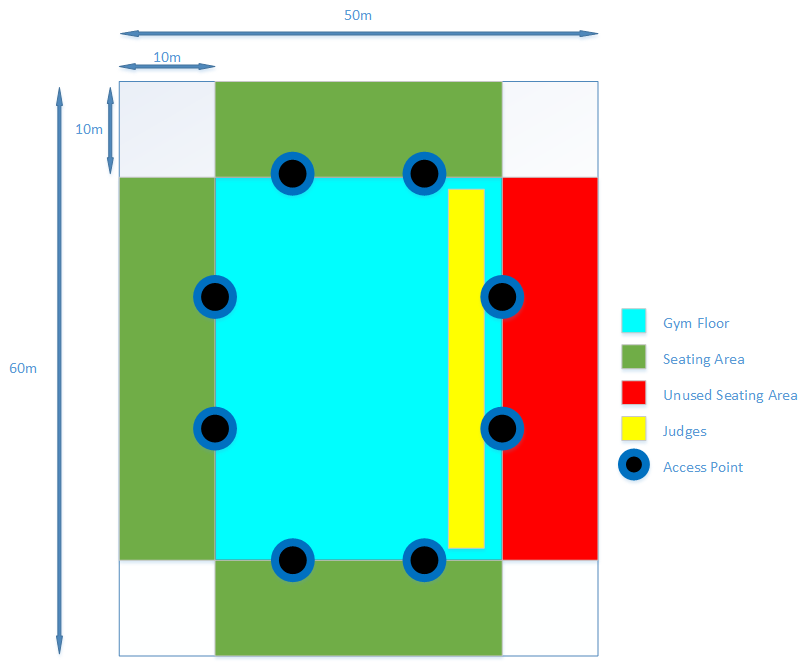


Figure 5. Diagram of AP distribution

These APs can support up to 400 clients each. However, to improve efficiency and reliability of the network, each AP should ideally support approximately 80 clients. Following this guideline, 8 APs would be needed for the network. If there are more than 500 users of the system, the 8 APs should be able to accommodate for them. In the future, if there is considerable growth in the attendance of MATHEX events, more APs could be deployed.

**Switch**

For the network switch, a 16-port switch with PoE capabilities is needed to support the APs. The WS-C2960L-16PS-LL switch by Cisco would provide this functionality. It is a basic switch with 16 ports. An 8-port switch wouldn’t be ideal as if there is a need for an extra AP in the future, the switch would have to be replaced with a 16-port switch. The switch will be able to power all APs with its PoE capabilities.

**Wireless Controller:**

The WLAN will need a wireless controller to manage the access points. There are 2 options for this:

* Utilise the Mobility Express feature in the Cisco Aironet 1850 series APs.
* Purchase and install a wireless controller device such as the Cisco 3504 or the Cisco 2504 devices.

***Note: This section is still in progress***

**References:**

ASB Stadium (n.d.). *Venue Hire for Sports*. Retrieved June 2, 2017, from <http://www.asbstadium.co.nz/sports-hire.html>

Cisco. (2017, May 8). *Cisco Aironet 1850 Series Access Points Data Sheet.* Retrieved June 2, 2017, from<http://www.cisco.com/c/en/us/products/collateral/wireless/aironet-1850-series-access-points/datasheet-c78-734256.html>

PostgreSQL. (n.d.). *PostgreSQL 9.2.21 Documentation Chapter 8. Data Types*. Retrieved June 2, 2017 from <https://www.postgresql.org/docs/9.2/static/datatype.html>

Intel. (n.d.). *Learn about Multiple-Input Multiple-Output.* Retrieved June 2, 2017 from <http://www.intel.com/content/www/us/en/support/network-and-i-o/wireless-networking/000005714.html>

Cisco. (2017, May 18). *Cisco Catalyst 2960-L Series Switches Data Sheet.* Retrieved June 2, 2017, from <http://www.cisco.com/c/en/us/products/collateral/switches/catalyst-2960-l-series-switches/datasheet-c78-737665.html>

Auckland Mathematics Association. (2014). *Casio MATHEX Entry Form 2014*. Retrieved June 2, 2017, from <http://www.aucklandmaths.org.nz/wp-content/uploads/2012/08/Mathex-Entry-form-and-Info-to-schools.doc>

Auckland Mathematics Association. (n.d.). *ASB Stadium Main Gym*. Retrieved from <http://www.asbstadium.co.nz/uploads/4/7/4/6/47460497/9258502_orig.jpg>