

Model United Nations



SCN:141881329

Jessica McClure

Abstract	5
Analysis	6
Project Proposal	6
Feasibility Study	8
Time Management	9
Environmental Concerns	13
Computing Research on Existing Products	14
Model UN App	14
wxMUN	15
Tikara	16
Survey Questions	17
Selection Strategies	27
Writing Tools	27
Programming Language	27
Programming Environment	29
Database	30
Scope	32
Boundaries	32
User Requirements	32
Chairs	32
Schools	33
Functional Requirements	33
Resources	35
Software	35
Hardware	35
Inputs	36
Outputs	36
Initial Test Plan	37
Design	39
Path of Program Flow Chart	39
Wireframes	41
Home page	41
Login	43
Registration Chair	46
Registration Delegation	49

Delete Profile	52
Voting	54
Award Pages	57
Personas	59
User Stories	60
Scenarios	61
Login Delegates	61
Login Chairs	61
Vote on delegates	61
Timer	62
View Results Delegates	62
View Results Chairs	62
User Centric Design - Use Case Diagrams	63
Class Diagram	64
Conceptual Model	66
Login	66
Display	66
Voting	67
Results	67
Sequence Diagrams	68
Connection to Database	68
Login	69
Timer	71
Voting	73
Results	75
Data Dictionary	76
Entity Relationship Diagram	82
Pseudocode	85
Implementation	118
Database	118
SQL Statements To Create Table	118
Java	120
User Interface	120
Connection to Database	120
Home Page	124
Login Page	126
Registration Chair	129

Registration Delegation	133
Delete Profile	137
Voting Form	140
Timer	145
Awards Committee	150
Awards Overall	155
Delegate Class	159
Delegation Class	163
Sorting Algorithm	165
Number of Countries	167
Total Score	168
Testing	171
Alpha	171
Home Page	171
Component Testing	171
Login Page	172
Component Testing	172
Deletion Of Profile	179
Component Testing	179
Chair Deletion	179
Delegation Deletion	182
Comprehensive Testing	183
Registration Chair	184
Component Testing	184
Comprehensive Testing	186
Registration Delegation	187
Component Testing	187
Voting Page	190
Timer Page	197
Component Testing	197
Comprehensive Testing	199
Awards Committee Page	200
Component Testing	200
Comprehensive Testing	204
Beta	206
Documentation	207

User Guide	207
Technical Guide	218
Evaluation	219
References	221

Abstract

This application is based around the extra-curricular club of Model United Nations(MUN). MUN is a club which is supposed to replicate the workings of the United Nations. This club allows people to represent a country and debate topics on various issues and create resolutions to solve and tackle the problems presented.

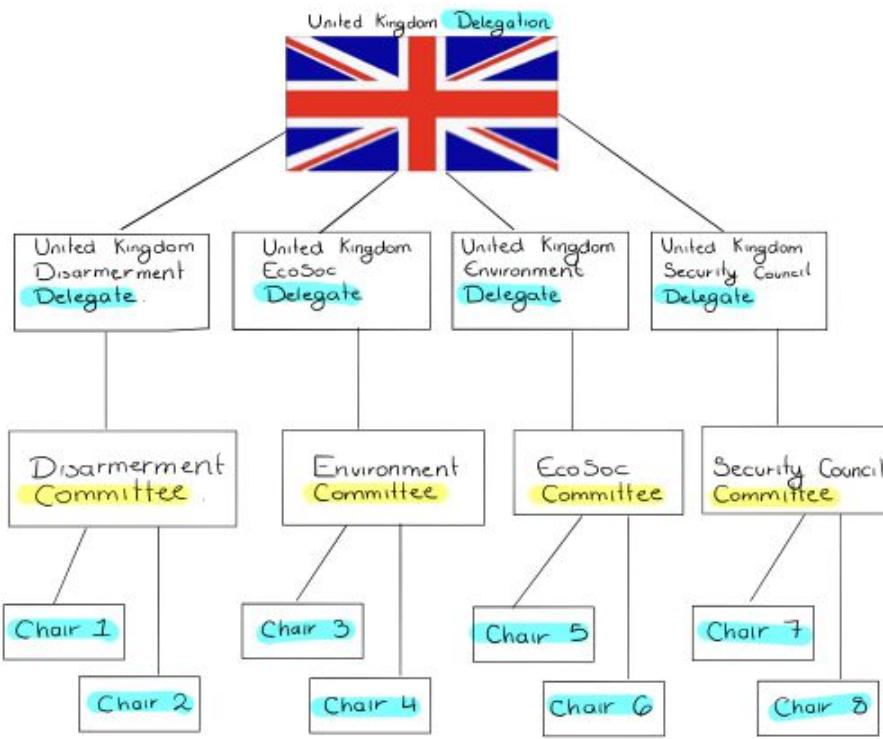
In the United Nations there are many different sub-branches called committee's. Take for example the well known Human Rights Committee. This committee is most well known for creating the *Declarations of Human Rights* and *The Geneva Conventions*. In my application the committee's which are at the conference is Disarmament, Economic And Social Committee, Environment and Security Council.

Each committee can have multiple Chairmen called chairs. The chairs job is to mediate the debate and ensure a good structure to the debate.

Each committee will have a representative from a country. This representative from that country is called a Delegate.

The country as a whole with all of the members from the different committees is called a delegation.

On the right is a diagram showing how the positions interact and come together:



Analysis

Project Proposal

I will create an app which is used at a Model United Nations (MUN) during conferences. This app will be used by the chairs (the people who run the committee sessions) to give an overall ranking of all countries and sort out of which delegate should deserve a prize. MUN is a club that is normally targeted at high school or university aged pupils so this will be my target audience. This ranking will be based on a points system in which there will be different categories for the chair to score the delegate who is speaking. All of the different categories will be given a different weighted depending on how important it is to the final score. The delegate with the highest overall score will receive the prize of best delegate. Other prizes such as best humor may also be awarded.

Each chair will have the ability to register and login into the app. When the chairs initially register they will have to say which committee they are chairing for since they can only vote on their committee. The chairs will have access to the voting page and a timer. The chairs will also have access to the overall awards but will only have access to their own committees awards. The delegation will have a chance to register and login into the app. The delegates will not have access to the voting or timer page but will have access to the awards overall and all the committee awards.

There is no current technology like this. The closest existing technology is a timer for the chairs to make sure the debate is well structured however there is nothing that actually helps in the giving of awards. This is important as it takes away any last minute biased views towards particular countries. This is why another feature that I would add is the ability for a random country to be picked for speaking so chairs aren't biased towards any particular country.

Delegations will also be able to register and login into the app. Once they login they will be able to view one of their delegates overall score in committee and the whole delegations combined score. They will not however be able to see the individual scoring on a delegate at a particular time. Total scores will be calculated by adding up the information in the voting forms submitted by the chairs throughout the conference. This information will be held in a database and can only be accessed by the teachers at the school.

My project can be split into four different sections:

1. Login and registration and deletion of profiles
2. Timer and structure section to put on the board for the chairs to help run the debate.
3. Voting section for the chairs to have on their phone or on a different laptop.
4. Awards and Ranking.

Feasibility Study

During my project I will have to consider three important factors of feasibility: technical, economical, legal and schedule.

I have access to a laptop which has the appropriate software I would need to complete my program. This software is Netbeans IDE 8.2. I also have enough storage space to run my project. Some of the features such as the ability to randomly select delegates during the conference would require the use of a camera to interact with the system and that is not a required task at this level nor do I have the equipment available.

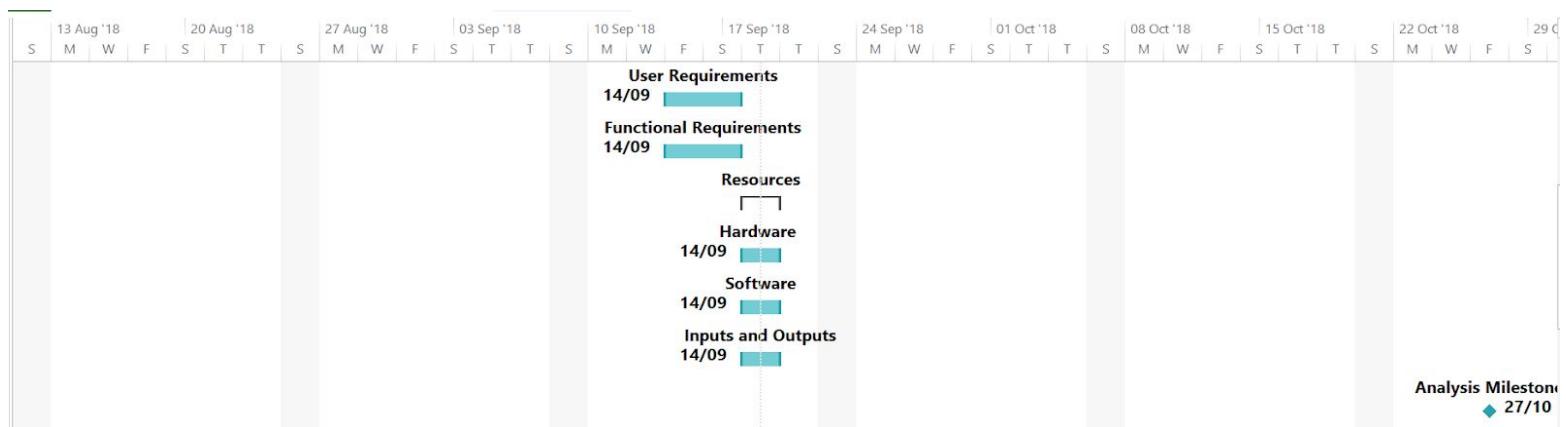
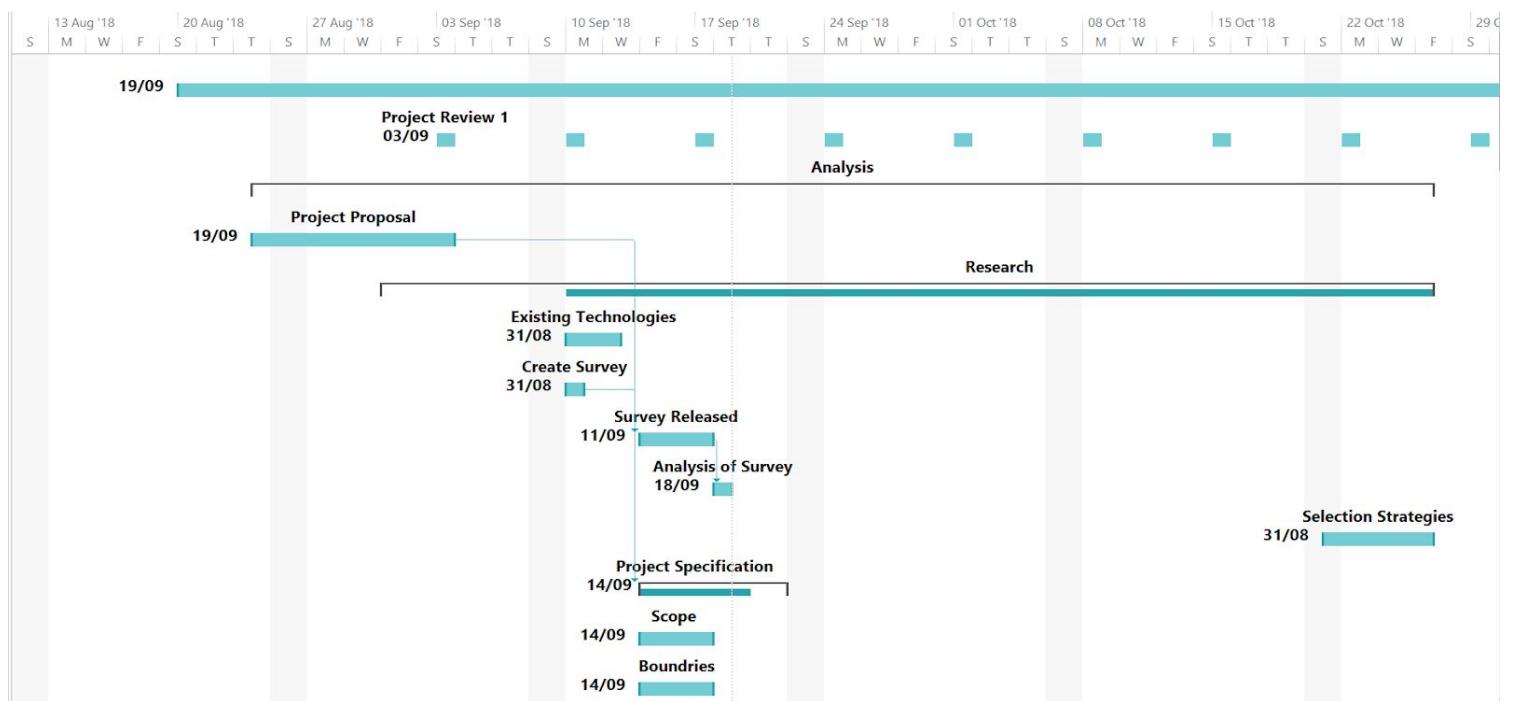
There is no economical benefit or loss in the creation of my project. If my project were to be released then it would require no purchase and would be freeware and therefore would have no financial reward. No money is lost either because all the equipment I required was purchased by the school and so there is no loss to my own budget.

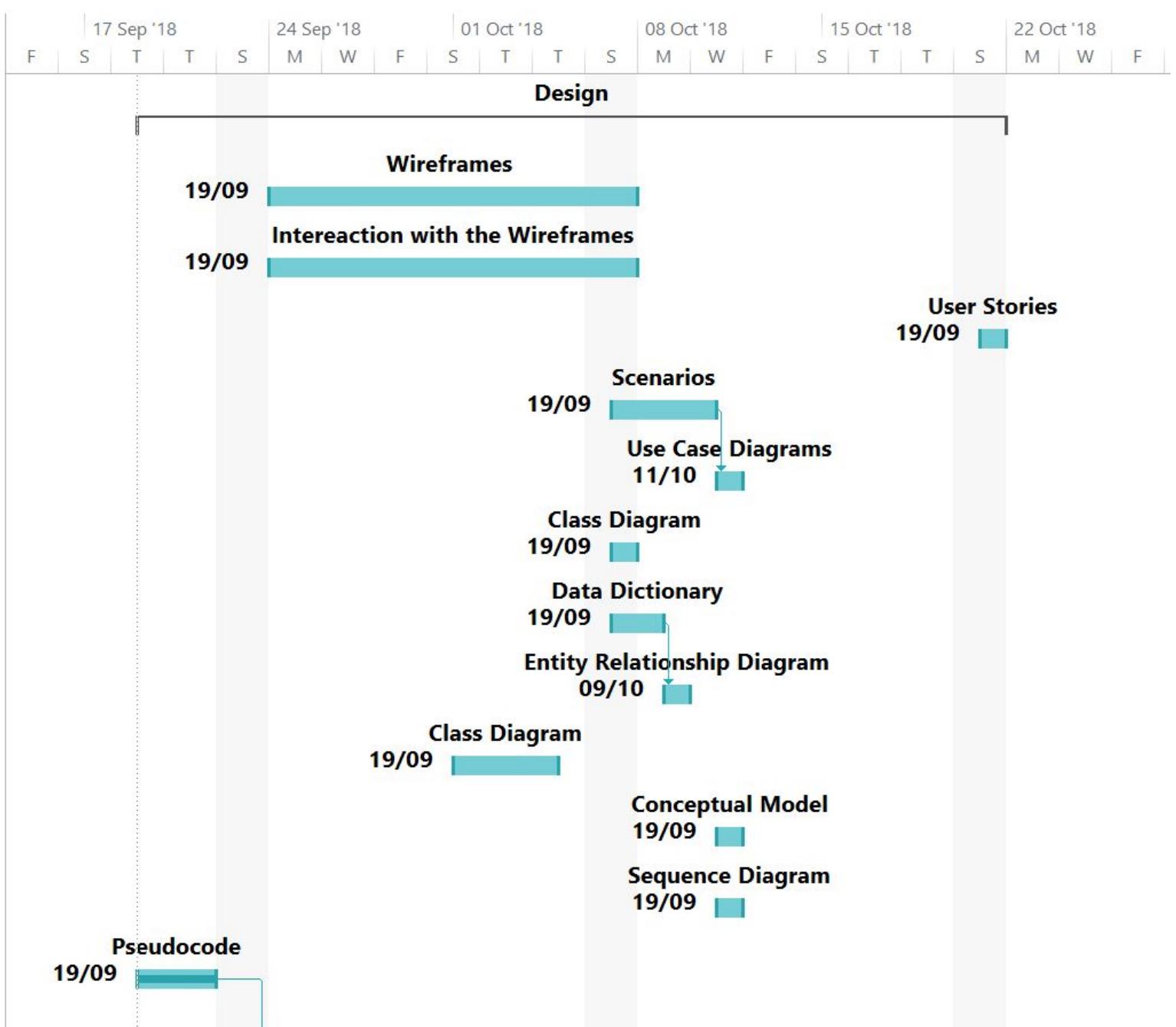
My program will adhere to existing laws. Since information is being stored in a database then one of the main laws of concern is the General Data Protection Regulation however when looking at what information is being stored in thata database none of it is sensitive. In the implementation of my program however, if I were to use any parts of code on the internet to help and assist me (such as the building of a timer since this wasn't included on the silybius) then I would need to make sure to edit the code enough to make it my own and make sure to reference the original creator of the code. If I were not to follow through on that then I would be infringing on *intellectual property rights*. If my idea wasn't original then I would also be in violation of *intellectual property rights* and perhaps even on the *copyright design and patents act*.

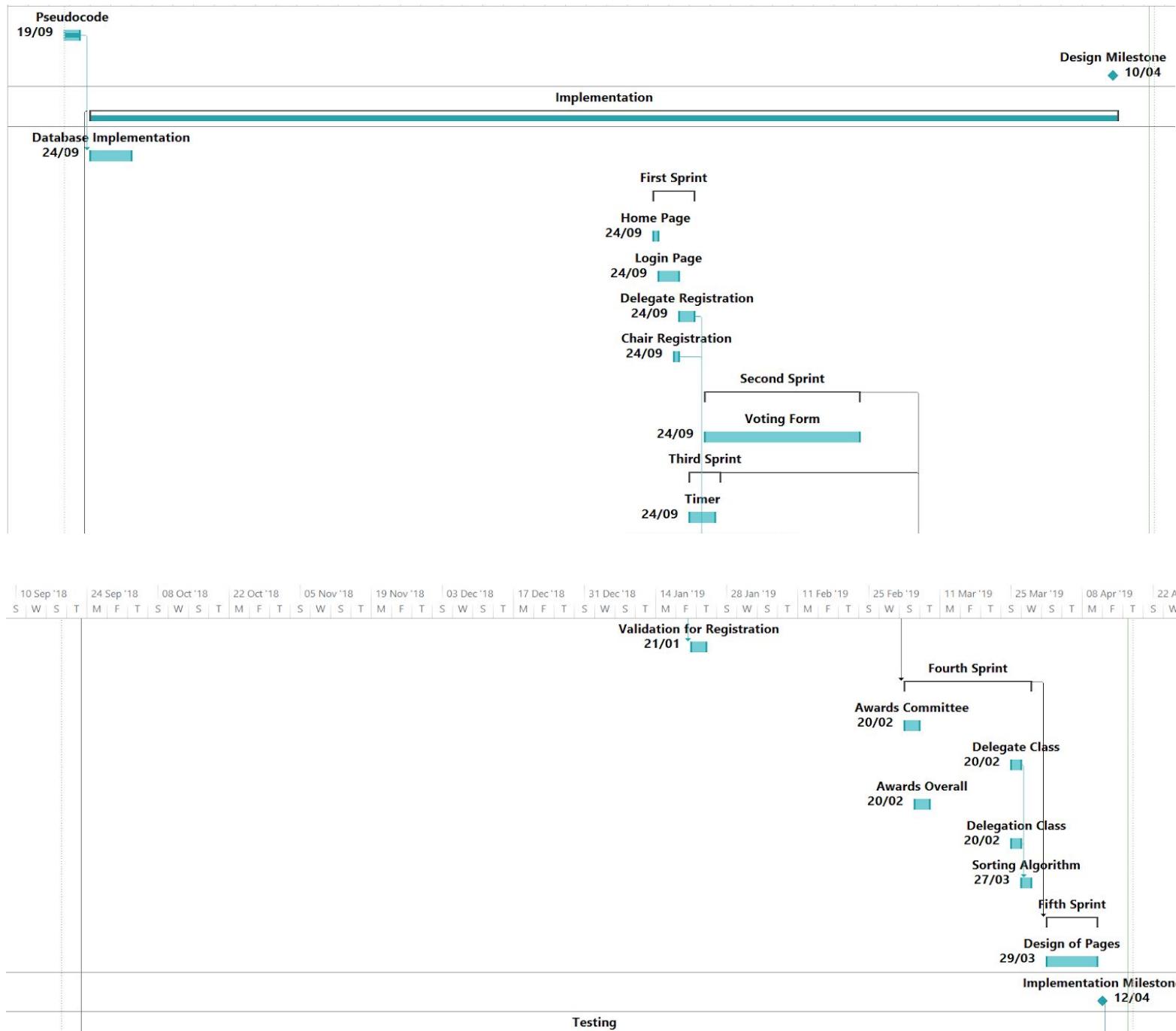
A main concern is the schedule. The project initially started on August the 24th and is due in on April the 17th. This gives about eight months to complete the whole project. However the whole eight months won't be used in the implementation of the project since all the knowledge required to built the program needed to be learnt in this time as well. All the analysis, design ect. also needs to be completed in this time as well. Due to the time scale that cannot be negotiated there may be features that I can't include do to the time constraints.

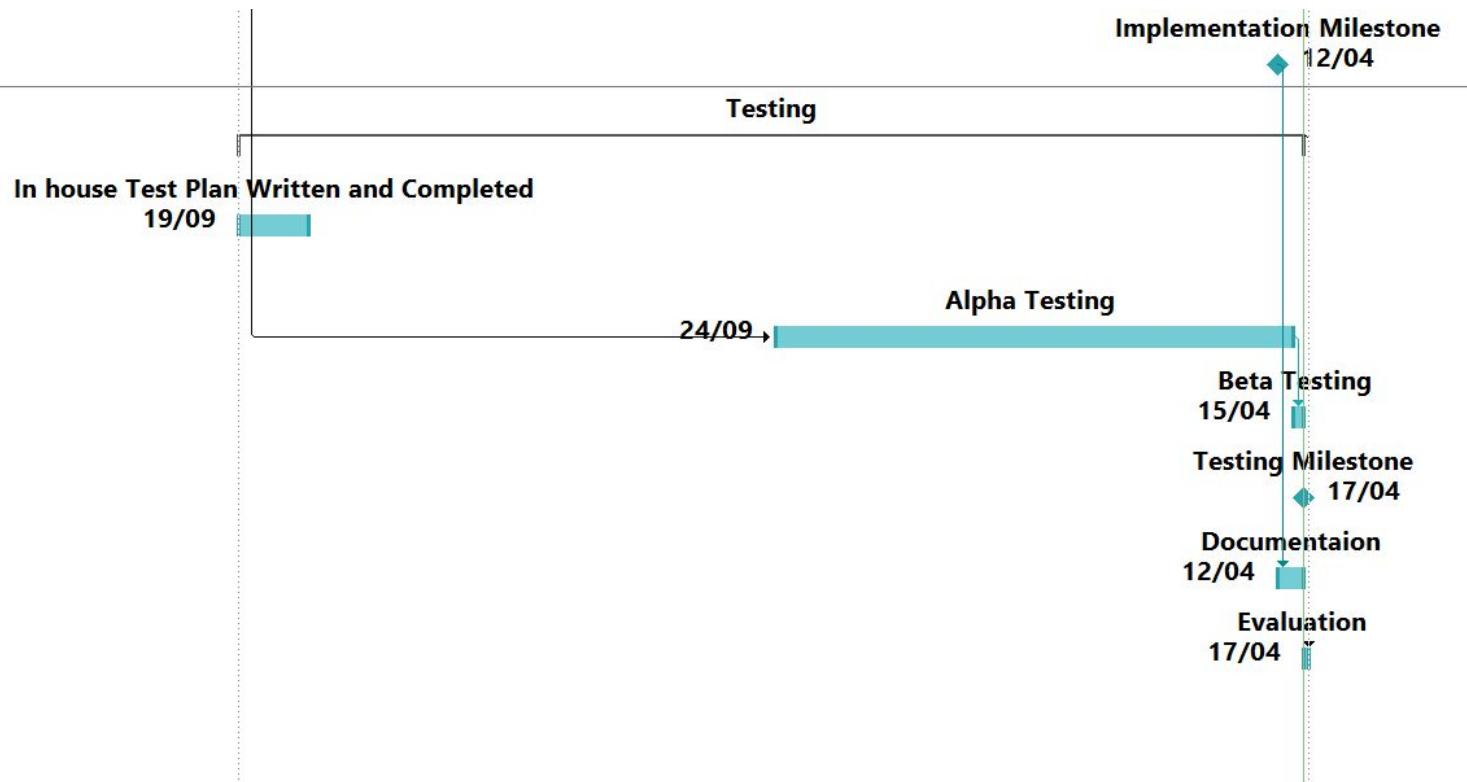
Time Management

To manage my time I create a Gantt chart on Microsoft Project. This is where I planned and scheduled all my tasks. This corresponds to my progress diary.









Environmental Concerns

Not only is there environmental concerns for the use and running of my application but also of the implementation and write up of my project.

MUN conferences last a long time, they usually last a couple days and the voting and timer page will be used almost consistently throughout that day. This means that the program will need to be running constantly and therefore a computer will need to be charged and connected to power. This means that the amount of electricity being used to charge the computer is contributing to the carbon footprint of the machine.

We would suggest that the machines used to run the program would be environmentally friendly by having eco-friendly processors. In 2007 the first environmentally friendly processor was released by intel called the Xeon processing chip which was the first processor which was halogen free.¹ This first environmental processor was released in 2007 and so there has only been environmental advancements in the industry since then.

Although this may not affect my application there are a lot of concept environmentally friendly computer items in which in the future could happen.² Solar paneled laptops would be effective in charging computers and therefore the carbon footprint would be dramatically reduced. This being said solar powered laptops wouldn't be that useful in the concept of my application as most MUN conference take place indoors.

¹ <https://www.tomshardware.co.uk/intel-Xeon-Eco-friendly.news-29117.html>

² <https://greendiary.com/7-eco-friendly-computer-concepts-show-greener-side-technology.html>

Computing Research on Existing Products

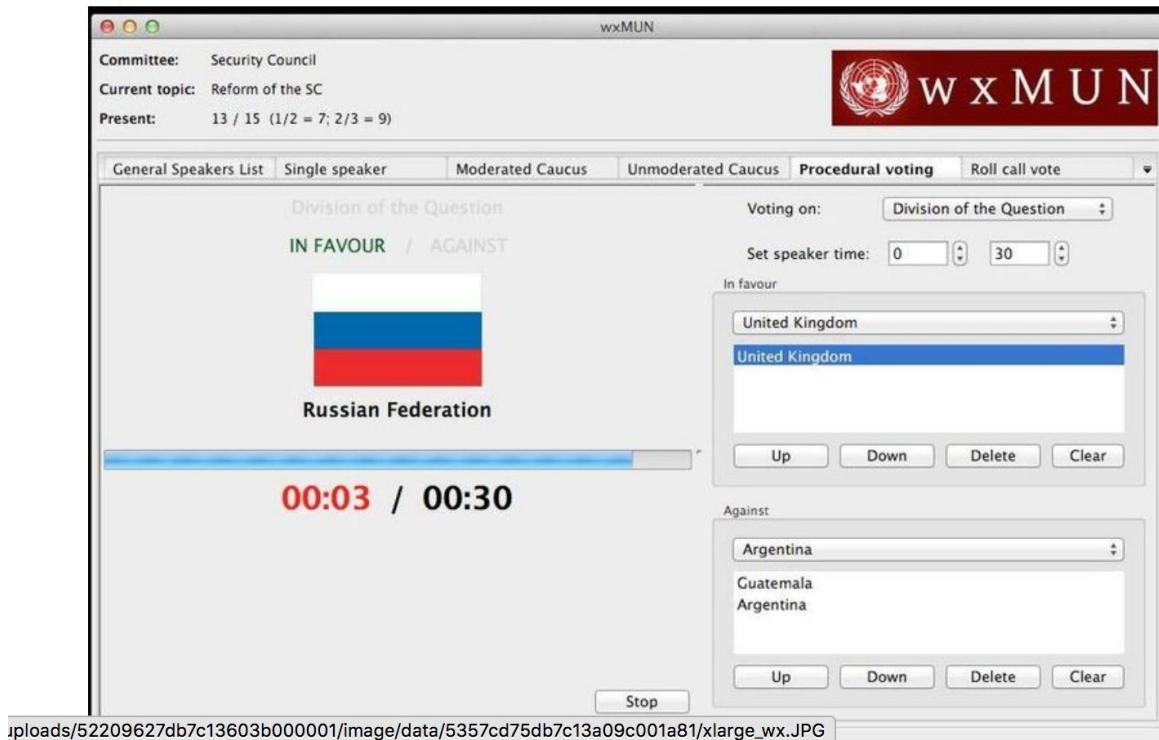
Model UN App³



This is an app used primarily for the delegates at the conference and not for the chairs. This app also isn't used for during a conference but just for MUN in general to show useful, basic, information. This app isn't for my desired audience and isn't aimed to use during the debate but more for use for research before the conference.

³ <http://bestdelegate.com/4-useful-apps-for-model-un-students/>

wxMUN⁴



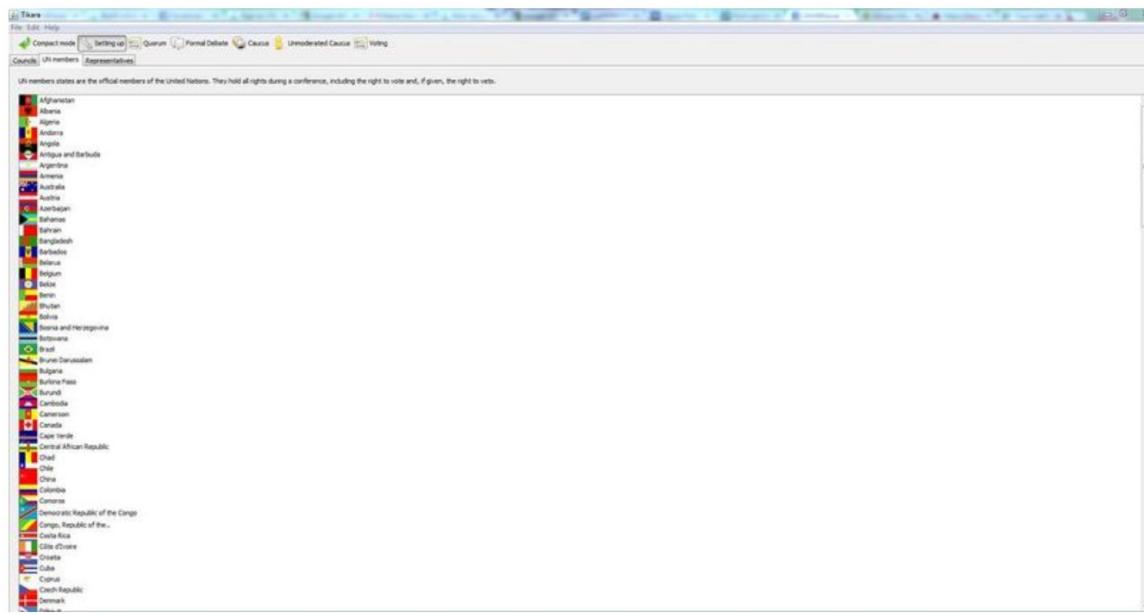
This app is more for the chairs. This has the use of a timer so that the debate is well structured a feature I would also like to incorporate into my app however it has no scoring system for the chairs. This app shows which committee the app is in use for however this is just typed manually and there is no login to the app. The app can only be used by one chair at the time as well and there is no history features to see what past times have been scored.

This piece of software was used during a previous conference and worked really well. I was initially unsure whether to use my app in correlation with this app or have a separate section like this in my software. I decided to have a separate timer section in my app so that the timer results could be stored on the database and used when calculating the total score.

⁴

<https://www.munplanet.com/questions/mun-technology/what-is-the-best-software-to-run-a-model-united-nations-committee>

Tikara⁵



This app has a timer mode for each individual section of the debate and gives you the ability to choose a country. Choosing a country is a feature I would like to incorporate however I would like the timer list to be niche depending on what countries were registered.

Although I will have a timer included I think this timer would be used all round and wouldn't be used for individual sections as this would seem to be too much of a hassle for the users.

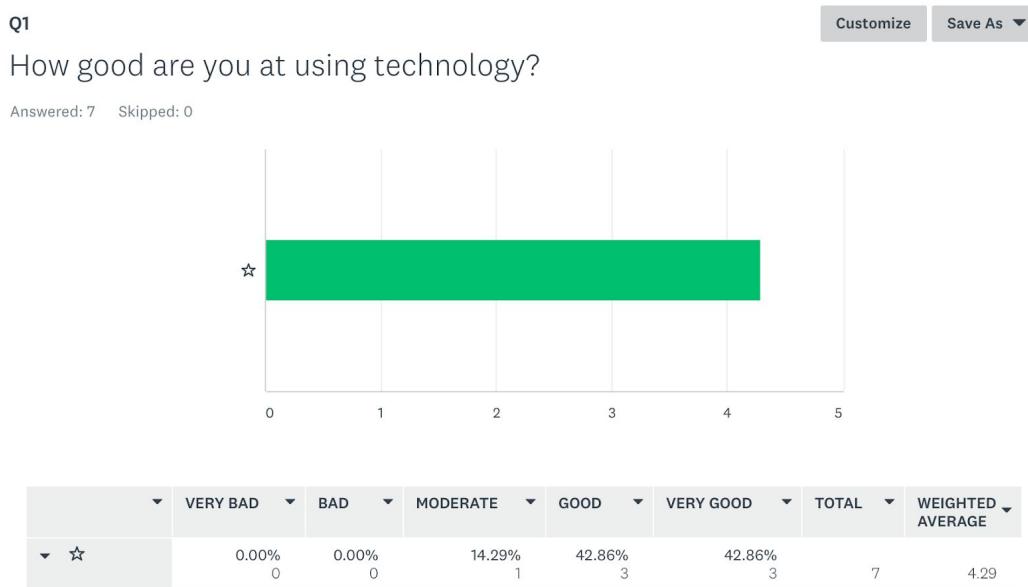
5

<https://www.munplanet.com/questions/mun-technology/what-is-the-best-software-to-run-a-model-united-nations-committee>

Survey Questions

Only aimed at MUN members who are aware of how a conference works and may have/will have in the near future experience in chairing.

1. How good are you at using technology?



The overall conclusion was that the majority of people felt good or very good about their ability to use technology. Since this application was aimed at high school MUN club members most felt pretty confident in using technology. This was expected. This means that I could perhaps put more complex features in my application however this will probably be unlikely as the simpler the application is the easier it will be to use especially during a debate when the application is used as a tool to make chairing a debate easier. The less concentration it takes to use the application the better.

2. Have you ever chaired a conference or are at a level in which you could chair a conference?

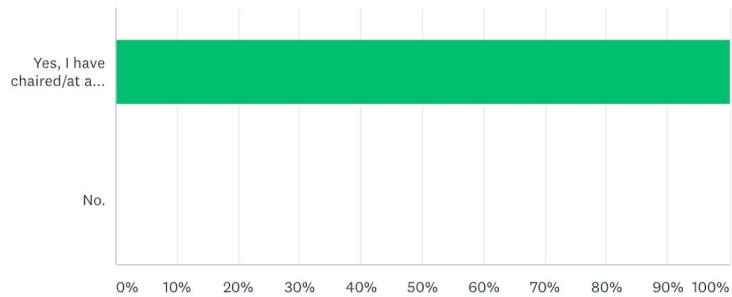
Q2

Customize

Save As ▾

Have you ever chaired a conference or are at a level in which you could chair a conference

Answered: 7 Skipped: 0



ANSWER CHOICES	RESPONSES
▼ Yes, I have chaired/at a level to chair a conference	100.00% 7
▼ No.	0.00% 0
TOTAL	7

This question was aimed to make sure that the desired audience I would be targeting (potential chairs of debates) were giving informed suggestions on how I could make the application. As seen by the 100% response to 'I have chaired/ at a level to chair a conference' the survey went to the desired audience and all of the responses were well informed and all responses should hold value.

3. Have you used a piece of software before when chairing a debate (if so please specify)?

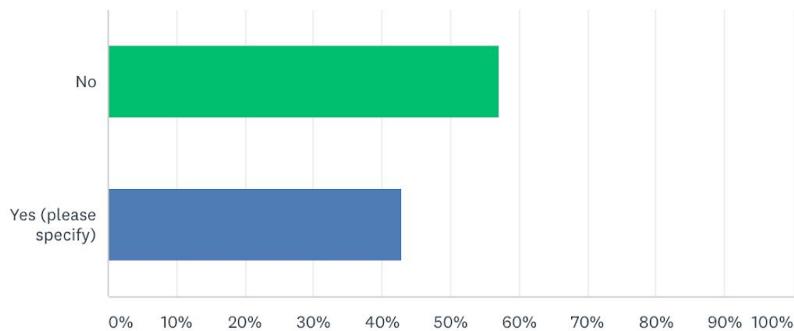
Q3

Customize

Save As ▾

Have you used a piece of software before when chairing a debate (if so please specify)

Answered: 7 Skipped: 0



ANSWER CHOICES	RESPONSES	
▼ No	57.14%	4
▼ Yes (please specify)	Responses	3
TOTAL		7

I wanted to see that if I were to make an app whether it would be useful or if the old method of pen and paper would be more used. I was surprised to find out that the amount of people using technology while chairing was only one less than those who haven't used technology. This was a positive answer as it showed that people would be willing to use and get to learn how to use my application.

The responses that people wrote about technology which they have used was looked at in the 'research existing technologies' section.

4. What feature would you find useful in this app (i.e. Timer) ?

Q4

Save As ▾

What feature would you find useful in this app (i.e. Timer)

Answered: 7 Skipped: 0

RESPONSES (7) TEXT ANALYSIS TAGS (0)



Timer

9/17/2018 1:57 PM

[View respondent's answers](#)



Timer, speaker list, being able to label the topic of debate,

9/15/2018 10:46 PM

[View respondent's answers](#)



Timer, speaker's list, ranking option(scoring delegates for quality of speech/amendments that then organised them) and resolution writing section

9/14/2018 5:20 PM

[View respondent's answers](#)



timer and warning for time

9/14/2018 5:00 PM

[View respondent's answers](#)



A decorum. A tally system that would only be shown to the chairs showing how many times each delegate has spoken.

9/14/2018 2:21 PM

[View respondent's answers](#)



Complexity of words

9/14/2018 1:25 PM

[View respondent's answers](#)

Although I already had my ideas of what features I wanted to include I wanted to know if other people thought the same or had any completely different and thought provoking ideas. Although a couple of responses included the 'timer' option (an important feature I wanted to include) I think this was mainly because it was prompted in the question. That being said people did like and agree with the idea.

A suggestion was a 'speaker list' this is included in my application in the drop down boxes from the timer and even in the voting page. Although this wasn't initially part of the design I decided this would be effective and increase the usability of my application.

I haven't included 'labelling the topic of debate' as I didn't think this was crucial in how my application ran. It would be a slightly unnecessary feature however could still be added in future updates of my app.

The 'ranking option' is included in my application as the ranking and sorting of the delegate and delegations total score will be ranked from first to last using a sorting algorithm. The response to the question may have meant that the chair wanted to be able to rank the delegates themselves manually instead of a total score being calculated however this would be defeating one of the main purposes of my app. If the chair was able to rank the delegates after the voting then they providing a biased viewpoint and the whole point of voting was to take away some of that initial biases.

A 'resolution writing section' would actually be useful for when the delegates login however considering currently I have a initial delegation login I don't know how this would be implemented. If I had time at the end however I would like to include a feature like this.

A 'warning for time' feature would be very good to include in my timer feature. The colour of the text on the count down could change once the delegate speaking gets to 10 seconds or something along those lines. This will be a feature I would like to include in the timer.

A 'decorum' would be useful point to judge on in the voting page however to have a whole section of the app dedicated to decorum seems unnecessary in this initial release of the app.

A 'tally system' would be a very useful and good section to have in the app however it might create and cause clutter on the screen and become stressful to use for the chairs. I might try and incorporate a feature like this if I have time at the end.

'Complexity of words' would be a good quality to include in the voting sheet however a whole section in the app wouldn't be useful.

5. Do you find yourself biased toward a particular delegate at a conference?

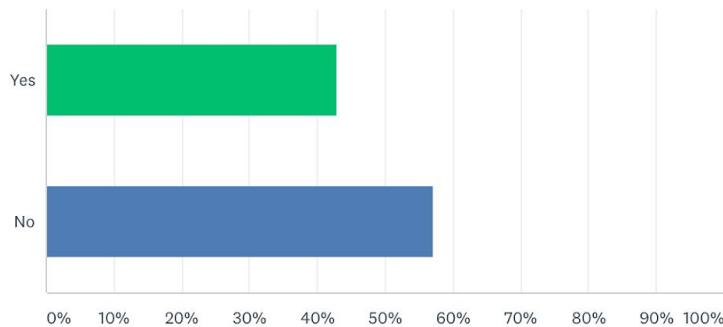
Q5

Customize

Save As ▾

Do you find yourself biased towards a particular delegate at a conference?

Answered: 7 Skipped: 0



ANSWER CHOICES	RESPONSES
▼ Yes	42.86%
▼ No	57.14%
TOTAL	7

This response was slightly unexpected as I thought that the majority, if not all the people, the response would be that they found themselves biased. This being said the 'no' response only gained majority by one more vote so biassed views can still be seen as an issue and so this is still an important feature to include in my app.

6. Would you find it useful for an app or software to randomly select a country (that has there plaque up) to get rid of any bias opinions?

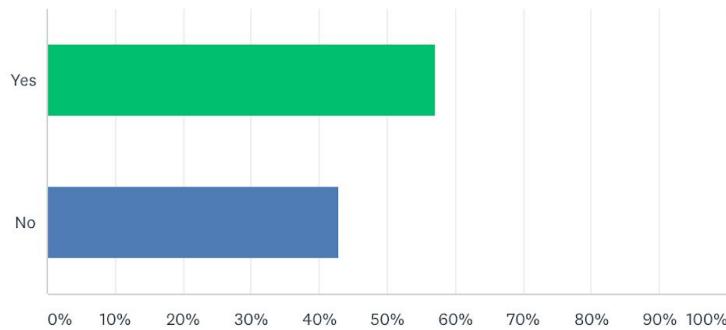
Q6

Customize

Save As ▾

Would you find it useful for an app or software to randomly select a country (that has there plague up) to get rid of any bias opinions ?

Answered: 7 Skipped: 0



ANSWER CHOICES	RESPONSES
▼ Yes	57.14%
▼ No	42.86%
TOTAL	7

I did pose the question and thought this would be a very good idea at the time however after I started implementation I realised that this wouldn't be very practical to code or an effective use of the program. Although the thought to randomly select countries is very good to select only countries who have a plaque card raised is very impractical. It would take longer to select which countries have their plaques up than to actually just pick the person. This is why I have decided to drop the feature even after the responses agreed and thought this would be a good idea.

7. Would you find a ranking system useful to help select a delegate for an award?

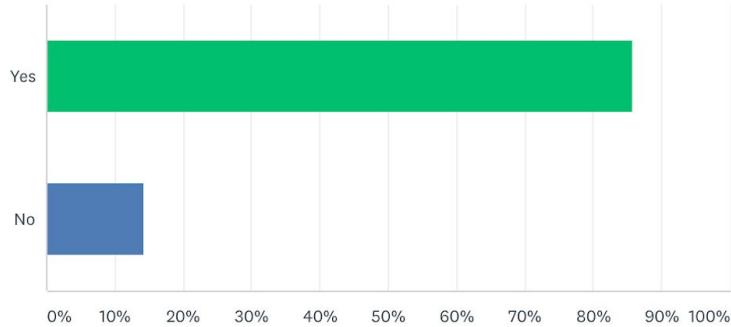
Q7

Customize

Save As ▾

Would you find a ranking system useful to help select a delegate for an award?

Answered: 7 Skipped: 0



ANSWER CHOICES	▼	RESPONSES	▼
▼ Yes		85.71%	6
▼ No		14.29%	1
TOTAL			7

This was an expected response as this is one of the key features of my program. The ranking of delegates will be viewed on a separate page. Awards will be based off of the ranking scores and both the rankings and awards will be displayed.

8. Which qualities do you think should the ranking be based off?

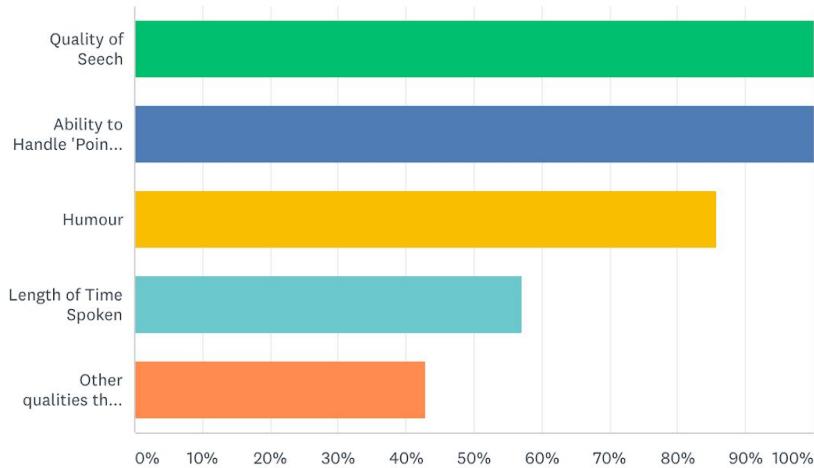
Q8

Customize

Save As ▾

Which qualities do you think should the ranking be based off?

Answered: 7 Skipped: 0



Knowing the topic very well, encouraging debate and newer delegates

9/17/2018 10:06 PM

[View respondent's answers](#)

Engagement in debate

9/17/2018 1:57 PM

[View respondent's answers](#)

Number of times they have instigated the debate (been the first one to speak up on a situation)

9/14/2018 2:21 PM

[View respondent's answers](#)

The main sections that I need to rank off of since there was a 100% agreement that these qualities should be included: 'Quality of Speech' and 'Ability to handle Point of information'. I will also include 'humour' as there will be a 'Funniest Delegate' award and therefore this quality will be used to give the award. I also decided to include the suggestion of 'number of times they have instigated debate'. I decided not to include the engagement in debate as this would be in parallel with the number of records and voting forms submitted.

9. Do you think you would find the entering of data during a conference tedious?

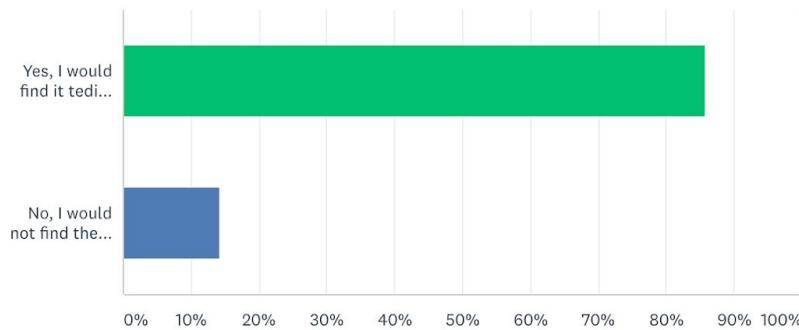
Q9

Customize

Save As ▾

Do you think you wold find the entering of data during a conference tedious?

Answered: 7 Skipped: 0



This question was very important because it would determine how often people would use the app. When creating the voting form I need to make sure that the number of clicks and buttons required to submit the form is minimum otherwise the whole process could become annoying and people wouldn't want to use my system.

Selection Strategies

Writing Tools

Name	Microsoft Word
Description	This software package is used to write up documents.
Advantages	Tools and options to write up report such as the referencing tool as well as the ability to apply footers.
Disadvantages	Can't use cloud computing.

Name	Google Docs
Description	This software package is used to write up documents.
Advantages	Cloud Computing therefore it will be easy to continue my work if I am on different computers.
Disadvantages	Not as many editing tools such as not being able to apply different footers to the pages.

I chose to use a combination of both Google Docs and Microsoft Word. I will use Google Docs while editing and creating my report so I can make use of the cloud computing aspect so I can access my report on multiple computers. I will use Microsoft Word before finally printing off my report so that I can make use of the footers and referencing tools.

Programming Language

Name	Java ⁶
Description	A high level object-oriented programming language. This language was developed by <i>Sun Microsystems</i> . This language is popular for creating web applications.

⁶ <https://techterms.com/definition/java>

Advantages	Java applications can run on multiple platforms due to the fact that Java programs are interpreted by a Java virtual machine instead of the the java program being run on the computing system. Object oriented therefore concepts such as classes, objects, inheritance, encapsulation and polymorphism can be used. Modular code can be used to make coding and the running of the program more efficient as well. ⁷
Disadvantages	More memory consuming and can be slower when compared to other languages such as C or C++. ⁵

Name	Visual Basics
Description ⁸	This is an event-driven language which is currently in its third generation. This language was created by Microsoft. The program promotes the use of Component Object Model (COM). COM enables inter-process communication object creation.
Advantages ⁹	Very simple and straightforward syntax and easy to initially understand. A lot of online help as this is application is made by Microsoft and this is a big company which will provide a lot of online services and support technicians. Supports the use of Rapid Application Development.
Disadvantages ¹⁰	High memory constraints as it is based around a Graphical User Interface development and therefore requires a lot of space to use this preset controls such as textbox. Not good when you longer applications need to be processed. Harder to use objects as this isn't an object-oriented language so you cannot achieve as complex programs.

⁷ <https://www.mindsmapped.com/java-advantages-and-disadvantages/>

⁸

⁹ <http://lavape.sourceforge.net/doc/html/AdvantVB.htm>

¹⁰ <https://www.techwalla.com/articles/the-advantages-disadvantages-of-visual-basic>

I chose to code in Java. This was mainly due to the fact it was an object-oriented programming language compared to Visual Basics which is an event-driven language.

Programming Environment

Name	Netbeans IDE
Description	Software which allows you to program in many different languages such as Java, C++, PHP, Groovy.
Advantages ¹¹	<p>Free and open source software.</p> <p>Integrated development modules which means there is access to features such as debugging tools.</p> <p>Includes SWING GUI tools which makes designing an interface very easy.</p> <p>Auto completion of code which makes coding easier as this gives coders possible endings to their code.</p> <p>Versioning tool is available which allows for the ability to look at past version of your code and what has been done to change it, what has been edited.</p>
Disadvantages ¹⁰	<p>Known to take longer to load initially.</p> <p>Application takes up more storage on memory to install.</p> <p>Using the advanced tools that are available can require more learning and can be too complicated to use and so the initial advantage of having the advanced tools is lost.</p>

Name	Visual Studio
Description	An environment created by Microsoft. This environment allows you to code in many different languages including languages such as C++, visual basics and C#.
Advantages	<p>Has many reviews saying how the environment is easy to use and therefore the environment is very accessible and user-friendly which is good.¹⁰</p> <p>Efficient compilers.</p>

¹¹ <http://infotech101.com/the-netbeans-ide-pros-cons/>

	Debugging tools .
Disadvantages	Some reviews have reported crashes whenever working on large programs. ¹²

I will use Netbeans IDE to implement my program. Netbeans IDE also allows for the ability to program in Java which was the programming language that I choice. The programming environment is also better when programming larger applications which my program will be and I don't want to run the risk of my program crashing which is why I will used Netbeans IDE.

Database

Name	MySQL
Description	This is a relational database management system. This is written in a structured query language.
Advantages	Free and open source. Quick processing, good reliability, easy and flexible to use. This is a portable system meaning it will run on any operating system.
Disadvantages	Difficult to debug as no debugging tools are provided. Difficult to maintain. Not good for complex and flexible business logic.

Name	Oracle Database
Description	Multi-model database management system created by the company, Oracle. This is most commonly written in languages such as C or C++.
Advantages	Since produced by the Oracle company there is great customer service if stuck. Used to store complex databases and can be found to be used in banking. High data storage therefore very reliable.

¹² <https://www.trustradius.com/products/visual-studio-ide/reviews/pros-and-cons?f=25>

Disadvantages	Have to pay for licensing to use the system. Since complicated to use it may be hard to find people who specialize and can manage a database in this language.
---------------	--

I have chosen to use MySQL. I decided primarily to use this database as I have worked with it before since and am familiar with the environment and the language.

Scope

My solution will be able to score delegates in their overall performance at a conference. They will then be ranked in their committees and prizes will be awarded accordingly. Prizes such as best delegate, best junior delegate, highly commended and commended will be awarded. The prizes will be given to those in the corresponding ranking position. Ranking will be determined by a delegates overall score. This will be calculated by the chairs scoring (1 to 5) on categories such as humour, quality of speech, ability to handle points of information and length of time spoken. My solution will also incorporate a timer that can be used during committee sessions so that debates run more efficiently. The chairs in the committee will also be able to randomly select countries when selecting who should speak at the front to reduce bias.

Another function that my solution will be able to do is that schools can look back on past conferences and see the schools overall score as a whole compared to other schools that attended that conference.

Boundaries

My solution will be most functional by the chairs and not the delegates. Delegates will have a login however they will not be able to view their scores and ranking until after the conference is completed. The delegates will be able to view the overall results as well as all the different committee results. Chairs will be able to complete voting forms and make use of a timer. Chairs will be able to view the overall results however will only be able to view their committee's results.

User Requirements

Encryption will be used to ensure security when accessing the different interfaces. This will be done in the form of a username and password.

Chairs

Chairs will use one of the interfaces as a timer and as a way to control the structure of the debate. They should display this on a screen that can be seen by all the delegates. The chairs will be able to set specific speaker times and will be able to use this as an overall voting of the procedure.

The chairs will be able to use the other interface on another computer. This will be used as a voting system. The delegates should not be able to view this system and although it will be protected via a username and password it is up to the chairs to make sure the delegates can't physically see it. The chairs will be able to keep a tally of who is making a point of information but most importantly rank delegates on how they present their speech. They will be able to vote on categories such as quality of speech, humour, ability to handle points of information. The length of time spoken will also be added in the calculation of the final score but the data will be taken from the first interface.

Schools

Each school will receive a login and password and will be able to access the third interface. This will allow them to view how well their delegates performed at the conference and will be able to see the exact scoring of the delegates. The school will be able to see how well their delegation did at the conference compared to other schools as this too will be ranked. It should be noted that the school will only be able to view their own delegates scores and no one else's from other schools.

The hosting school will be able to see all the scores and an output of best delegates in every committee will be displayed on the screen along with any other awards.

Functional Requirements

The chairs and delegates should be able to login to their respective interfaces and this login should be validated so only the correct people get into the interfaces. The chairs should be able to vote on every delegate multiple times and judge them on categories such as humour, quality of speech, ability to handle points of information. The program should calculate scores from all of these categories plus the time spoken by the delegate. The program should then output the ranking of all the delegates per individual committee using the final score to the scoring interface which can be accessed immediately by the hosting school. The sum of the delegates score from a single delegation(country) will be calculated and the delegations ranking and school will be outputted.

- Login (chairs and delegates go to respective pages)
- Chair registration
- Delegation registration
- Voting Page
 - Score on humour

- Score on quality
 - Score on ability to handle POI
 - Did they Instigate Debate
- Timer Page
- Ranking of each individual delegate in their committee
- Awards (for each committee)
 - Best Junior Delegate
 - Best Overall Delegate
 - Highly Commended Delegate
 - Commended Delegate
- Overall rankings for delegations as a whole
- Awards (for the whole country)
 - Best Overall Delegate
 - Highly Commended Delegate
 - Commended Delegate

Resources

Software

I plan to write my solution in Java NetBeans. And build the interface in Java Netbeans as well using Java Swing. PHPmyAdmin will be used to host my database for holding the rankings scores as well as the logins and passwords. SurveyMonkey was used to build and the link was used to distribute my survey. Microsoft Project 2013 was used to produce my Gantt Chart. Microsoft Visio 2013 was used to make some of my diagrams my diagrams apart from the wireframes in which <https://mockflow.com/apps/wireframepro/> was used. Microsoft Word 2016 and Google Docs were used to type the report.

Hardware

The hardware I will be using to build the solution is an 'Ultrabook ASUS laptop'. A SQL server will be required to host my database to store the history of delegates scores in the conference as well as hold all account details.

Inputs

- Username
- Password
- Timings
- Chairs scoring values (1 to 5, one being the worst), judged on categories such as:
 - Humour
 - Quality of speech
 - Ability to handle points of information

Outputs

- Best Delegate award (per committee) on screen
- Highly Commended Delegate award (per committee) on screen
- Commended Delegate award (per committee) on screen
- Best Junior Delegate award (per committee) on screen
- Best Delegation Overall award on screen
- Highly Commended Delegation Overall award on screen
- Commended Delegation Overall award on screen
- Ranking of all delegations on screen
- Ranking of all delegates (per committee) on screen
- Delegates individual scores

Initial Test Plan

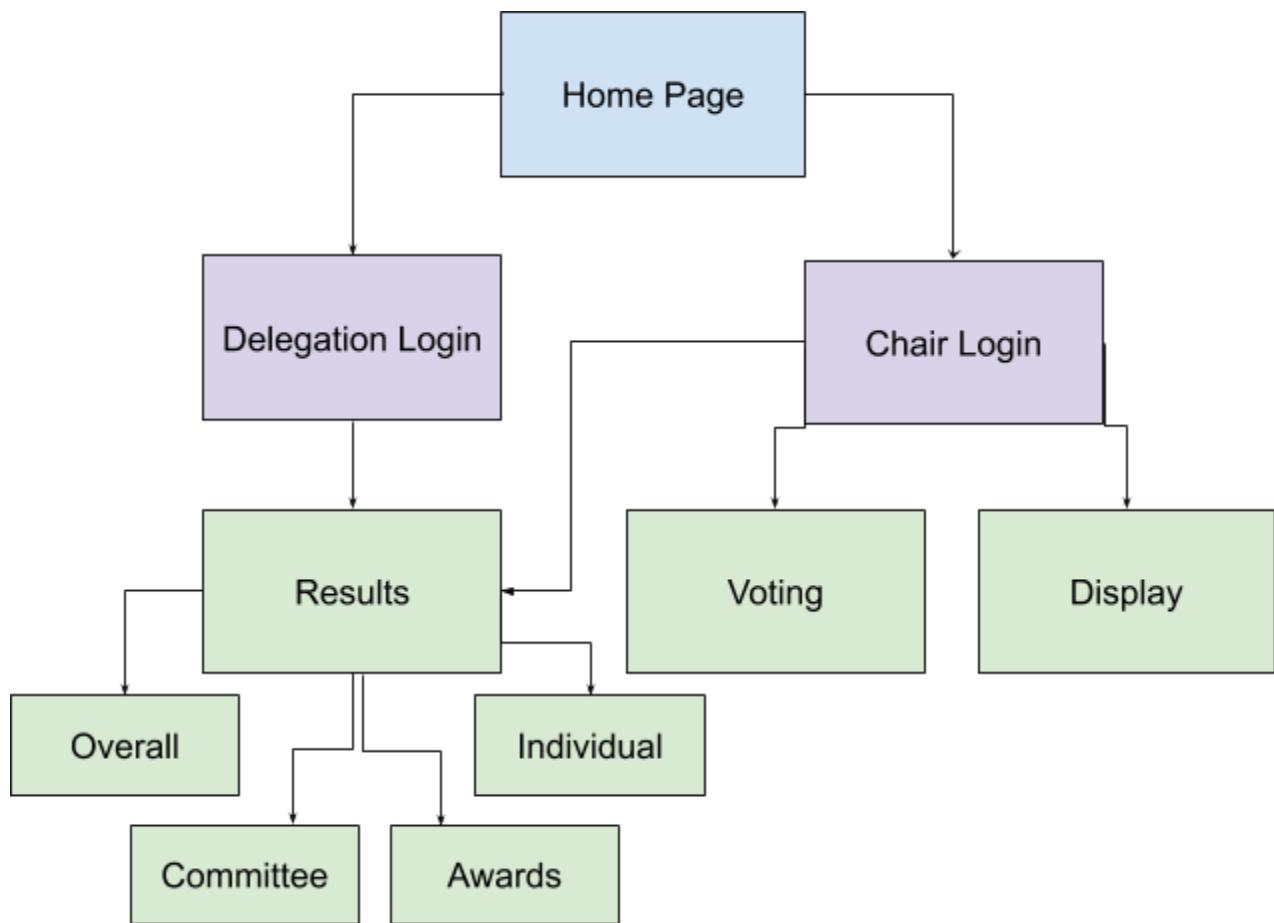
What is to be tested?	How is it to be tested?	Test data/ expected data?	How results will be recorded?
The login section for both interfaces.	Setting up usernames and passwords that are encrypted and validated when entered.	If the username or password doesn't match what it is in the database an access denied message will appear. Also not all logins will work for all the sections of the program.	In a test plan with screenshots of the results.
The scoring values are correctly inputted and are held in the corresponding variables.	I will input random scores into the different categories as if I was a chair and using breakpoints I will see if the values correspond.	The values inputted will be held in the corresponding category variables.	Through the use of a trace table.
The calculation for the final score for a delegate.	I will use the randomly inputted scores from the testing above and calculate the final score by hand and see if it corresponds with the value held in the variable using breakpoints.	My calculation is to be the same as the one that is shown. This will be done using normal and exceptional test data.	Through the use of a trace table.
The rankings of delegates in one committee	I will check to see if the ranking goes from highest score to lowest score.	Highest to lowest score.	In a table with delegate and score heading.
The calculation for	I will add up all the	My calculation is to	Through the use of

the final school for a whole delegation(country)	scores of a single delegation by hand and see if the whole school score total variable corresponds with my value.	be the same as the one that is shown.	debugging tools within netbeans.
The rankings of all the delegations at that conference	I will check to see if the ranking goes from highest score to lowest score.	Highest to lowest score.	In a table with delegation and score heading.
The awards are outputted onto the screen and an external file.	See if awards match the ranking status and correct awards are outputted	View the outputted results.	In a table.
A overall test for the program.	The whole program will be used at a mini MUN conference at my school.	Beta testing	By hand and by the program so a comparison can be made.

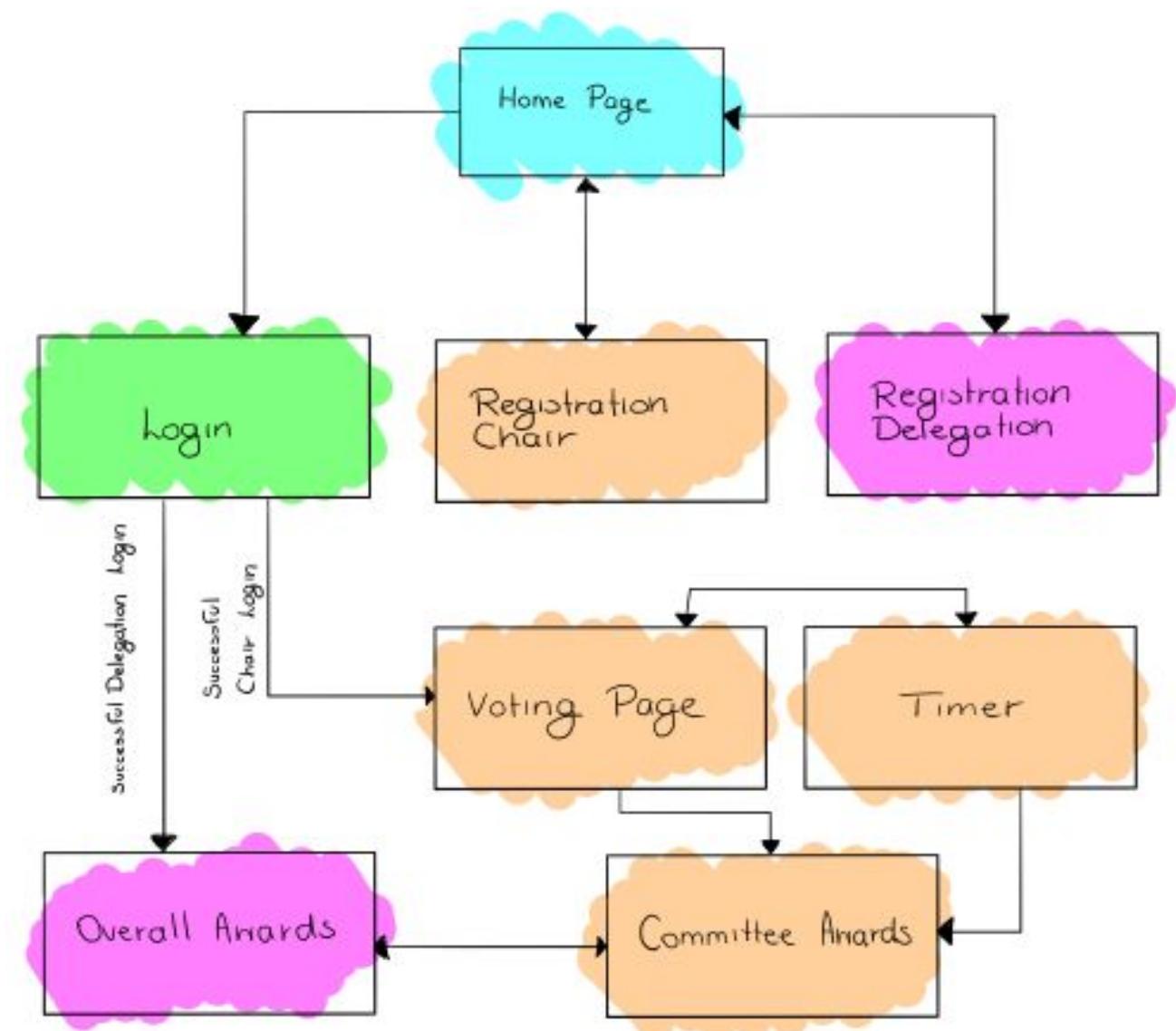
Design

Path of Program Flow Chart

First Stage before the implementation of my application

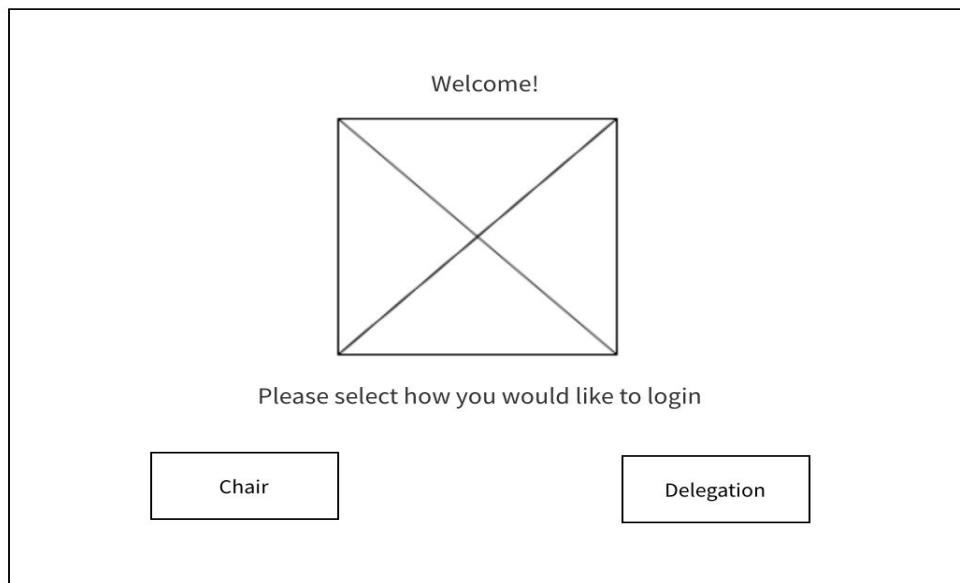


Second Stage after the implementation of my program



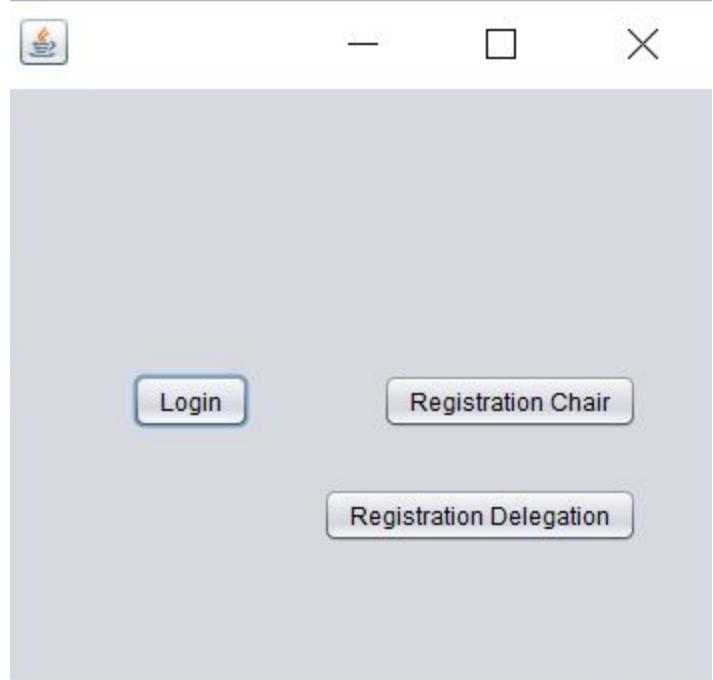
Wireframes

Home page

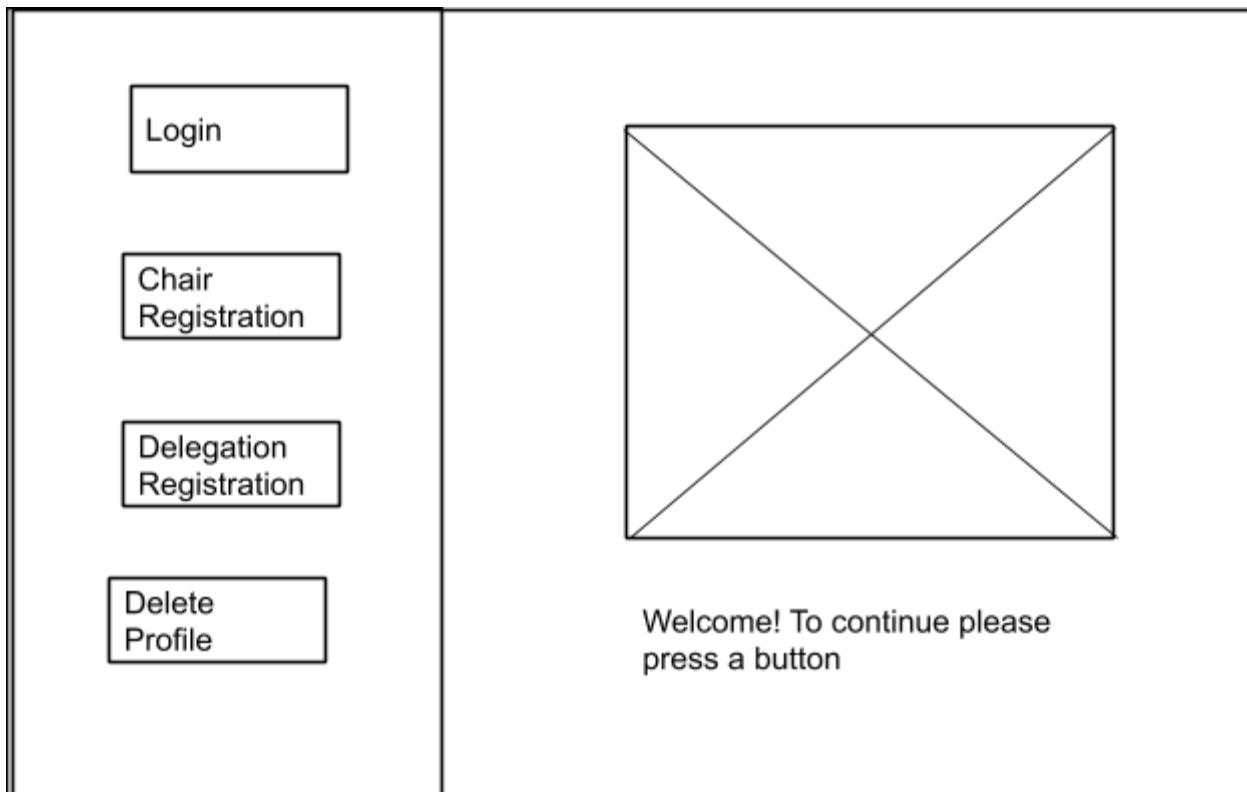


Stage 1: The user selects how they want to use the program; either viewing the results as a delegate or as a chair. They will select the button of whether they are a delegate or a chair.

Stage 2: I decided to change the whole concept of the home page. There will be a single login for both the chair and the delegation and therefore only a login button is needed. I realised that I needed a registration button and so included this in the home page. I was going to have a single registration button however since both the registrations were so different I decided just to have separate registration pages and so there are separate buttons to take the user to either the chair registration or the delegation registration.



Stage 3: Since I later added the ideas of having a delete profile button my wireframe changed slightly and so I decided to redraw my design. I also decided that it would be a good idea to have a navigation bar in all of my pages so that this would create consistency across my application. The user will still interact with the program in the same way by pressing the buttons.



Login

Stage 1: There will be a separate login for both the Delegation and the Chair. The delegate will get a delegation login and inputs their username and password. Once the username and password are entered the user will have to click the enter button. Once this happens the username and password are validated to make sure they match the usernames in the database. If the username and passwords match and exist then they will be able to move on however if they don't then an error message will appear and they won't be able to progress from the login page. There will also be a 'return to home page' button in which when it is pressed it take the user to the home page.

Title 

Welcome Delegation! Please login below.

Username:

Password:

Title 

Welcome Chairs! Please login below.

Username:

Password:

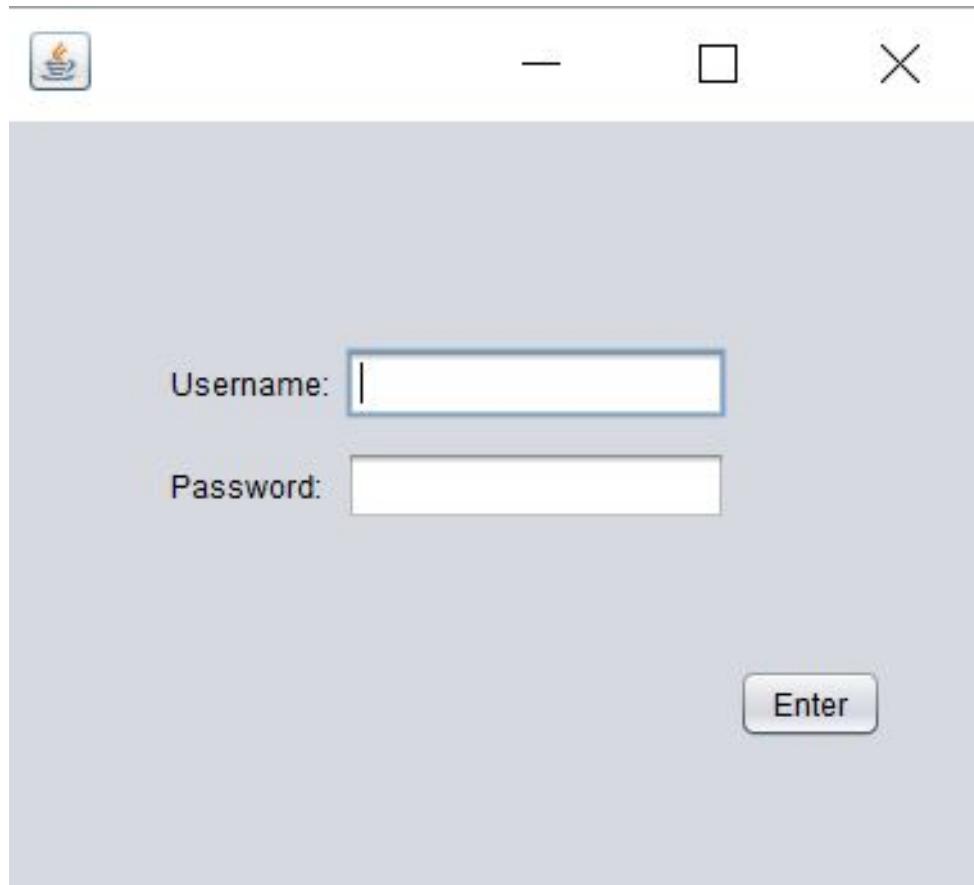
Please select which section you would like to access:

Display Voting

The chair login will work in a similar way to the delegation login however as well as entering the username and password the chair will also interact with a checkbox and click whether they want the timer option (display) or the voting page.

Stage 2: Instead of having two separate login pages I thought it would be more efficient just to have a single login page and that depending on the username and password entered the program automatically knows whether the user is a chair or a delegation and will take the user to the according page.

The login page is very simple and not busy so that it is easy to use and not distracting. The user will type in the username into the username box and will type the password into the password box. The password box is protected as anything typed into the box appears as asterisk (*) and therefore increases the security of the application. Once the username and password is entered the user can press the enter button to which if the login is successful the user will be taken to the corresponding page and if unsuccessful an error message will appear and the user will have another opportunity to enter the username and password again.



Stage 3: I decided to update my wireframe by making sure a navigation bar was present so that there was consistency throughout my webpage. It would also make the overall application nicer looking and increase the usability as it would be less confusing to operate.

	<p>Login</p> <p>Username: <input type="text"/></p> <p>Password: <input type="password"/></p> <p><input type="button" value="Press to login to the application"/></p>
<input type="button" value="Back"/>	

Registration Chair

Stage One: An initial wireframe for this page wasn't created as this page wasn't initially part of my plan however after discussions and thinking the program out I realised it would be easiest if delegations and chairs could register on the application itself instead of someone having to input all the information into the database using SQL statements. This wouldn't be user friendly. However before the implementation I wanted to have some idea of what the registration would look like so created this wireframe. I knew I needed all of these fields because when creating a chair record in the database I knew which values I needed so as to create the record. The user will interact with the program by typing the desired item (described above the field) into the textboxes. The user will select a committee by selecting one of the checkboxes. It should be noticed that only one should be selected at a time. When the user is finished the submit button should be pressed and the registration will be completed and a successful message will appear.

Chair Registration

Username:

Password:

Re-Enter Password:

Please select the committee you would like to chair for

Disarmament EcoSoc
Environment Security Council

Submit Registration

Stage Two: There is no difference between these two stages. Only that this is an implemented version of the wireframe and there are very slight differences in spacing.

The image shows a Java Swing application window titled "Registration Form". The window has a standard OS X-style title bar with minimize, maximize, and close buttons. The main content area is a light gray panel containing the following elements:

- Username:** A text field containing "jTextField1".
- Password (must be at least 5 characters):** A password field containing "*****".
- Please ReEnter Password:** A password field containing "*****".
- Committee:** A group of four radio buttons:
 - Disarmament
 - Environment
 - EcoSoc
 - Security Council
- Submit Registration:** A button at the bottom of the panel.

Stage Three: The wireframe was updated to show a navigation bar which will be used in all of my pages so that the application is consistent.

<p>Back</p>	<p>Chair Registration</p> <p>Username: <input type="text"/></p> <p>Password: <input type="password"/></p> <p>Re-Enter Password: <input type="password"/></p> <p>Please select the committee you would like to chair for</p> <p>Disarmament <input type="checkbox"/> EcoSoc <input type="checkbox"/> Environment <input type="checkbox"/> Security Council <input type="checkbox"/></p> <p>Submit Registration</p>
--------------------	---

Registration Delegation

Stage One: The user will manually have to type into all of the boxes (heavily outlined) to input their information apart from when they are selecting their country. Even though the users will have to manually type in a lot of their information there will be validation surrounding these inputs. For example there will be validation on the username to make sure this is unique. When selecting their country a drop down selection list will appear, this means that the people in charge of the conference have control over which countries there are at the conference. This also stops any spelling errors of countries being entered.

Delegation Registration	
Username:	<input type="text"/>
Password:	<input type="text"/> Re-Enter Password: <input type="text"/>
Country:	<input type="text"/> Drop down select list
<input type="checkbox"/> If you wish to be a security council country please check this box	
Ages of Delegates within the delegation	
Disarmament:	<input type="text"/> EcoSoc: <input type="text"/>
Environment:	<input type="text"/> Security Council: <input type="text"/>
<input type="button" value="Click Button To Submit Registration"/>	

Stage Two: The original design stayed mostly the same however I decided to space things out a bit more and included more description on the labels so that users would have a greater understanding of what to do .

The screenshot shows a registration form window with the following fields:

- Username:** A text input field.
- Country:** A dropdown menu set to "Argentina".
- Password (must be at least 5 characters):** A text input field.
- Please ReEnter Password:** A text input field.
- Please Enter the Ages of the Girls in their respective delegaiton:** A text input field.
- Disarmament:** A text input field.
- Please check the button if you would like a security council country:** A label with a checkbox below it labeled "Security Council Country".
- Environment:** A text input field.
- EcoSoc:** A text input field.
- Security Council (If applicable):** A text input field.
- Submit Registratiton:** A button at the bottom right.

The user interacts with the system by typing into the textboxes the values required. A checkbox allows the user to say whether or not they wish for the delegation to be a security council country (this is only a yes or no answer so is why it is a listbox). A pre-set country list is displayed in a dropdown list so that the user can select a country. If the country has already been chosen it should be noted that an error message will appear as the country needs to be unique. A drop-down list was used so that there would be no spelling mistakes in country's requested by the user and so that only a select list of countries were made available. To change the list the users who are running the conference would need to hard code the country list.

Stage Three: A navigation bar was added to my initial wireframe so as to add consistency to my overall application.

<input type="button" value="Back"/>	<p>Delegation Registration</p> <p>Username: <input type="text"/></p> <p>Password: <input type="password"/> Re-Enter Password: <input type="password"/></p> <p>Country: <input type="text"/> Drop down select list <input type="checkbox"/> If you wish to be a security council country please check this box</p> <p>Ages of Delegates within the delegation</p> <p>Disarmament: <input type="text"/> EcoSoc: <input type="text"/></p> <p>Environment: <input type="text"/> Security Council: <input type="text"/></p> <p><input type="button" value="Click Button To Submit Registration"/></p>
-------------------------------------	---

Delete Profile

Stage One: This wasn't initially apart of the plan however after much consideration I thought this would be an essential part to my program in making it more accessible. This looks and acts much like the login to the application. The user will enter their username and password into the boxes using a keyboard. They will then press the 'Delete Profile' button and their profile will be deleted.

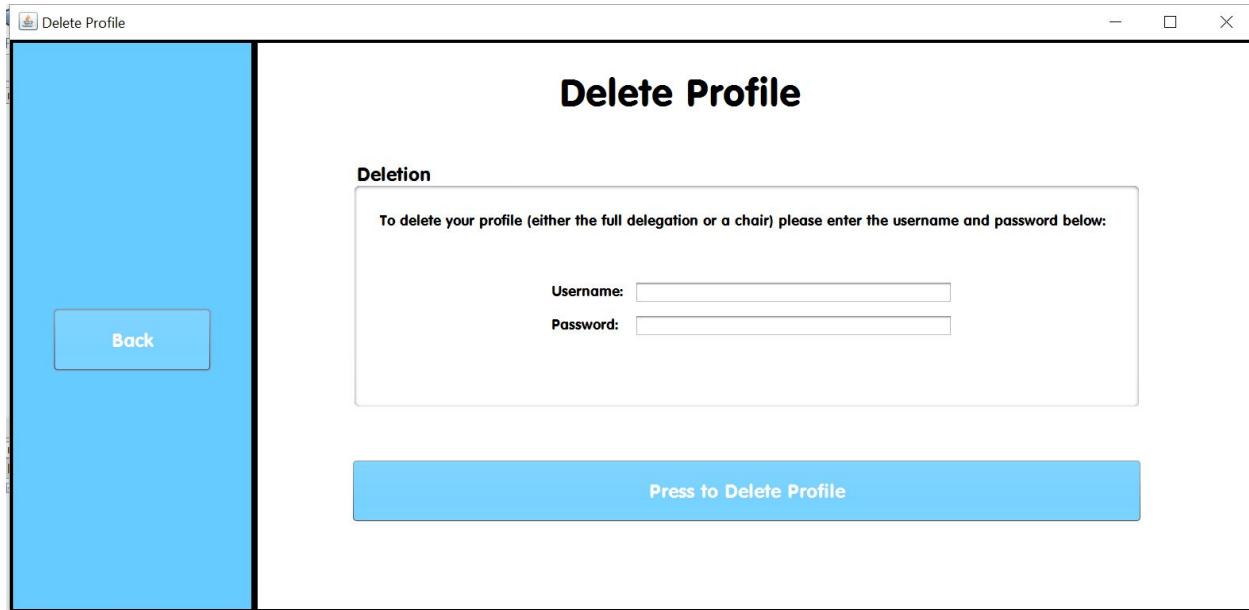
Delete Profile

Username:

Password:

Press To Delete Profile

Stage Two: Since I decided to add this functionality later on in the design stage I have no 'rough' implementation of this page and only the final implementation. The main change within these two stages were programing changes and not design changes.



Voting

Please select the delegate's country

Country

Instigate debate (bring a new idea to the table)

Yes No

Quality of Speech: Poor ★ ★ ★ ☆ ☆ Excellent

Humour: ★ ☆ ☆ ☆ ☆

Ability to handle Points of Information: ★ ★ ★ ★ ★

Points of information given :

United Kingdom Checkbox 6

USA Checkbox 7

Checkbox 3 Checkbox 8

Checkbox 4 Checkbox 9

Checkbox 5 Checkbox 10

Stage 1: The chairs will interact with this page during the conference. The chairs will be able to select which country is speaking from and the information about quality of speech, humour and ability to handle points of information will be recorded for that person. During my survey a quality which was suggested was whether or not that delegate instigated the debate. If they did extra points would be rewarded.

If delegates give a point of information then this will be noted by the chairs through the use of checkboxes. This will then hold some weight when calculating a delegates overall score.

After the main delegate

who is speaking finishes the chairs can submit their voting forms and this will clear the form for a new one. When the chair presses the 'leave voting' button a message will appear asking what page they would like to go to: the display section or back to the homepage.

Stage 2: This is a partially completed voting form that has the basic structures of what the form will look like in a high fidelity wireframe. There is a drop down list in which the chair will be able to select the country that is speaking and who the voting form is about. This is a drop down list so that there is no misspelling on any countries name (a feature that would cause a problem especially when retrieving results from a database). This drop down list will change automatically depending on which countries are in the committee (security council doesn't have all the countries) and which new countries register.

There are slider bars which allows the chair to vote on the aspects of: quality of speech, humour in speech and ability to handle points of information. The slider bars automatically 'snap' to the nearest whole number and so this will be easy for the chair to do in a fast pace environment like an MUN conference and this will make the voting process much easier.

There is a checkbox titled 'added new information to debate' which if clicked will add more points to the delegates total score. This was put in a checkbox as its just a 'yes or no' response.

There is a section for countries who gave points of information and the chairs can then just select the countries from the list if those countries gave a point of information to the speaker.

Country of Delegate

Item 1

Quality of Speech

1 2 3 4 5

Added New Information to Debate

Humour in Speech

1 2 3 4 5

Countries who Gave POI's

Ability to Handle Points of Information

1 2 3 4 5

jCheckBox1

jButton1

Stage 3: Not much has changed since the here and the stage 2 however I have just cleaned up and finished the design more (colours and overall design aesthetically still has to be improved). I decided to remove the checkboxes of the countries who gave point of information feature as this would have cluttered up the voting form and I felt this feature wasn't important so removed this.

I included a button that would submit the voting form and once this button is pressed then a message will appear either saying 'successful' or 'unsuccessful' depending whether or not the form has been successfully submitted into the database.

If the timer button is pressed then the user will be taken to the timer page and the voting form will close. If the awards page button is clicked then the user will be taken to the awards committee page and the voting form will close.

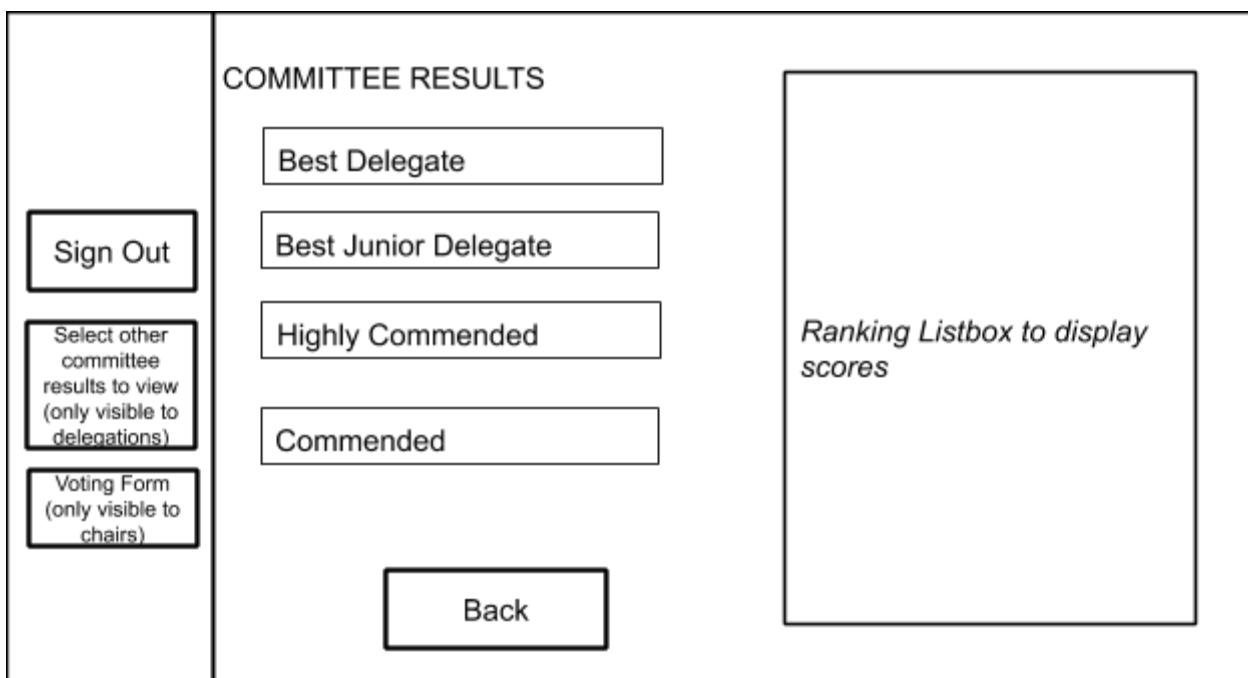
The screenshot shows a Windows application window with a title bar containing a logo, a minimize button, a maximize button, and a close button. The main area of the window is titled "Country of Delegate" and contains a dropdown menu set to "Russia". Below this, there are three horizontal sliders for rating different aspects of a speech:

- Quality of Speech:** A slider from 1 to 5, with the handle positioned at 5.
- Humour in Speech:** A slider from 1 to 5, with the handle positioned at 5.
- Ability to Handle Points of Information:** A slider from 1 to 5, with the handle positioned at 5.

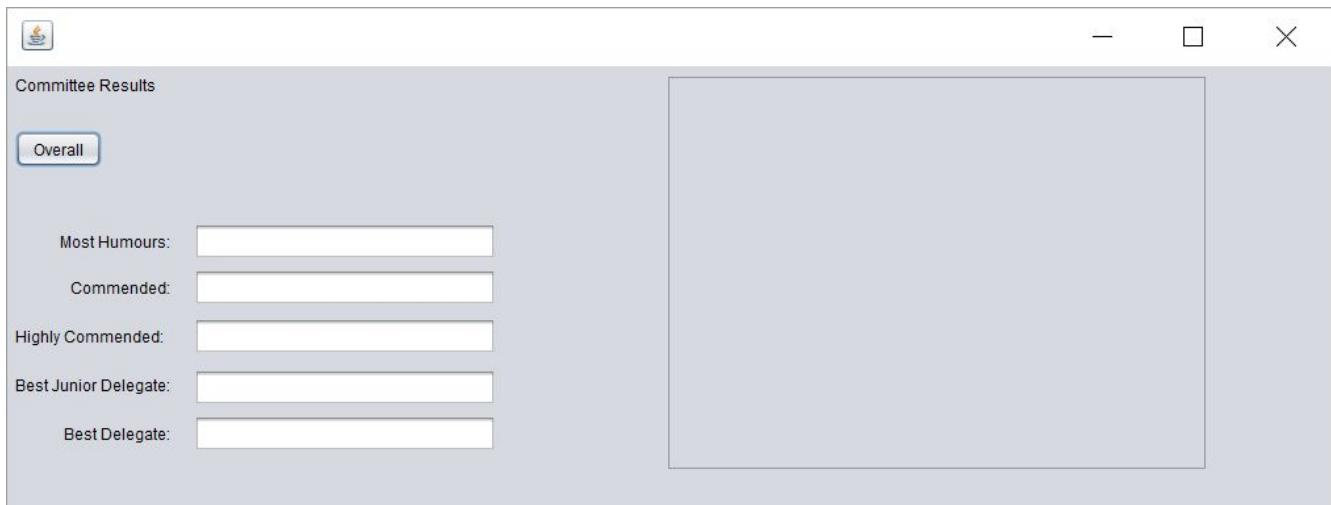
At the bottom of the window, there are two buttons: "Submit Vote" and "Awards Page". Below these buttons, a message box displays the text "Added New Information to Debate".

Award Pages

Stage One: This will contain the output and rankings of all committees as well as the overall ranking of the delegation and all the rewards will also be present. As well as that country (who logged in) individual scores in their committee. The delegates will be able to view their ranking as a delegation as a whole, view their ranking in the committee as a whole and they will be able to access and view their own individual breakdown scores. The chairs will be able to view the rankings of all the delegations, view their committees ranking and view all the individual breakdowns in their committee. The overall and committee awards will be displayed on separate pages however will look very similar.



Stage Two: This is the development of the committee awards. I decided at this stage as well that the implementation for the humours award and the actual sorting of this award would be too complicated at this stage as it would be too time consuming.



Personas



Name: Sarah Smart

User Type: Chair

Committee: Disarmament

Country: N/A

Background: Sarah is a student in her last year of secondary education. Sarah has played an important role in her schools MUN team and is even president of the club. Although Sarah is very smart she isn't the best at using technology and sometimes finds using computers difficult and challenging. This being said however she is a teenager and so knows the basics of a computer.



Name: Alexander Walker

User Type: Delegate

Committee: Environment

Country: France

Background: Alexander is a student in secondary education. Alexander is a very competitive person and takes debates and other such conferences very seriously. Alexander wants to do well at the next MUN conference. Alexander has a lot of general knowledge about the world and in the future wants to become a politician.

User Stories

Sarah Smart: This is Sarah's first time chairing a debate. She wants to use this system to help her with the timing of the debate and to make sure it runs smoothly. Since this is Sarah's first time chairing a debate she is unsure how she should rank delegates and what qualities qualify for a prize. The system will help guide Sarah through this as it tells her about specific aspects she should vote on.

Alexander Walker: Alexander is a very experienced delegate and knows how MUN conferences work. Alexander finds the screen that is displayed to delegates very useful as he can see how much time he has to talk. In previous conferences he has attended he found the use of a computer system useful to make sure the discussions stays on track. Alexander is always worried in conferences that the chairs will pick someone for the Best Delegate Award because they know them personally and not due to their MUNing ability. This is why Alexander is looking forward to the use of the voting system for the chairs. Alexander is looking forward to being able to view his delegations and his own individual total score so he can see where him and his delegation ranked due to others at the conference.

Scenarios

Login Delegates

Actor: Delegate

Scenario: When the program loads they will be taken to the Home Screen(as shown in the wireframes). They will then press a button to select to go to the delegation login. They will then put in their respective username and password into the input boxes. After they finish inputting the data they will press the submit button. Input validation will then be required by the system to make sure the passwords and usernames match the ones in the database and have the correct access rights and if the usernames and password matches they will be able to continue.

Login Chairs

Actor: Chairs

Scenario: When the program loads they will be taken to the Home Screen(as shown in the wireframes). They will then press a button to select to go to the chairs login. They will then put in their respective username and password into the input boxes. The chairs during their login will also have to select which page they would like to go to (voting or display); this will be done by selecting a button. After they finish inputting the data they will press the submit button. Input validation will then be required by the system to make sure the passwords and usernames match the ones in the database and have the correct access rights and if the usernames and password matches they will be able to continue.

Vote on delegates

Actor: Chair

Scenario: The chair uses the voting form to input the data about the delegate who is currently speaking. The form is then submitted and values are added to any existing score (running total for all the different categories). These scores are held in a database.

Timer

Actor: Chair

Scenario: The chair can select the length of the timer and speaker timer they want and can name the timer (i.e 'Time for amendment'). They can then select from the list of countries for countries who wish to speak. When the click the play button the timer for the first country will begin. Throughout a pause and play button can be clicked on for when points of information are asked. If the delegate hands back over to the chair before the timer is done the chair can click the finished button and the timer will reset to the next country.

View Results Delegates

Actor: Delegate

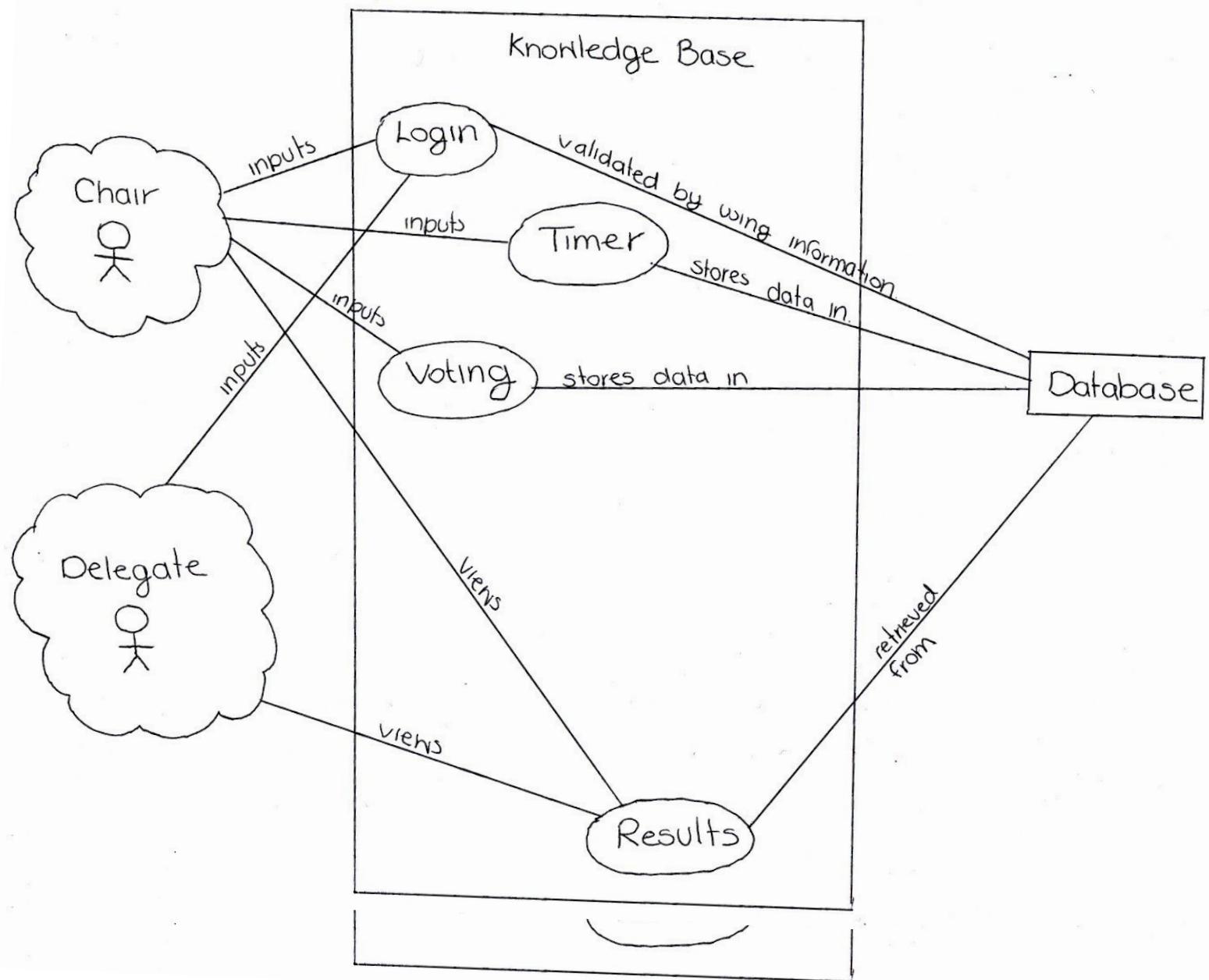
Scenario: The delegates can use the password provided to the school and look at the rankings compared to all other schools for the committees as a whole. They will also be able to view any awards given. The delegate can look at any of their delegations individual scores ubt won't be able to see any other delegations individual scores they will however be able to see where they are ranked against other delegations. This area is mainly just for viewing and the Delegate will mainly interact through buttons or hyperlinks to get to new areas.

View Results Chairs

Actor: Chair

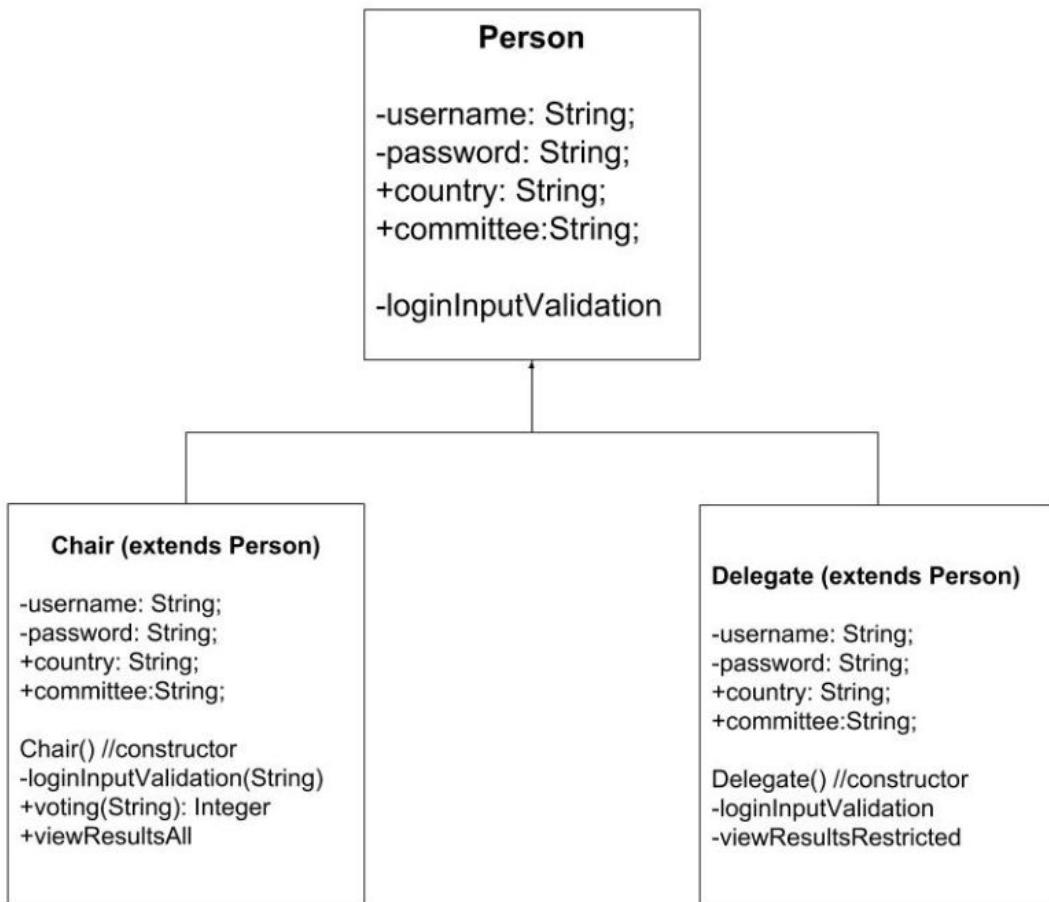
Scenario: The chairs can choose to go to the results page in which they will be able to view the results for all the delegations as a whole and the overall best delegation award. The chairs will only be able to view their committees result and see all the delegates in their committees individual breakdowns and rankings. They will also be able to see the awards in their own committees.

User Centric Design - Use Case Diagrams



Class Diagram

Stage 1 before the implementation of my application



Stage 2 After the implementation of my program.

During the implementation of my program I realised that there was no need to have a chairs class as the chair's data wasn't manipulated that often, if any. I decided to make a delegate class so that an array of objects could be made so as to sort the delegates. I however also realised that I needed to create a delegation class in order to sort the delegation as a whole. The Wrapper classes were used so that all the information stored in an object was swapped during the sorting algorithm and not just an individual data result.

Delegate

```
- countryName: String;  
- age: Int;  
+totalScore: Int;  
  
+setTotalScore(Int);  
+setCountryName(String);  
+setAge(Int);  
+getCountryName() : String;  
+getAge(): Int;  
+getTotalScore(): Int;  
+displayMessgae(): String;  
+compareTo(Delegate): Int ;
```

Delegation

```
- countryName: String;  
- countryID: Int;  
+totalScore: Int;  
  
+setTotalScore(Int);  
+setCountryName(String);  
+setCountryID(Int);  
+getCountryName() : String;  
+getTotalScore(): Int;  
+displayMessgae(): String;  
+compareTo(Delegation): Int ;
```

DelegateWrapper

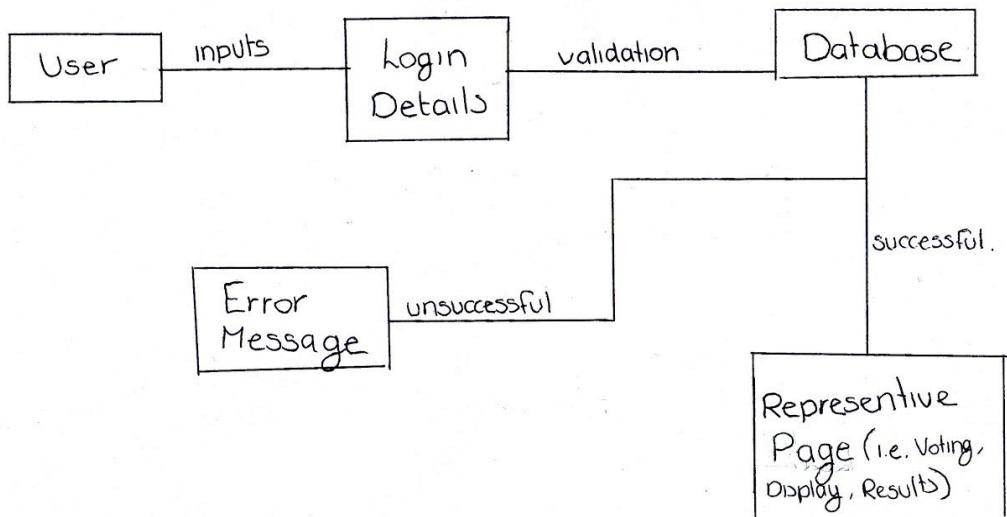
```
+d: Delegate;  
  
+delegateWrapper(Delegate);
```

DelegationWrapper

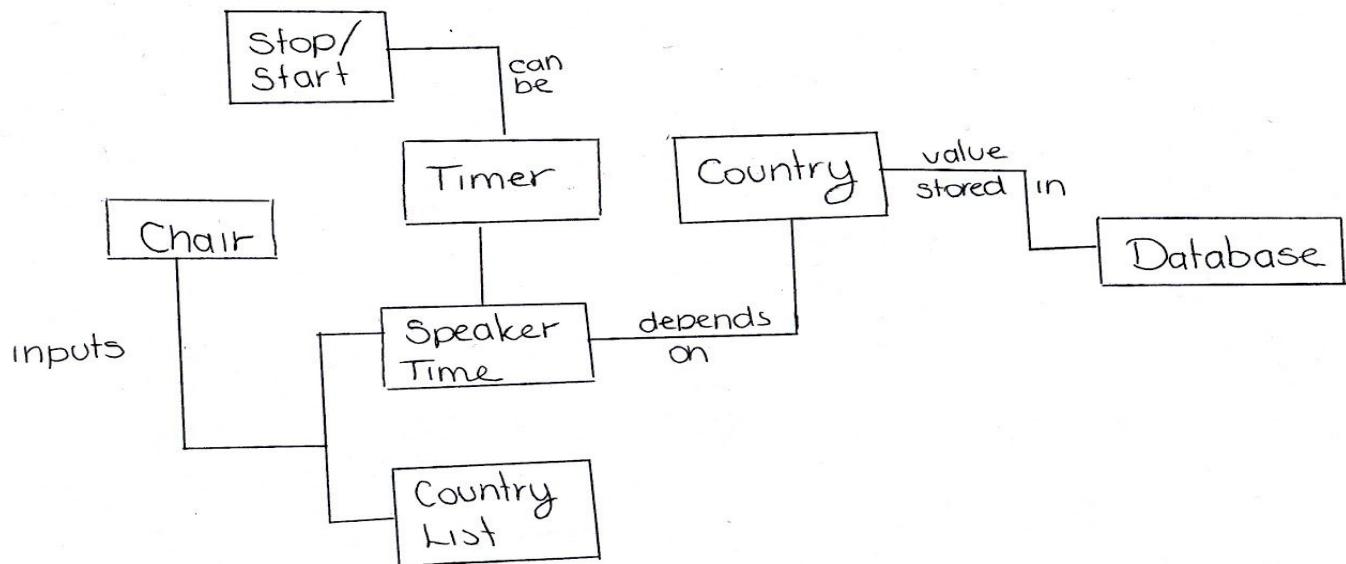
```
+d: Delegation;  
  
+delegationWrapper(Delegation);
```

Conceptual Model

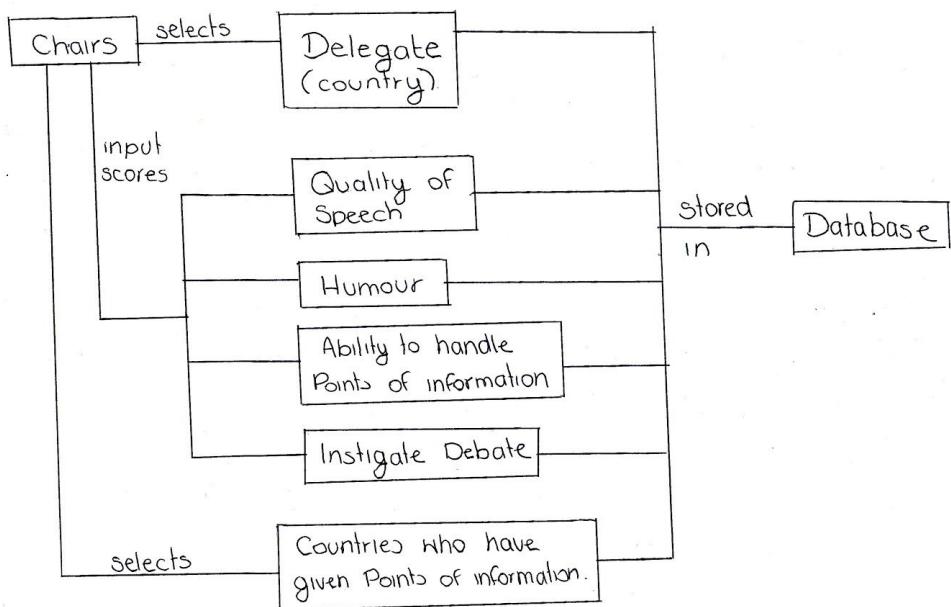
Login



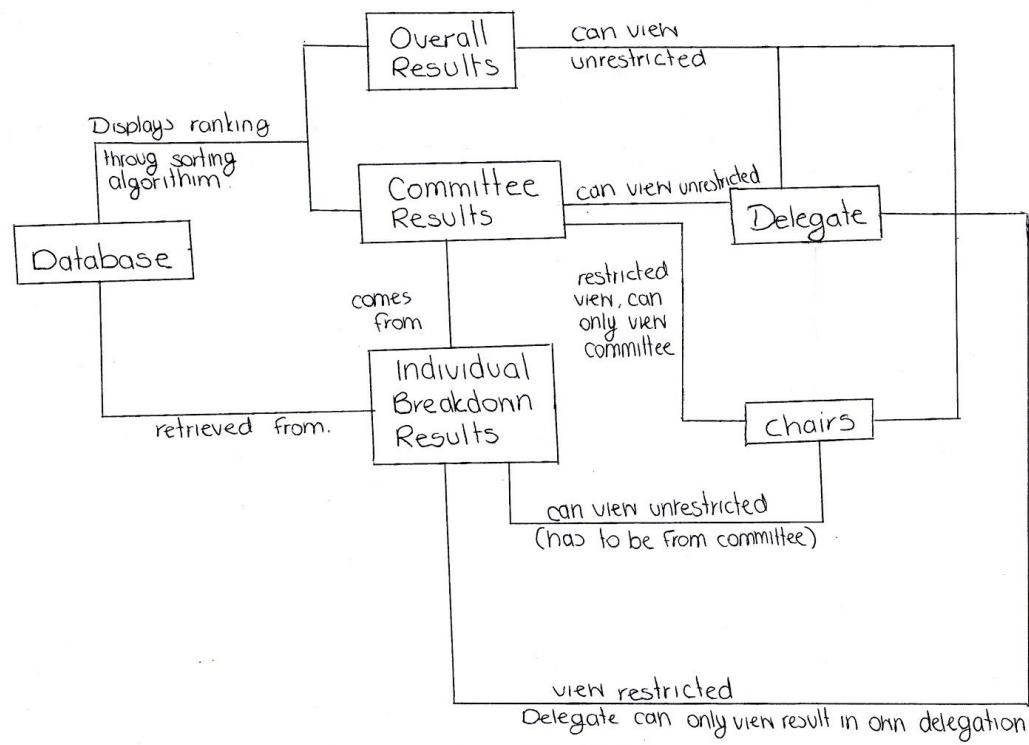
Display



Voting

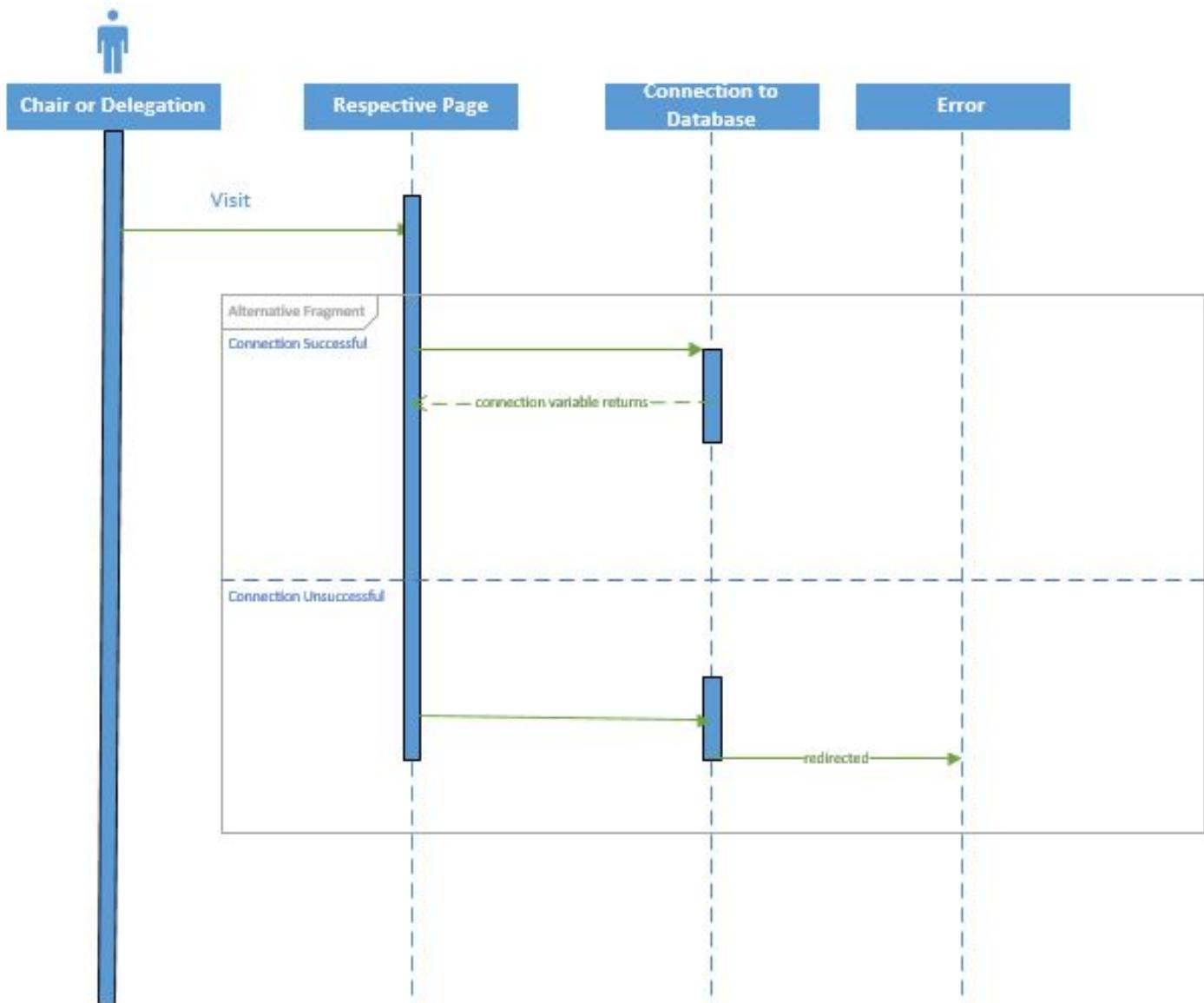


Results



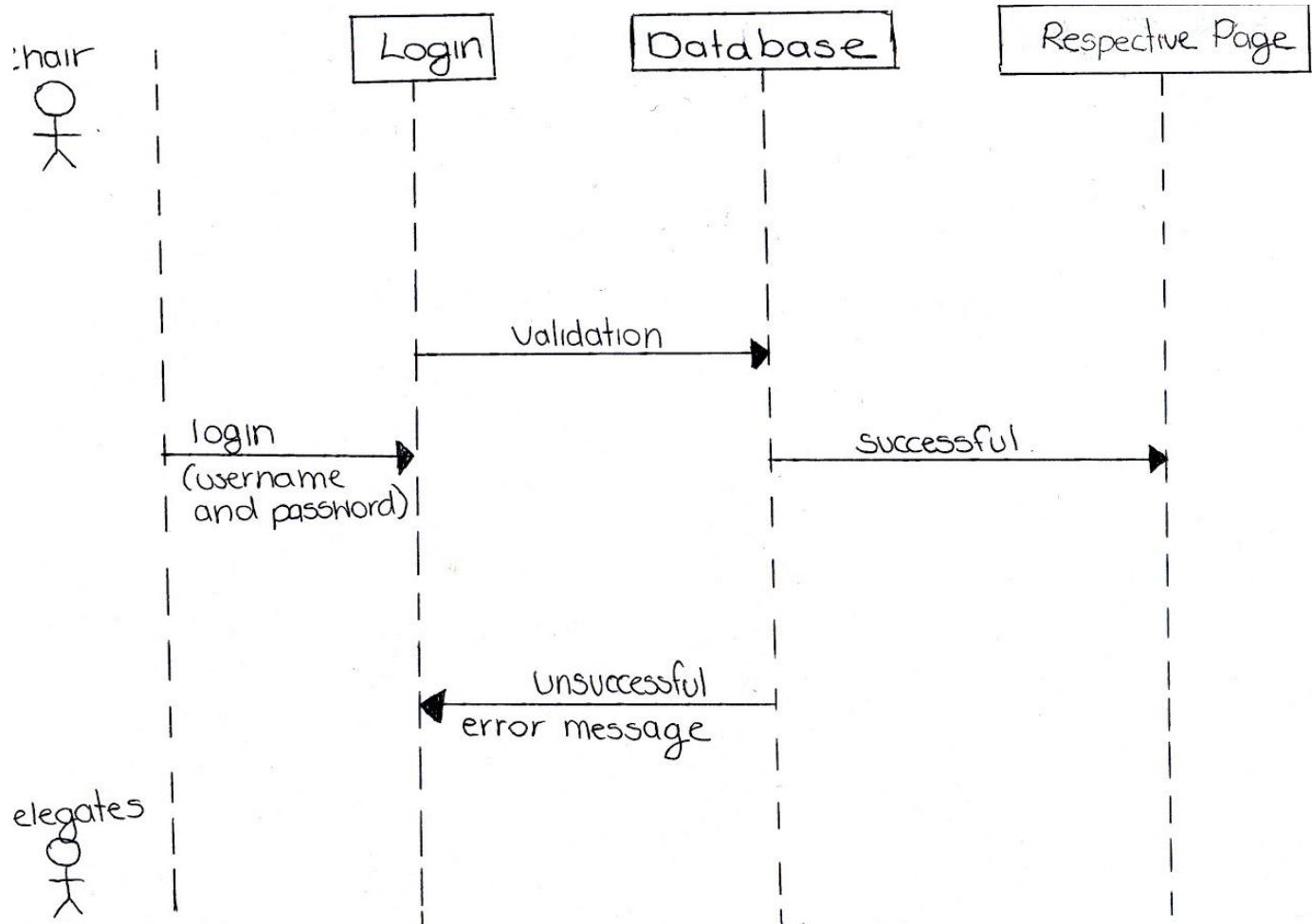
Sequence Diagrams

Connection to Database

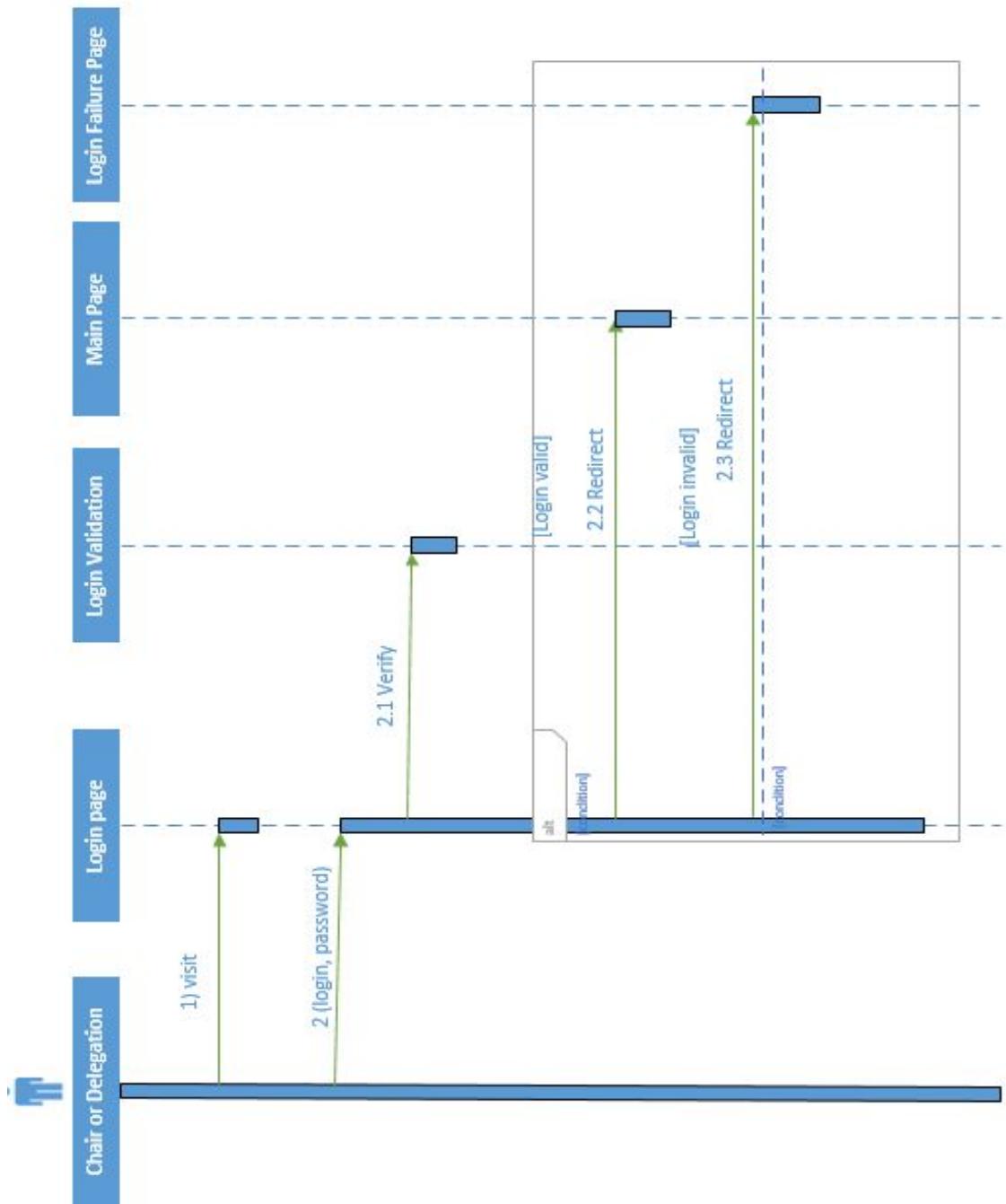


Login

Stage One before the implementation of the program

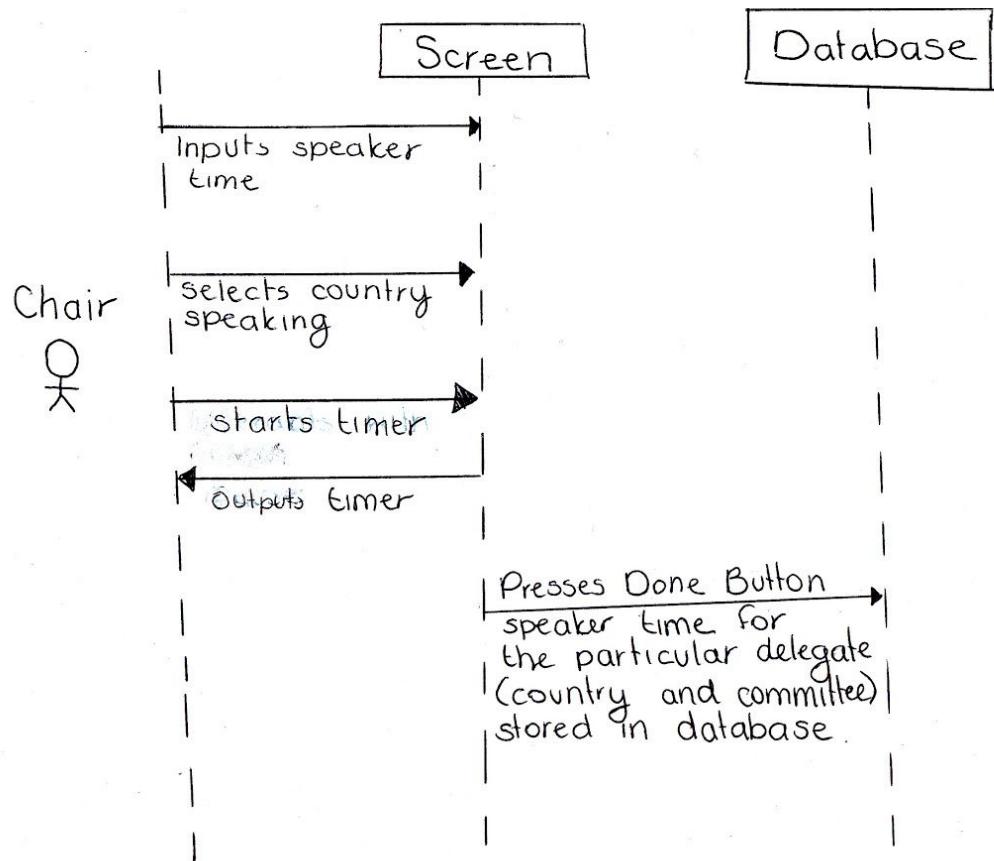


Stage Two after the implementation of the program

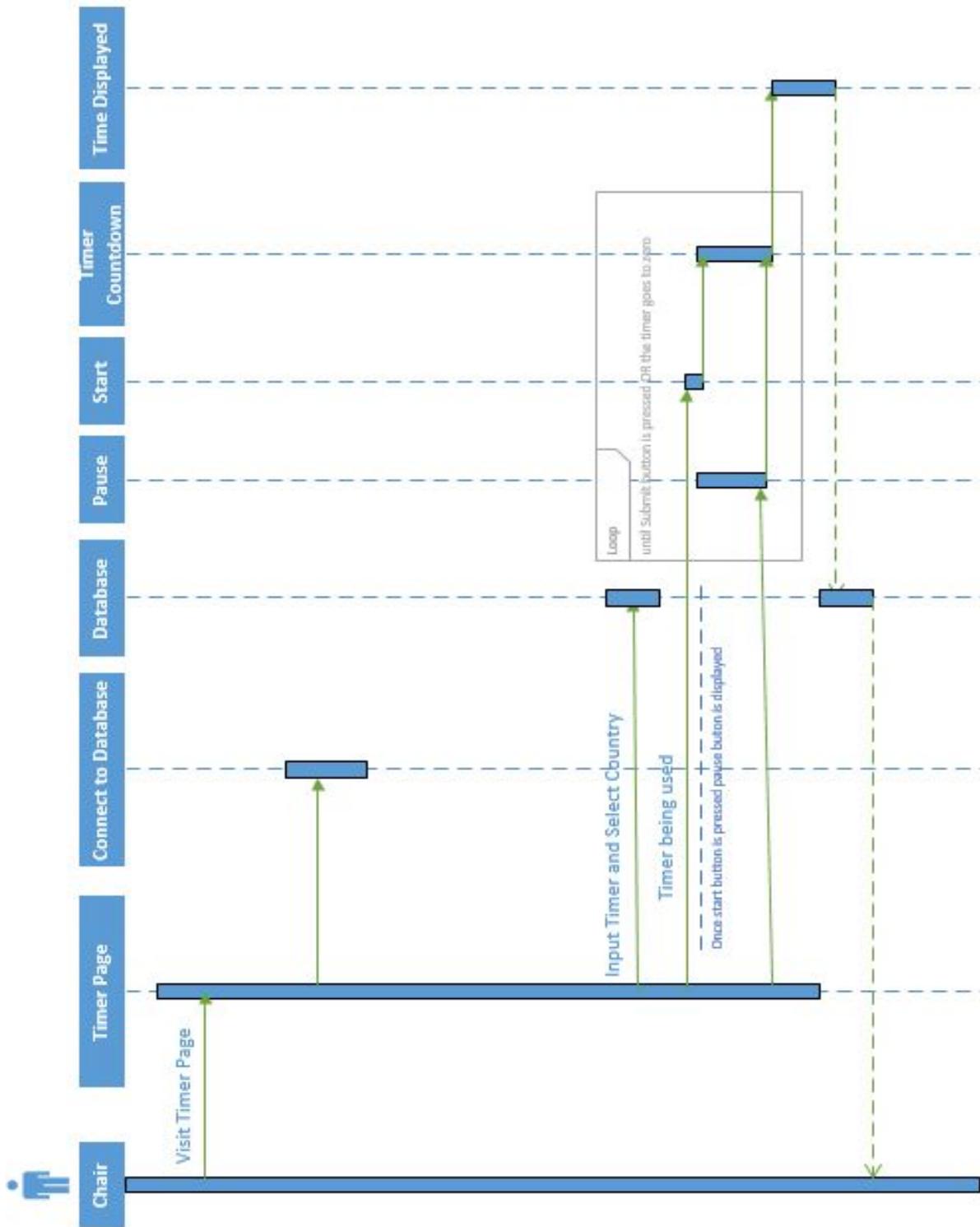


Timer

Stage One before the implementation of the program

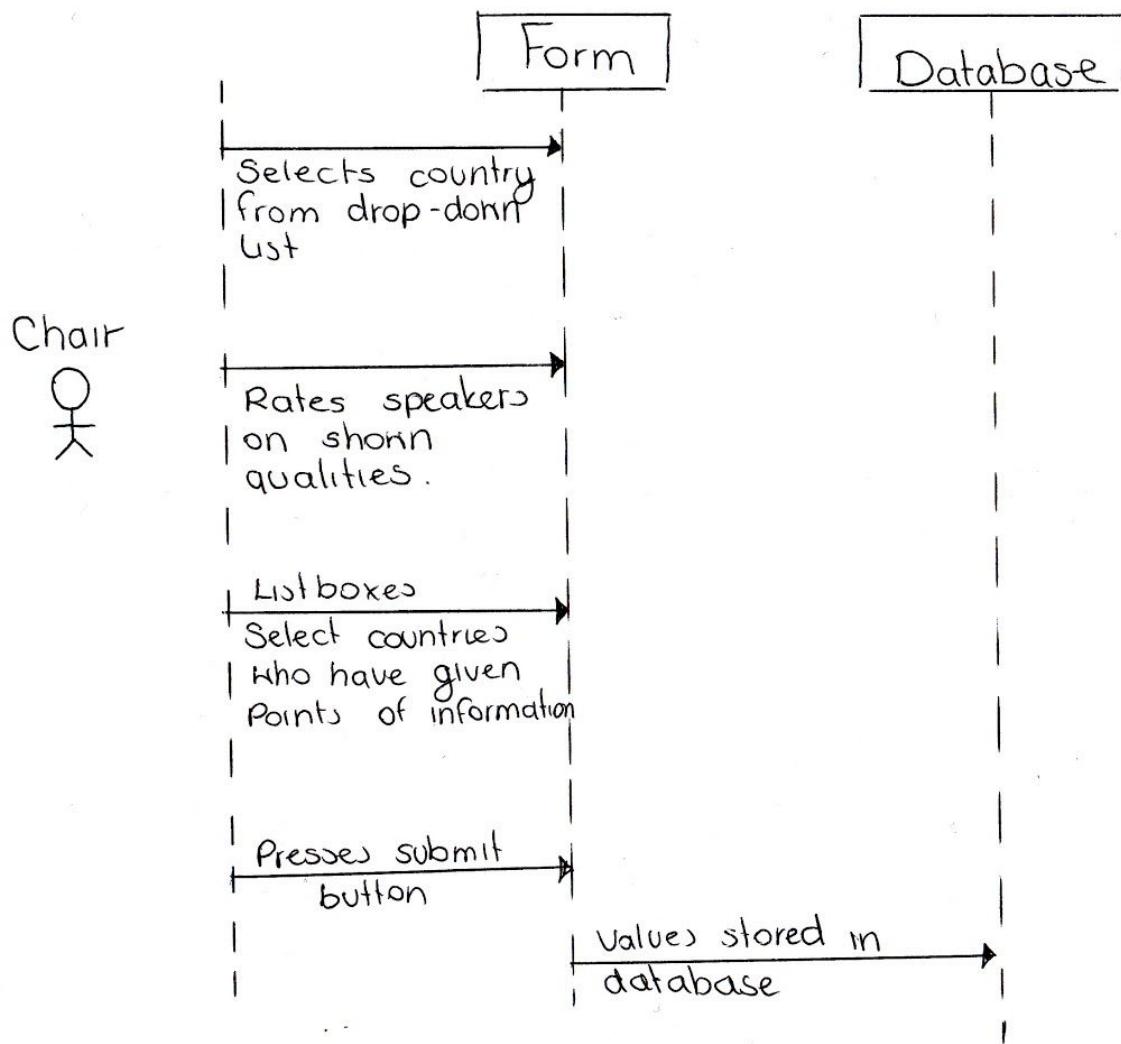


Stage Two after the implementation of the program

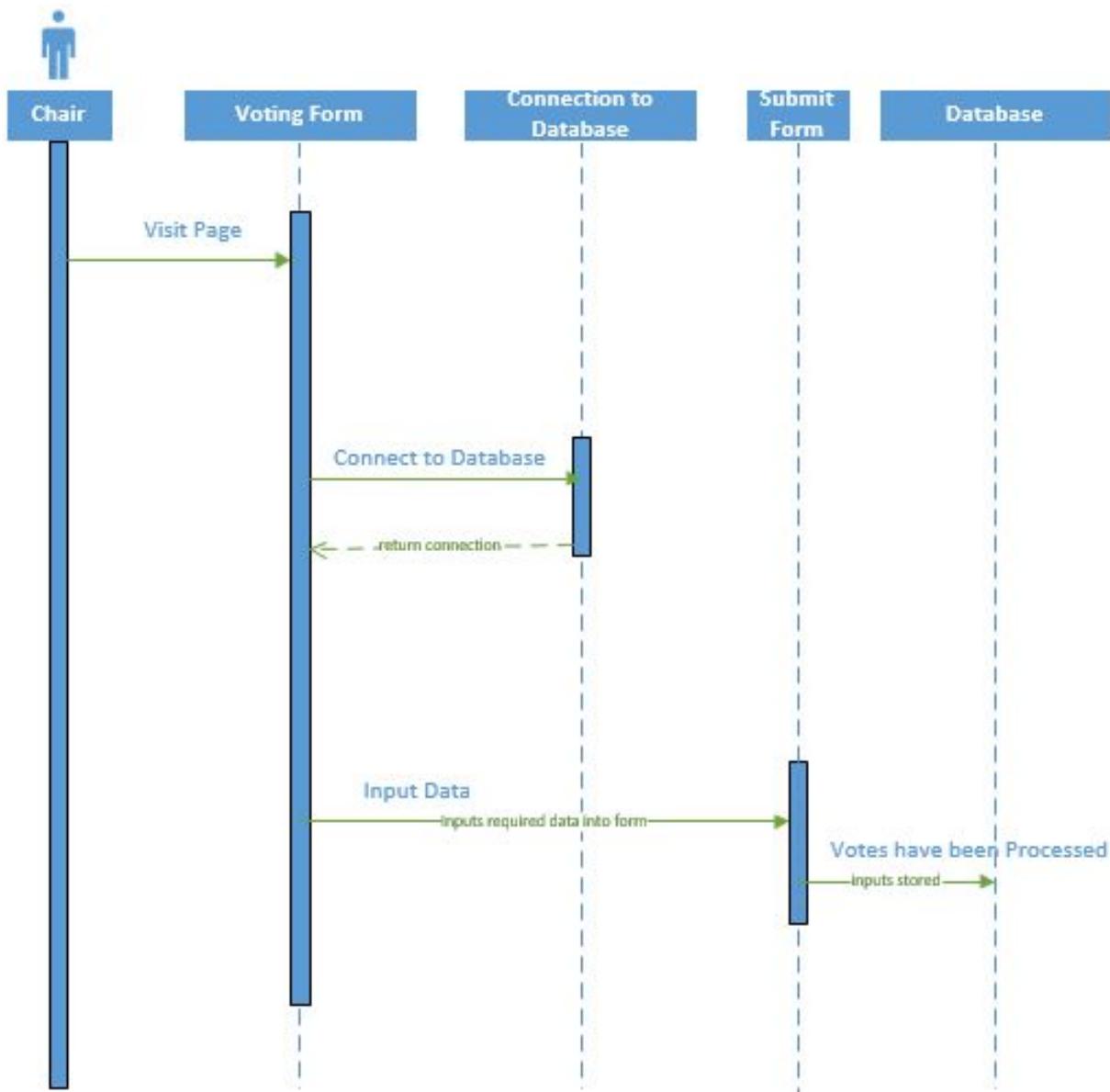


Voting

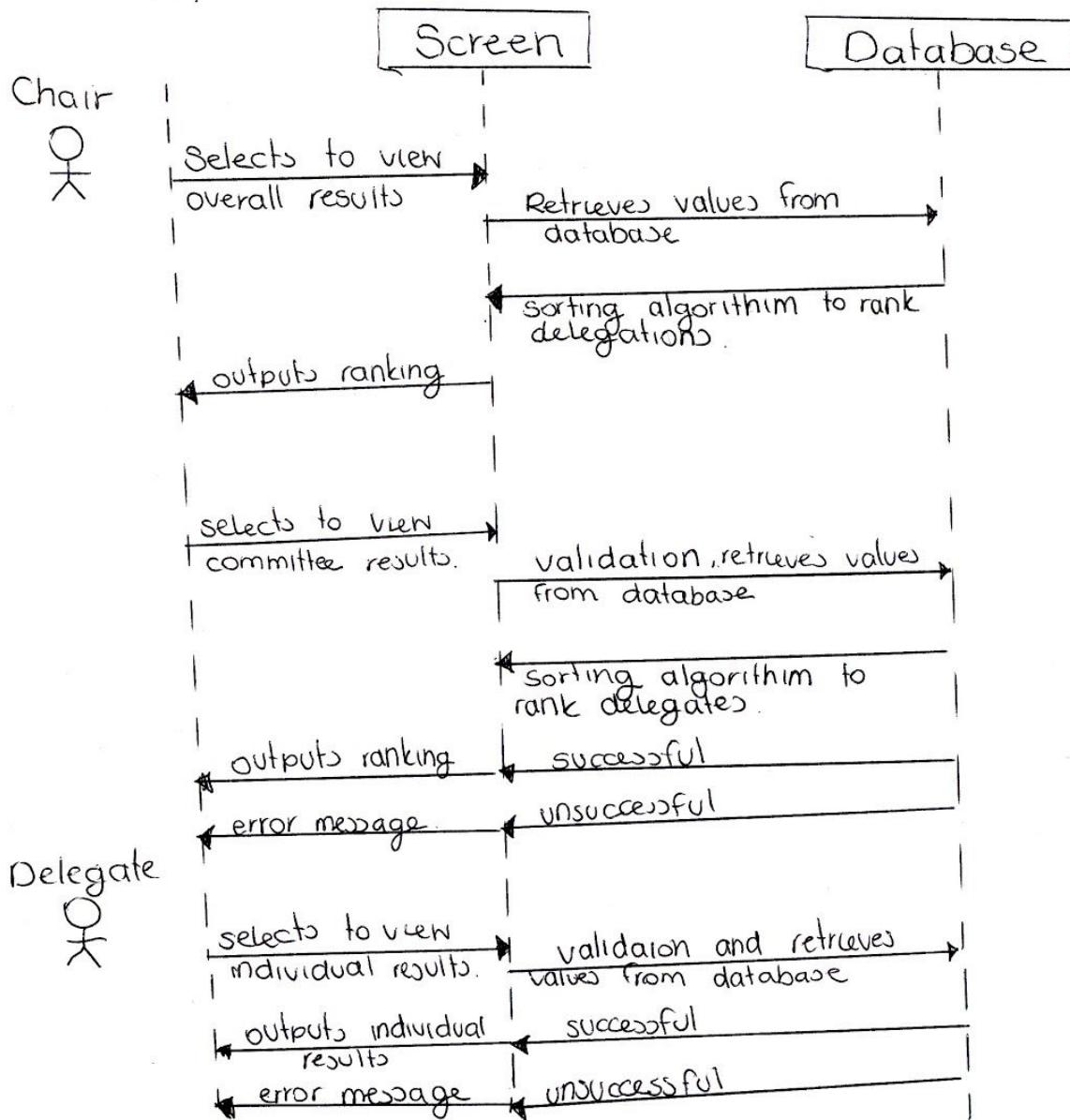
Stage One before the implementation of the program



Stage Two after the implementation of the program



Results



Data Dictionary

Stage One: This was designed before the implementation of my program so a lot had changed once I started implementation.

Name of Field	Data Type	Data Size (byte)	Key (if any which type?)	Unique (Y/N)	Required (Y/N)	Validation	Sample Data
Delegate							
<u>DelegateID</u>	Auto Number	8	Primary	Y	Y		1
username	String	100		N	Y		MegzGal2
password	String	100	-	N	Y	Must be between 6 and 12 characters and include a capital letter, symbol and number.	pass4Word
age	Integer	8		N	Y	Between 12 and 30	16
country	String	50		N	N		Israel
committeeid*	Auto Number	8	Foreign	N	Y	Auto Lookup from Committees table	5
totalScore	Integer	50		N	N		100
Chair							
ChairID	Auto Number	8	Primary	Y	Y		1
username	String	100		Y	Y		CoolChair5
password	String	100		N	Y	Must be between 6 and 12	pass4Word

						characters and include a capital letter, symbol and number.	
committeeid*	Auto Number	8	Foreign	N	Y	Auto Lookup from Committees table	5

Voting

<u>DelegateID*</u>	Auto Number	8	Primary and Foreign	N	Y	Auto Lookup from Delegates table	Israel
qualityScore	Integer	50		N	N	Between 1 and 5	5
humourScore	Integer	50		N	N	Between 1 and 5	2
abilityPOIScore	Integer	50		N	N	Between 1 and 5	3
instigateDebate	Boolean	50		N	Y		Yes
timeSpoken(min)	Real Number	50		N	N		23 mins

Committee

<u>committeeName</u>	String	50	Primary	N	Y		DISEC
ChairsID*	Auto Number	8	Foreign	Y	Y	Auto Lookup from Chairs table	1C
DelegateID*	Auto Number	8	Foreign	Y	Y	Auto Lookup from Delegates table	ID

Stage 2: This data dictionary was created after the implementation so that all the actual implemented database could match the data dictionary. There were some small changes on what type of data values were used for example 'VarChar' was used instead of 'String' this was due to the fact that in phpMyAdmin it was more reliable to use this data type.

The implemented data dictionary is shown below.

chair

Column	Type	Null	Default	Links to	Comments	MIME
chairID (<i>Primary</i>)	int(8)	No				
username	varchar(50)	No				
password	varchar(50)	No				
committeeID	int(8)	No		committee > committeeID		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	B TREE	Yes	No	chairID	4	A	No	
chairID	B TREE	Yes	No	chairID	4	A	No	
committeeID	B TREE	No	No	committeeID	4	A	No	

committee

Column	Type	Null	Default	Links to	Comments	MIME
committeeID (<i>Primary</i>)	int(8)	No				
committeeName	varchar(20)	No				

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	committeeID	4	A	No	

country

Column	Type	Null	Default	Links to	Comments	MIME
countryID (<i>Primary</i>)	int(11)	No				
countryName	varchar(50)	No				
username	varchar(50)	No				
password	varchar(20)	No				
securityCouncil	tinyint(1)	No				

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	countryID	2	A	No	
countryName	BTREE	Yes	No	countryName	2	A	No	
countryName_2	BTREE	Yes	No	countryName	2	A	No	
username	BTREE	Yes	No	username	2	A	No	

delegate

Column	Type	Null	Default	Links to	Comments	MIME
delegateID (<i>Primary</i>)	int(8)	No				
age	smallint(8)	No				
countryID	int(11)	No		country -> countryID		
committeeID	int(8)	No		committee -> committeeID		
totalScore	int(50)	No				

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	B TREE	Yes	No	delegateID	7	A	No	
delegateID	B TREE	Yes	No	delegateID	7	A	No	
countryID	B TREE	No	No	countryID	7	A	No	
committeeID	B TREE	No	No	committeeID	7	A	No	

timer

Column	Type	Null	Default	Links to	Comments	MIME
delegateID	int(8)	No		delegate -> delegateID		
timeSpoken	int(50)	No				

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
FOREIGN	BTREE	No	No	delegateID	2	A	No	

voting

Column	Type	Null	Default	Links to	Comments	MIME
delegateID	int(8)	No		delegate -> delegateID		
committeeID	int(8)	No		committee -> committeeID		
chairID	int(8)	No		chair -> chairID		
qualityScore	int(50)	No				
humourScore	int(50)	No				
abilityPOIScore	int(50)	No				
instigateDebate	tinyint(1)	No				

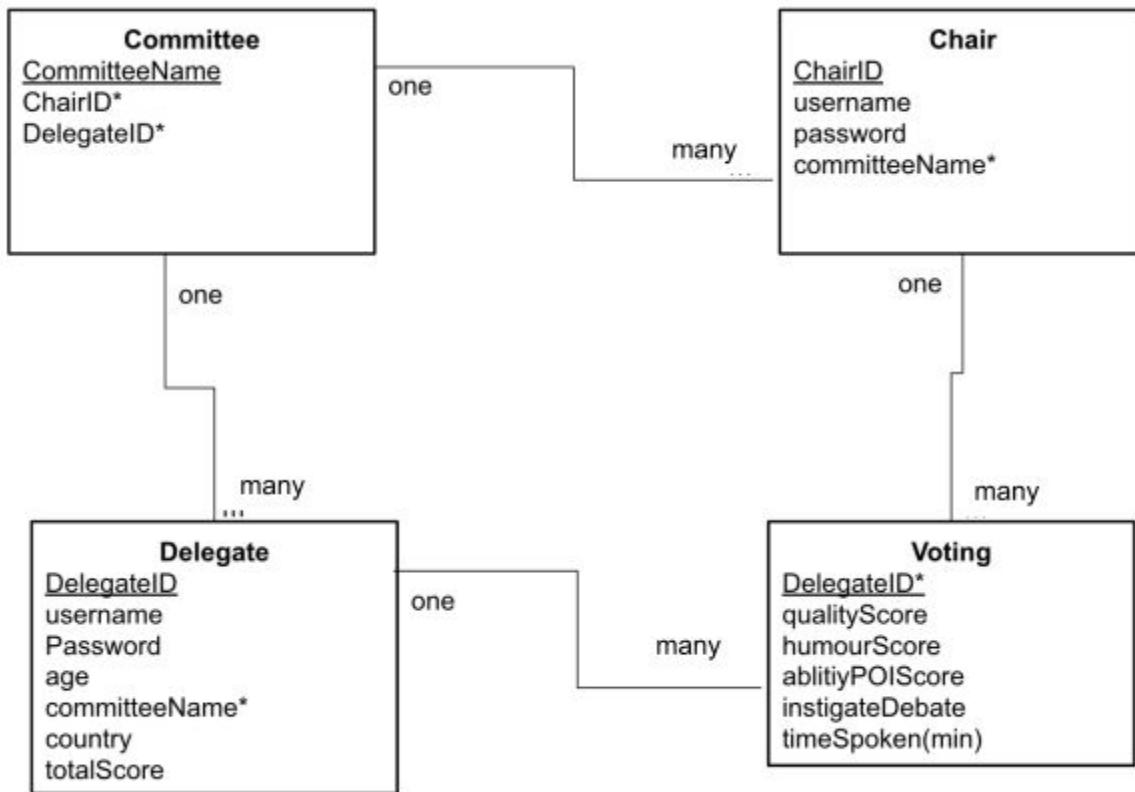
Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
chairID	BTREE	No	No	chairID	2	A	No	
delegateID	BTREE	No	No	delegateID	2	A	No	

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
FOREIGN	BTREE	No	No	committeeID	2	A	No	

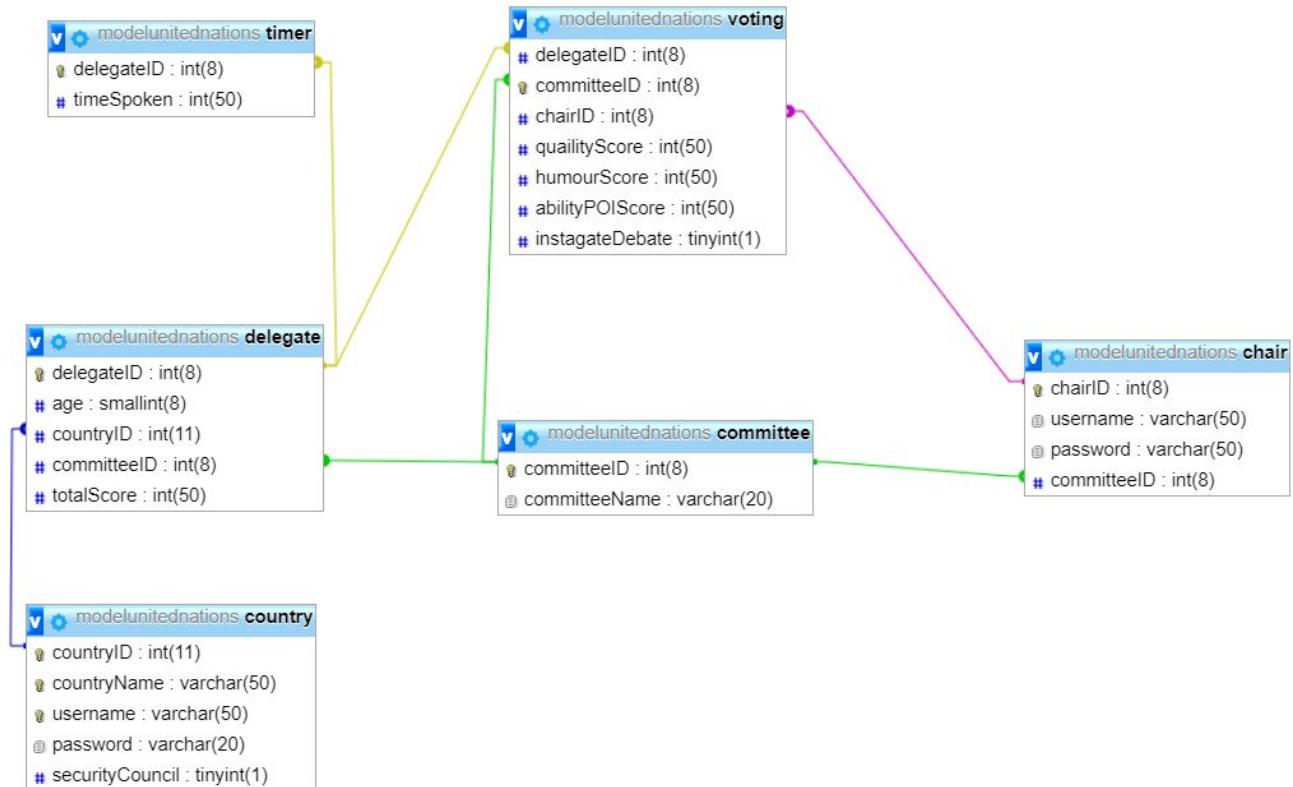
Entity Relationship Diagram

Stage 1: This was created and designed before the implementation of my program. I thought through what I wanted my program to do and how I wanted to achieve my ideas and so came up with this database to store all my data.

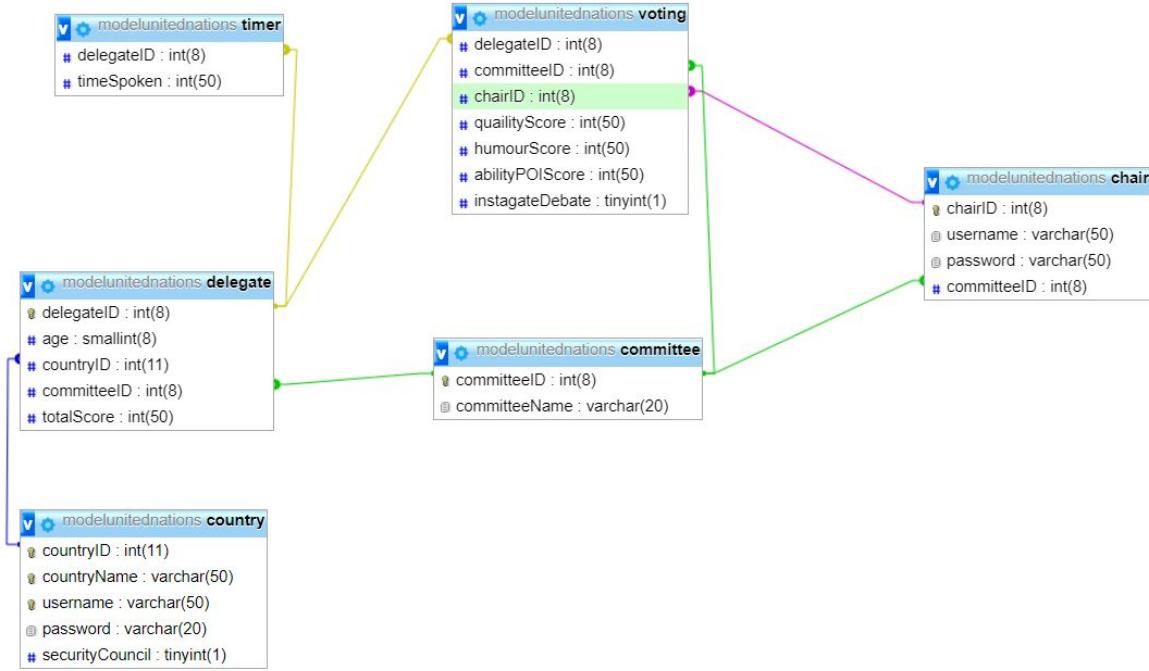


Stage 2: After the implementation of my database and throughout the implementation of my program I realised that I needed to create new tables and add relationships. I realised this once I started creating my program and thinking more about how I would achieve tasks. This was particularly true whenever I needed to use SQL statements to read from the database and whether I could get the information I desired from the database correctly.

I decided to separate the ‘timeSpoken’ from the ‘voting’ table and instead put this in a new table called ‘timer’. I then created a one to many relationship from the delegate table to the timer table. I also decided to create a one-to-many relationship between the ‘committee’ and ‘voting’ table. I also created another table called ‘country’ which had a one-to-many relationship between that table and the ‘delegate’ table.



Stage 3: My database changed very slightly again because I encountered a problem in the code and realised that I had made a mistake in the voting table. I realised I had made the '*committeeID*' field a primary key when I hadn't meant to. This was then fixed and changed.



Pseudocode

1. Home Page
2. Login
3. Delete Profile
4. Registration Chair
5. Registration Delegation
6. Voting Page
7. Timer
8. Awards Committee
9. Awards Overall
10. Delegate
11. Delegate Wrapper
12. Delegation
13. Delegation Wrapper
14. Number of Countries
15. Total Score
16. Sorting

1. **CLASS** HomePage

- 1.1. **IF** button_login **IS** pressed **THEN**
- 1.2. *<set up for class 2 >*
- 1.3. **ELSE IF** button_deleteProfile **IS** pressed **THEN**
- 1.4. *<set up for class 3 >*
- 1.5. **ELSE IF** button_registrationChair **IS** pressed **THEN**
- 1.6. *<set up for class 4>*
- 1.7. **ELSE IF** button_registrationDelegation **IS** pressed **THEN**
- 1.8. *<set up for class 5>*
- 1.9. **END IF**

END CLASS

2. **CLASS** Login

2.1. **PROCEDURE** button_enter_is_pressed()
2.2. *<connect to database>*
2.3. **RECIEVE** username **FROM (STRING)** usernameInputBox
2.4. **RECIEVE** password **FROM (STRING)** passwordInputBox

2.5. **IF** username = null **OR** password = null **THEN**
2.6. **SEND** "Please fill out the username and/or password fields **TO**
 DISPLAY
2.7. **ELSE**
2.8.
2.9. **TRY**
2.10. *<SQL Statement matching username and password to a record>*

2.11. **IF** SQL Statement returns a chair instance **THEN**
2.12. **SEND** "Welcome chair" **TO DISPLAY**
2.13. *<set up for class 6 >*
2.14. **ELSE IF** SQL Statement returns a delegation instance **THEN**
2.15. **SEND** "Welcome Delegation" **TO DISPLAY**
2.16. *<set up for class 9 >*
2.17. **ELSE**
2.18. **SEND** "I'm sorry there is no record matching your username and
 password please try again" **TO DISPLAY**
2.19. **END IF**
2.20. **CATCH**
2.21. **SEND** "Error Message" **TO DISPLAY**
2.22. **END CATCH**
2.23. **END IF**
2.24. **END PROCEDURE**

//Back Button

2.25. **PROCEDURE** button_back_is_pressed()
2.26. *<set up for class 1>*
2.27. **END PROCEDURE**

END CLASS

3. **CLASS** DeleteProfile

3.1. **PROCEDURE** button_submit_is_pressed()

//User Inputs

3.2. **RECIEVE** username **FROM (STRING)** usernameInputBox

3.3. **RECIEVE** password **FROM (STRING)** passwordInputBox

//Checking Values with Database

3.4. <connect to database>

3.5. **TRY**

3.6. <SQL Statement matching username and password to a record>

3.7. **IF** SQL Statement returns a chair instance **THEN**

3.8. **SEND** “Are you sure you want to delete your profile” **TO DISPLAY**

3.9. **RECEIVE** safetyCheck **FROM (STRING)** KEYBOARD

//Validation

3.10. **IF** safetyCheck = “Yes” **THEN**

//Profile Deletion

3.11. <SQL Delete Statement>

3.12. **SEND** “Profile Successfully Deleted” **TO DISPLAY**

3.13. **ELSE**

3.14. **SEND** “Profile NOT Deleted” **TO DISPLAY**

3.15. **END IF**

3.16. **ELSE IF** SQL Statement returns a delegation instance **THEN**

3.17. **SEND** “Are you sure you want to delete your profile” **TO DISPLAY**

3.18. **RECEIVE** safetyCheck **FROM (STRING)** KEYBOARD

//Validation

3.19. **IF** safetyCheck = “Yes” **THEN**

//Profile Deletion

```
3.20.      <SQL Delete Statement>
3.21.      SEND "Profile Successfully Deleted" TO DISPLAY
3.22.      ELSE
3.23.          SEND "Profile NOT Deleted" TO DISPLAY
3.24.      END IF

3.25.      ELSE
3.26.          SEND "I'm sorry there is no record matching your username and
            password please try again" TO DISPLAY
3.27.      END IF

3.28.      CATCH
3.29.          SEND "Error Message" TO DISPLAY
3.30.      END CATCH

3.31.      END PROCEDURE
3.32.
3.33.      PROCEDURE button_back_is_pressed()
3.34.          <set up for class 1>
3.35.      END PROCEDURE

END CLASS
```

4. **CLASS** RegistrationChair
- 4.1. **IF** radioButton_Disarmarment **IS selected THEN**
 - 4.2. **SET** committee **TO 1**
 - 4.3. **END IF**
 - 4.4. **IF** radioButton_EcoSoc **IS selected THEN**
 - 4.5. **SET** committee **TO 2**
 - 4.6. **END IF**
 - 4.7. **IF** radioButton_Environment **IS selected THEN**
 - 4.8. **SET** committee **TO 3**
 - 4.9. **END IF**
 - 4.10. **IF** radioButton_SecurityCouncil **IS selected THEN**
 - 4.11. **SET** committee **TO 4**
 - 4.12. **END IF**

 - 4.13. **PROCEDURE** button_submit_is_pressed()
 - 4.14. *<connect to database>*
 - 4.15. **RECEIVE** username **FROM (STRING)** usernameInputBox
 - 4.16. **RECEIVE** password1 **FROM (STRING)** password1InputBox
 - 4.17. **RECEIVE** password2 **FROM (STRING)** password2InputBox
 - 4.18. **TRY**
 - 4.19. **SET** numberOfUsernames **TO** *<SQL Count Statement for the number of records with the same username>*
 - 4.20. **CATCH**
 - 4.21. **SEND** error message **TO DISPLAY**
 - 4.22. **SET** numberOfUsernames **TO 0**
 - 4.23. **END CATCH**

//Validating Username and Password

 - 4.24. **IF** numberOfUsernames = 0 **THEN**
 - 4.25. **IF** password1 = password2 **THEN**
 - 4.26. *<Make sure password is above four characters and has either a special character or a digit>*

//Creating New Chair Record

 - 4.27. **TRY**
 - 4.28. *<SQL Insert Statement>*

4.29. **SEND** “New Chair Registered” **TO DISPLAY**
4.30. **CATCH**
4.31. **SEND** “Error Message” **TO DISPLAY**
4.32. **END CATCH**
4.33. **ELSE**
4.34. **SEND** “Two passwords don’t match. Please try again.” **TO DISPLAY**
4.35. **END IF**
4.36. **ELSE**
4.37. **SEND** “Not a unique username, please try again.” **TO DISPLAY**
4.38. **END IF**
4.39. **END PROCEDURE**

//Back button

4.40. **PROCEDURE** button_back_is_pressed()
4.41. *<set up for class 1>*
4.42. **END PROCEDURE**
END CLASS

5. **CLASS** RegistrationDelegation

5.1. <connect to database>
5.2. **DECLARE** countryID **AS INTEGER**
5.3. **IF** checkBox_SecurityCouncil_is_selected **THEN**
5.4. **SET** securityCouncil **TO** 1
5.5. **END IF**

5.6. **PROCEDURE** button_submit_is_pressed()

//User Input Values Being Set

5.7. **RECEIVE** username **FROM (STRING)** usernameInputBox
5.8. **RECEIVE** password1 **FROM (STRING)** password1InputBox
5.9. **RECEIVE** password2 **FROM (STRING)** password2InputBox
5.10. **RECEIVE** countryName **FROM (STRING)** comboBox_country
5.11. **RECEIVE** environmentAge **FROM (INTEGER)** environmentInputBox
5.12. **RECEIVE** disarmamentAge **FROM (INTEGER)** disarmamentInputBox
5.13. **RECEIVE** ecosocAge **FROM (INTEGER)** ecosocInputBox
5.14. **IF** secuirtyCouncil = 0 **THEN**
5.15. **RECEIVE** securityCouncilAge **FROM (INTEGER)**
 secuirtyCouncillInputBox
5.16. **END IF**
5.17. GetCountryID()

//Validating Username and Password

5.18. **TRY**
5.19. **SET** numberofUsernames **TO** <SQL Count Statement for the
 number of records with the same username>
5.20. **CATCH**
5.21. **SEND** error message **TO DISPLAY**
5.22. **SET** numberofUsernames **TO** 0
5.23. **END CATCH**

5.24. **IF** numberofUsernames = 0 **THEN**
5.25. **IF** password1 = password2 **THEN**
5.26. <Make sure password is above four characters and has either a
 special character or a digit>

//Creating New Chair Record

```
5.27.    TRY
5.28.        <SQL Insert Statement>
5.29.        SEND "Registration Successful" TO DISPLAY
5.30.    CATCH
5.31.        SEND "Error Message" TO DISPLAY
5.32.    END CATCH
5.33.    ELSE
5.34.        SEND "Two passwords don't match. Please try again." TO
DISPLAY
5.35.    END IF
5.36.    ELSE
5.37.        SEND "Not a unique username, please try again." TO DISPLAY
5.38.    END IF
5.39. END PROCEDURE

5.40. PROCEDURE GetCountryID()
5.41.    TRY
5.42.        <SQL Query Statement>
5.43.        SET countryID TO <SQL Query Statement>
5.44.    CATCH
5.45.        SEND "Error Message" TO DISPLAY
5.46.    END CATCH
5.47. END PROCEDURE

//Back button
5.48. PROCEDURE button_back_is_pressed()
5.49.     <set up for class 1 >
5.50. END PROCEDURE

END CLASS
```

6. **CLASS** Voting

6.1. <connect to database>

//Global Variables

6.2. **DECLARE** chairID **AS INTEGER**

6.3. **DECLARE** committelD **AS INTEGER**

6.4. **DECLARE** countryID **AS INTEGER**

6.5. **DECLARE** delegateID **AS INTEGER**

//Mutators

6.6. **PROCEDURE** setChairID(**INTEGER** chairID)

6.7. **SET** this.chairID **TO** chairID

6.8. **END PROCEDURE**

6.9. **PROCEDURE** setCommitteelD(**INTEGER** committeelD)

6.10. **SET** this.committeelD **TO** committeelD

6.11. **END PROCEDURE**

6.12. **PROCEDURE** button_submit_is_pressed()

//Get User Inputs

6.13. **RECEIVE** qualityOfSpeech **FROM INTEGER**(sliderQuality)

6.14. **RECEIVE** humour **FROM INTEGER** (sliderHumour)

6.15. **RECEIVE** poi **FROM INTEGER** (sliderPOI)

6.16. **RECEIVE** country **FROM STRING**(comboBox_country)

6.17. **GetCountryID()**

6.18. **GetDelegateID()**

6.19. **IF** radioButton_Instagate is selected **THEN**

6.20. **SET** instagateDebate **TO** 2

6.21. **ELSE**

6.22. **SET** instagateDebate **TO** 0

6.23. **END IF**

//Creating New Voting Record

6.24. **TRY**

```

6.25.      <SQL Insert Statement>
6.26.      SEND "Your vote has been processed" TO DISPLAY
6.27.      CATCH
6.28.          SEND "Error Message" TO DISPLAY
6.29.      END CATCH
6.30.
6.31.
6.32.      END PROCEDURE

6.33.      PROCEDURE GetCountryID()
6.34.          TRY
6.35.              <SQL Query Statement>
6.36.              SET countryID TO <SQL Query Statement>
6.37.          CATCH
6.38.              SEND "Error Message" TO DISPLAY
6.39.          END CATCH
6.40.      END PROCEDURE

6.41.      PROCEDURE GetDelegateID()
6.42.          TRY
6.43.              <SQL Query Statement>
6.44.              SET delegateID TO <SQL Query Statement>
6.45.          CATCH
6.46.              SEND "Error Message" TO DISPLAY
6.47.          END CATCH
6.48.      END PROCEDURE

                                //Sign Out button
6.49.      PROCEDURE button_signOut_is_pressed()
6.50.          <set up for class 1 >
6.51.      END PROCEDURE

                                //Timer button
6.52.      PROCEDURE button_timer_is_pressed()
6.53.          <set up for class 7 >
6.54.      END PROCEDURE

                                //Awards Committee button

```

6.55. **PROCEDURE** button_awardsCommittee_is_pressed()
6.56. *<set up for class 8>*
6.57. **END PROCEDURE**

END CLASS

7. **CLASS** timer

7.1. **DECLARE** minMax **AS INTEGER**
 7.2. **DECLARE** secMax **AS INTEGER**

7.3. **CONSTRUCTOR()**
 7.4. *<connect to database>*
 7.5. **SET** counter **TO** 0
 7.6. **END CONSTRUCTOR**

7.7. **PROCEDURE** setCommitteeID(**INTEGER** committeeID)
 7.8. **SET** this.committeeID **TO** committeeID
 7.9. **END PROCEDURE**

7.10. **PROCEDURE** setChairID(**INTEGER** chairID)
 7.11. **SET** this.chairID **TO** chairID
 7.12. **END PROCEDURE**

7.13. **PROCEDURE** setComboBox()
 7.14. *<SQL Query Statement>*
 7.15. **SET** comboBoxCountryNames **TO** *<SQL Query Statement>*
 7.16. **END PROCEDURE**
 7.17.
 7.18. **PROCEDURE** button_start_is_pressed()

//User Input

7.19. **RECEIVE** minMax **FROM INTEGER**(jSpinnerMin)
 7.20. **RECEIVE** secMax **FROM INTEGER**(jSpinnerSec)

//Validation

7.21. **IF** minMax = 0 **OR** secMax = 0 **THEN**
 7.22. **SEND** “Error Message” **TO DISPLAY**
 7.23. **ELSE**
 7.24. *<timer begins>*
 7.25. **END IF**
 7.26. **END PROCEDURE**

7.27. **PROCEDURE** button_pause_is_pressed()
7.28. *<timer stops>*
7.29. **END PROCEDURE**
7.30. **PROCEDURE** button_submit_is_pressed()

//Setting Local Variables

7.31. **DECLARE** timeActuallySpoke **AS INTEGER**
7.32. **RECEIVE** minLeft **FROM (INTEGER) Timer**
7.33. **RECEIVE** secLeft **FROM (INTEGER) Timer**
7.34. **RECEIVE** countryName **FROM (STRING) comboBox**

//Calculating Time Actually Spoken In Seconds

7.35. **SET** timeActuallySpoke **TO** ((minMax - minLeft) * 60) + (secMax - secLeft)

//Inserting Time Value Into Database

7.36. **SET** countryID **TO** GetCountryID(countryName)
7.37. **SET** delegateID **TO** GetDelegateID(countryID)
7.38. **TRY**
7.39. *<SQL Insert Statement>*
7.40. **SEND** “The time has been noted” **TO DISPLAY**
7.41. **CATCH**
7.42. **SEND** “Error Message” **TO DISPLAY**
7.43. **END CATCH**
7.44. **END PROCEDURE**
7.45.
7.46. **FUNCTION** GetCountryID(countryName) **RETURNS INTEGER**
7.47. **TRY**
7.48. *<SQL Query Statement>*
7.49. **SET** countryID **TO** *<SQL Query Statement>*
7.50. **CATCH**
7.51. **SEND** “Error Message” **TO DISPLAY**
7.52. **SET** countryID **TO** 0
7.53. **END CATCH**

7.54. **RETURN** countryID

7.55. **END FUNCTION**

7.56. **FUNCTION** GetDelegateID(countryID) **RETURNS INTEGER**

7.57. **TRY**

7.58. *<SQL Query Statement>*

7.59. **SET** delegateID **TO** *<SQL Query Statement>*

7.60. **CATCH**

7.61. **SEND** “Error Message” **TO DISPLAY**

7.62. **SET** delegateID **TO** 0

7.63. **END CATCH**

7.64. **RETURN** delegateID

7.65. **END FUNCTION**

7.66.

7.67. **PROCEDURE** button_voting_is_pressed()

7.68. *<set up for class 6 >*

7.69. **END PROCEDURE**

7.70. **PROCEDURE** button_signOut_is_pressed()

7.71. *<set up for class 1 >*

7.72. **END PROCEDURE**

7.73. **END CLASS**

8. **CLASS** AwardsCommittee

- 8.1. **DECLARE** count **AS INTEGER**
- 8.2. **DECLARE** delegates **AS Delegate[]**

- 8.3. **CONSTRUCTOR()**
- 8.4. *<connect to database>*
- 8.5. **END CONSTRUCTOR**

//Main Procedure were sub-procedures are mostly called upon

- 8.6. **PROCEDURE** ranking()
- 8.7. **DECLARE** committeeName **AS STRING INITIALLY AS " "**

- 8.8. **IF** committeeID = 1 **THEN**
- 8.9. **SET** committeeName **TO "Disarmarment"**
- 8.10. **ELSE IF** committeeID = 2 **THEN**
- 8.11. **SET** committeeName **TO "EcoSoc"**
- 8.12. **ELSE IF** committeeID = 3 **THEN**
- 8.13. **SET** committeeName **TO "Envrionment"**
- 8.14. **ELSE IF** committeeID = 4 **THEN**
- 8.15. **SET** committeeName **TO "Security Council"**
- 8.16. **END IF**
- 8.17. **SEND** "Viewing Results for: " & committName & " " **TO DISPLAY**

- 8.18. SetDelegateArray()
- 8.19. **DECLARE** sortObjects **AS Sorting**
- 8.20. sortObjects.bubbleSortDelegates(delegates, count)
- 8.21. DisplayResults()
- 8.22. Awards()
- 8.23. **END PROCEDURE**

//Creating Array of Objects

- 8.24. **PROCEDURE** SetDelegateArray()
- 8.25. **DECLARE** numCountries **AS NumberCountries**
- 8.26. **SET** count **TO** numCountries.NumberCountries(committeeID)

//Initialise Delegate Object Array

8.27. **DECLARE delegates AS Delegate[count]**

//Create Array Placeholder Variable

8.28. **DECLARE i AS INTEGER INITIALLY 0**

//Retrieve Needed Values From The Database

8.29. **TRY**

8.30. **<SQL Query Statement>**

8.31. **WHILE (<Still more SQL Query Results>)**

8.32. **SET delegateID TO <SQL Query Statement>**

8.33. **SET countryName TO <SQL Query Statement>**

8.34. **SET age TO <SQL Query Statement>**

//Assigning Delegate Object Value

8.35. **DECLARE delegates[i] AS Delegate**

8.36. delegates[i].setCountryName(countryName)

8.37. delegates[i].setAge(age)

8.38. delegates[i].setTotalScore(delegateID)

8.39. **SET i TO i + 1**

8.40. **END WHILE**

8.41.

8.42. **CATCH**

8.43. **SEND "Error Message" TO DISPLAY**

8.44. **END CATCH**

8.45. **END PROCEDURE**

8.46. **PROCEDURE DisplayResults()**

8.47. **FOR i FROM 0 TO count - 1 DO**

8.48. **SEND delegates[i].displayMessage() TO RankingListBox**

8.49. **END FOR**

8.50. **END PROCEDURE**

8.51. **PROCEDURE Awards()**

8.52. **SET** bestDelegate **TO** delegate[0].getCountryName
8.53. **SEND** bestDelegate **TO** BestDelegateTextBox
8.54. **SET** highlyCommended **TO** delegate[1].getCountryName
8.55. **SEND** highlyCommended **TO** HighlyCommendedTextBox
8.56. **SET** commended **TO** delegate[2].getCountryName
8.57. **SEND** commended **TO** CommendedTextBox

8.58. juniorDelegateAward()

8.59. **END PROCEDURE**

8.60. **PROCEDURE** juniorDelegateAward()

8.61. **DECLARE** found **AS BOOLEAN INITIALLY** False
8.62. **DECLARE** counter **AS INTEGER INITIALLY** 0
8.63. **WHILE** found = false **OR** counter = count **DO**
8.64. **IF** delegates[counter].getAge() < 16 **THEN**
8.65. **SET** juniorDelegate **TO** delegates[counter].countryName
8.66. **SEND** juniorDelegate **TO** JuniorDelegateTextBox
8.67. **SET** found **TO** True
8.68. **END IF**
8.69. **SET** counter **TO** counter + 1
8.70. **END WHILE**

//Ensuring Program is Robust

8.71. **IF** found = False **THEN**
8.72. **SEND** "No Juniors in Committee" **TO** JuniorDelegateTextBox
8.73. **END IF**

8.74. **END PROCEDURE**

//Mutators

8.75. **PROCEDURE** setCommitteeID(**INTEGER** committeeID)
8.76. **SET** this.committeeID **TO** committeeID
8.77. **END PROCEDURE**

8.78. **PROCEDURE** setChairID(**INTEGER** chairID)
8.79. **SET** this.chairID **TO** chairID
8.80. **END PROCEDURE**

//Buttons

8.81. **PROCEDURE** button_voting_is_pressed()
8.82. *<set up for class 6 >*
8.83. **END PROCEDURE**

8.84. **PROCEDURE** button_signOut_is_pressed()
8.85. *<set up for class 1 >*
8.86. **END PROCEDURE**
8.87. **PROCEDURE** button_awardsOverall_is_pressed()
8.88. *<set up for class 9 >*
8.89. **END PROCEDURE**

END CLASS

9. **CLASS** AwardsOverall

- 9.1. **DECLARE** count **AS INTEGER**
- 9.2. **DECLARE** delegations **AS Delegation[]**

- 9.3. **CONSTRUCTOR()**
- 9.4. *<connect to database>*
- 9.5. **END CONSTRUCTOR**

//Main Procedure were sub-procedures are mostly called upon

- 9.6. **PROCEDURE** ranking()
- 9.7.
- 9.8. SetDelegationArray()
- 9.9. **DECLARE** sortObjects **AS Sorting**
- 9.10. sortObjects.bubbleSortDelegations(delegations, count)
- 9.11. DisplayResults()
- 9.12. Awards()
- 9.13. **END PROCEDURE**

//Creating Array of Objects

- 9.14. **PROCEDURE** SetDelegationArray()
- 9.15. **DECLARE** numCountries **AS NumberCountries**
- 9.16. **SET** count **TO** numCountries.NumberCountries(*<any committeeID apart from Security Council>*)

//Initialise Delegate Object Array

- 9.17. **DECLARE** delegations **AS Delegation[count]**

//Create Array Placeholder Variable

- 9.18. **DECLARE** i **AS INTEGER INITIALLY 0**

//Retrieve Needed Values From The Database

- 9.19. **TRY**
- 9.20. *<SQL Query Statement>*

9.21. **WHILE** (<Still more SQL Query Results>)
9.22. **SET** countryID **TO** <SQL Query Statement>
9.23. **SET** countryName **TO** <SQL Query Statement>
9.24. **SET** totalScore **TO** <SQL Query Statement>

//Assigning Delegate Object Value

9.25. **DECLARE** delegations[i] **AS Delegation**
9.26. delegations[i].setCountryName(countryName)
9.27. delegations[i].setTotalScore(delegateID)
9.28. **SET** i **TO** i + 1
9.29. **END WHILE**
9.30.
9.31. **CATCH**
9.32. **SEND** “Error Message” **TO DISPLAY**
9.33. **END CATCH**

9.34. **END PROCEDURE**

9.35. **PROCEDURE** DisplayResults()
9.36. **FOR** i **FROM** 0 **TO** count - 1 **DO**
9.37. **SEND** delegations[i].displayMessage() **TO RankingListBox**
9.38. **END FOR**
9.39. **END PROCEDURE**

9.40. **PROCEDURE** Awards()

9.41. **SET** bestDelegation **TO** delegation[0].getCountryName
9.42. **SEND** bestDelegation **TO BestDelegationTextBox**
9.43. **SET** highlyCommended **TO** delegation[1].getCountryName
9.44. **SEND** highlyCommended **TO HighlyCommendedTextBox**
9.45. **SET** commended **TO** delegation[2].getCountryName
9.46. **SEND** commended **TO CommendedTextBox**

9.47. **END PROCEDURE**

//Mutators

9.48. **PROCEDURE** setCommitteeID(**INTEGER** committeeID)

9.49. **SET** this.committeeID **TO** committeeID
9.50. **END PROCEDURE**

9.51. **PROCEDURE** setChairID(**INTEGER** chairID)
9.52. **SET** this.chairID **TO** chairID
9.53. **END PROCEDURE**

//Buttons

9.54. **PROCEDURE** button_voting_is_pressed()
9.55. *<set up for class 6 >*
9.56. **END PROCEDURE**

9.57. **PROCEDURE** button_signOut_is_pressed()
9.58. *<set up for class 1 >*
9.59. **END PROCEDURE**

9.60. **PROCEDURE** button_awardsCommittee_is_pressed()
9.61. **IF** committeeID = 0 **THEN**

//Delegation Login Would Get This Option

9.62. **SEND** "Please pick a committee result you would like to view" **TO**
DISPLAY
9.63. **RECEIVE** committeeID **FROM (INTEGER) KEYBOARD**
9.64. *<set up for class 8 >*
9.65. **ELSE**
 //Chair Login Would Get This Path

9.66. *<set up for class 8>*
9.67. **END IF**
9.68. **END PROCEDURE**

END CLASS

10. **CLASS** Delegate

- 10.1. **DECLARE** countryName **AS PRIVATE STRING**
- 10.2. **DECLARE** age **AS PRIVATE INTEGER**
- 10.3. **DECLARE** totalScore **AS INTEGER**

- 10.4. **CONSTRUCTOR()**
- 10.5. **END CONSTRUCTOR**

//Mutators

- 10.6. **PROCEDURE** setTotalScore(**INTEGER** delegateID)
 DECLARE score **AS TotalScore** //Creating an object from a class
 score.calculatingTotalScore(delegateID)//Accessing methods from
 that clas
- 10.7. **SET** totalScore **TO** score.getTotalScore()
- 10.8. **END PROCEDURE**

- 10.9. **PROCEDURE** setCountryName(**STRING** name)
 SET countryName **TO** name
- 10.10. **END PROCEDURE**

- 10.11. **PROCEDURE** setAge(**INTEGER** age)
 SET this.age **TO** age
- 10.12. **END PROCEDURE**

//Accessors

- 10.13. **FUNCTION** getTotalScore() **RETURNS INTEGER**
 RETURN totalScore
- 10.14. **END FUNCTION**

- 10.15. **FUNCTION** getCountryName() **RETURNS STRING**
 RETURN countryName
- 10.16. **END FUNCTION**

- 10.17. **FUNCTION** getAge() **RETURNS INTEGER**
 RETURN age
- 10.18. **END FUNCTION**

//Other Methods

10.27. **FUNCTION** displayMessage() **RETURNS STRING**
10.28. **RETURN** " " & countryName & ":" & totalScore & ""
10.29. **END FUNCTION**
10.30. **FUNCTION** compareTo(**DELEGATE** compareDelegate) **RETURNS INTEGER**
10.31. **DECLARE** compareQuantity **AS INTEGER INITIALLY ((DELEGATE)**
 compareDelegate).getTotalScore()
10.32. **RETURN** compareQuantity - this.totalScore
10.33. **END FUNCTION**

END CLASS

11. **CLASS** delegateWrapper

11.1. **DECLARE d AS DELEGATE**

11.2. **CONSTRUCTOR(DELEGATE d)**

11.3. **SET this.d AS d**

11.4. **END CONSTRUCTOR**

END CLASS

12. **CLASS** Delegation

- 12.1. **DECLARE** countryName **AS PRIVATE STRING**
- 12.2. **DECLARE** countryID **AS PRIVATE INTEGER**
- 12.3. **DECLARE** totalscore **AS INTEGER**

- 12.4. **CONSTRUCTOR()**
- 12.5. **END CONSTRUCTOR**

//Mutators

- 12.6. **PROCEDURE** setTotalScore(**INTEGER** totalScore)
- 12.7. **SET** this.totalScore **TO** totalScore
- 12.8. **END PROCEDURE**

- 12.9. **PROCEDURE** setCountryName(**STRING** name)
- 12.10. **SET** countryName **TO** name
- 12.11. **END PROCEDURE**

- 12.12. **PROCEDURE** setCountryID(**INTEGER** countryID)
- 12.13. **SET** this.countryID **TO** countryID
- 12.14. **END PROCEDURE**

//Accessors

- 12.15. **FUNCTION** getTotalScore() **RETURNS INTEGER**
- 12.16. **RETURN** totalScore
- 12.17. **END FUNCTION**

- 12.18. **FUNCTION** getCountryName() **RETURNS STRING**
- 12.19. **RETURN** countryName
- 12.20. **END FUNCTION**

//Other Methods

- 12.21. **FUNCTION** displayMessage() **RETURNS STRING**
- 12.22. **RETURN** " " & countryName & ":" & totalScore & "
- 12.23. **END FUNCTION**

```
12.24.  FUNCTION compareTo(DELEGATION compareDelegation) RETURNS
        INTEGER
12.25.    DECLARE compareQuantity AS INTEGER INITIALLY
            ((DELEGATION) compareDelegation).getTotalScore()
12.26.    RETURN compareQuantity - this.totalScore
12.27. END FUNCTION

END CLASS
```

13. **CLASS** delegationWrapper

13.1. **DECLARE d AS DELEGATION**

13.2. **CONSTRUCTOR(DELEGATION d)**

13.3. **SET this.d AS d**

13.4. **END CONSTRUCTOR**

END CLASS

14. **CLASS** NumberCountries

14.1. **FUNCTION** NumberCountries(**INTEGER** committeeID) **RETURNS**
INTEGER

14.2. *<connect to database>*

14.3. **DECLARE** numberOfCountries **AS INTEGER INITIALLY 0**

14.4. **TRY**

14.5. *<SQL Query Count Statement>*

14.6. **SET** numberOfCountries **TO** *<SQL Query Count Statement>*

14.7. **CATCH**

14.8. **SEND** “Error Message” **TO DISPLAY**

14.9. **END CATCH**

14.10. **RETURN** numberOfCountries

14.11. **END FUNCTION**

END CLASS

15. **CLASS** TotalScore

15.1. **CONSTRUCTOR()**
15.2. <connect to database>
15.3. **END CONSTRUCTOR**

15.4. **PROCEDURE** calculatingTotalScore(**INTERGER** delegateID)

//Retrieving Values From Database

15.5. **TRY**
15.6. <SQL Query Statement>
15.7. **SET** qualityScore **TO** <SQL Query Statement>
15.8. **SET** humourScore **TO** <SQL Query Statement>
15.9. **SET** abilityPOIScore **TO** <SQL Query Statement>
15.10. **SET** instagateDebate **TO** <SQL Query Statement>
15.11. **SET** timer **TO** <SQL Query Statement>
15.12. **CATCH**
15.13. **SEND** “Error Message” **TO DISPLAY**
15.14. **END CATCH**

//Calculating Total Score

15.15. **SET** totalScore **TO** ((2*qualityScore) + humourScore +
 (2*abilityPOIScore) + instagateDebate + timer)

//Inputting Results Into Database

15.16. **TRY**
15.17. <SQL Update Statement>
15.18. **CATCH**
15.19. **SEND** “Error Message” **TO DISPLAY**
15.20. **END CATCH**

15.21. **END PROCEDURE**

15.22. **FUNCTION** getTotalScore() **RETURNS INTEGER**
15.23. **RETURN** this.totalScore
15.24. **END FUNCTION**

END CLASS

16. CLASS Sorting

//Delegate

- 16.1. **PROCEDURE** bubbleSortDelegates(**ARRAY OF Delegate** delegates, **INTEGER** length)
- 16.2. **FOR** outerloop **FROM** (length - 2) **TO** 0 **STEP** -1 **DO**
- 16.3. **FOR** counter **FROM** 0 **TO** outerloop **DO**
- 16.4. **IF** delegates[counter].compareTo(delegates[counter + 1]) > 0
THEN

//Placing objects in wrappers so the whole of the object can be swapped

- 16.5. **DECLARE** wrapper **AS DelegateWrapper**(delegates[counter])
- 16.6. **DECLARE** wrapper2 **AS DelegateWrapper**(delegates[counter + 1])
- 16.7. **swap**(wrapper, wrapper2)

//Reassigning the objects in the wrappers to the actual objects

- 16.8. **SET** delegates[counter] **TO** wrapper.d
- 16.9. **SET** delegates[counter + 1] **TO** wrapper2.d
- 16.10. **END IF**
- 16.11. **END FOR**
- 16.12. **END FOR**
- 16.13. **END PROCEDURE**
- 16.14. **PROCEDURE** swap(**DelegateWrapper** dw, **DelegateWrapper** dw2)
- 16.15. **DECLARE** temp **AS Delegate** **INITIALLY** dw.d
- 16.16. **SET** dw.d **TO** dw2.d
- 16.17. **SET** dw2.d **TO** temp
- 16.18. **END PROCEDURE**

//Delegation

- 16.19. **PROCEDURE** bubbleSortDelegations(**ARRAY OF Delegation** delegations, **INTEGER** length)

16.20. **FOR** outerloop **FROM** (length - 2) **TO** 0 **STEP** -1 **DO**
16.21. **FOR** counter **FROM** 0 **TO** outerloop **DO**

16.22. **IF** delegations[counter].compareTo(delegations[counter + 1]) > 0
 THEN

//Placing objects in wrappers so the whole of the object can be swapped

16.23. **DECLARE** wrapper **AS**
 DelegationWrapper(delegations[counter])
16.24. **DECLARE** wrapper2 **AS**
 DelegationWrapper(delegations[counter + 1])
16.25. swap2(wrapper, wrapper2)

//Reassigning the objects in the wrappers to the actual objects

16.26. **SET** delegations[counter] **TO** wrapper.d
16.27. **SET** delegations[counter + 1] **TO** wrapper2.d

16.28. **END IF**
16.29. **END FOR**
16.30. **END FOR**
16.31. **END PROCEDURE**

16.32. **PROCEDURE** swap2(**DelegationWrapper** dw, **DelegationWrapper** dw2
)
16.33. **DECLARE** temp **AS** **Delegation** **INITIALLY** dw.d
16.34. **SET** dw.d **TO** dw2.d
16.35. **SET** dw2.d **TO** temp
16.36. **END PROCEDURE**

END CLASS

Implementation

Database

SQL Statements To Create Table

```
CREATE TABLE chair (
    chairID INT(8) NOT NULL AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(50) NOT NULL,
    password VARCHAR(50) NOT NULL,
    CONSTRAINT committeeID FOREIGN KEY(committee),
    CONSTRAINT delegateID FOREIGN KEY(delegate)
)ENGINE=InnoDB DEFAULT CHARSET=latin1;;
```

```
CREATE TABLE committee (
    committeeID INT(8) NOT NULL AUTO_INCREMENT PRIMARY KEY,
    committeeName VARCHAR(20) NOT NULL
)ENGINE=InnoDB DEFAULT CHARSET=latin1;;
```

```
CREATE TABLE country (
    countryID INT(11) NOT NULL AUTO_INCREMENT PRIMARY KEY,
    countryName VARCHAR(50) NOT NULL,
    username VARCHAR(50) NOT NULL,
    password VARCHAR(20) NOT NULL,
    securityCouncil TINYINT(1) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE delegate (
    delegateID INT(8) NOT NULL AUTO_INCREMENT PRIMARY KEY,
    age SMALLINT(8) NOT NULL,
    countryID INT(11) NOT NULL,
    committeeID INT(8) NOT NULL FOREIGN KEY(committee),
    totalScore INT(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE timer (
```

```
delegateID INT(8) NOT NULL FOREIGN KEY(delegate),
timeSpoken INT(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE voting (
    delegateID INT(8) NOT NULL FOREIGN KEY(delegate),
    committeeeID INT(8) NOT NULL FOREIGN KEY(committee),
    chairID INT(8) NOT NULL FOREIGN KEY(chair),
    quailityScore INT(50) NOT NULL,
    humourScore INT(50) NOT NULL,
    abilityPOIScore INT(50) NOT NULL,
    instagateDebate TINYINT(1) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Java

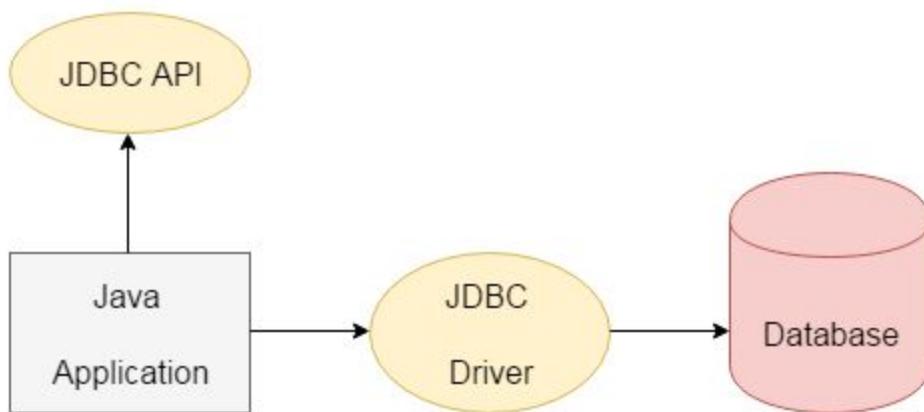
User Interface

Since I was using java netbeans to implement my program the easiest tool to develop the user interface was Java Swing. This wasn't taught as part of the course and I had to do plenty of research when I was creating the interface. I had to look at many different websites and watched videos which is referenced in my *Record of Progress*. Although we didn't get taught this it gave me a chance to self learn and enhance my adaptive ability of coding since I have designed user interfaces in other programming environments before.

Connection to Database

During the implementation of my program I realised I would need to connect my program to database so that the information stored on the database could be reflected in my program. This criteria was not apart of the *Advanced Higher Computing Science* course so a lot of research was done to complete a connection.

A diagram for what the connection to a database looks like can be shown in the diagram below:



When creating a connection to a database you need to realise that in order for the database on phpMyAdmin to communicate with the program you would need to make use of the *JDBC* functions. *JDBC* stands for 'Java Database Connectivity'.¹³ *JDBC* is a

¹³ <https://www.javatpoint.com/java-jdbc>
Image: <https://www.javatpoint.com/java-jdbc>

type of *Java API* that allows you to connect and interact with the database. A *Java API* is a shorthand for *Java Application Programming Interface*. The *Java API* is just predefined classes and functions that have already been tested that are already pre-installed or downloaded onto *Java Netbeans*.

When connecting to the database there are two main requirements needed: *JDBC API* and *JDBC Driver*. The *JDBC API* is used specifically for accessing tabular data that is stored in relational databases. The use of *JDBC API* means that data can be added, updated, deleted or fetched from the database. The *JDBC Driver* is the software component which allows the java application to directly interact with a database. A certain driver named *MySQL* was used in my database as I was interacting with my database using SQL statements.

```
//db parameters  
String url = "jdbc:mysql://localhost:3306/modelunitednations";
```

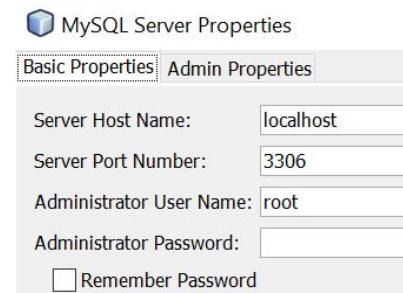
Above is a line of my code which shows the url for my database. It can be seen that the use of the *JDBC* function is being used.

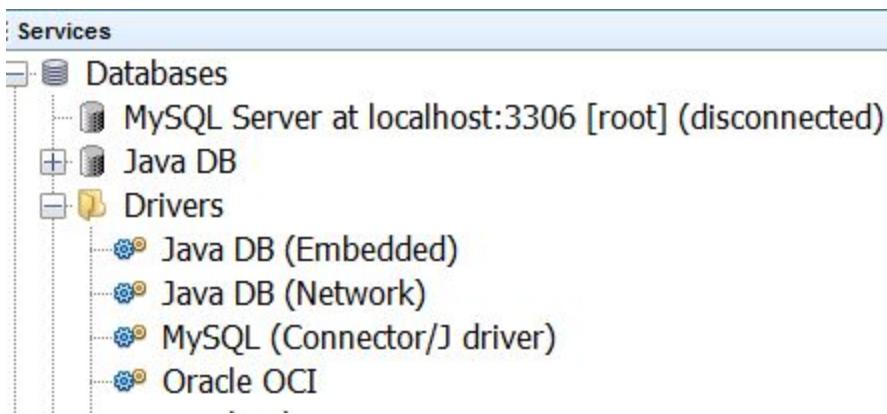
In some applications a *JDBC DriverManager* can be used. A *DriverManager* just manages any different drivers that are being used. This was an unnecessary addition as I only had an interaction with one database and therefore only a driver was necessary to complete my application's database connection however I also ended up using a *DriverManager* incase I needed to use the connection for other databases as well.

```
//create a connection  
connection = DriverManager.getConnection(url, db_username, db_password);
```

Above is a line of my code which shows how I have used the *DriverManager* to get a connection to my database.

When setting up the connection I needed to make sure that the server I was using to host the database was connected and so I needed I went to the server settings of *Java Netbeans* and inputted the information shown on the form to the right. The information for the server was all found on *Xampp* and also found on *phpMyAdmin*.





In my application you can see how I have needed select the server I am working with and how I am interacting with it (MySQL) and it can be seen that there is a connection made with the database, there is only a disconnection

currently as the server I am running my computer on hasn't started. It can also be seen how I have made use of the Drivers so as to create the connection to the database.

Since I knew I wanted to make use of my connection multiple times throughout the program, to increase efficiency, I created a class for the connection of the database and then had a subclass of the connection being made and then an accessor which would be used to access the connection so as this connection variable could be used in other modules of code throughout the program.

The code below is how I connected to the database:

```

1 import java.sql.Connection;
2 import java.sql.DriverManager;
3 import java.sql.SQLException;
4
5 /*
6  * To change this license header, choose License Headers in Project Properties.
7  * To change this template file, choose Tools | Templates
8  * and open the template in the editor.
9 */
10
11 /**
12 *
13 * @author j.mcclure
14 */
15 public class connectToDatabase {
16     Connection connection;
17     //method
18     public void connectToDatabase(){
19
20
21     try{
22         connection = null;
23         //db parameters
24         String url = "jdbc:mysql://localhost:3306/modelunitednations"; //last section in URL shows which table is being connected to
25         String db_username = "root";
26         String db_password = "";
27
28         //create a connection
29         connection = DriverManager.getConnection(url, db_username, db_password);
30
31     } catch(SQLException e){
32         System.out.println(e.getMessage());
33     } finally {
34         try{
35             if(connection == null)
36             {
37                 connection.close();
38
39             }
40             }catch(SQLException ex){
41                 System.out.println(ex.getMessage());
42             }
43         }
44
45     public Connection getConnectionVariable(Connection connection ){
46         this.connection = connection;
47         return connection;
48     }
49
50
51 }
52

```

Home Page



The user will be welcomed onto this screen when the application is ran. This screen is used for a 'base' throughout the program and whenever the user presses wants to sign out they are taken back to this page. The main code in this program is the coding for the buttons and bringing the user to the respective page to whichever button they press.

The code of the following page is shown below:

```

public class homePage extends javax.swing.JFrame {

    /**
     * Creates new form homePage
     */
    public homePage() {
        initComponents();
    }

}

@SuppressWarnings("unchecked")
Generated Code

private void jButtonLoginActionPerformed(java.awt.event.ActionEvent evt) {
    //close current form
    this.setVisible(false);
    //open corresponding form
    login1 jframe_sub = new login1();
    jframe_sub.setVisible(true);

}

private void jButtonRegistrationChairActionPerformed(java.awt.event.ActionEvent evt) {
    //close current form
    this.setVisible(false);
    //open corresponding form
    registrationChair jframe_sub = new registrationChair();
    jframe_sub.setVisible(true);

}

private void jButtonRegistrationDelegationActionPerformed(java.awt.event.ActionEvent evt) {
    //close current form
    this.setVisible(false);
    //open corresponding form
    registrationDelegation jframe_sub = new registrationDelegation();
    jframe_sub.setVisible(true);
}

private void jButton_deleteActionPerformed(java.awt.event.ActionEvent evt) {
    this.setVisible(false);

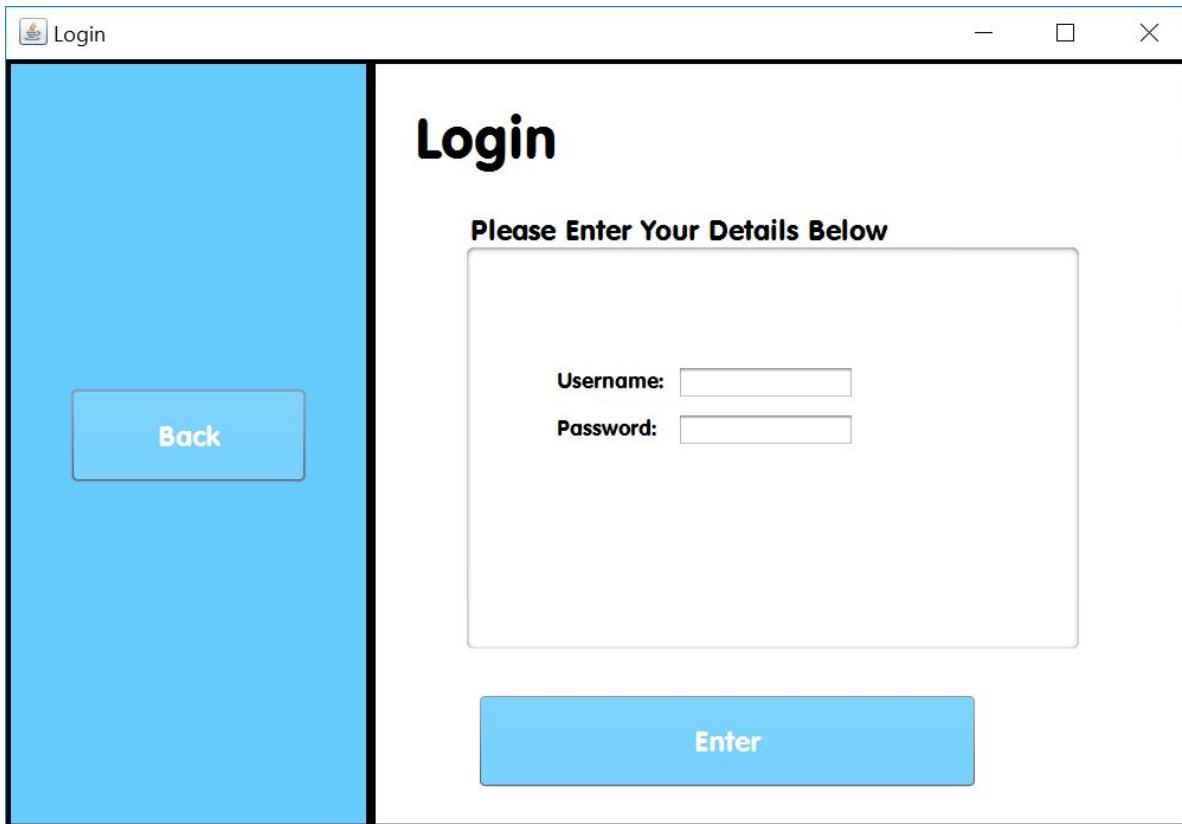
    deleteProfile form = new deleteProfile();
    form.setVisible(true);
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    Look and feel setting code (optional)

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            //get committeeCountryArray to get set up and read before program has to do anything with
            //CommitteeCountry2DArray delegateArraySetup = new CommitteeCountry2DArray();
            //delegateArraySetup.CommitteeCountry2DArray();
            //load home page
            new homePage().setVisible(true);
        }
    });
}
}

```

Login Page



Once brought to this page the user will enter their username and password which will be checked against the database. If there is a record with the same username and password then the user will be allowed into the system however what they can do will depend on their access rights (whether they are a chair or a delegate)

In my code I have firstly connected to the database by creating an object from the 'connect to database' class. A query has been made and has sent to the database and the results have been accessed. If there is a record then the appropriate login successful message will be displayed. Since there are two different actors in my program two different queries were made to check against two tables in the database to decide whether the login was a chair or a delegation.

The code of the following page is shown below:

```

7 import java.sql.Connection;
8 import java.sql.PreparedStatement;
9 import java.sql.ResultSet;
10 import java.sql.SQLException;
11 import javax.swing.JOptionPane;
12
13 /**
14 *
15 * @author j.mcclure
16 */
17 public class login1 extends javax.swing.JFrame {
18
19     //create global instance variables
20     Connection connection;
21     ResultSet rs_chair = null;
22     ResultSet rs_delegate = null;
23     PreparedStatement ps = null;
24
25     public login1() {
26         initComponents();
27     }
28
29     @SuppressWarnings("unchecked")
30     Generated Code
178
179     private void jButton2MouseClicked(java.awt.event.MouseEvent evt) {
180
181
182         //connect to database
183         connectToDatabase Test = new connectToDatabase();
184         Test.connectToDatabase();
185         //update connection variable
186         connection = Test.getConnectionVariable(connection);
187
188         String sql_chair = "SELECT `username`, `password`, `committeeID`, `chairID` FROM `chair` WHERE `username` = ? AND `password` = ?";
189         String sql_delegate = "SELECT `username`, `password` FROM `country` WHERE `username` = ? AND `password` = ?";
190
191     try {
192         //creating chair query
193         ps = connection.prepareStatement(sql_chair);
194         ps.setString(1, jTextField1.getText());
195         ps.setString(2, String.valueOf(jPasswordField1.getPassword()));
196         rs_chair = ps.executeQuery();
197         //creating delegate query
198         ps = connection.prepareStatement(sql_delegate);
199         ps.setString(1, jTextField1.getText());
200         ps.setString(2, String.valueOf(jPasswordField1.getPassword()));
201         rs_delegate = ps.executeQuery();
202
203         // Checking for empty field username
204         if(jTextField1.getText().length()==0)
205         {
206             JOptionPane.showMessageDialog(null, "Empty fields detected ! Please fill up all fields");
207         }else if(jPasswordField1.getPassword().length==0) // Checking for empty field password
208         {
209             JOptionPane.showMessageDialog(null, "Empty fields detected ! Please fill up all fields");
210         }
211         //cheking username and password match
212         if(rs_chair.next()) //if match and login is a chair
213         {
214
215             JOptionPane.showMessageDialog(null,"Login Successful. Welcome chair");
216
217             //get parameter values needed to pass in
218             int committeeID = rs_chair.getInt("committeeID");
219             int chairID = rs_chair.getInt("chairID");
220             // String username = rs_chair.getString("username");
221             // String password = rs_chair.getString("password");
222
223             //closing the current page
224             this.setVisible(false);

```

```

223 //opening corresponding page
224 votingPage1 jframe_sub = new votingPage1();
225 jframe_sub.setChairID(chairID);
226 jframe_sub.setCommitteeID(committeeID);
227 jframe_sub.setCombobox();//set values for voting page combobox
228
229 jframe_sub.setVisible(true);
230
231 } else if(rs_delegate.next()) //if match and login is a delegate/ delegation
{
232
233 JOptionPane.showMessageDialog(null,"Login Successful. Welcome delegate");
234
235 //closing the current page
236 this.setVisible(false);
237
238 AwardsOverall form = new AwardsOverall();
239 form.setCommitteeID(0);
240 form.ranking();
241 form.setVisible(true);
242 }else{
243 JOptionPane.showMessageDialog(null,"Login Unsuccessful");
244 }
245
246 } catch (SQLException e) {
247 JOptionPane.showMessageDialog(null, e);
248 }
249
250 }
251
252 private void jButton_backActionPerformed(java.awt.event.ActionEvent evt) {
253 this.setVisible(false);
254
255 homePage form = new homePage();
256 form.setVisible(true);
257 }

```

```

259 /**
260 * @param args the command line arguments
261 */
262 public static void main(String args[]) {
263 /* Set the Nimbus look and feel */
264 Look and feel setting code (optional)
265
266 /* Create and display the form */
267 java.awt.EventQueue.invokeLater(new Runnable() {
268 public void run() {
269 new login1().setVisible(true);
270 }
271 });
272 }

```

Registration Chair

The screenshot shows a Windows application window titled "Chair Registration". The window has a blue sidebar on the left containing a "Back" button. The main area is titled "Profile Information" and contains fields for "Username" (empty), "Password (must be at least 5 characters and include a digit or a special character)" (containing asterisks), "Please ReEnter Password" (containing asterisks), and a "Committee" section with four radio buttons: "Disarmament", "Environment", "EcoSoc", and "Security Council". A large blue "Submit Registration" button is at the bottom.

Once the user presses the 'submit registration' button. The code will start running. A lot of the code contains validation on the fields. For example makes sure the username entered is unique, the passwords both match and are at least 5 characters in length and contains a digit or a special character. There is also code on the checkbox buttons which means that only one can be selected at a time so the chair isn't accidentally registered in two committee's. If all of the fields are correct and there are no errors then the program will create a new record within the 'chair' table. A message saying that the registration was successful will appear. All the information in the field will then return to the default look (shown above).

The code of the following page is shown below:

```

2 import java.sql.Connection;
3 import java.sql.PreparedStatement;
4 import java.sql.ResultSet;
5 import java.sql.SQLException;
6 import java.sql.Statement;
7 import java.util.regex.Matcher;
8 import java.util.regex.Pattern;
9 import javax.swing.JOptionPane;
10
11 public class registrationChair extends javax.swing.JFrame {
12
13     //global variables
14     private String username; //private variables used in encapsulation to increase security of sensitive information
15     private String password1;
16     private String passwordAgain;
17
18     public int committee = 0;
19     Connection connection;
20     PreparedStatement ps;
21     ResultSet rs;
22
23     public registrationChair() {
24         initComponents();
25     }
26
27
28     @SuppressWarnings("unchecked")
29     Generated Code
277
280     private void jRadioButton_DisarmamentActionPerformed(java.awt.event.ActionEvent evt) {
281         committee = 1;
282
283         //if one clicked then the others can't be selected
284         jRadioButton_EcoSoc.setSelected(false);
285         jRadioButton_Environment.setSelected(false);
286         jRadioButton_SecurityCouncil.setSelected(false);
287     }

```

```

287     private void jButton_SubmitActionPerformed(java.awt.event.ActionEvent evt) {
288
289         //connect to database
290         connectToDatabase Test = new connectToDatabase();
291         Test.connectToDatabase();
292         //update connection variable
293         connection = Test.getConnectionVariable(connection);
294
295
296         String inputUsername = jTextField_Username.getText();
297         String inputPassword = String.valueOf(jPasswordField_1.getPassword());
298         String inputPassword2 = String.valueOf(jPasswordField_Again.getPassword());
299
300         int num_Username = 1;
301         try{
302             String matchUsername = "SELECT COUNT(username) AS num_username FROM chair WHERE username = ?"; //sql statement to count the number of rows with the same username
303             ps = connection.prepareStatement(matchUsername);
304             ps.setString(1, inputUsername);
305             rs = ps.executeQuery();
306
307             while (rs.next())
308             {
309                 num_Username = rs.getInt("num_username");
310             }
311
312         }catch (SQLException e){
313             JOptionPane.showMessageDialog(null, e);
314         }
315
316         if(num_Username == 0) //check to see if username is unique
317         {
318             if(inputPassword.equals(inputPassword2))
319             {
320                 //looking for digits and special characters
321                 Pattern digit = Pattern.compile("[0-9]");
322                 Pattern special = Pattern.compile ("[^#%$&_*+=|<>?()\\\"\\~]");
```

```

324     Matcher hasDigit = digit.matcher(inputPassword);
325     Matcher hasSpecial = special.matcher(inputPassword);
326
327     if (hasDigit.find() == false && hasSpecial.find() == false)
328     {
329         JOptionPane.showMessageDialog(null, "I am sorry the Password you entered doesn't include a digit or special character. Please re-enter a password");
330     }
331     else if (inputPassword.length() <= 4)
332     {
333         JOptionPane.showMessageDialog(null, "I am sorry the Password you entered does not meet the required length of 5 characters. Please try again.");
334     }
335     else if (inputUsername.equals("null"))
336     {
337         JOptionPane.showMessageDialog(null, "I am sorry you need to input a username.");
338     }
339     else if (inputPassword.equals("null"))
340     {
341         JOptionPane.showMessageDialog(null, "I am sorry you need to input a password.");
342     }
343     else if (committee == 0)
344     {
345         JOptionPane.showMessageDialog(null, "I am sorry you need to select the committee you wish to chair.");
346     }
347     else
348     {
349         try{
350             Statement state=(Statement)connection.createStatement();
351
352             String insert="INSERT INTO `chair`(`username`, `password`, `committeeID`) VALUES ('"+inputUsername+"','"+inputPassword+"','"+committee+"')";
353             state.executeUpdate(insert);
354
355             JOptionPane.showMessageDialog(null, "New Chair Registered");
356
357             //defaulting all fields
358             jTextField_Username.setText("");
359             jTextField_Password_1.setText("12345678");
360             jTextField_Password_Again.setText("12345678");
361             jRadioButton_Disarmament.setSelected(false);
362             jRadioButton_Environment.setSelected(false);
363             jRadioButton_EcoSoc.setSelected(false);
364
365             jRadioButton_SecurityCouncil.setSelected(false);
366
367         } catch (Exception e){
368             JOptionPane.showMessageDialog(null, e.getMessage(), "Error", 1);
369         }
370     }
371
372 }
373
374
375 private void jRadioButton_EcoSocActionPerformed(java.awt.event.ActionEvent evt) {
376     committee = 2;
377     //if one clicked then the others can't be selected
378     jRadioButton_Disarmament.setSelected(false);
379     jRadioButton_Environment.setSelected(false);
380     jRadioButton_SecurityCouncil.setSelected(false);
381 }
382
383 private void jRadioButton_EnvironmentActionPerformed(java.awt.event.ActionEvent evt) {
384     committee = 3;
385     //if one clicked then the others can't be selected
386     jRadioButton_Disarmament.setSelected(false);
387     jRadioButton_EcoSoc.setSelected(false);
388     jRadioButton_SecurityCouncil.setSelected(false);
389 }

```

```
391  private void jRadioButton_SecurityCouncilActionPerformed(java.awt.event.ActionEvent evt) {  
392      committee = 4;  
393      //if one clicked then the others can't be selected  
394      jRadioButton_Disarmament.setSelected(false);  
395      jRadioButton_Environment.setSelected(false);  
396      jRadioButton_EcoSoc.setSelected(false);  
397  }  
398  
399  private void jButton_backActionPerformed(java.awt.event.ActionEvent evt) {  
400      this.setVisible(false);  
401  
402      HomePage form = new HomePage();  
403      form.setVisible(true);  
404  }  
405  
406  private void jPasswordField_AgainActionPerformed(java.awt.event.ActionEvent evt) {  
407      // TODO add your handling code here:  
408  }  
409  
410  /**  
411   * @param args the command line arguments  
412   */  
413  public static void main(String args[]) {  
414      /* Set the Nimbus look and feel */  
415      Look and feel setting code (optional)  
416  
417      /* Create and display the form */  
418      java.awt.EventQueue.invokeLater(new Runnable() {  
419          public void run() {  
420              new registrationChair().setVisible(true);  
421          }  
422      });  
423  }  
424
```

Registration Delegation

The screenshot shows a Windows application window titled "Delegation Registration". On the left, there is a vertical blue sidebar with a "Back" button. The main content area has two sections: "Profile Information" on the left and "Country Information" on the right. The "Profile Information" section contains three text input fields: "Username", "Password (must be at least 5 characters)", and "Please ReEnter Password". The "Country Information" section contains several input fields and dropdowns: a checkbox for "Security Council Country", a dropdown for "Country" set to "Argentina", and four text input fields for "Disarmament", "Environment", "EcoSoc", and "Security Council (If applicable)". At the bottom is a large blue "Submit Registration" button.

The program will start running officially once the 'submit registration' button is pressed. The main purpose of the code is for the validation of the user inputs, this includes making sure the username is unique, both passwords match and contain at least five characters one either being a digit or special character. There is also validation on whether the country selected is unique. A place which has no validation is where the ages of the delegates are entered. If all of the inputs are correct and there are no errors then a new record in the 'country' table in the database will be created. Three or four new records will also be created in the 'delegates' table within the database. Once this is complete a message will be displayed saying that the registration was successful.

The code of the following page is shown below:

```

1 import java.sql.Connection;
2 import java.sql.PreparedStatement;
3 import java.sql.ResultSet;
4 import java.sql.SQLException;
5 import java.sql.Statement;
6 import java.util.regex.Matcher;
7 import java.util.regex.Pattern;
8 import javax.swing.JOptionPane;
9 /**
10 *
11 * @author j.mcclure
12 */
13 public class registrationDelegation extends javax.swing.JFrame {
14     //global variables
15     Connection connection;
16     ResultSet rs_country = null;
17     PreparedStatement ps = null;
18     ResultSet rs_countryID = null, rs = null;
19
20     int securityCouncil = 0;
21     int countryID;
22
23     String countryName;
24
25
26     public registrationDelegation() {
27
28         initComponents();
29     }
30
31     @SuppressWarnings("unchecked")
32     Generated Code
350
351     private void jCheckBox_SecurityCouncilActionPerformed(java.awt.event.ActionEvent evt) {
352
353     }

```

```

355     private void jButton_registrationActionPerformed(java.awt.event.ActionEvent evt) {
356
357         //connect to database
358         connectToDatabase Test = new connectToDatabase();
359         Test.connectToDatabase();
360         //update connection variable
361         connection = Test.getConnectionVariable(connection);
362
363         String inputUsername = jTextField_Username.getText();
364         String inputPassword = String.valueOf(jPasswordField_1.getPassword());
365         String inputPassword2 = String.valueOf(jPasswordField_Again.getPassword());
366         int num.Username = 1;
367
368         try{
369             //see if country username is unique
370             String matchUsername = "SELECT COUNT(username) AS num_username FROM country WHERE username = ?"; //sql statement to count the number of rows with the same
371             ps = connection.prepareStatement(matchUsername);
372             ps.setString(1, inputUsername);
373             rs = ps.executeQuery();
374
375             while (rs.next())
376             {
377                 num.Username = rs.getInt("num_username");
378             }
379
380         }catch(SQLException e){
381             JOptionPane.showMessageDialog(null, e);
382         }
383
384         int environmentAge = Integer.parseInt(jTextField_Environment.getText());
385         int disarmamentAge = Integer.parseInt(jTextField_Disarmament.getText());
386         int ecosocAge = Integer.parseInt(ecosoc.getText());
387         int scAge = 0;
388         if (securityCouncil == 1)
389         {
390             scAge = Integer.parseInt(jTextField_SC.getText());
391         }

```

```

392 countryName = (String)jComboBox_country.getSelectedItem();
393
394 //make sure country is unique, no code needed as error message comes up since the SQL server has an error
395
396 if(num_Username == 0) //check to see if username is unique
{
    if(inputPassword.equals(inputPassword2))
    {
        //looking for digits and special characters
        Pattern digit = Pattern.compile("[0-9]");
        Pattern special = Pattern.compile ("[@#$%^&_*+=|<>?{}\\[\\]\\~]");

        Matcher hasDigit = digit.matcher(inputPassword);
        Matcher hasSpecial = special.matcher(inputPassword);

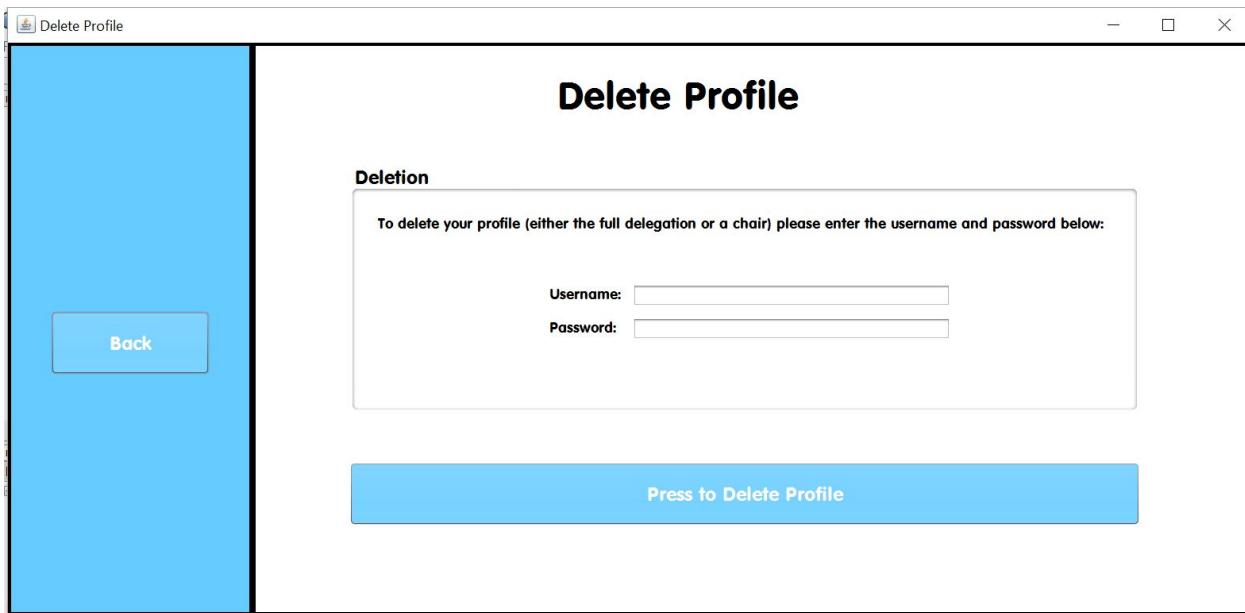
        if (hasDigit.find() == false && hasSpecial.find() == false)
        {
            JOptionPane.showMessageDialog(null, "I am sorry the Password you entered doesn't include a digit or special character. Please re-enter a password");
        }
        else if(inputPassword.length() <= 4)
        {
            JOptionPane.showMessageDialog(null, "I am sorry the Password you entered does not meet the required length of 5 characters. Please try again.");
        }
        else if(inputUsername.equals("null"))
        {
            JOptionPane.showMessageDialog(null, "I am sorry you need to input a username.");
        }
        else if (inputPassword.equals("null"))
        {
            JOptionPane.showMessageDialog(null, "I am sorry you need to input a password.");
        }
    }
}
422
423
424 try{
425     Statement state=(Statement)connection.createStatement();
426     //inserting delegation information into database
427     String insertDelegation="INSERT INTO `country`(`username`, `password`, `countryName`, `securityCouncil`) VALUES ('"+inputUsername+"','"+inputPassword+"','"+countryName+"','"+securityCouncil+"')";
428     state.executeUpdate(insertDelegation);
429     //get countryID after the country record is created
430     getCountryID(); //country ID is held in global variable to access
431
432     //inserting individual delegate information into database
433     String insertDelegateDis = "INSERT INTO `delegate`(`age`, `countryID`, `committeeID`, `totalScore`) VALUES('"+disarmamentAge+"', '"+countryID+"', '1', '0')";
434     state.executeUpdate(insertDelegateDis);
435     String insertDelegateEnv = "INSERT INTO `delegate`(`age`, `countryID`, `committeeID`, `totalScore`) VALUES('"+environmentAge+"', '"+countryID+"', '3', '0')";
436     state.executeUpdate(insertDelegateEnv);
437     String insertDelegateEco = "INSERT INTO `delegate`(`age`, `countryID`, `committeeID`, `totalScore`) VALUES('"+ecosocAge+"', '"+countryID+"', '2', '0')";
438     state.executeUpdate(insertDelegateEco);
439
440     if(securityCouncil == 1)
441     {
442         String insertDelegateSC = "INSERT INTO `delegate`(`age`, `countryID`, `committeeID`, `totalScore`) VALUES('"+scAge+"', '"+countryID+"', '4', '0')";
443         state.executeUpdate(insertDelegateSC);
444     }
445
446     JOptionPane.showMessageDialog(null,"Registration Successful");
447
448 } catch (Exception e){
449     JOptionPane.showMessageDialog(null, e.getMessage(), "Error", 1);
450
451     }
452
453     } else {
454         JOptionPane.showMessageDialog(null, "I am sorry the two passwords you entered do not match. Please try again.");
455     }
456
457 }

```

```
475     private void jButton_backActionPerformed(java.awt.event.ActionEvent evt) {
476         this.setVisible(false);
477
478         homePage form = new homePage();
479         form.setVisible(true);
480
481     }
482
483     /**
484      * @param args the command line arguments
485     */
486
487     public void getCountryID()
488     {
489
490         try{
491             //get list of countries so that combobox can update automatically depending on the committee
492             String sql_countryID = "SELECT `countryID` FROM `country` WHERE `countryName` = ? ";
493             ps = connection.prepareStatement(sql_countryID);
494             ps.setString(1, countryName);
495             rs_countryID = ps.executeQuery();
496
497             if(rs_countryID.next()) //if match and login is a chair
498             {
499                 countryID = rs_countryID.getInt("countryID");
500             }
501         } catch (Exception e){
502             JOptionPane.showMessageDialog(null, e.getMessage() , "Error", 1);
503         }
504     }
505 }
```

```
506     public static void main(String args[]) {
507         /* Set the Nimbus look and feel */
508         Look and feel setting code (optional)
509
510
511         /* Create and display the form */
512         java.awt.EventQueue.invokeLater(new Runnable() {
513             public void run() {
514                 new registrationDelegation().setVisible(true);
515             }
516         });
517     }
518 }
```

Delete Profile



The main part of the code will start running after the button 'Press to Delete Profile' is pressed. Once this happens the program will compare the inputs of the username and password against the database and determine whether or not the passwords match any records within the database. This is done by creating two queries; one which checks the inputs against the other chair records and the other against the delegation records. If there is a match then a message will be displayed asking again through a inputbox whether the user wishes to delete their profile. If the user types 'Y' then the profile will be deleted and a confirmation message will be displayed. If the user types 'N' or anything else then a message saying that the deletion of the profile didn't go ahead will be displayed.

The code of the following page is shown below:

```

1 import java.sql.Connection;
2 import java.sql.PreparedStatement;
3 import java.sql.ResultSet;
4 import java.sql.SQLException;
5 import javax.swing.JOptionPane;
6
7 public class deleteProfile extends javax.swing.JFrame {
8
9
10    //global variables
11    Connection connection;
12    PreparedStatement ps;
13    ResultSet rs_chair, rs_delegate;
14
15    public deleteProfile() {
16        initComponents();
17    }
18
19    /**
20     * This method is called from within the constructor to initialize the form.
21     * WARNING: Do NOT modify this code. The content of this method is always
22     * regenerated by the Form Editor.
23     */
24    @SuppressWarnings("unchecked")
25    Generated Code
26
27    private void jButton_backActionPerformed(java.awt.event.ActionEvent evt) {
28        this.setVisible(false);
29
30        HomePage form = new HomePage();
31        form.setVisible(true);
32    }

```

```

201 private void jButton2MouseClicked(java.awt.event.MouseEvent evt) {
202
203     //connect to database
204     connectToDatabase Test = new connectToDatabase();
205     Test.connectToDatabase();
206     //update connection variable
207     connection = Test.getConnectionVariable(connection);
208
209     String sql_chair = "SELECT `username`, `password`, `committeeID`, `chairID` FROM `chair` WHERE `username` = ? AND `password` = ?";
210     String sql_delegate = "SELECT `username`, `password` FROM `country` WHERE `username` = ? AND `password` = ?";
211     try {
212         //creating chair query
213         ps = connection.prepareStatement(sql_chair);
214         ps.setString(1, jTextField1.getText());
215         ps.setString(2, String.valueOf(jPasswordField1.getPassword()));
216         rs_chair = ps.executeQuery();
217         //creating delegate query
218         ps = connection.prepareStatement(sql_delegate);
219         ps.setString(1, jTextField1.getText());
220         ps.setString(2, String.valueOf(jPasswordField1.getPassword()));
221         rs_delegate = ps.executeQuery();
222
223         // Checking for empty field username
224         if(jTextField1.getText().length()==0)
225         {
226             JOptionPane.showMessageDialog(null, "Empty fields detected ! Please fill up all fields");
227         }else if(jPasswordField1.getPassword().length==0) // Checking for empty field password
228         {
229             JOptionPane.showMessageDialog(null, "Empty fields detected ! Please fill up all fields");
230         }
231         //local variable
232         String deletion_statement = "";
233         boolean wantDeletion = false;

```

```

235 //cheching username and password match
236 if(rs_chair.next()) //if match and login is a chair
{
    //input box
    String delete = JOptionPane.showInputDialog("Are you sure you would like to delete your chair profile? (Y/N)");
    //answer depending on input boxes statement
    if (delete.equalsIgnoreCase("Y"))
    {
        deletion_statement = "DELETE FROM `chair` WHERE `username` = ? AND `password` = ?; //SQL statement
        wantDeletion = true;
    }else{// when user doesnt want profile deleted
        JOptionPane.showMessageDialog(null, "Chair profile NOT deleted");
        wantDeletion = false;
    }
}
249
250 else if(rs_delegate.next()){//if match and login is delegation
    //input box
    String delete = JOptionPane.showInputDialog("Are you sure you would like to delete the whole delegation profile? (Y/N)");
    //answer depending on input boxes statement
    if (delete.equalsIgnoreCase("Y"))
    {
        deletion_statement = "DELETE FROM `country` WHERE `username` = ? AND `password` = ?; //SQL statement
        wantDeletion = true;
    }else{// when user doesnt want profile deleted
        JOptionPane.showMessageDialog(null, "Delegation profile NOT deleted");
        wantDeletion = false;
    }
}
263 else{ //when password and username don't match
    JOptionPane.showMessageDialog(null,"Password OR/AND Username is not correct");
}
266
267 if (wantDeletion == true){

    try{
        //deletion statement being executed
        ps = connection.prepareStatement(deletion_statement);
        ps.setString(1, jTextField1.getText());
        ps.setString(2, String.valueOf(jPasswordField1.getPassword()));
        ps.executeUpdate();
        JOptionPane.showMessageDialog(null, "Profile successfully deleted");
    }catch (SQLException e) {
        JOptionPane.showMessageDialog(null, "Profile couldn't be deleted :" + e + "");
    }
}
279 } catch (SQLException e) {
    JOptionPane.showMessageDialog(null, e);
}
281
282 }
283
284 private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
285     // TODO add your handling code here:
286 }
287
288 /**
289 * @param args the command line arguments
290 */
291 public static void main(String args[]) {
292     /* Set the Nimbus look and feel */
293     LookAndFeelSettingCode(optional)
294
295     /* Create and display the form */
296     java.awt.EventQueue.invokeLater(new Runnable() {
297         public void run() {
298             new deleteProfile().setVisible(true);
299         }
300     });
301 }
321

```

Voting Form

The screenshot shows a voting form window. On the left, there's a sidebar with three buttons: 'Sign Out', 'Awards Page', and 'Timer'. The main content area is titled 'Voting Form'. At the top right, there's a 'Country' section with a dropdown menu set to 'United States of America'. Below this is a 'Categories' section containing three horizontal sliders. The first slider, 'Quality of Speech', has its handle at the value 4. The second slider, 'Humour in Speech', has its handle at the value 1. The third slider, 'Ability to Handle Points of Information', also has its handle at the value 1. At the bottom of the categories section, there's a small message: 'Added New Information to Debate'. To the right of the categories section is a large blue button labeled 'Submit Vote'.

When the page is loaded the page is loaded as shown above. Before the page is loaded when an object of the page is created in the previous form (either login, timer or awards page) the values of ChairID and CommitteeID are set using a mutator. It should also be noted that the values in the dropdown ‘country of delegate’ list is set during this time. The dropdown list only contains countries in the chair’s committee and only countries which are registered. This means that the chair’s in the security council committee will have a slightly shorter list than those chairing other committees since not every delegation needs a member within the security council committee.

After the user inputs their information (adjust the sliders, checkbox and selects the country) they must press the ‘submit vote’ for the vote to be recorded. The vote is recorded by inserting a new record into the ‘voting’ table. Once this is complete and there are no errors a message will be displayed saying the vote was processed. The voting form will then go to a default with all the sliders adjusting back to the third position.

The code of the following page is shown below:

```
1 import javax.swing.JOptionPane;
2
3 import java.sql.Connection;
4 import java.sql.PreparedStatement;
5 import java.sql.ResultSet;
6 import java.sql.SQLException;
7 import java.sql.Statement;
8
9
10 public class votingPage1 extends javax.swing.JFrame {
11
12     //global variables
13     Connection connection;
14     Statement state; //insert
15
16     ResultSet rs = null; //select
17     PreparedStatement ps = null; //select
18
19     int instagateDebate;
20
21     int chairID;
22     int delegateID;
23     int committeeID;
24     int countryID;
25
26     public void setChairID(int chairID){
27         this.chairID = chairID;
28     }
29
30     public void setCommitteeID(int committeeID){
31         this.committeeID = committeeID;
32     }
33 }
```

```

34  public void setComboBox() {
35
36      try{
37
38          //get country names and add to drop down list
39          String country = "SELECT `countryName`, country.`countryID`, `committeeID` FROM `country`, `delegate` WHERE country.countryID = delegate.countryID AND `committeeID` = "+countryID;
40          ps = connection.prepareStatement(country);
41          rs = ps.executeQuery();
42
43          while (rs.next()) {
44              jComboBox1.addItem(rs.getString("countryName"));
45          }
46      }catch(SQLException e) {
47          JOptionPane.showMessageDialog(null, e);
48      }
49  }
50  public votingPage1(){
51
52      //create connection
53      //connect to database
54      connectToDatabase Test = new connectToDatabase();
55      Test.connectToDatabase();
56      //update connection variable
57      connection = Test.getConnectionVariable(connection);
58
59      //put interface in place
60      initComponents();
61
62  }
63 }
64 private void submitActionPerformed(java.awt.event.ActionEvent evt) {
65
66     //get values for slider qualities
67     int qualityOfSpeech = jSliderQuality.getValue();
68     int humour = jSliderHumour.getValue();
69     int poi = jSliderPOI.getValue();
70     String country = jComboBox1.getSelectedItem().toString();
71
72     try{
73
74         //countryName to countryID
75         String quereyCountry = "SELECT `countryID` FROM `country` WHERE `countryName` = ?";
76         ps = connection.prepareStatement(quereyCountry);
77         ps.setString(1, country);
78         rs = ps.executeQuery();
79
80         while (rs.next()) {
81             countryID = rs.getInt("countryID");
82         }
83
84         //get delegateID
85         String quereydelegate = "SELECT `delegateID` FROM `delegate` WHERE `countryID` = " + countryID+ " AND `committeeID` = "+committeeID+"";
86         ps = connection.prepareStatement(quereydelegate);
87         rs = ps.executeQuery();
88
89         while (rs.next()) {
90             delegateID = rs.getInt("delegateID");
91         }
92     }catch(SQLException e) {
93         JOptionPane.showMessageDialog(null, e);
94     }
95 }

```

```

397 try{
398     Statement state=(Statement)connection.createStatement();
399     String sql_insertVotingRecord = "INSERT INTO `voting` VALUES(`+delegateID+`,`+committeeID+`,`+chairID+`,`+qualityofspeech+`,`+humour+`,`+poi+`,`+instigateDebate+`)";
400     state.executeUpdate(sql_insertVotingRecord);
401
402     JOptionPane.showMessageDialog(null,"Your vote has been processed");
403     //default all voting slides
404     jSliderQuality.setValue(3);
405     jSliderHumour.setValue(3);
406     jSliderPOI.setValue(3);
407     jComboBox1.setSelectedItem(0);
408     jRadioButtonInstigate.setSelected(false);
409
410
411 } catch (Exception e){
412     JOptionPane.showMessageDialog(null, e.getMessage() , "Error cannot insert voting record", 1);
413 }
414
415

```

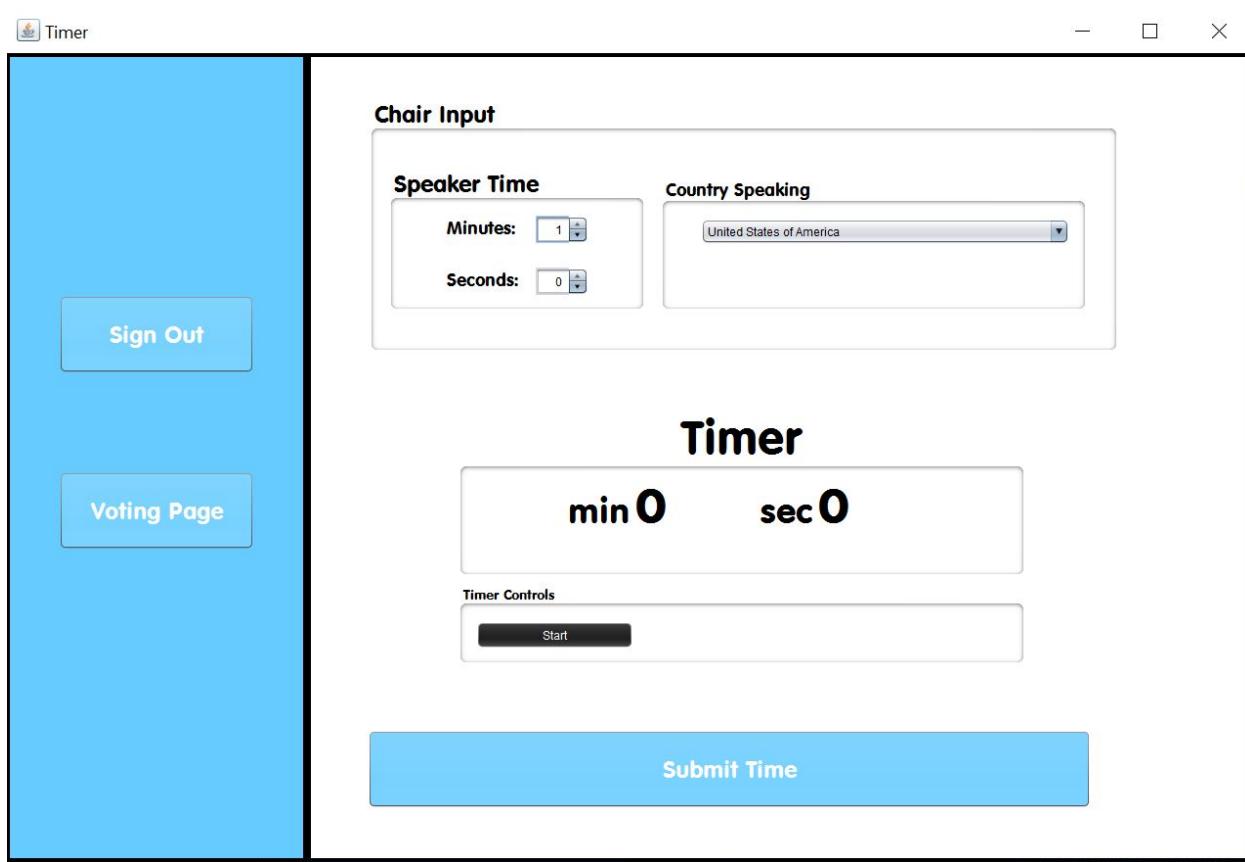
```

416 }
417
418 private void jRadioButtonInstigateActionPerformed(java.awt.event.ActionEvent evt) {
419     instigateDebate = 2;
420 }
421
422 private void jComboBox1ActionPerformed(java.awt.event.ActionEvent evt) {
423     // TODO add your handling code here:
424 }
425
426 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
427     // TODO add your handling code here:
428
429     this.setVisible(false); //get rid of the current form
430     UNTimer form = new UNTimer();
431
432     form.setCommitteeID(committeeID);
433     form.setChairID(chairID);
434     form.setCombobox();
435
436     form.setVisible(true); //display timer page
437
438 }
439
440 private void jButtonAwardsActionPerformed(java.awt.event.ActionEvent evt) {
441     this.setVisible(false);
442
443     AwardsCommittee form = new AwardsCommittee();
444     form.setVisible(true);
445     form.setCommitteeID(committeeID);
446     form.setChairID(chairID);
447     form.ranking();
448 }

```

```
450     private void jButton_logOutActionPerformed(java.awt.event.ActionEvent evt) {  
451  
452         this.setVisible(false);  
453  
454         HomePage form = new HomePage();  
455         form.setVisible(true);  
456  
457     }  
458  
459     /**  
460      * @param args the command line arguments  
461      */  
462     public static void main(String args[]) {  
463         /* Set the Nimbus look and feel */  
464         Look and feel setting code (optional)  
465  
466         /* Create and display the form */  
467         java.awt.EventQueue.invokeLater(new Runnable() {  
468             public void run() {  
469                 new votingPage1().setVisible(true);  
470             }  
471         });  
472     }  
473  
474 }
```

Timer



When the timer page is loaded it looks like the above image. The same process of having a unique dropdown country select list from the voting form is used in this page. The use of mutators is also used to set the chairID and committeeID. The user inputs the time they would like the speaker to speak for using the scroll boxes and once they have done this and also selected a speaker country they press the start button on the timer controls.

This activates the timer and the timer starts counting down. If the time of 0 minutes and 0 seconds is entered then a validation error appears asking the user to input a time. Once the timer is activated however the start button essentially disappears and instead a 'pause' button becomes visible. If the user presses the 'pause' button then the timer stops and the start button becomes visible again. This continues until either the timer has stopped or until the user selects the 'submit time' button. Once the 'submit time' button is pressed then a message is displayed saying that the time has been recorded. In the program this means that a new record was created and inserted into the 'timer'

table in the database using the delegateID from the delegate of the country and using the times in seconds. The time the speaker spoke for is determined by subtracting the time remaining (which is visible on the timer itself) away from the time initially set by the chairs.

Once all of this is complete the time on the actual timer is set back to 0 minutes and 0 seconds and the start button becomes visible again.

The inclusion of a timer was never studying in the advanced higher course and so I learnt how to implement this myself. I looked at a piece of code, that I referenced in my record of progress, on how to complete this and then tweaked this code to make it more suitable to my program and also make the timer run more smoothly.

The code of the following page is shown below:

```
1 import java.sql.Connection;
2 import java.sql.PreparedStatement;
3 import java.sql.ResultSet;
4 import java.sql.SQLException;
5 import java.sql.Statement;
6 import java.util.Timer;
7 import java.util.TimerTask;
8 import javax.swing.JOptionPane;
9
10 public class UNTimer extends javax.swing.JFrame {
11     //global variables
12     Connection connection;
13     Statement state; //insert
14
15     ResultSet rs = null; //select
16     PreparedStatement ps = null; //select
17
18     int instigateDebate;
19
20     int chairID;
21     int delegateID;
22     int committeeID;
23     int countryID;
24
25     //My timer variables:
26     Boolean isIt = false;
27     int sec;
28     int min;
29     int secMax;
30     int minMax;
31     int counter;
32
33     public void setCommitteeID(int committeeID){
34         this.committeeID = committeeID;
35     }
36 }
```

```

37  public void setChairID(int chairID){
38      this.chairID = chairID;
39  }
40
41  public void setCombobox(){
42
43      //add a section for the timer that won't count towards anything
44      jComboBox1.addItem("Unmoderated Caucus");
45
46      try{
47
48          //get country names and add to drop down list
49          String country = "SELECT `countryName`, `countryID` FROM `country`, `delegate` WHERE country.countryID = delegate.countryID AND `committeeID` = "+ 
50          ps = connection.prepareStatement(country);
51          rs = ps.executeQuery();
52
53          while (rs.next()) {
54              jComboBox1.addItem(rs.getString("countryName"));
55          }
56      }catch(SQLException e) {
57          JOptionPane.showMessageDialog(null, e);
58      }
59  }
60
61  public UNTimer() {
62
63      //create connection
64      //connect to database
65      connectToDatabase Test = new connectToDatabase();
66      Test.connectToDatabase();
67      //update connection variable
68      connection = Test.getConnectionVariable(connection);
69

```

```

70      //put interface in place
71      initComponents();
72      btnStart.setVisible(true);
73      btnPause.setVisible(false);
74
75      counter = 0;
76
77  }
78
79  @SuppressWarnings("unchecked")
80  Generated Code
392
393  private void btnStartMouseClicked(java.awt.event.MouseEvent evt) {
394
395      //get user inputs from box
396      minMax = Integer.parseInt(jSpinnerMin.getValue().toString());
397      secMax = Integer.parseInt(jSpinnerSec.getValue().toString());
398
399      if ((minMax == 0) && (secMax == 0)) //And operator
400      {
401          JOptionPane.showMessageDialog(null, "Error, please type in the time length (minute and second)");
402      }else{
403          btnStart.setVisible(false); //displaying buttons
404          btnPause.setVisible(true); //displaying buttons
405
406          Timer timer = new Timer(); //new timer
407
408          counter = counter + 1;
409
410          //set the timer, second and minute
411          if (counter == 1){
412              min = minMax;
413              sec = secMax;
414              timeLeftMin.setText(Integer.toString(min));
415              timeLeftSec1.setText(Integer.toString(sec));
416          }else{

```

```

4 476
4 477    while (rs.next()) {
4 478        countryID = rs.getInt("countryID");
4 479    }
4 480
4 481    //get delegateID
4 482    String quereydelegate = "SELECT `delegateID` FROM `delegate` WHERE `countryID`=" + countryID+ " AND `committeeID`="+committeeID+"";
4 483    ps = connection.prepareStatement(queredydelegate);
4 484    rs = ps.executeQuery();
4 485
4 486    while (rs.next()) {
4 487        delegateID = rs.getInt("delegateID");
4 488    }
4 489 }catch(SQLException e) {
4 490     JOptionPane.showMessageDialog(null, e);
4 491 }
4 492
4 493 try{
4 494     Statement state=(Statement)connection.createStatement();
4 495     String sql_insertTime = "INSERT INTO `timer` VALUES("+delegateID+"," + timeActualSpoke +")";
4 496     state.executeUpdate(sql_insertTime);
4 497
4 498     JOptionPane.showMessageDialog(null,"Your time has been noted");
4 499
4 500 } catch (Exception e){
4 501     JOptionPane.showMessageDialog(null, e.getMessage() , "Error cannot insert time into Timer table", 1);
4 502 }
4 503
4 504
4 505
4 506
4 507    //clear everything so that the timer can be used again
4 508    counter = 0;
4 509    timeLeftMin.setText("0");
4 510    timeLeftSec1.setText("0");
4 511    btnStart.setVisible(true);
4 512
4 513
4 514 }
4 515
4 516 private void jButtonVotingActionPerformed(java.awt.event.ActionEvent evt) {
4 517     // TODO add your handling code here:
4 518     this.setVisible(false); //get rid of the current form
4 519     votingPage1 form = new votingPage1();
4 520
4 521     form.setCommitteeID(committeeID);
4 522     form.setChairID(chairID);
4 523     form.setCombobox();
4 524
4 525     form.setVisible(true); //display voting page
4 526 }
4 527
4 528 private void jButton_signOutActionPerformed(java.awt.event.ActionEvent evt) {
4 529     this.setVisible(false);
4 530
4 531     homePage form = new homePage();
4 532     form.setVisible(true);
4 533 }
4 534     rs = ps.executeQuery();

```

```
538 [-] public static void main(String args[]) {  
539 [-]     /* Set the Nimbus look and feel */  
540 [+]  
541         Look and feel setting code (optional)  
561  
562 [-]     /* Create and display the form */  
563 [-]     java.awt.EventQueue.invokeLater(new Runnable() {  
564 [-]  
565             public void run() {  
566                 new UNTimer().setVisible(true);  
567             }  
568         } );  
569 }
```

Awards Committee

The screenshot shows a web application window titled "Results for the EcoSoc Committee". On the left, there is a sidebar with three buttons: "Sign Out", "Overall Results", and "Voting Form". The main content area is divided into two sections: "Awards" and "Ranking". The "Awards" section contains four entries: "Best Delegate" (United States of America), "Best Junior Delegate" (No Juniors In Committee), "Highly Commended" (Russia), and "Commended" (China). The "Ranking" section displays a table titled "Ranking of Countries According To Total Score" with the following data:

Rank	Country	Total Score
1.	United States of America	2194
2.	Russia	1830
3.	China	188
4.	Venezuela	116

This page is displayed when showing the committee awards.

This page creates an array of objects from the delegates class and also used the sorting class to sort the ranking. The main processes in this form is that the object values are set in this page and calls on the classes mentioned above. This page also selects the awards by selecting the first object in the array of objects as the 'best delegate' the second as 'highly commended' and so forth. The best delegate process was slightly more complicated as the first delegate in the ranking couldn't be picked but rather the age of the delegate also had to be looked at.

Noticeable traits in this page is that the label of the title will change depending on what committee the chair or delegate is viewing. It should be mentioned that a chair only has the restrictions to view their own committee and none of the other committees results whereas a delegation login can view all the different committee's results. So when logged in as a delegation when viewing the committee awards page there is another button visible which lets you change the committee you are viewing. When this button is pressed it brings up a choices box which lets you select a new committee to view (picture top right on this page). This option isn't available to chairs who login. However the 'voting' button on the navigation bar isn't visible to delegation logins. All of this is shown in the screenshot below which was taken after using a delegation login.



Sign Out
Overall Results

Results for the Disarmerment Committee

Awards

Best Delegate:	United States of America
Best Junior Delegate:	No Juniors In Committee
Highly Commended:	Russia
Commended:	Argentina

Ranking

Ranking of Countries According To Total Score
1. United States of America: 158
2. Russia: 44
3. Argentina: 0
4. China: 0
5. Venezuela: 0

[Choose New Committee Results To View](#)

The code of the following page is shown below:

Page 152 Of 220

```
1 import java.sql.Connection;
2 import java.sql.PreparedStatement;
3 import java.sql.ResultSet;
4 import java.sql.SQLException;
5 import javax.swing.DefaultListModel;
6 import javax.swing.JOptionPane;
7
8 /*
9  * To change this license header, choose License Headers in Project Properties.
10 * To change this template file, choose Tools | Templates
11 * and open the template in the editor.
12 */
13
14 /**
15 *
16 * @author j.mcclure
17 */
18 public final class AwardsCommittee extends javax.swing.JFrame {
19
20     //global variables
21     Connection connection;
22     ResultSet rs;
23     PreparedStatement ps;
24
25     int delegateID;
26     String countryname;
27     int age;
28     int chairID;
29     int committeeID;
30     int count;
31     //array of object
32     Delegate[] delegates;
33 }
```

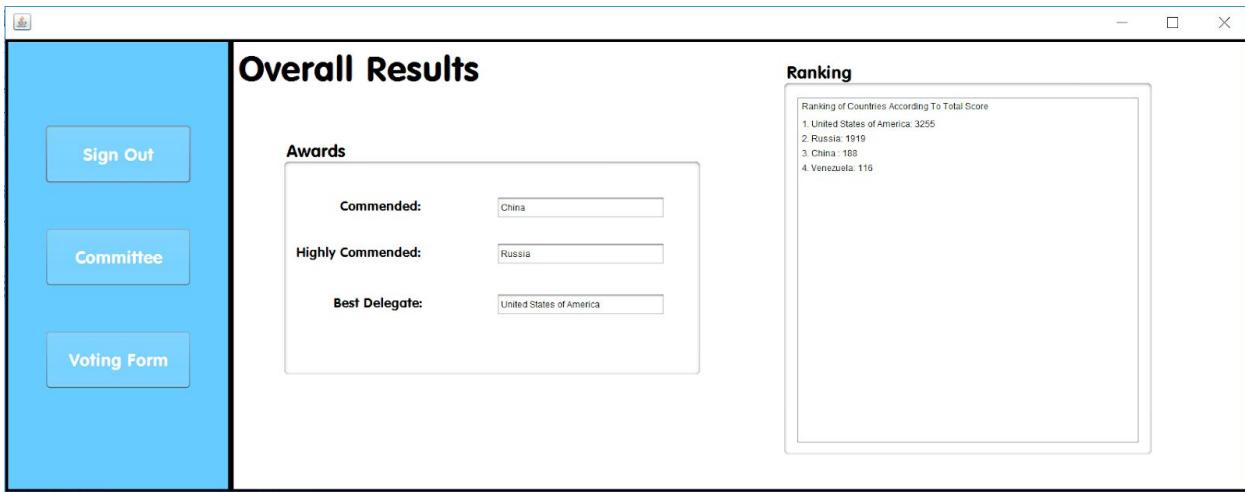
```
34 public AwardsCommittee() {
35     initComponents();
36
37     //connect to database
38     connectToDatabase Test = new connectToDatabase();
39     Test.connectToDatabase();
40     //update connection variable
41     connection = Test.getConnectionVariable(connection);
42
43 }
44
45 //use total
46 public void ranking() //this is what displays in the listbox needs to be seperate from main class section so committeeID can be set
47 {
48     String committeeName = "";
49
50     if(committeeID == 1){
51         committeeName = "Disarmerment";
52     }else if(committeeID == 2){
53         committeeName = "EcoSoc";
54     }else if(committeeID == 3){
55         committeeName = "Environment";
56     } else if (committeeID == 4) {
57         committeeName = "Security Council";
58     }
59     //set display for what committee it is
60     jLabel1.setText("Results for the " + committeeName +" Committee");
61
62     setDelegateArray(); //private since shouldn't be accessed by general
63
64     //create an object so this can access the sorting algorithm
65     Sorting sortObjects = new Sorting();
66     sortObjects.bubbleSortDelegates(delegates, count);
67     displayResults();
68     awards();
69 }
70 }
```

```
72 public void setCommitteeID(int committee)
73 {
74     this.committeeID = committee;
75
76 }
77
78 public void setChairID(int chairID){
79     this.chairID = chairID;
80
81     //whether or not a button appears since only chairs should have access to button
82 }
```



```
468     public int chooseCommittee()
469     {
470         String[] choices = { "Disarmament", "Ecosoc", "Environment", "Security Council" };
471         String input = (String) JOptionPane.showInputDialog(null, "Choose which Committee", "Committee Awards you would like to review:", JOptionPane.QUESTION_MESSAGE, null,
472             choices[0]); // Initial choice
473
474         if (input == choices[0])
475             l
476             return 1;
477         else if (input == choices[1]){
478             return 2;
479         else if (input == choices[2]){
480             return 3;
481         else{
482             return 4;
483         }
484     }
485
486     /**
487      * @param args the command line arguments
488     */
489     public static void main(String args[]) {
490         /* Set the Nimbus look and feel */
491         LookAndFeel setting code (optional)
492         //
```

Awards Overall



This page is ran very similarly to the committee awards page although instead of creating an array of objects from the delegate class it creates an array of objects from the delegation class. All the other process are nearly identical. The button 'voting form' on the navigation panel is only visible to the users logged in as chairs. There is also no 'best junior delegation' award as there can't be a junior delegation, only junior delegates within a delegation.

The code of the following page is shown below:

```
1 import java.sql.Connection;
2 import java.sql.PreparedStatement;
3 import java.sql.ResultSet;
4 import java.sql.SQLException;
5 import javax.swing.DefaultListModel;
6 import javax.swing.JOptionPane;
7
8 /**
9 *
10 * @author j.mcclure
11 */
12 public class AwardsOverall extends javax.swing.JFrame {
13
14     //global variables
15     Connection connection;
16     ResultSet rs;
17     PreparedStatement ps;
18
19
20     int count;
21     int committeeID;
22     int chairID;
23     //array of object
24     Delegation[] delegations;
25
26     public AwardsOverall() {
27
28         initComponents();
29
30         //connect to database
31         connectToDatabase Test = new connectToDatabase();
32         Test.connectToDatabase();
33         //update connection variable
34         connection = Test.getConnectionVariable(connection);
35     }
36 }
```

```

37
38     public void ranking() //this is what displays in the listbox needs to be seperate from main class section so committeeID can be set
39     {
40         setDelegateArray(); //private since shouldn't be accessed by general
41
42         Sorting sortObjects = new Sorting();
43         sortObjects.bubbleSortDelegations(delegations, count);
44         displayResults();
45
46         awards();
47     }
48
49     public void setCommitteeID(int committee)
50     {
51         this.committeeID = committee;
52
53         if (committeeID == 0)
54         {
55             jButton_Voting.setVisible(false);
56         }else{
57             jButton_Voting.setVisible(true);
58         }
59     }
60
61     public void setChairID(int chairID){
62         this.chairID = chairID;
63     }
64
65     private void setDelegateArray()
66     {
67         //get number of countries as well
68         NumberCountries numCount = new NumberCountries();
69         count = numCount.NumberCountries(1);
70
71         //local variables
72         String countryName;
73         int countryID;
74         int totalScore;

```

```

74
75         //initialise delegate object size.
76         delegations = new Delegation[count];
77
78         // create array placeholder variable
79         int i = 0;
80
81         try{
82             //alias used since a union needed to be made
83             String valuesNeeded = "SELECT c.countryName, d.'countryID', SUM(d.'totalScore') AS countryTotalScore FROM `delegate` AS d, `country` AS c WHERE d.'countryID' = c.'count
84             ps = connection.prepareStatement(valuesNeeded);
85             rs = ps.executeQuery();
86
87             while (rs.next()) { //conditional loop
88                 countryID = rs.getInt("countryID");
89                 countryName = rs.getString("countryName");
90                 totalScore = rs.getInt("countryTotalScore");
91
92                 delegations[i] = new Delegation();
93                 delegations[i].setCountryName(countryName);
94                 delegations[i].setTotalScore(totalScore);
95                 i = i + 1; //running total
96             }
97         }catch (SQLException e) {
98             JOptionPane.showMessageDialog(null, e);
99         }

```

```

100
101
102     public void displayResults()//add results to listbox
103     {
104
105         DefaultListModel listModel = new DefaultListModel();
106
107         //to neaten and clean display by giving a title and space

```

```
401 public int chooseCommittee()
402 {
403     String[] choices = { "Disarmament", "Ecosoc", "Environment", "Security Council" };
404     String input = (String) JOptionPane.showInputDialog(null, "Choose which Committee", "Committee Awards you would like to review:", JOptionPane.QUESTION_MESSAGE, null,
405             choices[0]); // Initial choice
406
407     if (input == choices[0])
408     {
409         return 1;
410     } else if (input == choices[1])
411     {
412         return 2;
413     } else if (input == choices[2])
414     {
415         return 3;
416     } else
417     {
418         return 4;
419     }
420 }
421 /**
422 * @param args the command line arguments
423 */
424 public static void main(String args[])
425 {
426     /* Set the Nimbus look and feel */
427     /* Look and feel setting code (optional)
428     //
```

Delegate Class

This class was created to store the values of delegates: Country Name, age and total score from all the vote records. The class is comprised of accessors and mutators as most of the variables are private variables. In this class the creation of a totalScore object is created so that the total score can be calculated by accessing a method within the 'Total Score' class. There is a method within the class called compareTo(). This is a comparative method which is used when sorting the array of delegate objects.

The code of the following page is shown below:

```
1 import java.util.Comparator;
2 /**
3  * 
4  * @author j.mcclure
5 */
6 public class Delegate implements Comparable<Delegate>{
7     //from country table
8     private String countryName;
9     //from delegate table
10    private int age;
11    int totalScore; //public
12
13    //constructor
14    public Delegate() {
15
16    }
17
18    //mutators
19    public void setTotalScore(int delegateID) //public access
20    {
21        TotalScore score = new TotalScore();
22        score.calculatingTotalScore(delegateID);
23        totalScore = score.getTotalScore();
24    }
25
26    public void setCountryName(String name)
27    {
28        countryName = name;
29    }
30
31    public void setAge(int age)
32    {
33        this.age = age;
34    }

```

```

36     //accessors
37     public String getcountryName()
38     {
39         return countryName;
40     }
41     public int getAge()
42     {
43         return age;
44     }
45     public int getTotalScore()
46     {
47         return totalScore;
48     }
49
50     //display object
51     public String displayMessage()
52     {
53         return ""+ countryName + ": " + totalScore + "";
54     }
55
56
57     //compareTo function so as to sort the delegate array to get the total score
58     public int compareTo(Delegate compareDelegate)
59     {
60         int compareQuantity = ((Delegate) compareDelegate).getTotalScore(); //comparing based on totalScore
61
62         //return in descending order
63         return compareQuantity - this.totalScore;
64     }
65
66 }
67

```

I had to create a delegate wrapper which is another class which is used during the sorting process to switch the whole object in the array with another object. Without the use of this wrapper class only the total score value would swap and therefore the wrong total score would be assigned to the wrong country.

The code of the following page is shown below:

```
11 public class DelegateWrapper {  
12  
13     Delegate d;  
14  
15     //constructor  
16     DelegateWrapper(Delegate d)  
17     {  
18         this.d = d;  
19     }  
20 }  
21 }
```

Delegation Class

This class was created to store the values of an overall delegation (a country). The values of the the Country ID, Country Name and the overall total score is held in an object of this class. Like the delegate class an array of objects will be created from this class to hold all the information of the delegations for the ranking process. The class contains mutators and accessors and has a display method which is what is displayed in the ranking listbox in the overall awards page. Another method is the compareTo() which is used to sort the delegations by comparing the total score of the delegations.

The code of the following page is shown below:

```
11 public class Delegation implements Comparable<Delegation> {
12
13     private String countryName;
14     private int countryID;
15     int totalScore; //public
16
17     //constructor
18     public Delegation(){
19
20     }
21
22     //mutators
23     public void setTotalScore(int totalScore) //public access
24     {
25         this.totalScore = totalScore;
26     }
27
28     public void setCountryName(String name)
29     {
30         countryName = name;
31     }
32
33     public void setCountryID(int countryID)
34     {
35         this.countryID = countryID;
36     }
37
38     //accessors
39     public String getcountryName()
40     {
41         return countryName;
42     }
```

```
44     public int getTotalScore()
45     {
46         return totalScore;
47     }
48
49     //display object
50     public String displayMessage()
51     {
52         return "+" + countryName + ": " + totalScore + "";
53     }
54
55
56     //compareTo function so as to sort the delegate array to get the total score
57     public int compareTo(Delegation compareDelegation)
58     {
59         int compareQuantity = ((Delegation) compareDelegation).getTotalScore(); //comparing based on totalScore
60
61         //return in descending order
62         return compareQuantity - this.totalScore;
63     }
64
65 }
```

I had to create a delegation wrapper which is used for the same purpose as the delegate wrapper. This is used to make sure that the whole object is swapped during the sorting process.

The code of the following page is shown below:

```
11     public class DelegationWrapper {
12
13         Delegation d;
14
15         //constructor
16         DelegationWrapper(Delegation d)
17         {
18             this.d = d;
19         }
20
21     }
```

Sorting Algorithm

This algorithm was used to rank all the delegates/delegations in order from highest to lowest based on their total score. A bubble sort was used to complete the sorting process. A bubble sort works by comparing adjacent pairs of objects in the array and swaps those objects if the conditions specified are right. I had to create two 'sets' of this algorithm as one was used when comparing the Delegate array while the other was used when comparing the Delegation array. This was needed as the array variable type needed to be typed into the parameter. I tried many different ways to get this method to work by only having one sorting method however due to time constraints I wasn't able to achieve this.

The code of the following page is shown below:

```

11 public class Sorting {
12
13
14     public void bubbleSortDelegates(Delegate[] delegates, int length)
15         //sorting delegates by total Score
16     {
17         //bubble sort from highest to lowest
18
19         for(int outerloop = length ; outerloop > 0; outerloop --)
20         {
21             for(int counter = 0 ; counter < (outerloop - 1) ; counter++)
22             {
23                 if( delegates[counter].compareTo(delegates[counter + 1]) > 0)
24                 {
25                     //create wrapper objects of the objects in the array so that all the information in the delegaes object is swapped
26                     DelegateWrapper wrapper;
27                     wrapper = new DelegateWrapper(delegates[counter]);
28                     DelegateWrapper wrapper2 = new DelegateWrapper(delegates[counter + 1]);
29                     //swapping function
30                     swap(wrapper , wrapper2);
31
32                     delegates[counter] = wrapper.d ;
33                     delegates[counter + 1] = wrapper2.d ;
34
35                 }
36             }
37         }
38     }
39
40
41     public void swap(DelegateWrapper dw, DelegateWrapper dw2)
42     {
43         Delegate temp = dw.d ; //temporarily holds the delegate values from the first parameter (a)
44         dw.d = dw2.d;
45         dw2.d = temp;
46     }
47
48
49     public void bubbleSortDelegations(Delegation[] delegations, int length)
50         //sorting delegates by total Score
51     {
52         //bubble sort from highest to lowest
53
54         for(int outerloop = length ; outerloop > 0; outerloop --)
55         {
56             for(int counter = 0 ; counter < (outerloop - 1) ; counter++)
57             {
58                 if( delegations[counter].compareTo(delegations[counter + 1]) > 0)
59                 {
60                     //create wrapper objects of the objects in the array so that all the information in the delegaes object is swapped
61                     DelegationWrapper wrapper;
62                     wrapper = new DelegationWrapper(delegations[counter]);
63                     DelegationWrapper wrapper2 = new DelegationWrapper(delegations[counter + 1]);
64                     //swapping function
65                     swap2(wrapper , wrapper2);
66
67                     delegations[counter] = wrapper.d ;
68                     delegations[counter + 1] = wrapper2.d ;
69
70                 }
71             }
72         }
73     }
74
75
76     public void swap2(DelegationWrapper dw, DelegationWrapper dw2)
77     {
78         Delegation temp = dw.d ; //temporarily holds the delegate values from the first parameter (a)
79         dw.d = dw2.d;
80         dw2.d = temp;
81     }
82 }
```

Number of Countries

During implementation I noticed that I kept counting the number of countries in the database and so instead of having that repetition I created a whole different class so any page could access the function which reads the number of delegates in a particular committee. This works by creating a query which counts the number of delegates from the particular committee. The query is then executed and the results are read from and stored as an integer which is returned by the function.

The code of the following page is shown below:

```
1 import java.sql.Connection;
2 import java.sql.PreparedStatement;
3 import java.sql.ResultSet;
4 import java.sql.SQLException;
5 import javax.swing.JOptionPane;
6
7 /**
8 *
9 * @author j.mcclure
10 */
11 public class NumberCountries {
12     //global variables for connection of database
13     Connection connection;
14     ResultSet rs;
15     PreparedStatement ps;
16
17     public int NumberCountries(int committeeID)
18     {
19         //connect to database
20         connectToDatabase Test = new connectToDatabase();
21         Test.connectToDatabase();
22         //update connection variable
23         connection = Test.getConnectionVariable(connection);
24
25         int numberOfCountries = 0;
26
27         try{
28
29             String count = "SELECT COUNT(`delegateID`) AS count FROM `delegate` WHERE `committeeID` = " + committeeID+ "";
30             ps = connection.prepareStatement(count);
31             rs = ps.executeQuery();
32
33             while (rs.next()) {
34                 numberOfCountries = rs.getInt("count");
35             }
36         }catch (SQLException e) {
37             JOptionPane.showMessageDialog(null, e);
38
39         }
40
41         return numberOfCountries;
42     }
43 }
```

Total Score

This class was created to calculate the total score of a delegate. This is done by creating a query and reading all the appropriate values (quality of speech, humour etc) from the database ‘vote’ table in the database for a particular delegate. All these values are summed together in the query and the outputs are read and stored in corresponding variables. There is another query which sums together all the times from the ‘timer’ table for a particular delegate. The result from this query is also stored in a corresponding variable.

All these values are added together (some like quality of speech holding greater weight than others) This value is called the total score which is then used in the delegate class and set as the total score of an object.

This however isn’t the end of the class as the delegate’s record (identified by the delegateID) is updated in the ‘delegates’ table by inserting the calculated total score.

It should be noted that this process doesn’t happen when finding the total score of a delegation due to the fact that the total score value for every individual delegate is now held in the database.

The code of the following page is shown below:

```
1 import java.sql.Connection;
2 import java.sql.PreparedStatement;
3 import java.sql.ResultSet;
4 import java.sql.SQLException;
5 import java.sql.Statement;
6 import javax.swing.JOptionPane;
7
8 /*
9 * To change this license header, choose License Headers in Project Properties.
10 * To change this template file, choose Tools | Templates
11 * and open the template in the editor.
12 */
13
14 /**
15 *
16 * @author j.mcclure
17 */
18 public class TotalScore {
19     Connection connection;
20     Statement state; //insert
21
22     ResultSet rs_votingForm = null; //select
23     PreparedStatement ps_votingForm = null; //select
24
25     ResultSet rs_timer = null; //select
26     PreparedStatement ps_timer = null; //select
27
28     ResultSet rs_humour = null; //select
29     PreparedStatement ps_humour = null; //select
30
31     int totalScore;
32     int humour;
33
34     public TotalScore()
35     {
36         //connect to database
```

```

37 |     connectToDatabase Test = new connectToDatabase();
38 |     Test.connectToDatabase();
39 |     //update connection variable
40 |     connection = Test.getConnectionVariable(connection);
41 |
42 |
43 |     public void calculatingTotalScore(int delegateID)
44 |     {
45 |         //querey statements held in variables
46 |         //String querey_votingForm = "SELECT `qualityScore`, `humourScore`, `abilityPOIScore`, `instagateDebate` FROM `voting` WHERE `delegateID` = "+delegateID+"";
47 |         //String querey_timer = "SELECT `delegateID`, SUM(`timeSpoken`) As totalTime FROM `timer` WHERE `delegateID` = "+delegateID+" GROUP BY `delegateID`";
48 |
49 |         String query_totalScoreValues = "SELECT voting.`delegateID`, SUM(`qualityScore`) AS qs, SUM(`humourScore`) AS humour, SUM(`abilityPOIScore`) AS POIScore, SUM(`instagateDebate`)";
50 |         //local variables
51 |         int qualityScore = 0, humourScore = 0, abilityPOIScore = 0, instagateDebate = 0, timer = 0;
52 |
53 |         //retrieving data from database
54 |         try{
55 |             ps_votingForm = connection.prepareStatement(query_totalScoreValues);
56 |             rs_votingForm = ps_votingForm.executeQuery();
57 |
58 |             //ps_timer = connection.prepareStatement(querey_timer);
59 |             //rs_timer = ps_timer.executeQuery();
60 |
61 |             while (rs_votingForm.next()) {
62 |                 //running total used so no group by statement required in SQL statement
63 |                 qualityScore = rs_votingForm.getInt("qs");
64 |                 humourScore = rs_votingForm.getInt("humour");
65 |                 abilityPOIScore = rs_votingForm.getInt("POIScore");
66 |                 instagateDebate = rs_votingForm.getInt("instagateDebate");
67 |                 timer = rs_votingForm.getInt("time");
68 |             }
69 |
70 |         }catch(SQLException e) {
71 |             JOptionPane.showMessageDialog(null, e);
72 |         }
73 |
74 |         totalScore = (2*qualityScore) + humourScore + (2*abilityPOIScore) + instagateDebate + (timer) ; //the quality and ablitiyPOI scores are
75 |         //multiplied since they are more desirable qualities and therefore should hold more weight in the total score calcuated
76 |
77 |         //update totalScore into the database delegate table
78 |         try{
79 |             Statement state=(Statement)connection.createStatement();
80 |             String sql_totalScore = "UPDATE `delegate` SET `totalScore`=" + totalScore +" WHERE `delegateID`="+ delegateID+"";
81 |             state.executeUpdate(sql_totalScore);
82 |
83 |         }catch(SQLException e) {
84 |             JOptionPane.showMessageDialog(null, e);
85 |         }
86 |
87 |     }
88 |
89 |     //getter since totalScore variable is private
90 |     public int getTotalScore()
91 |     {
92 |         return this.totalScore;
93 |     }
94 |

```

Testing

Alpha

Throughout the implementation of my program I was continually testing each component after I finished the section. Most of my testing was done using breakpoints and manually looking and thinking out what the code should do. I didn't make frequent use of stubs and drivers as I thought this would be an ineffective use of my time as by the time I created the stubs and drivers it would have just been easier to run the code normally. It would have taken a while to create these stubs and drivers because a lot of the values used at the initial login was essential throughout the rest of my program. A lot of my program was continuous and relied on me to finish certain aspects before I could move on to other ones as well.

Home Page

Component Testing

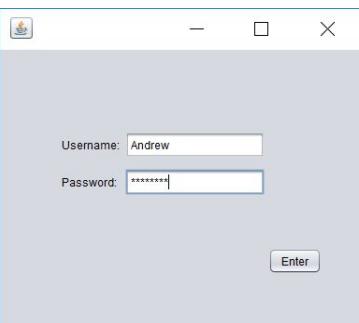
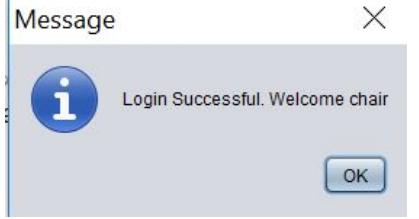
1. Login Button Performs how expected and takes you to the login page.
2. Registration Delegation Button performs how expected and takes you to the Registration Delegation page.
3. Registration Chair Button performs how expected and takes you to the Registration Chair page.
4. Delete Profile Button performs how expected and takes you to the Delete Profile page.

Login Page

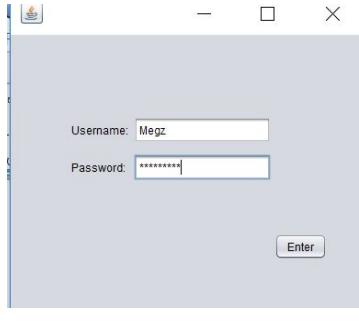
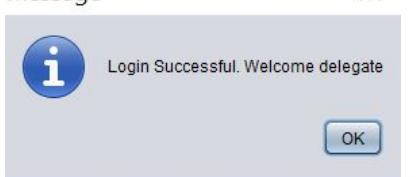
Component Testing

During testing of my login I wanted to look at all the different variations of a username and Password I could enter and see the expected vs actual result. This is component testing on the login section.

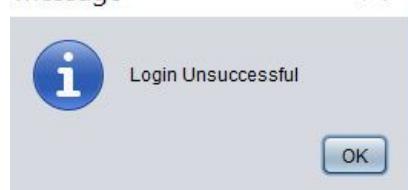
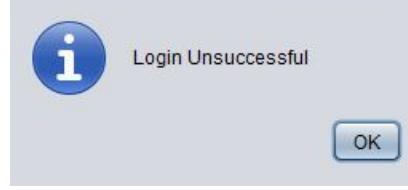
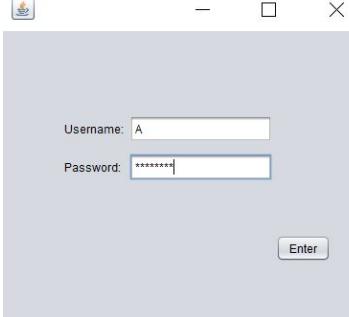
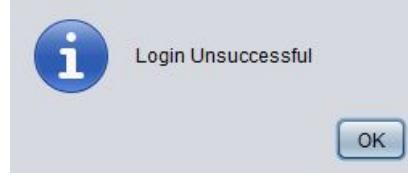
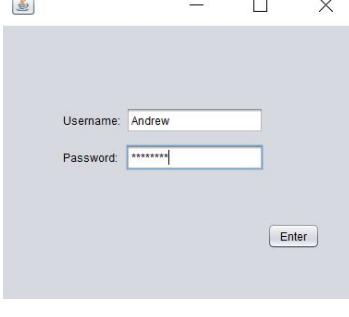
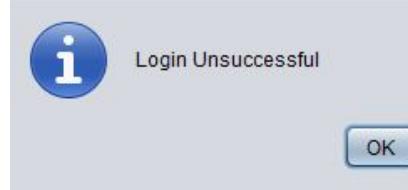
1. If username and password of Chair is entered correctly then the user should be prompted with a message saying 'Welcome Chair'

Input Screenshot	Username Password	Expected Result	Output Screenshot
	Andrew AndyPass	Login Successful message	

2. If username and password of Delegation is entered correctly then the user should be prompted with a message saying 'Welcome delegate'

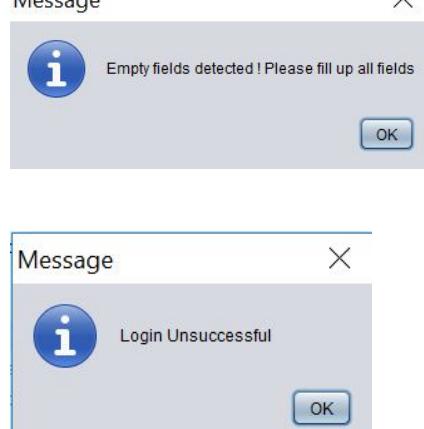
Input Screenshot	Username Password	Expected Result	Output Screenshot
	Megz Password1	Login Successful	

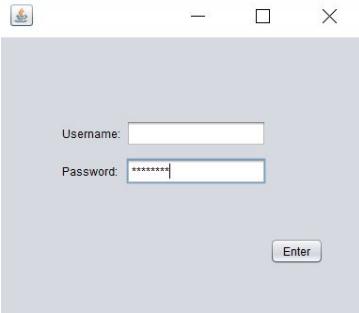
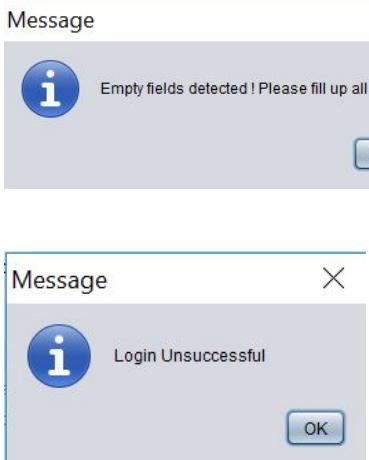
3. If username and password do not match then an error message asking to re-input username and password will be displayed.

Input Screenshot	Username Password	Expected Result	Output Screenshot
 A screenshot of a Java Swing application window titled "Java Login". It contains two text fields: "Username: WrongUsername" and "Password: ****". A blue "Enter" button is at the bottom right.	WrongUsername Pass	Neither the username or password are values within the database and so login should be unsuccessful	 A modal dialog box titled "Message" with a blue info icon. The text "Login Unsuccessful" is displayed, and an "OK" button is at the bottom right.
 A screenshot of the same Java Swing application window. The "Username" field now contains "Andrew" and the "Password" field contains "****". The "Enter" button is at the bottom right.	Andrew Pass	The username is an actual username however the password doesn't match and so the login should be unsuccessful.	 A modal dialog box titled "Message" with a blue info icon. The text "Login Unsuccessful" is displayed, and an "OK" button is at the bottom right.
 A screenshot of the Java Swing application window. The "Username" field contains "A" and the "Password" field contains "*****". The "Enter" button is at the bottom right.	A AndyPass	The password is an actual password however the username doesn't match and so the login should be unsuccessful.	 A modal dialog box titled "Message" with a blue info icon. The text "Login Unsuccessful" is displayed, and an "OK" button is at the bottom right.
 A screenshot of the Java Swing application window. The "Username" field contains "Andrew" and the "Password" field contains "*****". The "Enter" button is at the bottom right.	Andrew PassWord	The username and password are both actual values in the database however they don't belong to the same instance therefore the login	 A modal dialog box titled "Message" with a blue info icon. The text "Login Unsuccessful" is displayed, and an "OK" button is at the bottom right.

		should be unsuccessful.	
--	--	----------------------------	--

4. If one or both of the fields (username and password) are left empty but the enter button is pressed then the user will be prompted to make sure to fill in all the fields.

Input Screenshot	Username Password	Expected Result	Output Screenshot
	Blank Blank	Error Message followed by a login unsuccessful message	
	Andrew Blank	Error Message followed by a login unsuccessful message	

	<p>Blank PassWord</p>	<p>Error Message followed by a login unsuccessful message</p>	
---	------------------------------	---	---

When login was successful the user was always brought successfully to the appropriate page depending on their user type (chair or delegation)

This type of testing for the login was successfully completed.

Comprehensive Testing

The main concerns during this testing was getting the connection to the database established efficiently first. Only after the connection was established was I able to run the code and check to see if the username and password was present in the database.

During Implementation I kept getting unsuccessful login and I thought it was because of my query and that the program wasn't comparing the right variables. It decided to make use of a 'try and catch' statement in the connection section to my database so that if the catch was triggered then I would know exactly what was wrong.

```
try{
    if(connection == null)
    {
        connection.close();
    }
} catch(SQLException ex){
    System.out.println(ex.getMessage());
}
```

I then started getting the message this message printed to me when I tried to run the program and so I realised that my program wasn't connecting to mySQL database. I then looked at various different people's version of connecting a mySQL database in Java Netbeans and edited it to suit my own program. At first I realised that I didn't ever specify which database my program was to connect to on the localhost. After I realised this I added the '/modelunitednations' to this section of code below which specified the path to the database. This would have been an execution error as the program physically stopped and crashed my application.

```
String url = "jdbc:mysql://localhost:3306/modelunitednations";
```

After I stopped getting the error message that the connection to my database failed I realised my results still weren't working. A common problem I normally have during coding is spelling and so I checked the spelling of my variable and table names. It was then that I realised in the query I created I was calling the non-existent table 'chairs' instead of 'chair'. This was a syntax error due to my poor spelling.

```
String sql_chair = "SELECT `username`, `password` FROM `chair` WHERE `username` = ? AND `password` = ?";
```

I wanted to make my program more efficient and so decided to create a class for the connection to a database so that I was able to call upon this class in other pages. Since I was used to coding in Visual Basics I forgot that I had to actually create an object out of that class and not just call upon a method.

```
170 // connect to database  
120 connectToDatabase.connectToDatabase();
```

I then realised that an object of the class would have to be created and after I solved this problem my program was successfully connecting to the database. This mistake was a logical error.

Another error I encountered was that when I was comparing the first password the user entered to the other password I used this statement (shown below):

```
if(inputPassword == inputPassword2)
```

However this wasn't given the expected results. When I set the password equal to the same I would still get the message that the passwords aren't equal. To make sure that I was inputting the same correct passwords I even used breakpoints to check the variable values of the first and second password:

```
170 //  
120     String inputUsername = jTextField_username.getText();  
120     String inputPassword = String.valueOf(jPasswordField_1.getPassword());  
120     String inputPassword2 = String.valueOf(jPasswordField_Again.getPassword());
```

I then decided to change my code to use the equals predefined function

```
if(inputPassword.equals(inputPassword2))
```

This then gave me my expected results. This error was a syntax error.

Deletion Of Profile

Component Testing

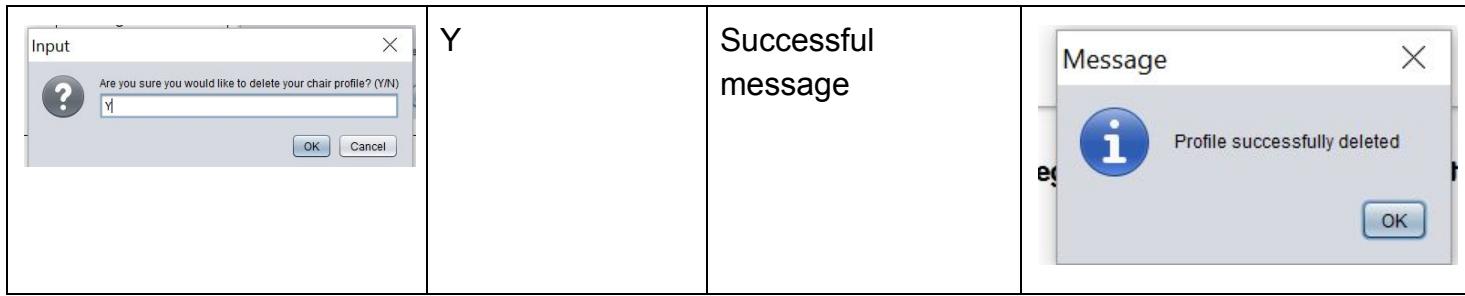
Chair Deletion

Component testing initially to see if the entered username and password gets accepted or not whether it is wrong or right.

Input Screenshot	Username Password	Expected Result	Output Screenshot
	Andrew AndyPASD	Error Message	
	Andrew AndyPass	Double Check Message Asking if I want to delete specific to chair.	

Testing to see if the login is right whether the input box of checking the deletion of the profile is correct and behaves as expected.

Input Screenshot	Answer To Choice Question	Expected Result	Output Screenshot
	N	Not successful message specific to the chair profile	



I tried to login after the deletion of the chair profile (ChairID 2) and this attempt was unsuccessful. I also viewed the voting table containing the votes made by the chair,

Voting Table Before deletion of Chair ID 2:

+ Options	delegateID	committeelID	chairID	qualityScore	humourScore	abilityPOIScore	instagateDebate
	6	1	2	5	5	2	0
	2	1	2	3	3	3	0
	6	1	2	5	3	4	2
	2	1	2	3	5	1	2
	6	1	2	2	1	3	0
	4	2	3	3	3	3	2
	4	2	3	3	3	3	0
	8	2	3	1	5	1	0
	8	2	3	1	5	1	0
	8	2	3	1	5	1	0
	8	2	3	1	1	2	2
	4	2	3	5	3	2	2
	4	2	3	5	3	2	0
	4	2	3	5	5	5	0
	8	2	3	5	3	2	2
	7	3	4	3	3	3	0

Voting Table After deletion of Chair ID 2:

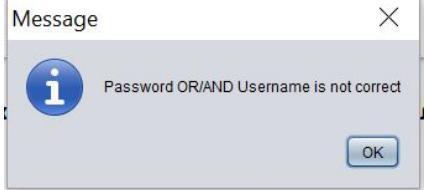
+ Options

delegateID	committeelID	chairID	▲ 1	qualityScore	humourScore	abilityPOIScore	instagateDebate
4	2	3		3	3	3	2
8	2	3		1	5	1	0
8	2	3		1	5	1	0
8	2	3		1	5	1	0
8	2	3		1	1	2	2
8	2	3		5	3	2	2
4	2	3		5	3	2	2
4	2	3		5	3	2	0
4	2	3		5	5	5	0
4	2	3		3	3	3	0
7	3	4		3	3	3	0

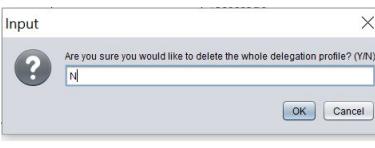
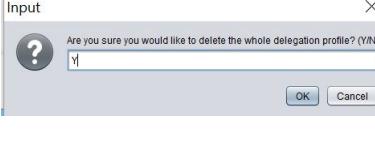
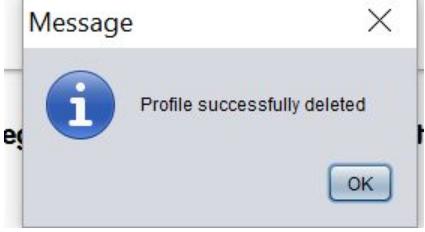
It can be seen that all the records containing the ChairID of 2 was also deleted. This was also expected and therefore the deletion of a chair works as expected.

Delegation Deletion

Component testing initially to see if the entered username and password gets accepted or not whether it is wrong or right.

Input Screenshot	Username Password	Expected Result	Output Screenshot
	Tullis Brassband8	Error Message	
	Tullis BrassBand8	Double Check Message Asking if I want to delete specific to delegation.	

Testing to see if the login is right whether the input box of checking the deletion of the profile is correct and behaves as expected.

Input Screenshot	Answer To Choice Question	Expected Result	Output Screenshot
	N	Not successful message specific to the delegation profile	
	Y	Successful message	

I tried to login after the deletion of the chair profile (countryID 21) and this attempt was unsuccessful. I also viewed the country table containing the records of the country,

Country Table Before deletion of Country ID 21:

+ Options						
	← T →	countryID	countryName	username	password	securityCouncil
<input type="checkbox"/>	Edit Copy Delete	19	United States of America	Megz	Password1	1
<input type="checkbox"/>	Edit Copy Delete	20	Russia	RGUMun	PasswordMun	1
<input type="checkbox"/>	Edit Copy Delete	21	Germeny	Tullis	BrassBand8	1

Country Table After deletion of Country ID 21:

+ Options						
	← T →	countryID	countryName	username	password	securityCouncil
<input type="checkbox"/>	Edit Copy Delete	19	United States of America	Megz	Password1	1
<input type="checkbox"/>	Edit Copy Delete	20	Russia	RGUMun	PasswordMun	1

There were also no delegates with the country ID of 21 any more in the delegate table. Since this all behaved as expected the deletion of a delegation profile was successful as it behaved as expected.

Back button works as expected as it brings the user back to the home page.

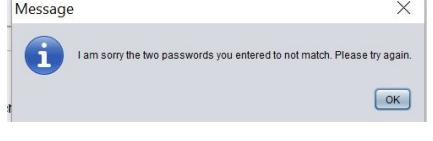
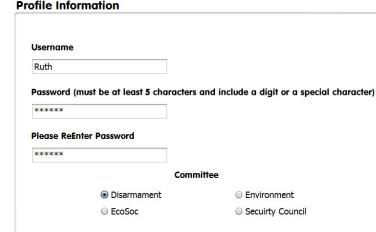
Comprehensive Testing

I had no errors when I was testing this code due to the fact that it was a similar process to the login page.

Registration Chair

Component Testing

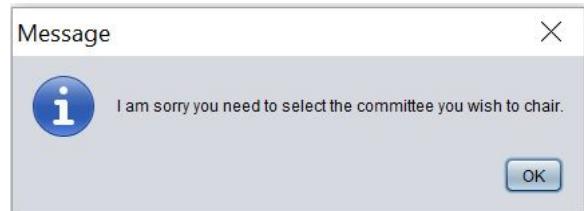
Component testing initially to see the validation on the fields and if they are correct and work as expected.

Input Screenshot	Username Password Re-Password Committee	Expected Result	Output Screenshot
	Danny Password1 Password1 Disarmament	Error message as username already exists.	
	Ruth gurney Disarmament	Error message to re-enter passwords since the two passwords don't match	
<p>There is a spelling error in the output screenshot, this has now been corrected. This wasn't an error but instead a design flaw.</p>			
	Ruth Gurney Gurney Disarmament	Error message since no number or special character is entered in username	

Profile Information Username: Ruth Password (must be at least 5 characters and include a digit or a special character): ***** Please ReEnter Password: ***** Committee: <input type="radio"/> Disarmament <input type="radio"/> Environment <input checked="" type="radio"/> EcoSoc <input type="radio"/> Security Council	Ruth Gurney1 Gurney1 None selected	Error message since committee needs to be selected	
---	--	--	---

This result was unexpected, I need to add another if statement to my clause so that if this happens again the chair isn't registered. This is an logical error as the program still worked but just didn't give me my expected result. After I make this change to my program I will delete this record so that the username of 'Ruth' is still unique.

New output:



therefore behaves as expected.

Profile Information Username: Ruth Password (must be at least 5 characters and include a digit or a special character): ***** Please ReEnter Password: ***** Committee: <input checked="" type="radio"/> Disarmament <input type="radio"/> Environment <input type="radio"/> EcoSoc <input type="radio"/> Security Council	Ruth Gurney1 Gurney1 Disarmament	Successful chair registered message	
---	---	-------------------------------------	---

Testing to see whether the form results were updated into the database

Chair table before registration:

+ Options					
	← T →	▼	chairID	username	password
	Edit	Copy	Delete		committeelID
			3	Chair2	PassWord
			4	Bella	HelloWorld5
			10	Danny	LovesDogs!

Chair table after registration:

+ Options					
	← T →	▼	chairID	username	password
	Edit	Copy	Delete		committeelID
			12	Ruth	Gurney1
			3	Chair2	PassWord
			4	Bella	HelloWorld5
			10	Danny	LovesDogs!

Since this now behaves as expected the registration of chairs is successful and works how it should.

Back button works as expected as it brings the user back to the home page.

Comprehensive Testing

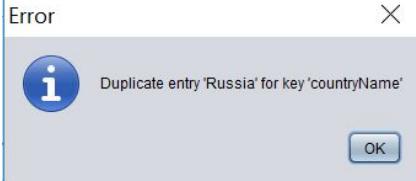
Everything worked well during implementation and there were no problems.

Registration Delegation

Component Testing

Initially testing the validation on the fields to see if they respond as expected.

Input Screenshot	Username Password Re-Password Committee Ages (disarmament, environment, ecosoc, security council) Security Council (Y/N) Country	Expected Result	Output Screenshot
	Megz Password! Password! (17,16, 15, 16) N Russia	Error Message as username isn't unique	
	DundeeHigh Pass Pass (17,16,15,16) N Russia	Error Message password doesn't include special digit	
	DundeeHigh Pas! Pas! (17,16,15,16) N Russia	Error Message password doesn't meet required length	

	<p>DundeeHigh Pass! Pass1 (17,16,15,16) N Russia</p>	<p>Error Message passwords don't match</p>	
	<p>DundeeHigh Pass! Pass1 (17,16,15,16) N Russia</p>	<p>Error Message already a 'Russia' delegation</p>	
	<p>DundeeHigh Pass! Pass1 (17,16,15,16) N Venezuela</p>	<p>Successful Message. The delegation has been registered</p>	

Back button works as expected as it brings the user back to the home page. The page performs as expected and therefore is successfully tested.

Comprehensive Testing

During the implementation I was trying to validate the countries selected by the delegation when registering. This meant that only one delegation could be a country.

```
.68 |     //make sure country is unique
.69 |     String sql_country = "SELECT `countryName` FROM `country` WHERE `username` = " +country +"";
.70 |     try {
.71 |         //creating chair query
.72 |         ps = connection.prepareStatement(sql_country);
.73 |
.74 |         rs_country = ps.executeQuery();
.75 |         int rowCount = 0;
.76 |         while ( rs_country.next() )
.77 |         {
.78 |             // Process the row.
.79 |             rowCount++;
.80 |         }
.81 |         System.out.println("There were " + rowCount + " records.");
.82 |
.83 |
.84 |     } catch (SQLException e) {
.85 |         JOptionPane.showMessageDialog(null, e);
.86 |     }
.87 | }
```

This line of code wasn't working as expected so through the use of breakpoints I noticed noticing that the code is running to the catch section and not completing the try. I later realised that my code kept jumping after the executeQuery() method. I realised that since the chosen country was defined as a unique key in my database. I didn't need to have an error message in my code since the SQL server will create an error message automatically that the countryName isn't unique. The same can be said for the username.

Voting Page

Component Testing

Testing that the input's for the voting form get correctly inputted into the database as expected.

Input:

Voting Form

Country

Country of Delegate: United States of America

Categories

Quality of Speech: 4

Humour in Speech: 1

Ability to Handle Points of Information: 2

Added New Information to Debate: 1

Submit Vote

Added New Information to Debate

Expected Results:

Logged in as ChairID 10, CommitteelD 4, CountryID 19

Quality of Speech: 4

Humour in Speech: 1

Ability to Handle Points: 2

Added New INformation: 1

Output:



delegateID	committeelD	chairID	qualityScore	humourScore	abilityPOIScore	instagateDebate
4	2	3	3	3	3	2
4	2	3	3	3	3	0
8	2	3	1	5	1	0
8	2	3	1	5	1	0
8	2	3	1	5	1	0
8	2	3	1	1	2	2
8	2	3	5	3	2	2
4	2	3	5	3	2	2
4	2	3	5	3	2	0
4	2	3	5	5	5	0
7	3	4	3	3	3	0
5	4	10	4	1	2	2

Since seen in voting table this means the vote has been processed.

Since logged in as a chair who is chairing for security council this means that not all the countries should be viewed in the dropdown country listbox.

countryID	countryName	username	password	securityCouncil
19	United States of America	Megz	Password1	1
20	Russia	RGUMun	PasswordMun	1
24	Venezuela	DundeeHigh	Pass!	0

Since only country ID's 19 and 20 are security council Chair ID 10 should only include these countries and not Venezuela:

Country

Country of Delegate

United States of America
United States of America
Russia

Chair ID 12 isn't a security council chair and therefore country list should include Venezuela:

Country

Country of Delegate

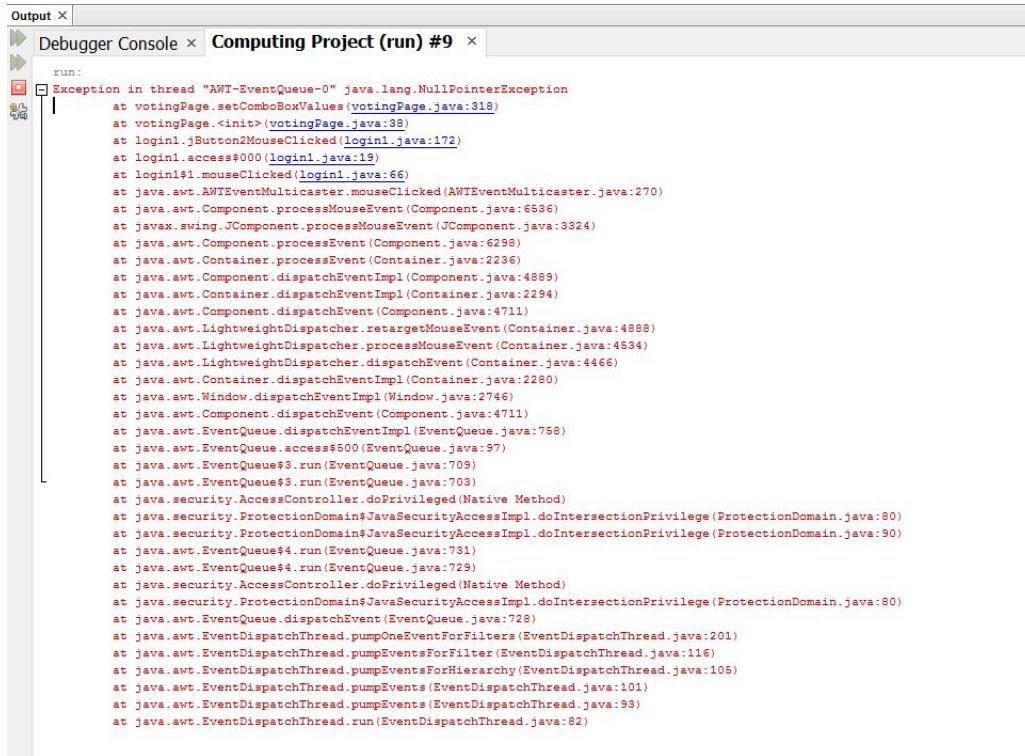
United States of America
United States of America
Russia
Venezuela

Sign Out, Awards Page and Timer buttons all work as expected.

Voting Page performs as expected.

Comprehensive Testing

During my implementation I noticed that when going back to my voting page every time I tried to run the page I would get this error.



```
run:
Exception in thread "AWT-EventQueue-0" java.lang.NullPointerException
    at votingPage.setComboBoxValues(votingPage.java:318)
    at votingPage.<init>(votingPage.java:38)
    at login1.Button2MouseClicked(login1.java:172)
    at login1.access$000(login1.java:19)
    at login1$1.mouseClicked(login1.java:66)
    at java.awt.AWTEventMulticaster.mouseClicked(AWTEventMulticaster.java:270)
    at java.awt.Component.processMouseEvent(Component.java:6536)
    at javax.swing.JComponent.processMouseEvent(JComponent.java:3324)
    at java.awt.Component.dispatchEvent(Component.java:6298)
    at java.awt.Container.dispatchEvent(Container.java:2236)
    at java.awt.Component.dispatchEventImpl(Component.java:4889)
    at java.awt.Container.dispatchEventImpl(Container.java:2294)
    at java.awt.Component.dispatchEvent(Component.java:4711)
    at java.awt.LightweightDispatcher.retargetMouseEvent(Container.java:4888)
    at java.awt.LightweightDispatcher.processMouseEvent(Container.java:4534)
    at java.awt.LightweightDispatcher.dispatchEvent(Container.java:4466)
    at java.awt.Container.dispatchEventImpl(Container.java:2280)
    at java.awt.Window.dispatchEventImpl(Window.java:2746)
    at java.awt.Component.dispatchEvent(Component.java:4711)
    at java.awt.EventQueue.dispatchEvent(EventQueue.java:758)
    at java.awt.EventQueue.access$500(EventQueue.java:97)
    at java.awt.EventQueue$3.run(EventQueue.java:709)
    at java.awt.EventQueue$3.run(EventQueue.java:703)
    at java.security.AccessController.doPrivileged(Native Method)
    at java.security.ProtectionDomain$JavaSecurityAccessImpl.doIntersectionPrivilege(ProtectionDomain.java:80)
    at java.security.ProtectionDomain$JavaSecurityAccessImpl.doIntersectionPrivilege(ProtectionDomain.java:90)
    at java.awt.EventQueue$4.run(EventQueue.java:731)
    at java.awt.EventQueue$4.run(EventQueue.java:729)
    at java.security.AccessController.doPrivileged(Native Method)
    at java.security.ProtectionDomain$JavaSecurityAccessImpl.doIntersectionPrivilege(ProtectionDomain.java:80)
    at java.awt.EventQueue.dispatchEvent(EventQueue.java:728)
    at java.awt.EventDispatchThread.pumpOneEventForFilters(EventDispatchThread.java:201)
    at java.awt.EventDispatchThread.pumpEventsForFilter(EventDispatchThread.java:116)
    at java.awt.EventDispatchThread.pumpEventsForHierarchy(EventDispatchThread.java:105)
    at java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:101)
    at java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:93)
    at java.awt.EventDispatchThread.run(EventDispatchThread.java:82)
```

I realised that while trying to change the country names automatically in the comboBox I instead created these many errors. I decided to isolate this process and have put it in its own separate testing class so I can solve the issue I had to make use of drivers to run the code since I didn't have the variables need. I also realised there was an issue with this specific part of the code because when I used breakpoints the SQLException was always set off during this part of the code:

```
try{
    //get list of countries so that comboBox can update automatically depending on the committee
    String sql_countryList = "SELECT `countryID` FROM `delegate` WHERE `committeeID` = "+ committeeID +"";
    ps = connection.prepareStatement(sql_countryList);
    rs = ps.executeQuery();

    while (rs.next()) {
        countryID = rs.getInt("countryID");
        //match countryID to countryName so that countryName is displayed since it is a more meaningful data value;
        String sql_countryName = "SELECT `countryName` FROM `country` WHERE `countryID` = "+ countryID +"";
        ps = connection.prepareStatement(sql_countryName);
        rs = ps.executeQuery();
        String countryNameAdd = rs.getString("countryName");
        System.out.println(countryNameAdd);
    }
} catch(SQLException e) {
    JOptionPane.showMessageDialog(null, e);
}
```

I used mutator to access the committeeID variable from the chairs however don't think this is working and the value isn't carrying on in the program as that page that I am accessing it from is closed by the time the voting page is opened.

```
//committeeID found using getter in login1 class
login1 committeeID = new login1(); //creating new object
committeeID.getCommitteeID();
//chairID found using getter in login1 class
login1 chairID = new login1(); //creating new object
chairID.getChairID();
```

After using breakpoints I noticed that the *committeeID* login class accessors has gone wrong since I am receiving a null value which is causing the errors. The value of committeeID isn't staying once page closes. I realized that instead of an accessor to get the values needed I would need to use a mutator instead in the class where I wanted the values to transfer.

```
public void setCommitteeID(int committeeIDFromLogin)
{
    committeeID = committeeIDFromLogin;
}
```

This was a logical error since the program still ran.

Even after this problem was solved however errors were still coming up but in a different section:

```
String sql_countryList = "SELECT `countryID` FROM `delegate` WHERE `committeeID` = "+ committeeID +"";
```

I realised after putting this SQL statement into *PHPMyAdmin* that I spelt committeld wrong, and therefore this new error was a syntax error.

Error

SQL query: 

```
SELECT `countryID` FROM `delegate` WHERE `committeeID` = 1 LIMIT 0, 25
```

MySQL said: 

```
#1054 - Unknown column 'committeeID' in 'where clause'
```

I corrected this spelling mistake. I also realised I never connected to the database in the testing class that I had created. I sorted this execution error.

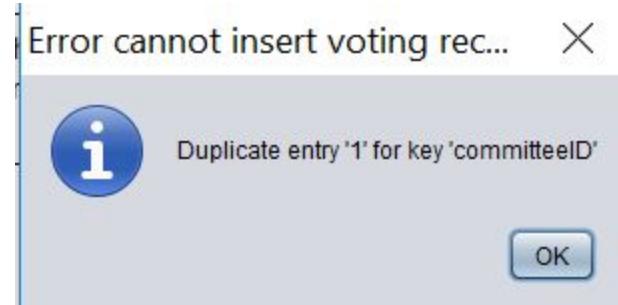
I had another error in this page when I was trying to run this particular query statement:

```
"SELECT `d.countryID`, `c.countryName` `d.committeeID`, `d.delegateID` FROM  
`delegate d`, `country c` WHERE (c.countryID = d.countryID);
```

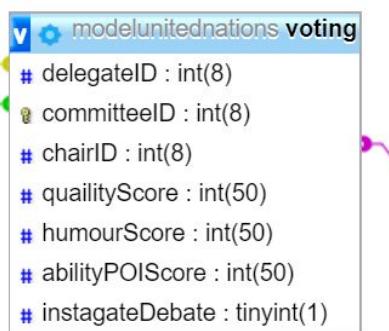
To solve this problem I ran the query through PhpMyAdmin and realised the SQL statement was wrong. This made the error a syntax error as I wasn't correctly doing aliases. I then corrected the statement to what is below which gave the expected result.

```
"SELECT d.countryID, c.countryName, d.committeeID, d.delegateID FROM `delegate`  
AS `d`, `country` AS `c` WHERE (c.countryID = d.countryID)".
```

When testing out the voting page I noticed that it only let me one record. This error then came up



I knew there was a problem with my data structure within the voting table and so went onto the design view.



I realised that the committeeID field is a primary key therefore cannot have duplicates of the number. I have to change this and if I can't create a table without a primary key then I need to include another field which is just auto incremented and call it votingID.

After deleting the committeeID and trying to recreate the relationship this error keeps appearing:

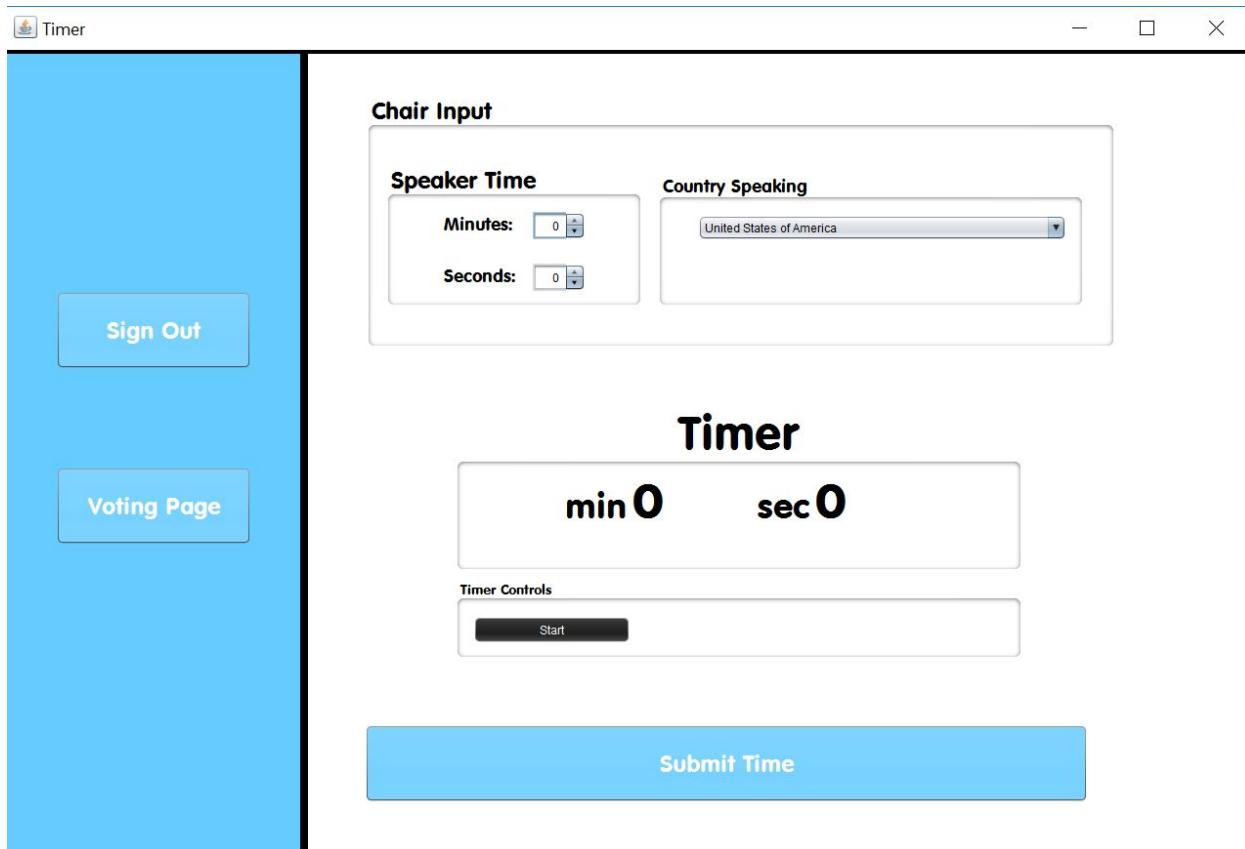
Error: Missing index on column(s).

I solved this problem by deleting the committeeID field altogether. When I recreated the field I made sure that the index type was INDEX and named it 'foreign'. This solved the problem and the committeeID no longer has a key next to the name, instead a hashtag. I also didn't need to include a votingID field. To fix this problem I looked the error up and a web forum (listed in my record of progress) helped. This error was an execution error.

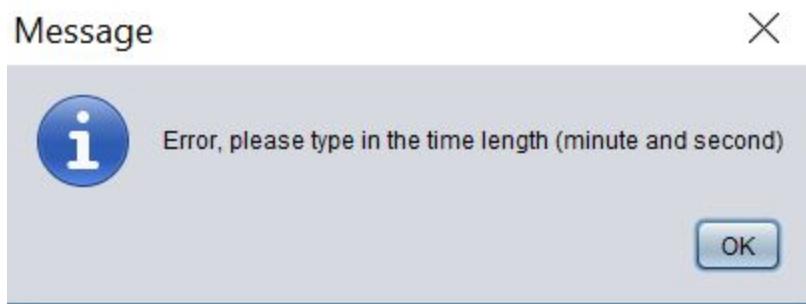
Timer Page

Component Testing

The main validation in this page is that if the input is 0 minutes and 0 seconds then an error message will appear.



The output for this was:



This was the expected result.

The committeeID is 4 and CountryID 19 so delegate 5 should have a time input for 90 seconds:

+ Options	
delegateID	timeSpoken
6	21
2	27
2	60
2	11
8	80
5	90

The correct information has been inputted into the table and therefore the insert method has worked.

If not all the full time is used then the time remaining gets subtracted from the initial total time, therefore if the initial total time is 2 minutes but the timer is stopped at 35 seconds left then the time spoken should be 85 seconds.

Input:

Chair Input

Speaker Time	Country Speaking
Minutes: <input type="text" value="2"/>	<input type="text" value="United States of America"/>
Seconds: <input type="text" value="0"/>	

Timer

min 0	sec 35
Timer Controls	
<input type="button" value="Start"/>	

Output:

+ Options

delegateID	time Spoken
6	21
2	27
2	60
2	11
8	80
5	90
5	85

All the buttons work and go back to the respective page.

Comprehensive Testing

When testing that the timers result entered into the database this error came up



I realised that this error was due to the fact I never received the input of the country name of the delegate speaking therefore the delegate ID couldn't be correctly received from the database and the values couldn't be inserted into the table. This was a logical error as it was due to my own fault that this error appeared.

Awards Committee Page

Component Testing

For the results (unless stated otherwise) I will test the results from the EcoSoc Committee.

If I use the values currently in the database I will manually calculate the total score for each delegation with the committee ID of 2.

To make the calculating process easier I created a query to get the sum of all the delegates scores in the EcoSoc committee. This query was:

```
SELECT voting.`delegateID`, SUM(`qualityScore`), SUM(`humourScore`),  
SUM(`abilityPOIScore`), SUM(`instagateDebate`), SUM(timer.timeSpoken)  
FROM `voting` , `timer`  
WHERE `committeID` = 2 AND voting.`delegateID` = timer.`delegateID`  
GROUP BY `delegateID`
```

+ Options	delegateID	SUM(`qualityScore`)	SUM(`humourScore`)	SUM(`abilityPOIScore`)	SUM(`instagateDebate`)	SUM(timer.timeSpoken)
	4	63	51	45	12	1915
	8	36	66	30	18	1614
	16	10	12	8	0	68
	19	6	8	8	0	152

When calculating the total score the equation below is used:

totalScore = (2*qualityScore) + humourScore + (2*abilityPOIScore) + instagateDebate + (timer)

This means that the expected total Score for:

DelegateID 4 is 2194, USA (Best Delegate)

DelegateID 8 is 1830, Russia (Highly Commended)

DelegateID 16 is 116, Venezuela

DelegateID 19 is 188, China (Commended)

The output was:

Results for the EcoSoc Committee

Awards

Best Delegate:	United States of America
Best Junior Delegate:	No Juniors in Committee
Highly Commended:	Russia
Commended:	China

Ranking

Ranking of Countries According To Total Score

1. United States of America: 2194
2. Russia: 1830
3. China: 188
4. Venezuela: 116

This was the expected results.

In order to see that the sorting algorithm was working correctly a trace table was used to follow the code the sorting algorithm for this page is shown below:

```

11  public class Sorting {
12
13
14  public void bubbleSortDelegates(Delegate[] delegates, int length)
15  //sorting delegates by total Score
16  {
17      //bubble sort from highest to lowest
18
19      for(int outerloop = length ; outerloop > 0; outerloop --)
20      {
21          for(int counter = 0 ; counter < (outerloop - 1) ; counter++)
22          {
23              if( delegates[counter].compareTo(delegates[counter + 1]) > 0)
24              {
25                  //create wrapper objects of the objects in the array so that all the information in the delegates object is swapped
26                  DelegateWrapper wrapper;
27                  wrapper = new DelegateWrapper(delegates[counter]);
28                  DelegateWrapper wrapper2 = new DelegateWrapper(delegates[counter + 1]);
29                  //swapping function
30                  swap(wrapper , wrapper2);
31
32                  delegates[counter] = wrapper.d ;
33                  delegates[counter + 1] = wrapper2.d ;
34
35              }
36          }
37      }
38  }
39
40
41  public void swap(DelegateWrapper dw, DelegateWrapper dw2)
42  {
43      Delegate temp = dw.d ; //temporarily holds the delegate values from the first parameter (a)
44      dw.d = dw2.d;
45      dw2.d = temp;
46  }

```

Line Number	Outerloop	Counter	Delegates[counter] Country Name and Total Score	Delegates[counter + 1] Country Name and Total Score
19	4			
21		0		
23			USA, 2194	Russia, 1830
21		1		
23			Russia, 1830	Venezuela, 116
21		2		
23			Venezuela, 116	China, 188
30			China, 188	Venezuela, 116

19	3			
21		0		
23			USA, 2194	Russia, 1830
21		1		
23			Russia, 1830	China, 188
19	2			
21		0		
23			USA, 2194	Russia, 1830
Code Terminates				

This trace table correctly matched the code and was checked using the breakpoints. As well is shown by the successful output of the country ranking's being in order

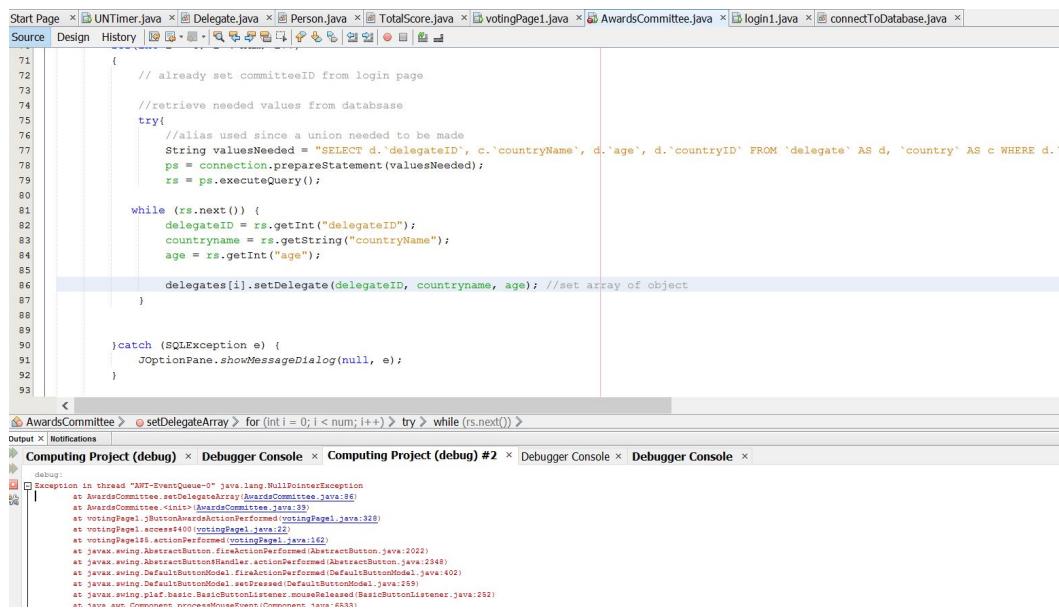
Overall the results for the awards also is how expected.

All the buttons work correctly and takes the user to the reperspective page as expected.

Since the Awards Overall committee is sorted in the exact same way there is no need to check to see whether the sorting algorithm works. The only difference is how the total score is calculated however this was used by creating a query and the query was checked using phpMyAdmin.

Comprehensive Testing

When testing my array of objects to make sure that the values were being set into the array an error keeps occurring whenever I reached the section of code highlighted in value.



The screenshot shows an IDE interface with several tabs at the top: Start Page, UNTimer.java, Delegate.java, Person.java, TotalScore.java, votingPage1.java, AwardsCommittee.java, login1.java, and connectToDatabase.java. The main pane displays Java code for setting delegate values in an array. A specific line of code, 'delegates[i].setDelegate(delegateID, countryname, age); //set array of object', is highlighted in blue. Below the code, the stack trace from the debugger console shows a null pointer exception:

```
Exception in thread "AWT-EventQueue-0" java.lang.NullPointerException
at AwardsCommittee.setDelegateArray(AwardsCommittee.java:86)
at AwardsCommittee.<init>(AwardsCommittee.java:39)
at votingPage1.JButtonAwardsActionPerformed(votingPage1.java:328)
at votingPage1.JButtonAwardsMouseReleased(votingPage1.java:22)
at votingPage1.JButtonAwardsMouseReleased(votingPage1.java:142)
at javax.swing.AbstractButton.fireActionPerformed(AbstractButton.java:2022)
at javax.swing.AbstractButton$Handler.actionPerformed(AbstractButton.java:2348)
at java.awt.DefaultButtonModel.fireActionPerformed(DefaultButtonModel.java:402)
at java.awt.DefaultButtonModel.mouseReleased(DefaultButtonModel.java:259)
at javax.swing.plaf.basic.BasicButtonUI$Listener.mouseReleased(BasicButtonUI$Listener.java:255)
at java.awt.Component.processMouseEvent(Component.java:653)
```

I realised that it would be easier to call each variable and make accessors and mutators for each of the different variables. This meant I could also make the variables private and that would increase the overall security in my project.

This error kept coming up after I called on the sub-class 'setDelegate' in the committee awards page.

```
36     //connect to database
37     connectToDatabase Test = new connectToDatabase();
38     Test.connectToDatabase();
39     //update connection variable
40     connection = Test.getConnectionVariable(connection);
41     setCommitteeID(committee);
42     setDelegateArray(); //private since shouldn't be accessed by general
43     sortDelegates();
44     displayResults();
45 }
46
47 private void setCommitteeID(int committee)
48 {
49     this.committeeID = committee;
50 }
51
52 private void setDelegateArray()
53 {
54
55     num = 0;
56 }
```

The screenshot shows an IDE interface. At the top is a code editor window displaying Java code. A specific line of code, `setDelegateArray();`, is highlighted with a red rectangular background. Below the code editor is a tab bar with three tabs: 'Output X', 'Notifications', and 'Debugger Console'. The 'Debugger Console' tab is currently selected and active. It displays a stack trace starting with 'Exception in thread "AWT-EventQueue-0" java.lang.ArrayIndexOutOfBoundsException: 0'. The stack trace lists several method calls, including `setDelegateArray` from 'AwardsCommittee.java:30', `<init>` from 'AwardsCommittee.java:42', and various methods from the Java Swing library like `AbstractButton.fireActionPerformed` and `DefaultButtonModel.fireActionPerformed`.

The array wasn't being initialised correctly which was causing the problem.

Beta

Two people were able to test my application and give feedback on it. The first person was an experienced MUNer who doesn't have that much experience in computing science and so was primarily looking at whether the functionality of the application was appropriate for MUN. The feedback that I received was that the 'Registration' (both delegation and chair) was *tedious* and *fiddly*. Although I couldn't do anything about this currently because all the fields the user needs to input in the registration process is absolutely necessary. Perhaps in the future chair accounts could be linked to Google and/or Facebook accounts and therefore the process of entering a username and password would be faster and therefore easier. Another complaint was that after registration (if successful) that user be brought to the respective page immediately and are able to forego the login process. Although this seems like a good idea and would be rather easy to implement I wouldn't like to include this feature currently due to privacy concerns or the chance that the user forget the username and password they just entered. A user forgetting their username and password is a great concern as there is no 'reset password' currently in my program and so the act of repetition of typing in username and password currently I feel is beneficial.

I also received positive feedback about how the application would be a positive experience and *help against bias chairs*. The person also felt that there were *good questions to help decide the winners* and that *the overall look of the application was very good*.

The second person who tested my application was a previous computing science student. I thought this would be useful as they have also studied about the importance of accessibility and design and would know what they are looking for in terms of the layout of the application instead of the content. I received feedback that *the buttons were clearly labelled* and that the application was *easy to use*. I also got positive feedback that the *navigation was easy* and *interface was clean/ clear*. Her overall statement was that *it functioned well with no errors*.

Overall my beta testing was a success and also made me think about updates I could install into the application if I were to develop it further.

Documentation

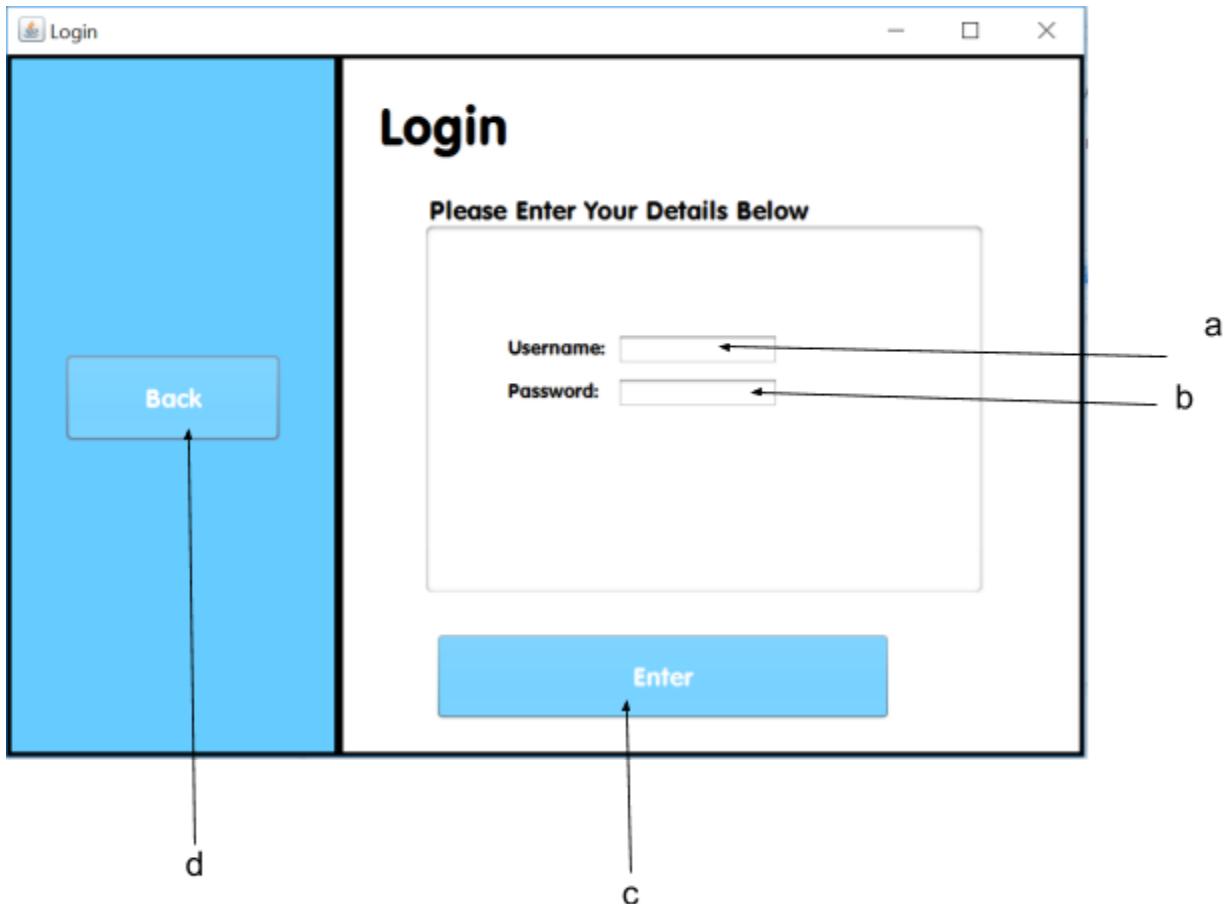
User Guide

Once the program is loaded the home page will be displayed and should look like the display shown below:



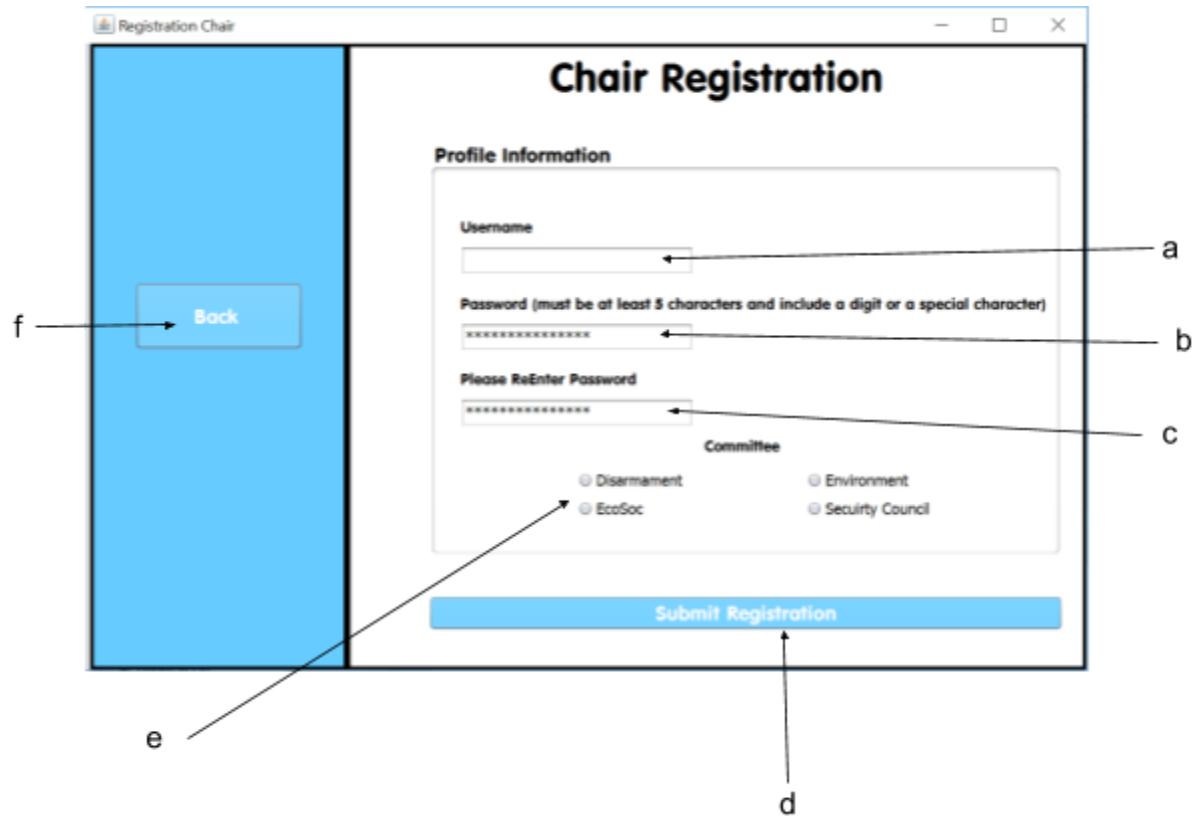
The user can then press one of the buttons shown on the navigation bar on the left (all labelled with numbers) and this will take the user to the respective page. Go to the number you want to learn more about once that button is pressed.

- When the login button is pressed the user will be taken to the login page. As shown below:



- This is where the user must type in their username using a keyboard.
- This is where the user must type in their password. The password is marked out in asterisk so that the password isn't visible for all to see.
- To complete the login process this button must be selected and if successful then the user will be taken to the respective page.
 - If the login was completed by a chair please see number 5.
 - If the login was completed by a delegation please see number 6.
- When pressed this button will take the user back to the home page.

2. When the Registration Chair button is pressed the user will be taken to the registration chair page as shown below:



- This is where the user types in their username they wish to register as using a keyboard. This has to be unique to the system and therefore if the name isn't unique then an error message will appear and the user will be prompted to change their username.
- This is where the user must create their password typed in using a keyboard. The password must be at least 5 characters in length and contain at least one number or special character.
- This is where the user must re-enter their password to make sure that it matched the password above.
- To complete the registration process this submit button must be pressed. If the registration has happened and there are no errors then a success message will appear.

- e. The user must select only one of the committee checkboxes to say which committee they are in. This selection can be done by clicking the circle on the screen with a mouse. If the circle is filled in black then that committee has been selected. If no committee is selected then the user will not be registered.
 - f. When this button is clicked the user will be brought back to the home page.
3. When the Registration Delegation button is pressed the user will be taken to the registration delegation page which is shown below:

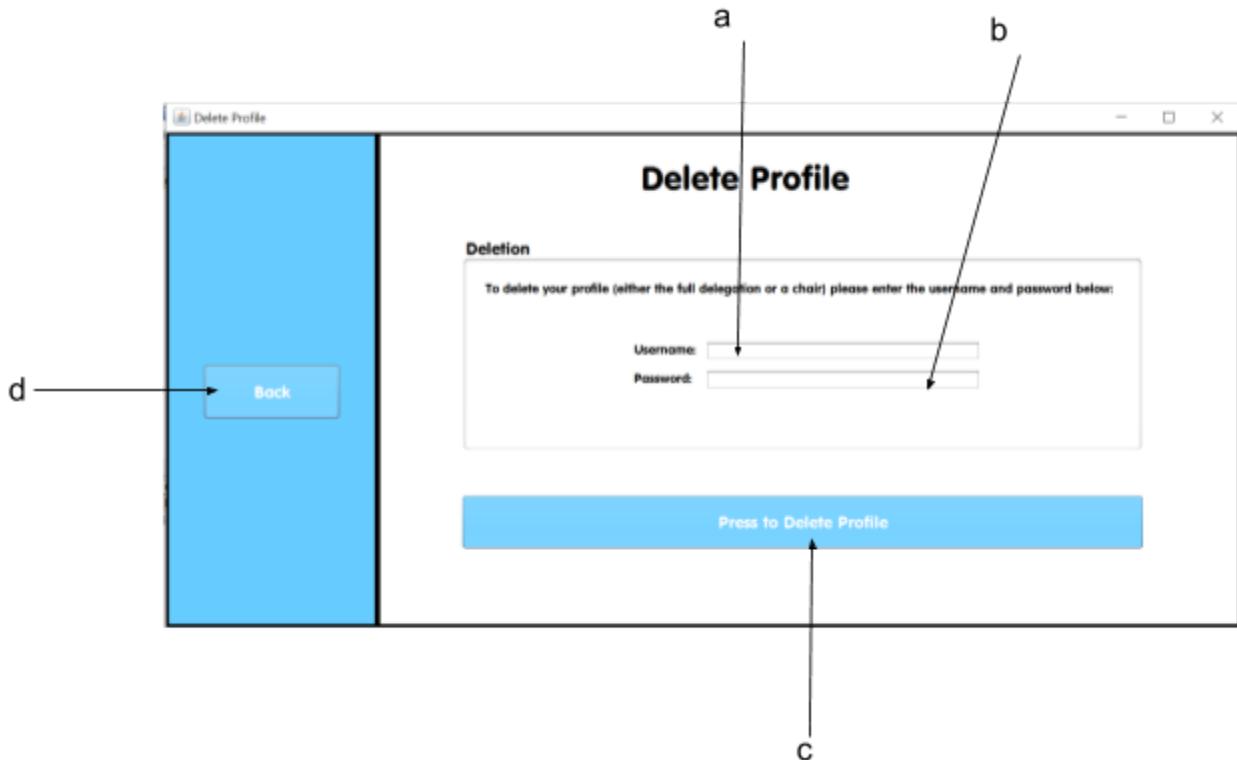
The screenshot shows a Windows application window titled "Delegation Registration". The window is divided into two main sections: "Profile Information" on the left and "Country Information" on the right. The "Profile Information" section contains fields for "Username", "Password (must be at least 8 characters)", and "Please Retype Password". The "Country Information" section contains a checkbox for "Please check the button if you would like a security council country" (with "Security Council Country" checked), and fields for "Delegation", "Environment", "Brief", and "Security Council (If applicable)". A "Submit Registration" button is located at the bottom of the right panel. The window has a standard title bar with minimize, maximize, and close buttons. On the left side, there is a vertical blue sidebar with a "Back" button. Callouts labeled a through h point to various elements on the screen:

- a: Points to the "Submit Registration" button.
- b: Points to the "Delegation Registration" title.
- c: Points to the "Security Council Country" checkbox.
- d: Points to the "Country" dropdown menu.
- e: Points to the "Security Council (If applicable)" field.
- f: Points to the "Country Information" section.
- g: Points to the "Profile Information" section.
- h: Points to the "Back" button on the sidebar.

- a. The user types in their desired username here using a keyboard. If the username isn't unique to the database system then an error message will appear and they will be prompted to select a new username.
- b. This is where the user must create their password typed in using a keyboard. The password must be at least 5 characters in length and contain at least one number or special character.

- c. If this box is selected then the delegation wishes to have four members instead of three the fourth member being a security council member.
- d. Drop-down list containing a list of country names. To access the drop down list the user just needs to click on the arrow on the right side of the grey box. If the country selected is already has a team then the user will be prompted to select a new country from the list.
- e. The user must type in the ages of the members of their delegation in relation to which committees the member will be in. This will be typed in on a keyboard
- f. To complete the registration process this submit button must be pressed. If the registration has happened and there are no errors then a success message will appear.
- g. This is where the user must re-enter their password to make sure that it matched the password above.
- h. When this button is clicked the user will be brought back to the home page.

4. When the Delete Profile button is pressed the user will be taken to the delete profile page as shown below:



- a. This is the username where the user must type in their username using a keyboard.
- b. This is where the user must type in their password. The password is marked out in asterisk so that the password isn't visible for all to see.
- c. To complete the deletion process this button must be pressed the user will then be prompted again making sure they wish to delete their respective profile (either chair or delegation) to which they must answer either 'Y' or 'N'. If answered 'Y' then the profile WILL be deleted. If answered 'N' or any other input then the profile WILL NOT be deleted.
- d. If this button is pressed then the user will be taken back to the home page.

5. If the login entered was a chair the user will be taken to this page, the Voting Form. This is one of the main components of my program and is where the user mostly inputs their information

The screenshot shows a software application window titled "Voting Form". On the left, there is a vertical sidebar with three buttons: "Sign Out" (labeled 'a'), "Awards Page" (labeled 'b'), and "Timer" (labeled 'c'). The main area is titled "Country" and contains a dropdown menu set to "United States of America" (labeled 'g'). Below this is a section titled "Categories" with three rating scales: "Quality of Speech" (ranging from 1 to 5), "Humour in Speech" (ranging from 1 to 5), and "Ability to Handle Points of Information" (ranging from 1 to 5). A message at the bottom states "Added New Information to Database" (labeled 'd'). On the right side, there is a large blue button labeled "Submit Vote" (labeled 'f'). A callout arrow labeled 'e' points from the "Added New Information to Database" message towards the "Submit Vote" button.

- a. When this button is pressed the user is logged out of their account and are taken back to the home page (number 1)
- b. When this button is pressed the user is taken to their own Committee Awards Page.

- c. When this button is pressed the user is taken to the timer page.
 - i. The timer page is loaded and looks like the image below.

1. This is the speaker time. The user must use the arrows (or type into the box) how many minutes and/or seconds they want the delegate to speak for. It should be noted that the time can only be submitted if these values aren't set at zero if that happens an error message will appear.
2. Drop-down list containing a list of country names in that particular committee. The country name is the name of the delegates country who is currently speaking and you are therefore timing. There is also an *unmoderated caucus* section. This is there for timing purposes i.e. a break and this time isn't noted in the database.
3. This is the where the timer controls are. To start the timer the start button should be pressed. Once the start button is pressed this button will no longer become visible however a pause button will become visible. If the pause button is pressed the timer will stop and the pause button will no longer become visible however the start button will re-appear. This will continue until the timer has run out or until the 'submit time' button is pressed (number 4)

4. To submit the time this button must be pressed. A message confirming that the vote has been processed will be displayed after which the timer will then be how it was the first time it was used.
 5. When this button is pressed the user will be taken back to the voting page (big number 5, current section)
 6. When this button is pressed the user will be logged out of the application and will be brought back to the home page, the very beginning of the application.
- d. This is a checkbox which can either be selected or not-selected. This is to say whether or not the delegate speaking has added new information to the debate or not.
 - e. These are the sliders in which the user can move to score the delegate speaking on the three aspects of quality, humour and ability to handle points of information. The slider 'snaps' into place on the nearest whole number so no decimal points can be awarded.
 - f. Once this button is pressed the vote will be processed and the values inputted will be registered and stored for later use. A message should display after the button is pushed confirming that the vote has been processed.
 - g. Drop-down list containing a list of country names in that particular committee. The country name is the name of the delegates country who is currently speaking.

6. When the Overall Awards page is loaded this is what the page will look like:

- a. When this button is pressed the user will be logged out of the application and will be brought back to the home page, the very beginning of the application.
- b. When this button is clicked the user will be brought to a committee Results page. Depending on the users access (chair or delegation) will depend on the small details of what happens.
- c. When this button is pressed the user will be brought ot the voting form. It should be noted that this button is only visible to the users with chair access rights.
- d. This is where the awards are displayed.
- e. This is where the ranking of the countries are displayed.

7. When viewing the Committee Results this is what the page will look like:

- a. When this button is pressed the user will be logged out of the application and taken to the home page.
- b. When this button is pressed the user will be taken to the overall results page. It should be noted that for result to be successful all of the committee's results must have been accessed and viewed.
- c. This button is only accessible and visible to the chair logins. When pressed the user will be taken to the voting form.
- d. This is where all the awards are displayed. The award is next to the country which received the award.
- e. This is the ranking box where the order of the delegates from best to worse is displayed. Shown is the country name and the delegates total score.
- f. Not shown on this diagram is a button which is available to delegation logins which allows the user to change to view a different committee results. This is done through a choices message box which will be displayed and the user will be prompted what to do.

Technical Guide

To run the application the user is required to have a few hardware and software equipment.

The export of the ‘modelunitednations’ database is required and stored as a SQL file. This file requires 8 KB of storage space. The user will then be required to use a server application (such as Xampp) to import the database and store it on their own local server.

The application is required and is stored as an Executable JAR File which is called *Computing_Project*. This requires 162 KB of storage space. The imported libraries for the application is also required which will take up 1,606 KB of storage space.

A monitor, keyboard and mouse is required as hardware to use the application.

Evaluation

I nearly managed to reach all my solution specifications apart from the inclusion of a the 'Best Humour' Award. I didn't include this because the code would have become really iterative as I would need to include a whole other sorting algorithm just to sort the humour section. Although there were probably more efficient ways this could have been done, time restraints meant that I didn't have time to explore different methods to complete this task.

I thought that I managed to make my application very user friendly due to all the different types of ways the user can input data. My data input ranged from textboxes to sliders, drop-down list boxes and checkboxes and spinners. I also felt that my application as a whole is very responsive and gives the user feedback if something goes wrong and how to fix that, of course this could still be improved upon.

Given no time constraints I would have included the 'Best Humour Award' but i would also generally go through my code and see how I could make it less repetitive. I feel like with the use of so many SQL statements this could have been reduced. Another main issue was with the delegation and delegate class with the sorting algorithm. I would hope that I would be able to condense the sorting algorithm to only two methods instead of four (double what I would ideally like). I would have also like to perform more extensive testing to make sure that my application was as reliable and robust as it could be. I would also like to look at the design of my application more and make all the forms one consistent size.

I was proud of the small details I managed to include. For example in the voting and timer page when you look at the country list only the countries which are currently registered in that committee appear in the drop-down box. This took a lot of work as I had never used drop-down boxes in java netbeans before. I am also extremely proud of my timer function as this wasn't something that was taught in the course or that was even required. Although I did use the 'backbone' of someone else's code I managed to tweak it by quite a lot and make it my own and improved the overall timer.

An improvement to be made was that I could have used inheritance in my delegate class and delegation class. A Lot of the methods were repeated and methods that I didn't want in one class could have been overridden in another. I could have also used more private methods in my classes so that there is a greater level of security in my program. I also noticed that I wasn't being consistent is that on most of the pages the

cursor is set to a 'hand' setting however on some of the pages I forgot to change the default cursor setting to the hand cursor and therefore there are some inconsistencies.

My application is fit for purpose as suggested by my beta testing. My application is robust due to the fact that the code is very responsive and that there is a lot of validation on the user inputs. My application is also reliable as the sorting algorithm always works correctly and expected. The only unreliable aspect is that for the overall ranking of the delegations to be accurate all the other previous committee's rankings need to be ran. My program isn't very efficient due to the fact that inheritance wasn't used in the classes (delegates and delegations) and as mentioned previously the two sets of sorting algorithm that needed to be used to sort the array of objects. My application is very portable as a JAR file is used to execute the code. This means that no compilers are needed to load the application as the application is already compiled in this form. The main problem which makes the application less portable is that the database needs to be exported and loaded by the user. The database at first was only accessible on my own localhost of phpMyAdmin and so this makes the application less portable however a SQL file is available to import the database into another server and therefore this makes the program very portable.

Maintainability of my program would be heard especially when releasing any future updates. This is because currently my program could either be freeware or shareware if it were to be released. I would probably make the program freeware with the application being loaded on the JAR file as this would mean that I no one could copy my code and try and sell it off as theirs in the future. I know that the *Copyright, Designs and Patents Act* could prevent this from happening however I would like to be extra cautious. Since once the code is out there it would be hard to get users to update the software if future updates were realised. However since no sensitive information is being stored in the database then security concerns within the program isn't very important and therefore feature updates would only be to add greater efficiency or more features onto the application.

During my project the development methodology used was agile methodology. This meant that the testing and implementation of the program was completed simultaneously and that sprints were performed. I choice this method over the waterfall method as this method wouldn't have been efficient due to the repetitive nature. The agile methodology worked well and I would repeat this methodology if I were to do the project again.

References

<http://bestdelegate.com/4-useful-apps-for-model-un-students/>

<https://www.munplanet.com/questions/mun-technology/what-is-the-best-software-to-run-a-model-united-nations-committee>

<https://www.techwalla.com/articles/the-advantages-disadvantages-of-visual-basic>

<https://netbeans.org/switch/why.html>

<http://infotech101.com/the-netbeans-ide-pros-cons/>

<https://www.siteground.com/tutorials/php-mysql/mysql/>

https://www.tutorialspoint.com/What-are-the-advantages-and-disadvantages-of-using-M_ySQL-stored-procedures

<https://www.techwalla.com/articles/the-advantages-of-oracle-databases>