Sectrics - Civil Engineering Software –Part A – HSC MAJOR PROJECT

By Shaan Khan – 3146 4145

Contents

Defining The Problem & Feasibility Study

Identification Of The Problem

Throughout high school and college careers civil engineering students are required to perform long and tedious and complicated hand calculations to calculate forces acting in each beam via truss analysis then further determine compressive & tension acting on each member and then further analyzing the problem using bending moment & shear force diagrams to further determine and analyzing the forces acting on a rigid body. During all these calculations students always have one major problem, determining if their answers are right or not. Even the beginnings of truss analysis require a minimum a page of calculation to determine the forces acting within a small system, with more complicated trusses the time taken to solve a question increases whilst the chance of getting the answer correct and the preciseness of the answer decreases. Depending on the application getting the incorrect answer may be unacceptable and can result in serious bodily harm or even death from incorrectly calculated forces resulting in the failure of a rigid body system.

To assist these future engineers to help validate their answers and fully master the art of truss analysis a new system must be implemented, by the creation of an application to help students & engineers alike to help quickly determine forces acting on a system students & engineers alike will be able to save countless man hours in comparison to slaving away at equations that may or may not contain an error being potentially fatal.

In the year 11 class of engineering studies after being introduced to truss analysis a problem was shown. Truss analysis is one of the hardest parts of the syllabus and there is no way to clearly validate answers after searching for a solution such as a truss analysis calculator a problem arose. There is no non-limiting free truss analysis piece of software for students. After approaching Mr Gill regarding this problem an idea was brought up, the creation & implementation of a new civil engineering software suite aimed towards HSC engineering students and thus this program was born.

Functionality Requirements

Mandatory Requirement

- Must be compatible with windows 10
- Section 1 must be completed before the 16/12/2018
- The user will be able to enter:
    - The x & y coordinates of each node
    - The order of members of which node connects to easy node
    - The X & Y magnitude & location of a load acting on the system
- The program when prompted will be able to:
    - Calculate both all forces acting in each member and then determine if it either in compressive or tension forces

Desirable Requirement

- The program will contain graphics allowing the user to easily and be able to enter in data and validate data
- Display X & Y coordinates of each member on the bridge in a visual method removing having to visualize the bridges node & force locations via viewing coordinates
- The program when prompted will be able to:
    - Generate a corresponding shear force & bending force diagram

Optional Requirement

- Have the graphics of this program be aesthetically pleasing
- Being able to display hand calculations to assist students
- Containing a tutorial and welcome screen to the user introducing them to the program in a quick useful manner
- Allowing the user to be able to zoom in and zoom out on graphs & the visual appearance of bridges determining if each item is correctly inputted
- Allowing the ability of users to save current models of their bridge for future use
- The program when prompted will be able to:
    - Being able to calculate the maximum bending stress acted upon onto each member

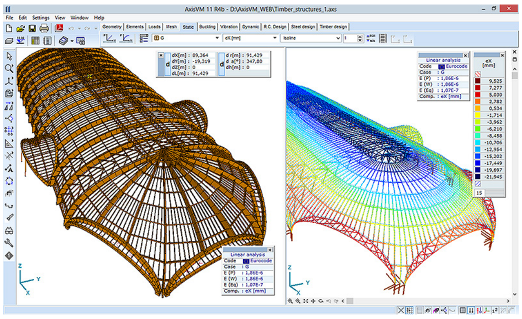
Possible Future Enhancement

- Being able to use machine learning to help determine points of failures and generate or improve a bridge to its maximum efficiency dependent on its specifications
- Displaying heat map locations of compressive & tension forces until failure using a time-lapse of forces further increasing on a specified point

Communication In Proposed System

In the Sectrics program, communication involving external entities excluding the user is to a minimal. The user interacts with the system by inputting information regarding the bridge including; node location, member order, forces on each node with both x & y magnitudes, support locations and support type. Whilst the system will process and then further output additional information regarding the bridge including; displacement, tensile/compressive forces in each member as well as reaction forces and torque forces. After this the system will output this data onto the user's screen, sending the information back to the user and if requested will also additionally communicate with a printer to print the information in tabulated form displaying all calculated forces in a neatly structured table with the force, displacement and reaction forces corresponding to each member.

Generation Of Ideas



Idea Generation

After determining the requirements of this project, I looked further at alternate solutions to this problem and further added to the requirements of this software and enhancing requirements further taking small ideas from which each piece of software did effectively. After creating the first prototype UI for this program I encountered a few issues. How would the user be able to determine in which location & order would to enter data. After prototyping with a few examples, I determined the best course of action would be a minimalistic design containing a grid to allow the user to view the bridge with a scale and in addition having the order of the UI focused on order rather than efficiency sacrificing machine efficiency for human efficiency focusing on a human-focused system rather than machine focused system. Throughout this process, this step was reiterated again and again until we concluded with our final solution leading to the most optimal human-centred design.

Solution Idea Generation

1. Method of joints– by using a previously pre-existing civil engineering method by discovering reaction forces, moments and then solving for beams we are able to solve for forces in each member. This, however, is a tedious operation requiring many calculations to be repeated to find the forces in one member and then again for the next and so on.
2. Method of sections – a similar idea to a method of joints, the user solves the system by first calculating the reaction forces and moments but then isolating a section of the bridge and assuming the section is in equilibrium as the entire system is in equilibrium and then solving isolated areas from there. This is a quicker method than the method of joints for a human to solve, however, this leads to more complicated patterns for a computer to solve and is still generally inefficient.
3. The simultaneous equation using distance coefficients – a more advanced and quicker way, especially for a computer to solve is to utilize a matrix and solving a large simultaneous equation on the whole system and then determining the tensile/compressive forces from there. This is the most efficient way to solve however requires the creation of several functions prior to the development of the program, regarding matrices.
4. Finite element method – this is the same as the simultaneous equation method but however has additional steps compared to the other three. For the first three operations, they are unable to easily solve or solve at all for dynamic systems. Using the finite element method, a dynamic system such as a bridge which opens, and closes can be calculated with the maximum force in each member at a specified time. This is the ideal and most flexible method for solving truss analysis problems. Additionally, this method has been specifically created to solve for problems on computers and there is additional course information from second year Civil Engineering universities on this method and therefore would be the most efficient and most optimal method of solving for the tensile/compressive forces.

Alternative Solutions

Possible Alternate Solutions

For the alternate solutions to this software problem several additional alternate proposals were made, some of the more notable ones include:

1. Phone application: This current program is made to work specifically with windows in mind however since the choice of software was C#, which allows cross-platform support to Linux and MacOS. With several major alterations, more heavily the changing of the UI & UX designs, this piece of software could instead be implemented as a phone application. This idea, however, has quickly been removed as it would same unusual or rare that a user would choose to conduct truss analysis on a phone due to it being slow and also dangerous for accuracy wise when inputting data. This in addition to windows being the dominant operating system has resulted in this idea being discarded.
2. Cloud-based software: Alternatively, instead of making this application dedicated and limited to the windows platform by removing this restraint and going towards an approach focused on a cloud-based platform we, in turn, increase our market share to nearly all devices which interact online. In addition, this would further allow users to store their bridges online or share with friends allowing interactivity between users as well as prevent the loss of data via misplacing it such as losing a hard-disk drive or USB.
3. Text Based Game: The addition of graphics required a major time investment to implement it instead of having graphics this application could have saved on time and money by not implementing it, this, however, is a con as the user would have a harder time navigating and understanding the program.
4. Do Nothing: Similar applications to this program currently exist and the requirements may have been edited to fit that application or the making of this software could be redundant as there is a dominant pre-existing market. By avoiding doing this project we save on time & money on the risk that this software will not be a successful commercial product.

Existing Alternatives Solutions

Existing alternative solutions generally follow the same solving method using the Finite Element Method, however, vary from how the data is inputted as well as how the program is running. Some pre-existing alternative solutions include:

1. SkyCiv | Cloud Structural Analysis Software – SkyCiv is a cloud structural analysis software which means it runs online on the cloud, the benefit of this is that programs can intensely utilize a big range of computing power fairly simultaneously and relatively cheaply. In the structural analysis, the number of supports, members and points of deflection can vary from tens of thousands to hundreds of thousands. Due to this by using cloud-based computing the program is able to utilize a vast range of computing power simultaneously to help quickly solve a structural analysis problem instantaneous without the user requiring a server farm for more complicated structural analysis solutions.
2. Robot Structural Analysis Professional – Robot is an all in one structural analysis piece of software that helps engineers perform simulations, analysis and designing of structures quickly and effectively.
3. COSMOS/M is an entry level complete self-contained finite element system designed for personal computers and workstations. This program solves both static and dynamic structural problems but however is limited as it is targeted towards new users or people requiring building a finite element method quickly due to this, some more advanced problems cannot be solved using this program.
4. ANSYS is a structural analysis software that enables engineers to solve complex structural engineering problems. This is targeted towards the more advanced users of structural analysis software and therefore is not as user-friendly to start learning, with this limitation, however, the program makes up for it with its flexibility allowing users to solve incredibly complex finite element models. But due to the flexibility, it lacks the ability to quickly create models and requires a long amount of time to solve simpler models that could have been done quicker in other software such as COSMOS.

Error Handling

Errors will be minimized with data validation checks prior to allowing data to be used by the system minimizing the occurrence if not eliminating the chance of the program to throw an exception, this will be further elevated by the utilization of a fail-fast orientated programming stance and will, in addition, all contain catch loops ensuring data is entered and that the data is valid.

If an error is to occur, however, the program will prompt the user notifying them of this issue and then once the user acknowledges this error, the program will terminate gracefully saving a [Recovered] copy of the save data in addition to its original saved data allowing the user to recover from this error.

In addition, the program will create an error file in its current directory containing the exception or error presented, this, in this case, the error file can be presented to the developer via means of contacting via GitHub allowing the error to be patched and easily identified.

Conventions / Standards

The programming convention used for this program are as followed:

- The comment convention
- Indentation style convention
- Characters per line convention
- Variable naming convention
- Coupling (computer programming) principles
- Deutsch limit
- Fail fast orientated programming

In addition, the Microsoft convention standards are followed heavily as a basis for this program as well as the; Input-Process-Output (IPO), Storyboards, Context Diagrams, Data Flow Diagrams (DFD), Structure Charts & System Flowcharts for documentation. This will ensure the idea of the program will be clear to start with allowing the programmer to easily create a program that will satisfy this program and more imploringly the Structure & System flowcharts will easily and clearly assist the coder ensuring the software contains scalability, the ability for addition and most importantly no logical errors.

Feasibility Of The Solution

Scope

Project Purpose & Justification

The assessment task given by Mr Haung is to plan, document & design an application of our choosing, this application is a piece of civil engineering software currently called Sectrics. The purpose of this project is to create a civil engineering tool to assist students studying civil engineering and civil engineers alike help calculate forces acting upon a rigid body in a quick efficient manner opposed to hand calculations. The function of this piece of software is to be able to collect user information of their specifications regarding a bridge, organize it in an appealing way to view it visually such as representation of the bridge and then be able to process the data to the users required specifications such as displaying compressive & tension forces in each member and display the reaction forces required on each support and so on.

High-Level Requirements

The civil engineering software, Sectrics has been approved for development. In order for it to satisfactorily fit the target markets specification, there are several requirements that must be met. Sectrics is a program targeted towards students and therefore must successfully address the requirements of students. By instead of listing complicated data outside of the scope of students' studies but only listing the bare essentials thus being a tool for question validation rather than focus on real-life applications such as proper civil engineering we prevent liability as well as save time and money due to further focusing the scope. For a successful execution of this project, several requirements must be met these include:

- Being able to easily allow users to enter data such as coordinates of nodes, coordinates & magnitude of forces etc.
- Being able to easily be able to receive & interpret output results given by the program. I.e. output of reaction forces in each support
- Being able to quickly respond to user feedback with a tactile response indicating that the program is still functioning whilst adding the factor of responsiveness into the program
- Having the ability to respond to users enter incorrect data via data validation and catching these errors and preventing them from occurring

Boundaries

The "Sectrics" project includes all work regarding planning, designing, building & releasing the product in a commercial environment. This project includes; Identification of the problem, functionality requirements, feasibility study, UI & screen designs, proposed algorithm design & implementation, refined system modelling & technical requirements regarding the software. Items **NOT** included in this project include:

- The full implementation of cross-platform or multi-platform functionality
- Future software support & patches
- Any ongoing patches
- Commercial popularity & campaign
- Collection & utilization of user feedback
- Compatibility for all hardware setups, emphasizing on the minimum hardware requirements
- Compatibility requirements for windows drivers' configurations leading to incompatibility
- Collection and utilization of private user data
- Control & rerouting of network traffic regarding what a system does preventing control of data from other systems
- Certification & reliability of Sectrics

Strategy

For Sectrics the project strategy will be a simple yet efficient one. As the team consists of one-person Shaan Khan (3146 4145) who will be acting as programmer, writer, designer & program tester the project strategy will be organized in a manner to revolve around this one user approach. As the size & project group of the project permits it a recommended approach would be following rapid application development (RAD) which is a subset of the agile approach, however, as the requirements set by the committee outsourcing this project demand the structured approach, the structured approach will be utilized. However, the prioritizing of each section may be edited to contrast with the size of the group for a more appropriate streamlined process. Non-functioning or semi-functioning throwaway prototypes will be utilized and then evaluated upon obtaining key features and ideas, after which the structured approach will ensure. Starting from the feasibility study the project will then divert into the planning & designing phases which will be done in parallel with the programming of the software solution following this method should ensure in a streamed project suitable to finishing each requirement milestone in its allocated time period. This in addition ensures if any additional errors occur or if the user specifications are required to be changed, they can be changed quicker and cheaper in contrast to acting upon in a later part of the development cycle saving a factor of x10-100 times the amount of money required to change the specifications in a later section of this task.

Deliverables

There are several deliverables which will be achieved as result to the successful completion of the Sectrics project. If all of the following deliverables are not met, then this project will not be considered successful. These deliverables include:

1. A written report assessing the feasibility of this project
2. A planning & designing report showing the processes & designing required and involved in the creation of this project
3. A complete and full troubleshooting guide/manual
4. A working copy of the software solution commercial packaged paired with correctly packaged documentation
5. A logbook journal documenting the documentation and coding aspect of this project including the time required for each aspect of this project and the progress ensured per hour containing all errors or hurdles ensured throughout this project

Assumptions

For this product, no outsourcing of work may be permitted with the exception of the creation of graphics UI & assisting from external vendors for code review and assistance in problems occurring in this software solution. This external work may not be approved unless correctly documented, logged and be in compliance with the NESA "All my own work" guidelines, rules & regulations. Furthermore, any changes outside of the task description must be discussed and evaluated by the teacher and project manager of this project, Mr. Haung.

Constraints

The final finished project proposal and design documentation must be submitted by term 4 weeks 10 2018 and the software solution packaged accordingly in commercial packaging must be presented in term 2 weeks 10 2019. This leads to time being an important factor in this project, as the team has relatively limited experience in programming however has successfully completed several projects prior, including the Master Mind application. Due to this team being relatively new the timeframe may or may not fluctuate especially with certain coding milestones such as the completion of the mathematics library. Due to this restriction & limited time frame the program may or may not contain optional features by the allocated time frame given, however, will be able to successfully complete the minimum viable product (MVP) within the timeframe given. Ways to improve this issue include the allocation of a greater time frame, the expansion of the team of more programmers & or more experienced programmers, the removal of the restriction on outsourcing or the changing of the method from the structured approach to agile system development. Excluding these constraints, there are no further constraints applied to this project hindering its production.

Financial

Cost Estimate

For the development of the Sectrics program, the expenditure of money is required to allow for the application to be successfully developed. The estimated cost for this project is included in the table below. As the project proceeds and any additional costs become know, this cost estimate will be refined and displayed below.

The price of this program will be extensive, due to this being a Civil Engineering piece of software the costs associated as this program must not have a percentage of error is critical. Due to this, it is critical that this program is extensively tested to ensure there are no errors. This, in addition, will unfortunately add to the cost of the program. Additional costs may arise due to the requirement of consultation of current Civil Engineers in assistance with the finite element analysis method in addition to surveys done to ensure this program satisfies the requirements.

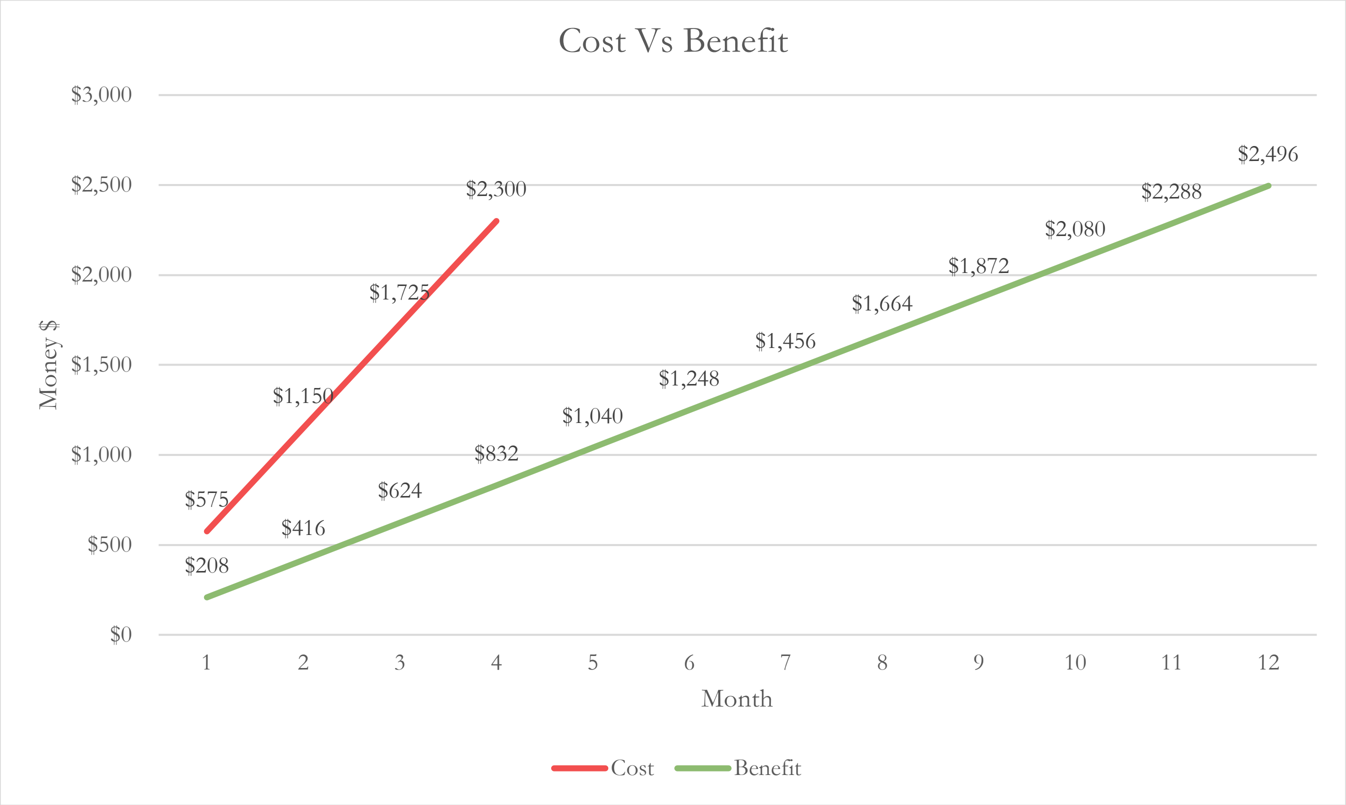
Expense	Description	Percentage Of Total Price	Estimated Budget	Expended To Date	Variance
1. Labor –Internal	Internal labour costs associated with the programming of the Sectrics program including the cost of programming and the additional costs associated with the employees of this project such as utensil use etc.	85%	$1700 (75-100 hours)	$300	-$900
2. Labor – External	External labour costs associated with the outsourcing of labour external to the employees hired for this project. External labour includes the hiring of professional UI & UX graphics designers and/or external help if money is exchanged for such help	0%	$0	$0	$0
3. Software	This includes all costs associated with the purchasing of external software programs in order to aid with the creation of this program. Such software includes other commercial products to inspect & computer-aided software engineering tools (CASE tools) for assistance during the structured approach	5%	$100	$20 (SkyCiv Student Edition)	-$80
4. Hardware	Hardware costs of purchasing additional hardware required for the development of this program. Note: general computers to program on are exempt. This includes additional custom hardware such as Raspberry Pi's and or specialist modules such as camera sensors or Intel Movidius neural compute stick.	0%	$0	$0	$0
5. Commercial Packaging	Commercial packaging costs outline the costs encountered whilst creating the final section of the final solution deployment section. This includes the creation & implementation of commercial packaging	5%	$100	$0	-$100
6. Printing Commercial Documentation	This section is an extension to commercial packaging and includes the cost encountered by the printing and releasing of commercial documentation. This includes the binding & printing of documentation regarding this project	5%	$100	$0	-$100
Total:		100%	$2000	$320	-$2000

Cost-Benefit Analysis

A cost-benefit analysis has been performed for the Sectrics Project. The successful completion of this project will provide significant benefits to the student body studying engineering in the HSC syllabus. It is important that all stakeholders know the benefits as well as the importance of the successful completion of this project.

The cost of this program is $2000 AUD with the successful implementation, HSC students will be able to conduct truss analysis at a highly competent level aiding them in their calculations as well as developing an understanding of the significance of truss analysis and structure analysis as a whole in Civil Engineering.

	With Sectrics Program	Without Sectrics Program
Cost of Project		$0
Recurring Cost	$0	$0
Non-Recurring Cost	$2300	$0
Capital Costs	$0	$0
Total Cost of Sectrics Project	$2300	$0
Benefits (1 year)	100+ hours saved per student ($2500)	0 hours saved by students
Total Benefits of Sectrics Project	100+ hours per student ($2500 P.a)	0 hours per student
Net Benefits of Sectrics Project	100+ hours per student ($300 + 2500 P.a after the first year)	-100 hours per student



Operational

The Sectrics program is targeted towards students studying civil engineering at a high school to university level. The target audience has been chosen for this program as this in turn reduces liability as our program will not be certified and thus may lead to unknown errors possibly causing death in extreme scenarios if a faulty calculation causes a rigid body to fail. Due to this the target audience has been chosen to be students who will in turn not be building projects resulting in liability and thus we will be free from liability in turn. Due to the target market being chosen as engineering students the program has been suited to fit in with the needs of students especially. Such possible features include; hand calculations and additional features such as shear force diagrams specifically designed towards the HSC syllabus.

The Windows operating system has been chosen as it is the most common operating system to the general public and in addition to the windows operating system being user-friendly allowing a user using the operating system to easily install our program in "one click of a button". If implemented successfully the Sectrics application will be a highly successful truss analysis piece of software imperviousness to liability caused by this software due to the target market. Due to this in addition to the relatively low cost for the development of the application, this program should be successful with having a relatively low risk or cost if commercial failure is to occur.

This software has been defined to be human-centered rather than machine centred, meaning that it will focus on users rather than solely on efficiency, due to this integration into an existing software should be relatively easy as there should be minimal to no training required for the implementation of the Sectrics program. Additionally, there is no communication for this program required and therefore does not require any specific hardware or software configurations to be installed prior to installation of the program. This in turn also reduces the impact of pre-existing infrastructure bottlenecking an organization such as a lack of electricity due to an inadequate power grid and many other factors.

Technical

The Sectrics program is targeted towards the windows platform, due to having the largest market share owning over 83% of the current market thus being the best market to focus on for development. If time permits and the Sectrics program is a commercial success an additional MacOS and/or Linux version may proceed with development. Since this program is targeted towards the window platform cross-platform compatibility is not supported and may or may not run smoothly if at all. However, since this program is to be coded in the C# programming language, cross-platform support is an easily feasible option as C# is a cross-platform language. The process of creating and running this software does not require any specialist hardware or software due to it containing a relatively light load on the CPU in standard usage, however, if the user does enter a complicated bridge design the time taken for processing will increase drastically.

This program should not require any additional drivers or dependencies from the user and will only require the standard windows library, any additional libraries required will be hand coded and made specifically tailored for this program thus removing any dependencies required for a user being all installed when the installation of the program is completed.

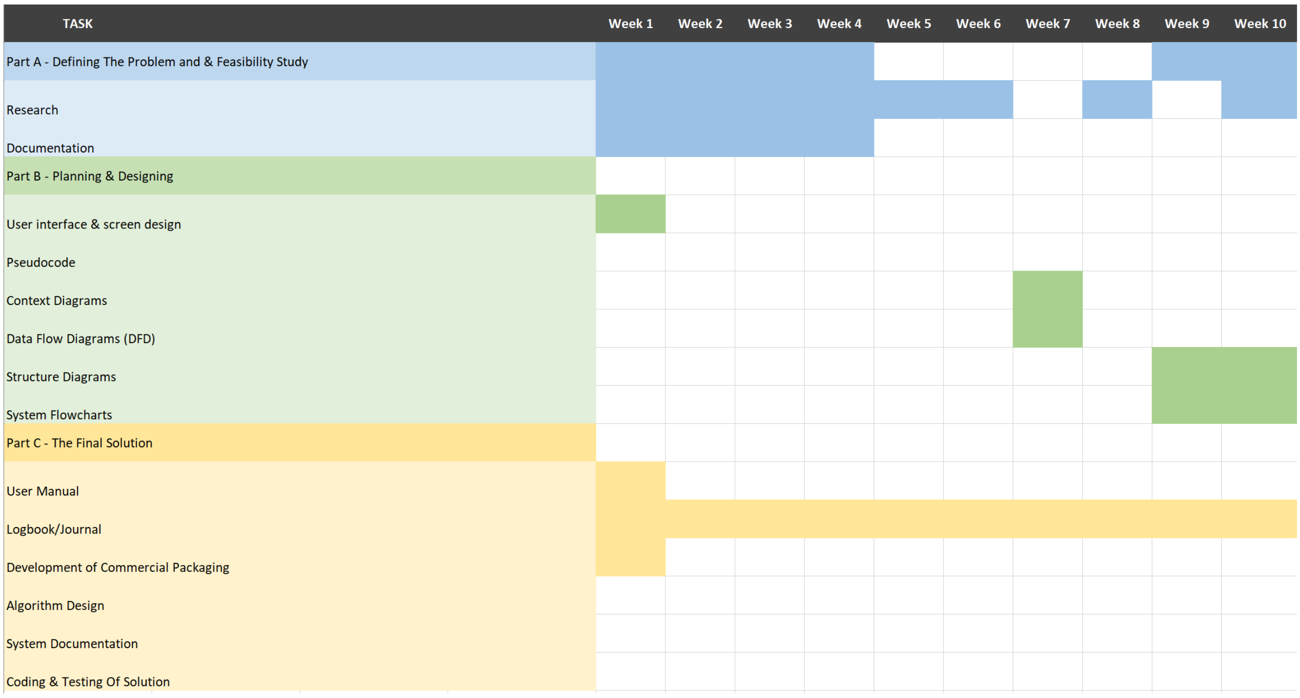
Project Plan

The time taken per module may vary, however, the recommended time taken for each module compared to the overall time frame should approach the proposed allocated time scheduling listed as below. This, in addition, is displayed further in depth in the project scheduling Gantt chart.

- 70% creation and implementation of the program
- 20% documentation and logging of work
- 5% bug testing and user feedback and user feedback data collection

Project Scheduling (Gantt Chart)

Each stage of the Sectrics projects development has been split up into several minor stages and the recommended allocated time slot for each objective to be completed by is as listed, this is in compliance with the required due dates. If this Gantt Chart is followed as proposed the software solution should be finished it in the time required. Several instances will be done with breaks throughout due to requiring a previous module to be finished or a reiteration process after becoming obsolete. Significant ones requiring many iterations include the proposed algorithm design as well as the Log Book/Journal being worked on throughout the entire duration of the project.



TASK	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10
Part A - Defining The Problem and & Feasibility Study	🟦	🟦	🟦	🟦					🟦	🟦
Research	🟦	🟦	🟦	🟦	🟦	🟦		🟦		🟦
Documentation	🟦	🟦	🟦	🟦						
Part B - Planning & Designing										
User interface & screen design	🟩									
Pseudocode										
Context Diagrams							🟩			
Data Flow Diagrams (DFD)							🟩			
Structure Diagrams									🟩	🟩
System Flowcharts									🟩	🟩
Part C - The Final Solution										
User Manual	🟨									
Logbook/Journal	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨
Development of Commercial Packaging	🟨									
Algorithm Design										
System Documentation										
Coding & Testing Of Solution										

Week 10	Week 11	Week 12	Week 13	Week 14	Week 15	Week 16	Week 17	Week 18	Week 19	Week 20
🟦										
🟦		🟦	🟦	🟦	🟦	🟦	🟦			
	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
🟩										
🟩										
🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨
	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨
	🟨	🟨	🟨			🟨	🟨	🟨	🟨	🟨

Week 20	Week 21	Week 22	Week 23	Week 24	Week 25	Week 26	Week 27	Week 28	Week 29	Week 30
🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩		
				🟨	🟨	🟨	🟨	🟨	🟨	🟨
🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨
								🟨	🟨	🟨
🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨
🟨	🟨				🟨	🟨	🟨	🟨	🟨	🟨
					🟨	🟨	🟨	🟨	🟨	🟨

Social & Ethical Considerations

For engineering software, software should be designed to avoid errors by disability and random errors to minimize the chance of causing errors. Due to this software being a piece of engineering software several special additional social & ethical issues arise when designing this piece of software. Unlike rockets or a life support system, this system will not directly because death or harm, however, indirectly faults with this software can lead to serious harm to companies and people alike both financially and physically potentially causing death due to the failure of a civil structure such as a bridge. Due to this when building the Sectrics program procedures & precautions should be instated prior to the usage of the software announcing that this software has limited liability & is not designed for commercial civil engineering applications but rather targeted towards students, due to limiting the liability we ensure that our software will not cause deliberate or indeliberate damage to civil structures and thus preventing injury due to the creation of our application.

Conversely rather than a negative benefit from the creation of the Sectrics program, inclusivity is also an issue. By making our software inclusive to all people including; vision impaired people & or color-blind people we maximize our market and ensure our software does not create any disadvantages for the disabled in addition, it helps a business further achieve their achieve corporate social responsibilities (CSR) thus helping them finish a requirement legally required by law. In addition, this adds a degree of marketing to our software by allowing a greater range of people to utilize our product and thus we also finish our ethical obligation to our users.

However, ultimately this program cannot solve all problems directly or indirectly caused by it. People with epilepsy or vision-impairing difficulties may still be affected by this program and may or may not be able to utilize it in a corporate environment therefore not fulfilling CSR in this rare case. In addition, this program may lead to extended sedentary behaviour due to it running on a computer possibly causing several health hazards due to this indirect consequence. Eye strain is also an additional concern for extended periods of use, especially at night due to bright contrasting colours can lead to eye strain generally being temporary. The solution generated to this issue is to take special consideration of this issue during the UI & UX designs and ensuring that the software was designed with a "dark mode" created allowing users to use it late at night with milder effects compared to using traditional bright white contrasting colours.

Furthermore ethically due to recent privacy issues arose by the Cambridge Analytica data breach, this software will contain no data collections method to prevent the possibility of our software being used illegally for data collection to be sold at a later date stealing the consumers privacy and in addition, of course, this software will also not contain any ransomware, spam ware or any sort of malicious executable within the software,

Additionally to users who are utilizing our software, it is essential that the user is able to use our software correctly and thus safely, if errors are to occur due to user incompetence that could have been prevented it is ultimately our fault, therefore to ensure errors caused by the misuse of our software is at a minimum the software must be easy to use. Manuals will be provided to the user, in addition, additional support will be provided via the GitHub page & website of this program.

Additional user confusion can also occur due to the inadequate design of user interfaces or the inconsistency of user interfaces. To further mitigate user error due to this, the user interface design has been specifically designed to ensure consistency to help orient and familiarize the user quickly. This is specifically important due to our emphasis on being a human-orientated system. Additionally, maintaining a consistent look also aesthetically aids in the appearance of the software.

Copyright, in addition, is also an important factor for the Sectrics program to be successfully completed. This program is to be distributed for free and is open source, additionally people are allowed to modify it and republish it however it must be open source falling under the same license as ours as well as an acknowledgement and crediting of us as a part of the creators, due to this the Sectrics program falls under the **Mozilla Public License 2.0.** The breach of this copyright may lead to legal action being acted upon to the infringer.

Refined Design & Modelling

User Interface & Screen Designs

The user interface & screen designs have been ergonomically chosen to be ergonomically stable for the user. Such factors contributing to the stability of ergonomics include the utilization and careful consideration of the colour pallet focusing on a primarily black theme reducing strain on the user's eyes, especially at night. Additionally, several other key considerations have been taken such as the primary focus on minimizing mouse movement to reduce repetitive strain injury.

The user starts the program in the main menu, from here he can click one of three options, the start button, the about button and the exit button. If the user chooses to exit the application asks for a confirmation and either terminates or returns back to the main menu. If the user presses about it will show additional information such as the creations name, GitHub and some additional notes until he presses back which will return him to the main menu.

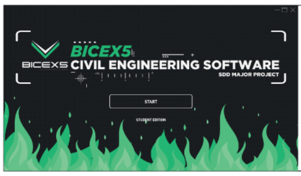
Alternatively, if the user clicks the start button the user will be transferred to the core section of the program where the user will start on the node's menu. From there the user enters the number of nodes he wants to enter and then confirms it after which he will be prompted to enter that many numbers of nodes, with its relative x & y coordinates. After the required node coordinates have been satisfied the member's menu will change colours to indicate to the user to go to the next screen.

After the use selects the member's menu the user will be prompted to enter the number of members they wish to enter and then will be prompted to enter the relative node "to connection" and relative node "from connection" to store the member's diagram. After which the next menu, supports will be highlighted prompting the user to progress to the next menu.

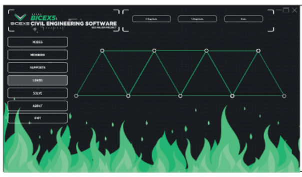
After the users selects supports, they again will be prompted to enter the number of supports they wish to enter, after which they will be prompted to enter the node number of the support and the support type going from either pin joint, roller joint or fixed support. After the user enters the correct number of supports the next menu loads will be highlighted then again prompting them to continue to the next menu of loads.

At the menu loads, the user will once again be prompted asking the user how many loads they would like to attach to the bridge, after which the user will select a specific node and then enter in a relative X & Y load with both up and right being positive and conversely down and left being negative. After the user enters the specified number of nodes the next menu, solve will be highlighted again but in a different colour, being green indicating to them that this is the final screen.

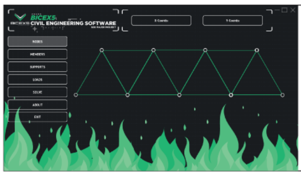
When the user decides to click onto solve the menu will change to one displaying the bridge constructed with its relative nodes, members as well as support locations. Additionally, the program will solve for both tensile and compressive forces in each member and display them in either red or blue signifying tension or compression forces in each member. Additionally, a table will be shown in the left displaying the relevant node locations, forces applied in each member as well as displacement and other miscellaneous properties.



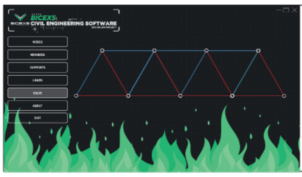
NAME: BICEX CIVIL ENGINEERING SOFTWARE
PROGRAMMER: SHAAN KHAN
DATE: 14/12/2018
SCREEN: MAIN MENU
NOTES: Main menu from the initialising of the program prompting the user to start



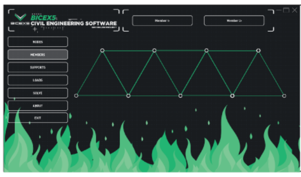
NAME: BICEX CIVIL ENGINEERING SOFTWARE
PROGRAMMER: SHAAN KHAN
DATE: 14/12/2018
SCREEN: LOADS
NOTES: Main menu where the loads are added in the x & y magnitudes on a selected node.



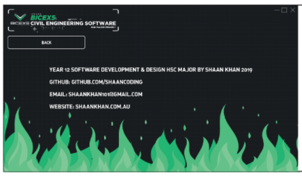
NAME: BICEX CIVIL ENGINEERING SOFTWARE
PROGRAMMER: SHAAN KHAN
DATE: 14/12/2018
SCREEN: NODES
NOTES: Main screen where determine truss analysis takes place, the addition of the positions of nodes to be added or removed is shown here



NAME: BICEX CIVIL ENGINEERING SOFTWARE
PROGRAMMER: SHAAN KHAN
DATE: 14/12/2018
SCREEN: SOLVE
NOTES: Solving of the stress / strain in each member additionally shows reaction forces



NAME: BICEX CIVIL ENGINEERING SOFTWARE
PROGRAMMER: SHAAN KHAN
DATE: 14/12/2018
SCREEN: MEMBERS
NOTES: Main menu from where the member connections are established, telling the system where each node is connected



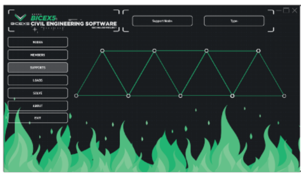
YEAR 12 SOFTWARE DEVELOPMENT & DESIGN HSC MAJOR BY SHAAN KHAN 2019

GITHUB: GITHUB.COM/SHAANCODING

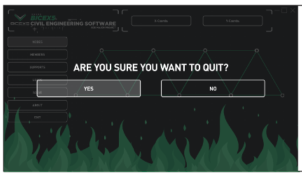
EMAIL: SHAANKHAN101@GMAIL.COM

WEBSITE: SHAANKHAN.COM.AU

NAME: BICEX CIVIL ENGINEERING SOFTWARE
PROGRAMMER: SHAAN KHAN
DATE: 14/12/2018
SCREEN: ABOUT SCREEN
NOTES: Misc. about screen giving detail of this project in addition to contact information if required



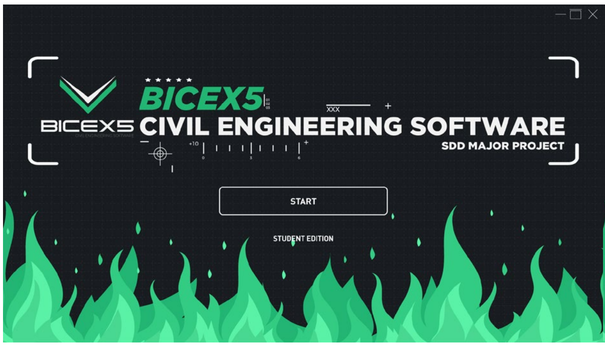
NAME: BICEX CIVIL ENGINEERING SOFTWARE
PROGRAMMER: SHAAN KHAN
DATE: 14/12/2018
SCREEN: SUPPORTS
NOTES: Main menu where the support locations & support types are added

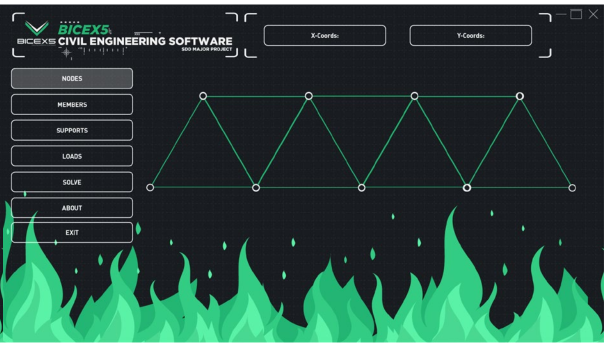


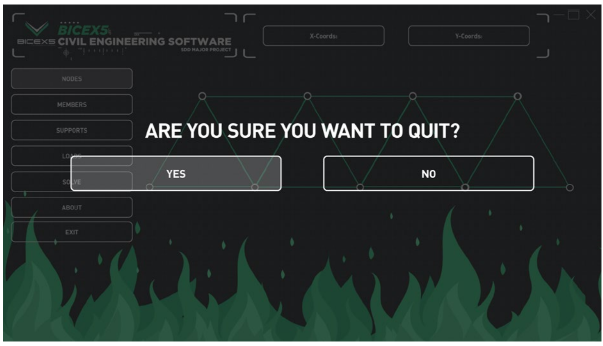
NAME: BICEX CIVIL ENGINEERING SOFTWARE
PROGRAMMER: SHAAN KHAN
DATE: 14/12/2018
SCREEN: MAIN SCREEN
NOTES: Exit conformation overlaying confirming to the user if he/she wishes to exit, if true the application terminates else it returns to MAIN SCREEN

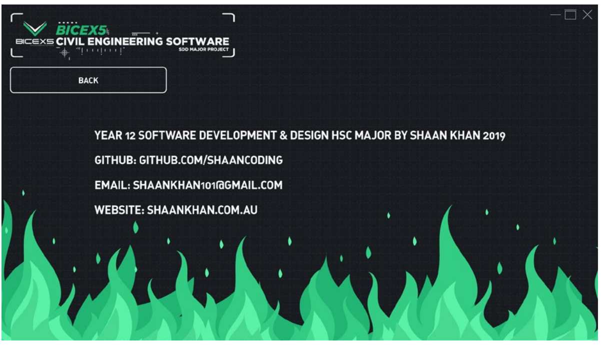
Story Board

Menu Functions






ARE YOU SURE YOU WANT TO QUIT?

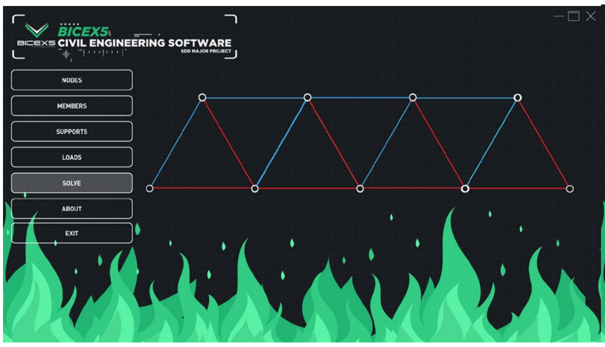


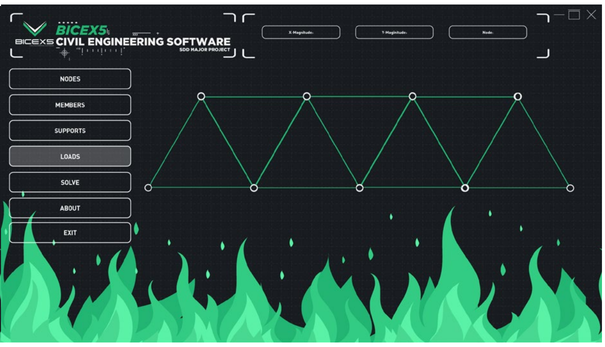
YEAR 12 SOFTWARE DEVELOPMENT & DESIGN HSC MAJOR BY SHAAN KHAN 2019

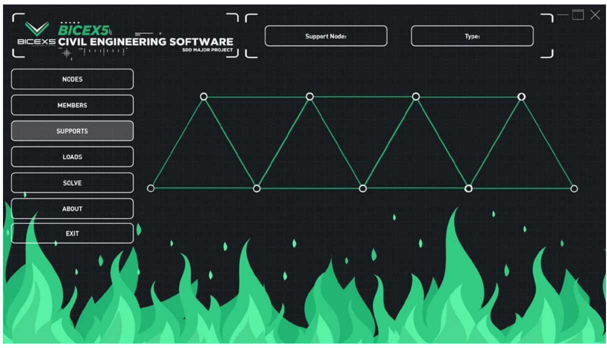
GITHUB: GITHUB.COM/SHAANCODING

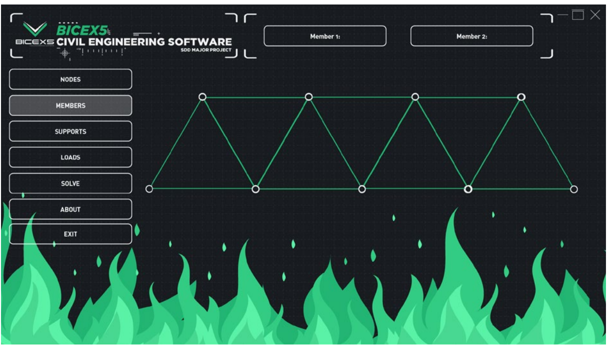
EMAIL: SHAANKHAN101@GMAIL.COM

WEBSITE: SHAANKHAN.COM.AU



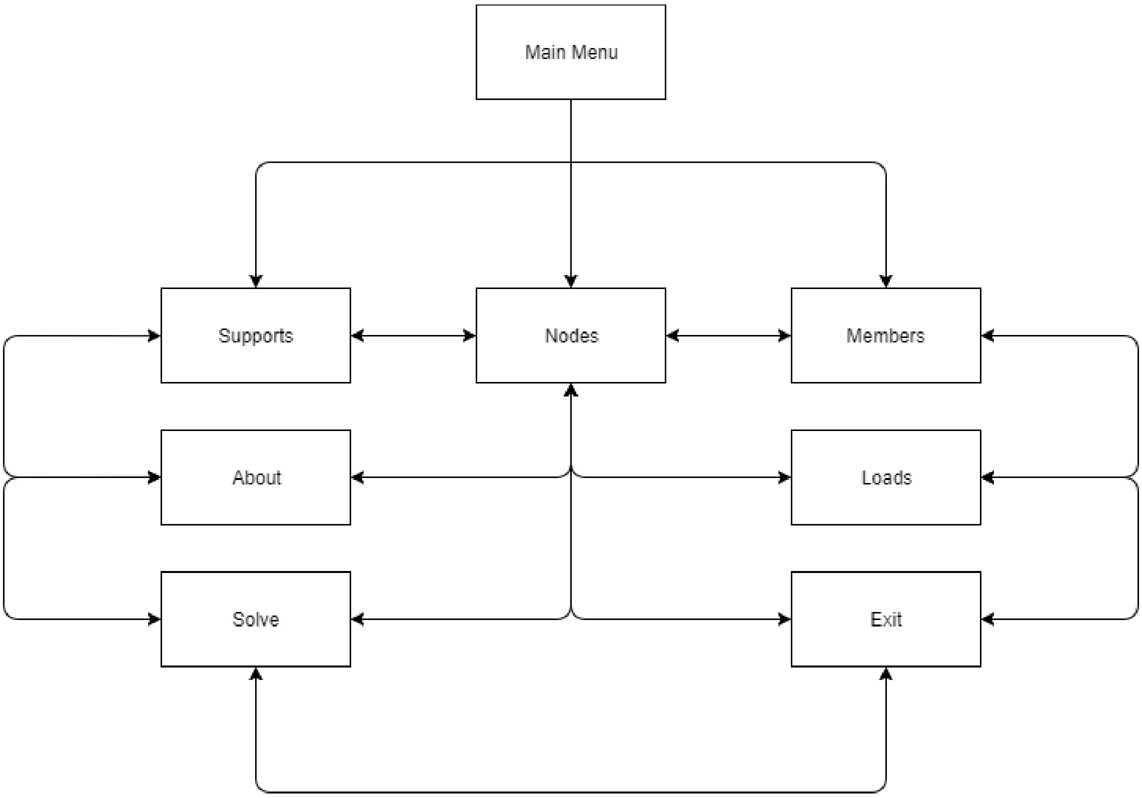






Networked Storyboard

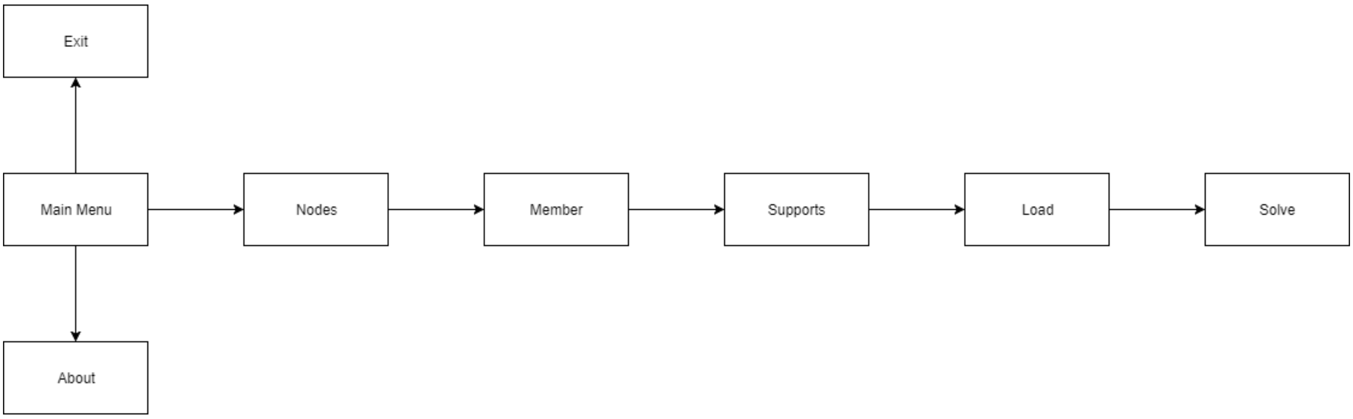
The **Main Menu** is the starting module of the program after the user presses start, he/she will be directed to the node's menu, after which all the menus are interconnected with each other, excluding the main menu where it is no longer accessible unless the program is restarted. The user is able to jump from: **Nodes, Supports, About, Solve, Members, Loads & the Exit Menus** as they please.



Linear Storyboard

An alternative representation of how the user should be using the program is as follows, in the previous diagram it is presented that they can return to any module as they please, but the software is intended to go in this linear hierarchal order.

The **Main menu** once again is the start location, where the user can view **About** or **Exit** and then return to the **Main Menu** or the user can start on **Nodes** and add the required nodes giving the coordinates of each beams turning point then to **Member** where the user chooses which **Node** connects to which **Node** afterwards the user can go to **Supports** and select a support type at a selected **Node** and go to **Loads** where the user can attach a required X & Y load on a specific **Node** then is finally able to go to **Solve** where it outputs the solved location. After which the user can terminate the program as its objective is finished.



Input Process Output (IPO) Diagrams

The Input Process Output Diagram (IPO) displays relevant information regarding the user input process and output. This shows what the user enters in each menu and what response will be given. This IPO will be divided into several key modules corresponding to each key menu. This will help assist in understanding and describing the scope of the project in addition to what the purpose is of each menu.

Main Menu

Input	Process	Output
Left mouse clicks	Intercept which menu to load	Load the desired menu

Tutorial

Input	Process	Output
Left mouse clicks	Respond to mouse click being either forward in the tutorial or backwards	Displays the appropriate screen related to the tutorial when the selected screen is selected

Credits

Input	Process	Output
Left mouse clicks	Loads credit menu	Credits

Quit

Input	Process	Output
Left mouse clicks	Quits the application	Closes the application

Node

Input	Process	Output
Keyboard & Mouse input of X & Y coordinates left click confirming or removing selected node	Adds or removes node via user input	Displays node in table format displaying appropriate X & Y coordinates. In addition, draws a bridge respectively to the X & Y coordinates.

Members

Input	Process	Output
Keyboard & Mouse input of Member1 & Member2 left click confirming or removing selected node	Adds or removes members selected	Displays members in table format displaying it in appropriate node 1 and node 2 formats. In addition, draws a bridge respectively to the Member1 & Member2 coordinates.

Supports

Input	Process	Output
Keyboard & Mouse Input of Support nodes left click confirming or removing the selected node	Adds or removes node selected with appropriate node type	Displays supports in a table format displaying which node it is attached to and the support type. In addition, draws a bridge respectively to the support location & support type.

Loads

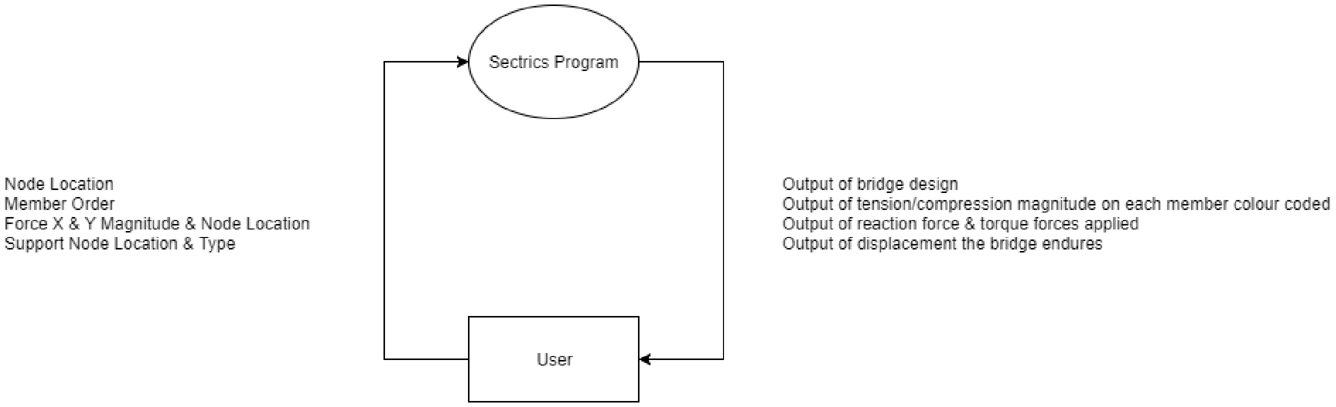
Input	Process	Output
Keyboard & Mouse Input of Loads. Loads are added by left click and then typing the X & Y values.	Adds or removes selected load with the appropriate node location	Displays the load in a table format displaying which node it is attached to and the x & y values. In addition, it draws a force vector with its respective magnitude and direction on the bridge.

Solve

Input	Process	Output
Mouse input activating the Solve function	Solves the bridge changing each beam element to blue, red or green signifying tension, compression as well as no force respectively.	Shows tension / compressive force in each member shown in blue, green or red signifying the force being compressive, tension or none. In addition, displays the reaction force as well as reaction moment to keep the bridge in static equilibrium

Context Diagram

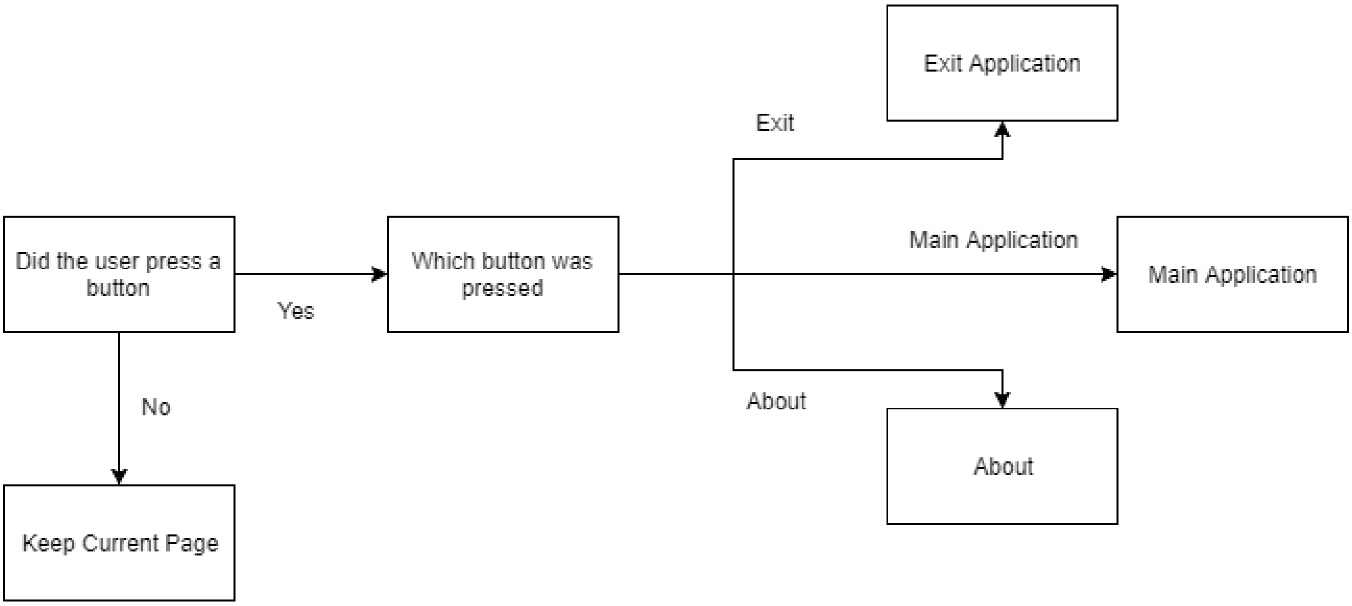
The purpose of this context diagram is to define the boundary between the system and its environment showing the entities that interact with it. In Sectrics there are no external entities acting with the software excluding the user itself. The user enters data regarding the bridge including; node location, member order, forces with both x & y magnitudes and support location & support type whilst the program outputs a drawing of the bridge showing both magnitude and direction of forces applied in each member, being tensile or compressive forces, and then colour coding it. In addition, the program outputs and shows the displacement diagram of the bridge, showing by how much the bridge moves by from its original position due to loads being applied.


Sectrics Program
Node Location
Member Order
Force X & Y Magnitude & Node Location
Support Node Location & Type
Output of bridge design
Output of tension/compression magnitude on each member colour coded
Output of reaction force & torque forces applied
Output of displacement the bridge endures
User

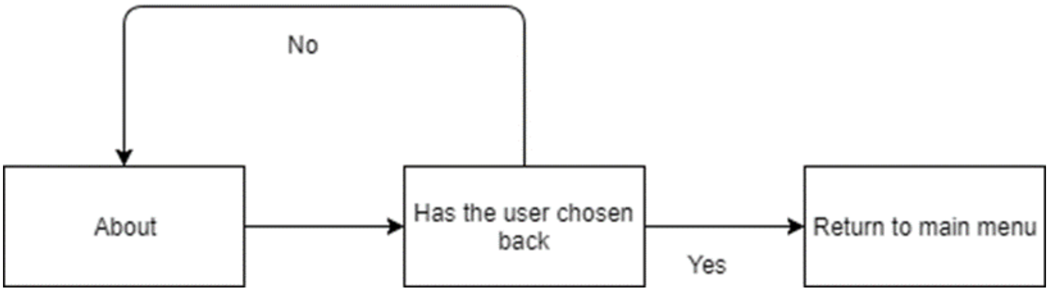
Decision Tree

A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences. Using a tree-like the model we can show all possible options the user can do in the program and therefore can ensure that they are no conditions that are left uncoded leading to errors. In addition, this diagram also ensures that our program also makes logical sense and that no logical method will lead to a dead end making the user unable to continue the program. A decision tree has been created for each key menu including the: Main Menu, About Screen, Exit Screen & Main Application Screen, where the entire program plays.

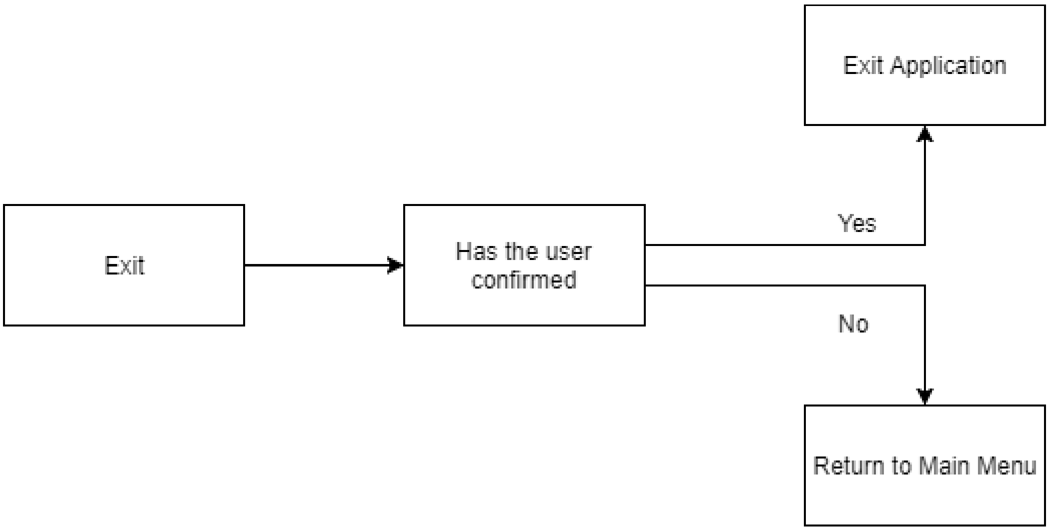
Main Menu



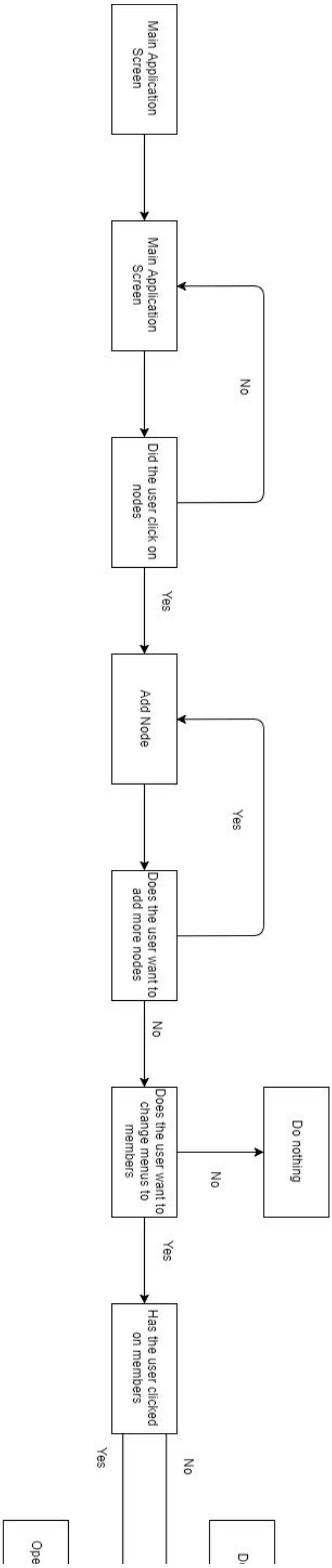
About Screen



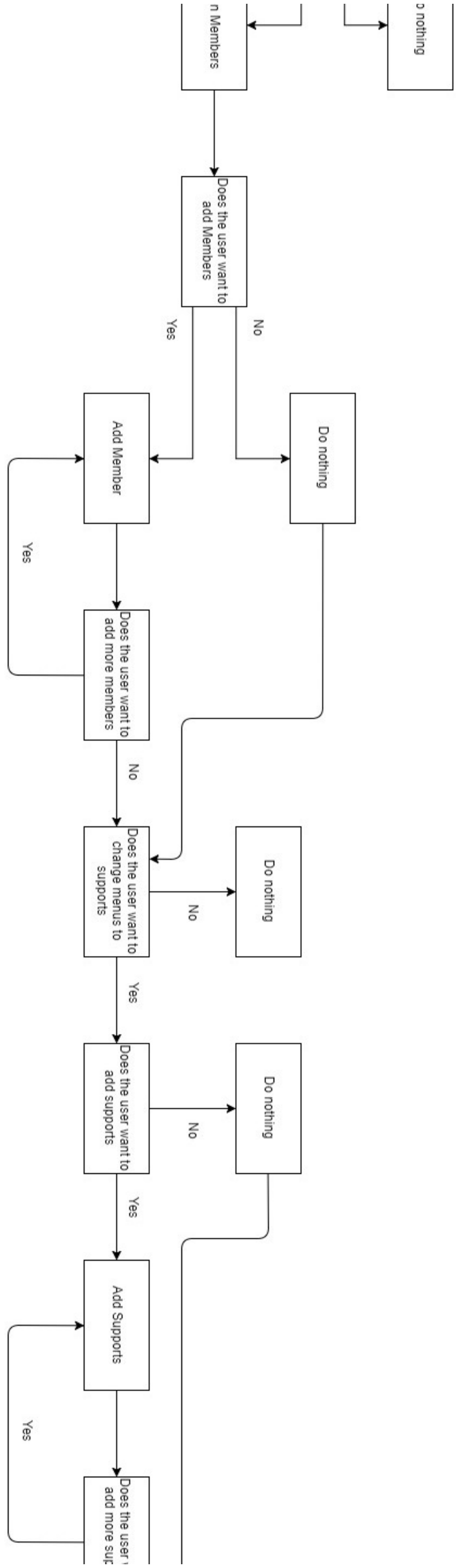
Exit Screen

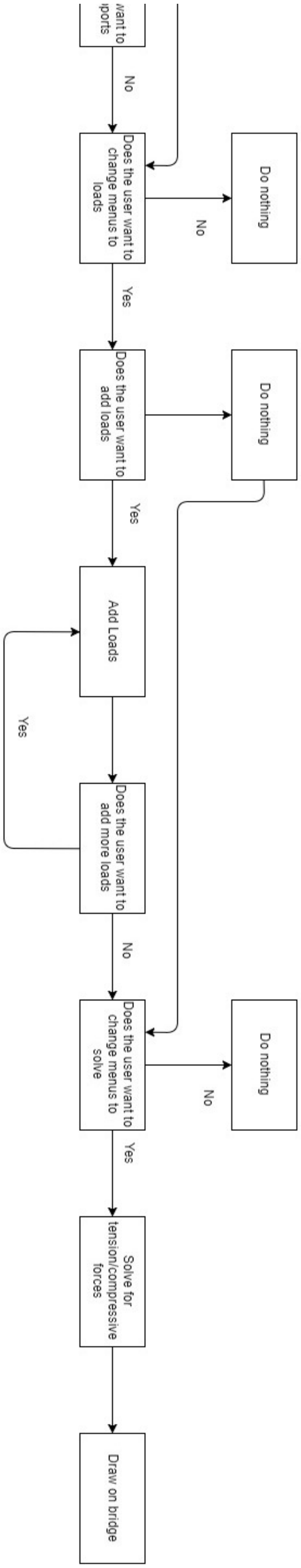


Main Application Screen



Main Application Screen
Main Application Screen
No
Did the user click on nodes
Yes
Add Node
Yes
Does the user want to add more nodes
No
Does the user want to change menus to members
No
Do nothing
Yes
Has the user clicked on members
Yes
No
Ope
D

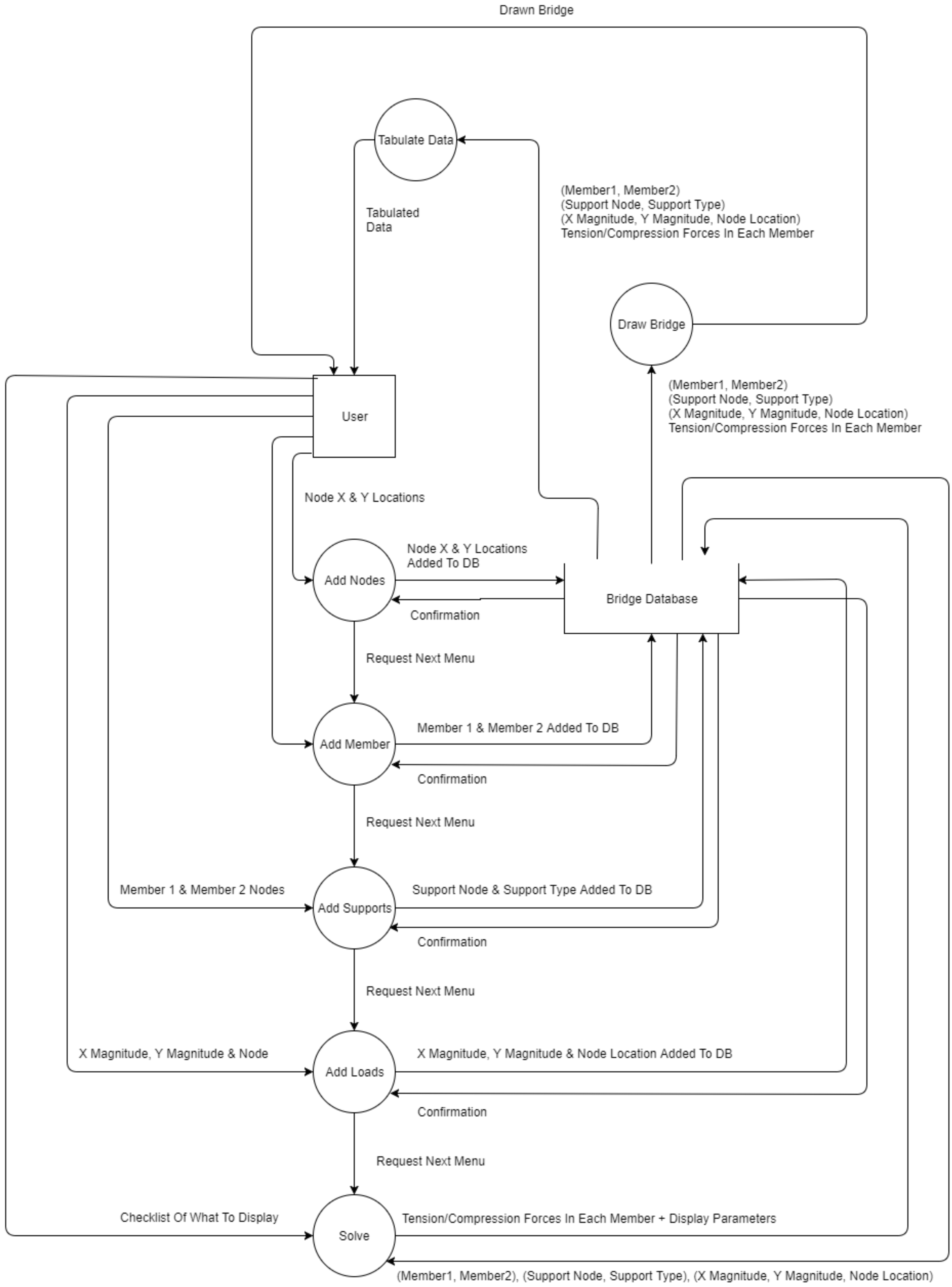

n Members
o nothing
Does the user want to add Members
Yes
No
Add Member
Do nothing
Yes
Does the user want to add more members
No
Does the user want to change menus to supports
No
Do nothing
Yes
Does the user want to add supports
No
Do nothing
Yes
Add Supports
Yes
Does the user add more sup



vant to
ıports
No
Does the user want to change menus to loads
No
Do nothing
Yes
Does the user want to add loads
Do nothing
Yes
Add Loads
Yes
Does the user want to add more loads
No
Does the user want to change menus to solve
No
Do nothing
Yes
Solve for tension/compressive forces
Draw on bridge

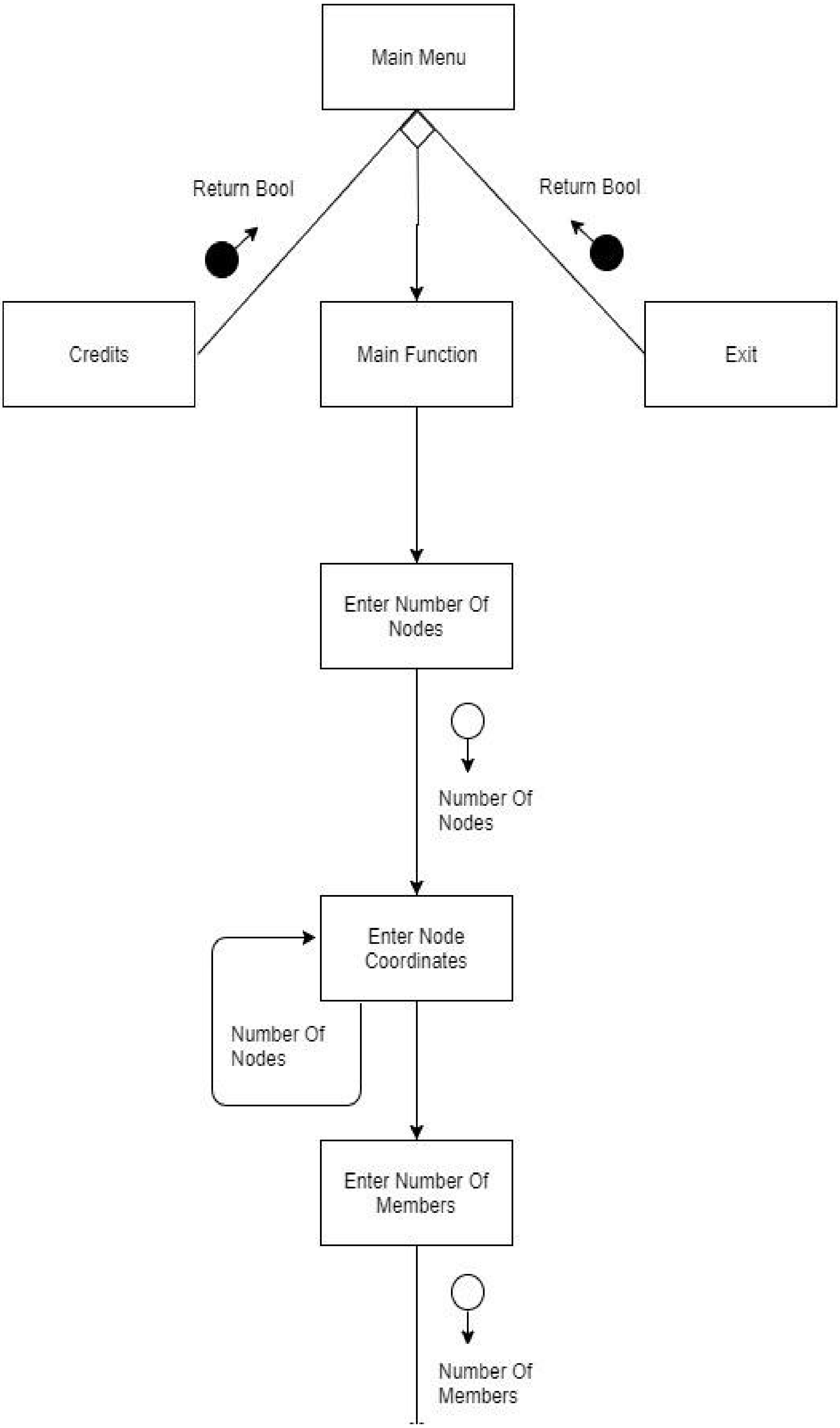
Data Flow Diagrams (DFD)

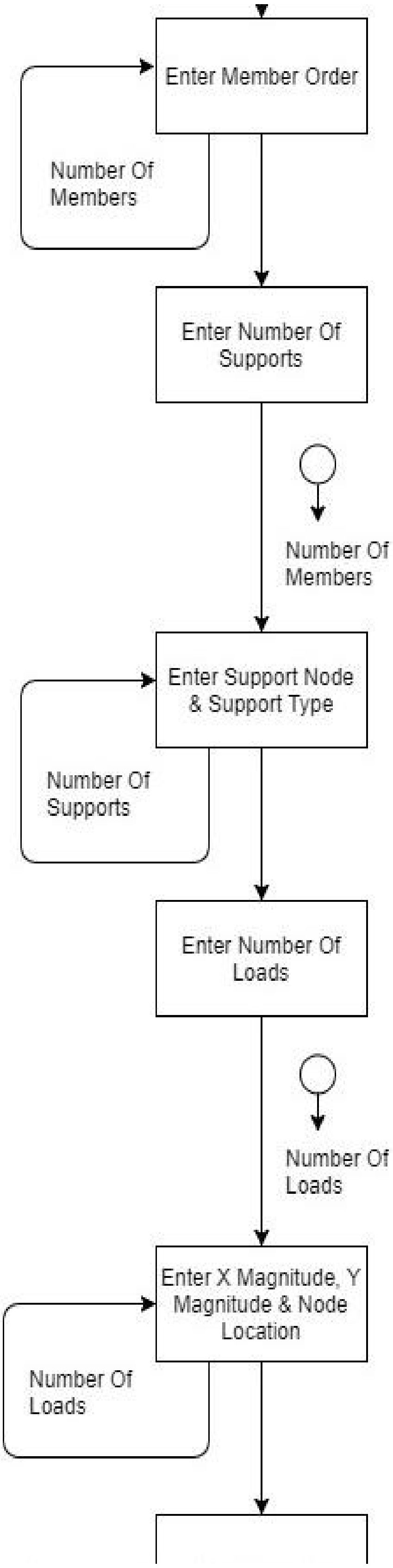
The purpose of a Data Flow Diagram is to show a graphical representation of the flow of data through an information system. This data flow shows what kind of information will be inputted and what information will be outputted through the system as well as what operations will take place.

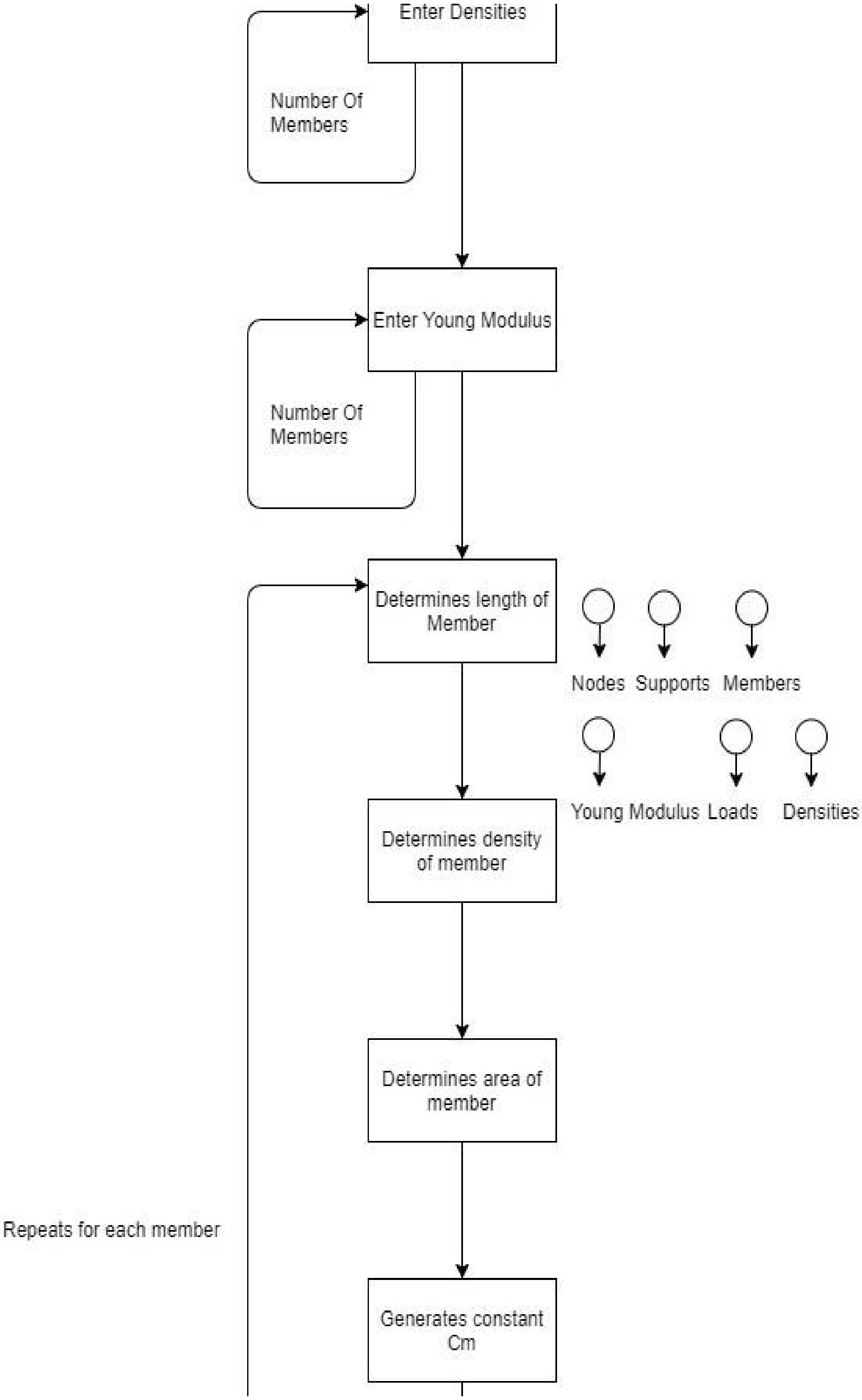

Drawn Bridge
Tabulate Data
Tabulated Data
(Member1, Member2)
(Support Node, Support Type)
(X Magnitude, Y Magnitude, Node Location)
Tension/Compression Forces In Each Member
Draw Bridge
User
(Member1, Member2)
(Support Node, Support Type)
(X Magnitude, Y Magnitude, Node Location)
Tension/Compression Forces In Each Member
Node X & Y Locations
Node X & Y Locations Added To DB
Add Nodes
Bridge Database
Confirmation
Request Next Menu
Member 1 & Member 2 Added To DB
Add Member
Confirmation
Request Next Menu
Member 1 & Member 2 Nodes
Add Supports
Support Node & Support Type Added To DB
Confirmation
Request Next Menu
X Magnitude, Y Magnitude & Node
Add Loads
X Magnitude, Y Magnitude & Node Location Added To DB
Confirmation
Request Next Menu
Checklist Of What To Display
Solve
Tension/Compression Forces In Each Member + Display Parameters
(Member1, Member2), (Support Node, Support Type), (X Magnitude, Y Magnitude, Node Location)

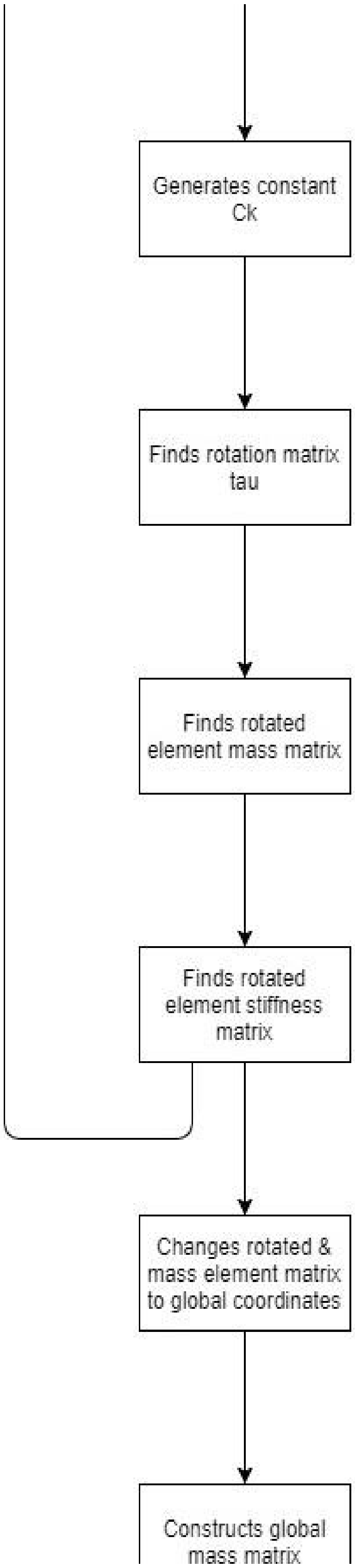
Structure Chart

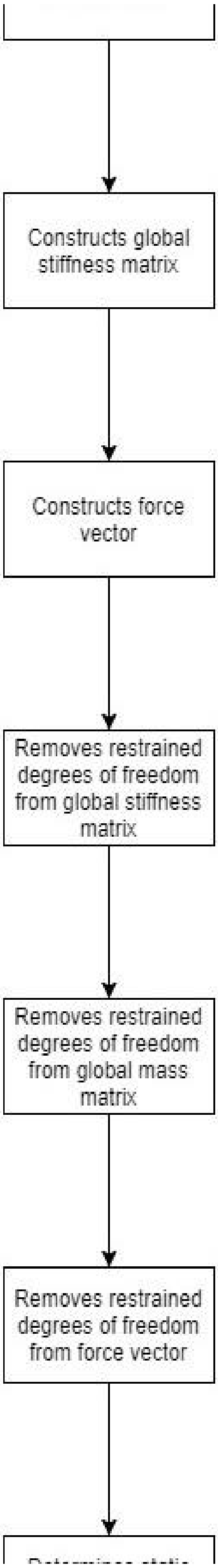
A structure chart is a chart which shows the breakdown of a system to its lowest manageable levels. This is similar to the tree diagram but in addition, also highlights and displays which variables will be passed to other modules. The key aspects of this is the utilization of control variables such as bools as well as the passing of normal variables between modules. The most important variables passed through the program includes; Nodes, Supports, Members, Young Modulus, Loads & Densities.

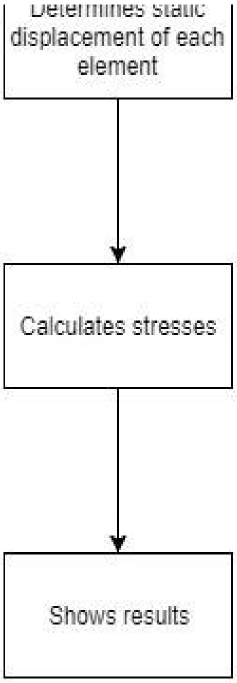



Enter Member Order
Number Of Members
Enter Number Of Supports
Number Of Members
Enter Support Node & Support Type
Number Of Supports
Enter Number Of Loads
Number Of Loads
Enter X Magnitude, Y Magnitude & Node Location
Number Of Loads


Enter Densities
Number Of Members
Enter Young Modulus
Number Of Members
Determines length of Member
Nodes
Supports
Members
Young Modulus
Loads
Densities
Determines density of member
Determines area of member
Repeats for each member
Generates constant Cm

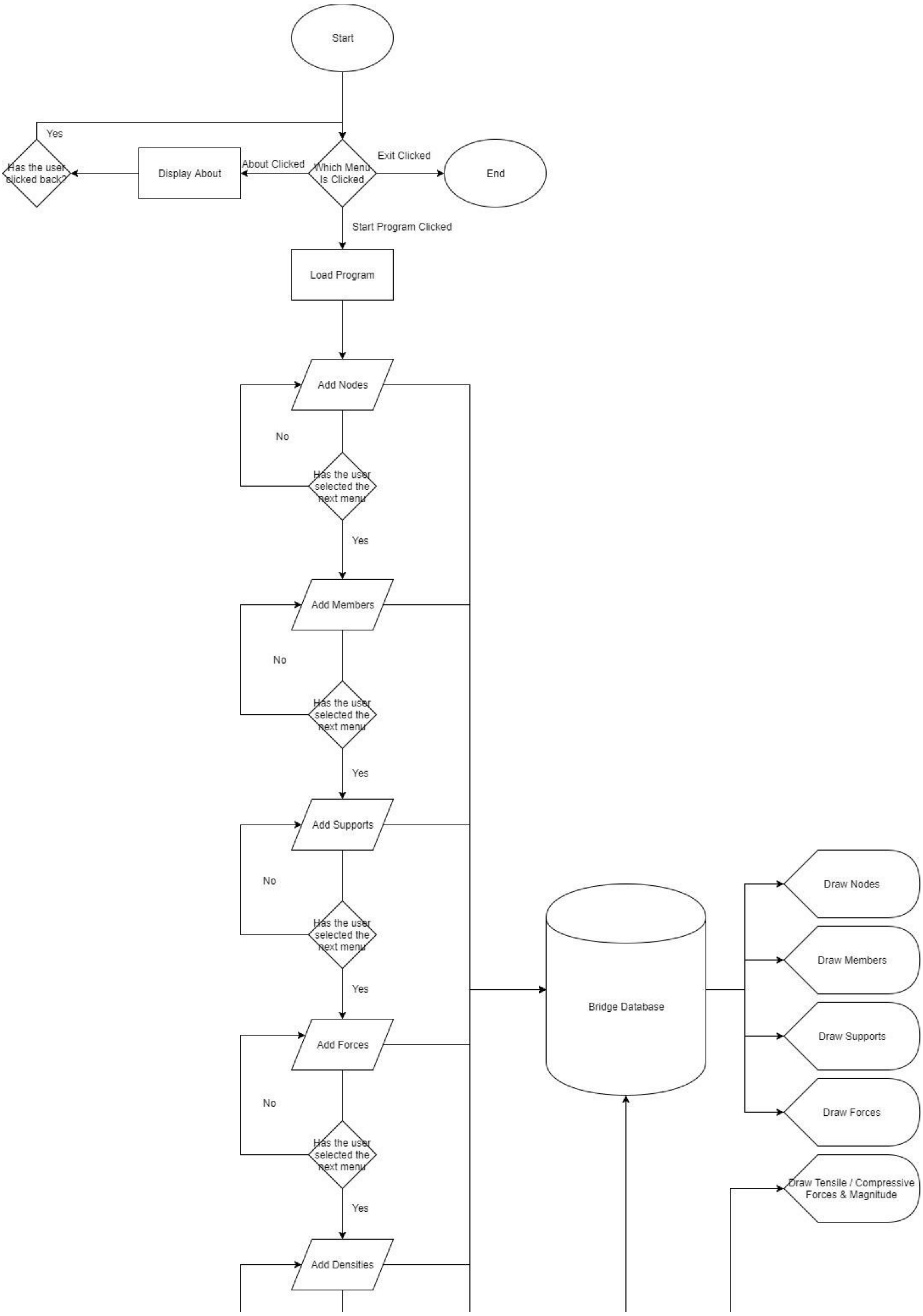

Generates constant Ck
Finds rotation matrix tau
Finds rotated element mass matrix
Finds rotated element stiffness matrix
Changes rotated & mass element matrix to global coordinates
Constructs global mass matrix

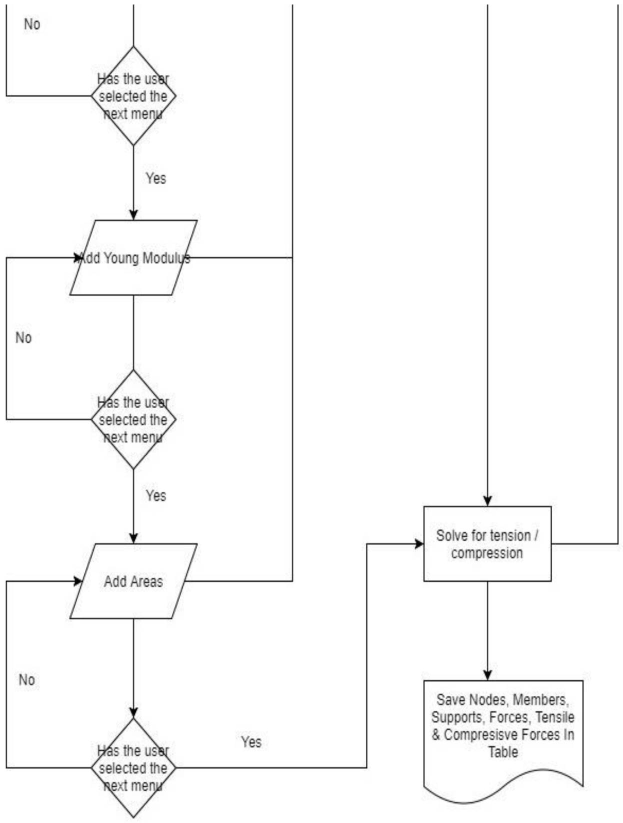

Constructs global stiffness matrix
Constructs force vector
Removes restrained degrees of freedom from global stiffness matrix
Removes restrained degrees of freedom from global mass matrix
Removes restrained degrees of freedom from force vector
Determines static


Determines static displacement of each element
Calculates stresses
Shows results

System Flowcharts

The purpose of a system flowchart is to display how data flows in a system and how decisions are made to control events. This diagram is similar to data flow diagrams however due to data flow diagrams not showing the flow of data based on conditions the diagram is limited to its functionality. For a better diagram to show the specific flow of data, in addition to decision trees, system flowcharts show the best of both worlds.




No
Has the user selected the next menu
Yes
Add Young Modulus
No
Has the user selected the next menu
Yes
Add Areas
No
Has the user selected the next menu
Yes
Solve for tension / compression
Save Nodes, Members, Supports, Forces, Tensile & Compresisve Forces In Table

Software Requirements

This application was coded in C# with Windows 10 in mind. As this program was made specifically for Windows 10 in mind some of the libraries may not exist on other platforms and thus may lead to other platforms containing incompatibility issues. Due to this to run the Sectrics program windows 10 is a required pre-requisite software.

Hardware Requirements

Minimum Specifications

The minimum hardware specifications for this application to run requires the minimum hardware requirements for Windows 10 to run. Due to this the minimum hardware requirement are:

- **Processor:** 1 gigahertz (GHz) or faster processor or SoC
- **RAM:** 1 gigabyte (GB) for 32-bit or 2 GB for 64-bit
- **Hard disk space:** 16 GB for 32-bit OS 20 GB for 64-bit OS
- **Graphics card:** DirectX 9 or later with WDDM 1.0 driver
- **Display:** 800x600

Recommended Specifications

Depending on the user's bridge requirements the specifications of the software varies, for commercial applications such as calculating the forces on a real world bridge with hundreds to thousands of nodes and members, the utilization of more powerful hardware is required, for programs requiring such processing power the recommended specifications are:

- **Processor:** Intel® Core™ i7-6700K @ 4.4GHz-Hi-Perf
- **Memory:** 16GB DDR4
- **Graphics:** NVIDIA® GeForce® GTX 1080 OC 8GB GDDR5X VRAM PCIe
- **Storage:** 512GB Solid State Drive NVMe Samsung® 950 PRO M.2 PCIe 3.0 x4 2500/1500MB/s
- **Display:** 1920x1080

Data Dictionary

A data dictionary is a collection of variable names, variable types and an example of its purpose. It shows the relationship between type, origin, usage and format. To ensure that our program is designed correctly the allocation of data types, data variables and the relationship between data must be displayed. To aid with this operation a data dictionary has been chosen for this purpose.

Direction Cosine

Data Name	Data Type	Example	Description	Data Structure
directionCosine	Double	0.481	Determines the direction cosine for a member	A double is selected by determining the dot product of both vectors and then dividing it by the distance formula

RotationMatrix

Data Name	Data Type	Example	Description	Data Structure
X_proj	Double	0.68	Finds the x projection as a double value	A double is selected using the direction cosine with the parameters deltaPoint and the x_axis
Y_proj	Double	0.68	Finds the y projection as a double value	A double is selected using the direction cosine with the parameters deltaPoint and the y_axis
rotationMatrix	Double[,]	[1,2,3,4] [5,6,7,8] [9,10,11,12] [13,14,15,16]	Determines the local rotation matrix for an element	A double[,] is selected by using the formula [x_proj,y_proj,0,0] [0,0,x_proj,y_proj]

Main

Data Name	Data Type	Example	Description	Data Structure
X_axis	Double[]	{1,0}	A predefined double variable for the x_axis coefficients	The double[] is a constant using the predefined numbers of [1,0]
Y_axis	Double[]	{0,1}	A predefined double variable for the y_axis coefficients	The double[] is a constant using the predefined numbers of [0,1]

Nodes	Double[,]	{0,10}
		{0,0}
		{10,5}
			Stores the nodes x & y location in a array list	The Double[,] is defined using user input to define the x coordinate and y coordinate of each node entered, each node is in array format

degreesOfFreedom	Int[,]	{1,2}
		{3,4}
		{5,6}
			Stores the degrees of freedom of each node in an array list	The Int[,] is defined automatically as an X & Y value increasing incrementally by one for each node location

Members	Int[,]	{0,2}
		{1,2}
			Stores the node array location for each node that connects to another	The Int[,] is defined using user inputs to define the "node from" connection and "node to" connection helping store how each member of the truss is made

restrainedDegreesOfFreedom	Int[]	{1,2,3,4}	Stores the restrainedDegreeOfFreedom locations indicating where the axis is fixed, such as a fixed support or roller joint.	The Int[] is defined using user inputs when they enter the type of support they wish to have; the support is then converted into their respective degreesOfFreedom values and then is converted to its restrainedDegreeOfFreedom value and is added here

Forces	Double[,]	{0,0}
		{0,0}
		{0,-200}
			Stores the Forces of each node in its respective X & Y magnitudes with its integer value corresponding to its respective node.	The Double[,] is defined using user inputs selecting which node they would put a force on then the x magnitude then y magnitude. The other nodes in this Double[,] is pre-populated with zeros.

Densities	Double[]	{0.284} {0.284}	Stores the Densities of each member respective to its Member integer	The Double[] is defined using user inputs typing in the density value for each member.

Areas	Double[]	{1.0}
		{2.0}
			Stores the Areas of each member respective to its Member integer	The Double[] is defined using user inputs typing in the area for each member

M	Double[,]	{0.85.-0.43,0,0,0.43,-0.21} {0.85.-0.43,0,0,0.43,-0.21} {0.85.-0.43,0,0,0.43,-0.21} {0.85.-0.43,0,0,0.43,-0.21} {0.85.-0.43,0,0,0.43,-0.21} {0.85.-0.43,0,0,0.43,-0.21}	Stores the global mass matrix of the entire bridge	The global mass matrix is defined using the element mass matrix and the constant Cm $Cm * M_{rG}$
K	Double[,]	{0.85.-0.43,0,0,0.43,-0.21} {0.85.-0.43,0,0,0.43,-0.21} {0.85.-0.43,0,0,0.43,-0.21} {0.85.-0.43,0,0,0.43,-0.21} {0.85.-0.43,0,0,0.43,-0.21} {0.85.-0.43,0,0,0.43,-0.21}	Stores the global stiffness matrix of the entire bridge	The global stiffness matrix is defined using the element mass matrix and the constant Ck $Ck * K_{rG}$
F	Double[]	{0,0,-200,0,0}	Stores the Force Double[,] to a vector	Converts the Force double to a vector
DOFS	Int[]	{0,0,10,5}	Stores the degreesOfFreedom for each member to a vector	Converts the degreesOfFreedom for each member to a vector
Length	Double	{1.78}	Stores the length of each member	Determines the length of each member using the distance formula i.e. $\sqrt{(x2-x1)^2+(y2-y1)^2}$
Cm	Double	{1.89}	Stores a constant coefficient for the element mass matrix	Determines the constant coefficient for the element mass matrix by using the formula $Cm = \frac{density[i] * length}{6}$
Ck	Double	{1.89}	Stores a constant coefficient for the element stiffness matrix	Determines the constant coefficient for the element stiffness matrix by using the formula $Ck = \frac{stiffness[i] * area}{length}$

m	Double[,]	{2,1}
		{1,2}
			Stores a constant matrix for the member	Determines the constant matrix by using the predefined coefficients of:
				{2,1}
				{1,2}

k	Double[,]	{1,-1}
		{-1,1}
			Stores a constant matrix for the stiffness	Determines the constant matrix by using the predefined coefficients of:
				{1,-1}
				{-1,1}

m_r	Double[,]	{1.6,-0.8,0.8,-0,4}
		{-0.8,0.4,-0.4,0.2}
		{0.8,-0.4,1.6,-0.8}
		{-0.4,0.2,-0.8,-.4}
			Stores the element mass rotation matrix	Determines the matrix by using the formula: $transpose(tau).m.tau$

k_r	Double[,]	{1.6,-0.8,0.8,-0,4}
		{-0.8,0.4,-0.4,0.2}
		{0.8,-0.4,1.6,-0.8}
		{-0.4,0.2,-0.8,-.4}
			Stores the element stiffness rotation matrix	Determines the matrix by using the formula: $transpose(tau).k.tau$

M_rG	Double[,]	{1.6,-0.8,0.8,-0,4}
		{-0.8,0.4,-0.4,0.2}
		{0.8,-0.4,1.6,-0.8}
		{-0.4,0.2,-0.8,-.4}
			Stores the global stiffness rotation matrix	Determines the matrix by using the formula $transpose(B).m_R.B$

K_rG	Double[,]	{1.6,-0.8,0.8,-0,4}
		{-0.8,0.4,-0.4,0.2}
		{0.8,-0.4,1.6,-0.8}
		{-0.4,0.2,-0.8,-.4}
			Stores the global stiffness rotation matrix	Determines the matrix by using the formula $transpose(B).K_R.B$

Test Data

To ensure boundary conditions are satisfied and that no errors occur with the incorrect entering of data a Test Data table has been created. This aids by checking the boundary conditions of where errors may occur. Such as checking positive numbers, zeros and negative all possible user input mistakes have been removed. Due to the restriction of numbers only being able to be entered by the system, the test data for the user entering a character or string is not required as it is not possible.

Test Data	Variable Tested	Output Results	Reason For Inclusion

{0,10}
{0,0}
{10,5}
	Nodes	Program functions as normal	This data is entered by the user and therefore is not automatic and is therefore not already pre-handled to avoid errors.

{-0,-10}
{-0,-0}
{-10,-5}
	Nodes	Program functions as normal	This data is entered by the user and therefore is not automatic and is therefore not already pre-handled to avoid errors.

{0,2}
{1,2}
	Members	Program functions as normal	This data is entered by the user and therefore is not automatic and is therefore not already pre-handled to avoid errors.

{-0,-2}
{-1,-2}
	Members	The program will not function as normal as a members node cannot be less than zero	This data is entered by the user and therefore is not automatic and is therefore not already pre-handled to avoid errors.

{0,0}
{0,0}
{0,-200}
	Forces	Program functions as normal	This data is entered by the user and therefore is not automatic and is therefore not already pre-handled to avoid errors.

{-0,-0}
{-0,-0}
{-0,-200}
	Forces	Program functions as normal	This data is entered by the user and therefore is not automatic and is therefore not already pre-handled to avoid errors.

{0.284} {0.284}	Densities	Program functions as normal	This data is entered by the user and therefore is not automatic and is therefore not already pre-handled to avoid errors.
{-0.284} {-0.284}	Densities	The program will not function as normal as density cannot be less than or equal to zero	This data is entered by the user and therefore is not automatic and is therefore not already pre-handled to avoid errors.

Logbook/Journal

Date & Time:	Difficulty:	Task Achieved:
Day/Month/Year	Difficulty (1-5)	Task Achieved – Challenge Encountered + Solution – Ideas & Thoughts – Reflection On Progress – Upcoming Tasks – (Bibliography)
09/25/2018	1	After drafting my first ideas for a major project I've decided to make a student study planner after looking at alternatives I've found that nothing integrates everything into an all in one bundle targeted towards HSC students by making an online based software students will be able to keep up to date with their work and not lose any progress if their computer or phone gets corrupted due to this idea I've decided to build a website, therefore, I've started to learn PHP, JS, MySQL as well as revising HTML & CSS
10/29/2018	2	Continued progress on learning PHP, JS & MySQL. After revision on HTML & CSS, I've decided to host a little static website to tinker on during my free time at shaancoding.com. Additionally, my HTML & CSS revision is finished.
10/30/2018	2	After getting semi-proficient in PHP I've decided to further work on the documentation of this project. Started with the initial prototyping phase of user interface designs as well as a storyboard and user story of how the application will work. In addition, I've started work on a data-flow diagram as well as a data dictionary.
10/31/2018	3	After further research and analysis when writing the scope of this project after completing the data flow diagram, I've decided that the scope of this project is too wide as well as the pre-requisites required are too many before, I can even attempt to begin this project. This in addition to the social & ethical issues that could potentially arise with an inadequately programmed piece of software, such as the leaking of user data, passwords & credit card information has ultimately made me decide to change projects to a Civil Engineering piece of software aimed towards students – Sectrics. So far initial research of truss analysis as well as the best way to solve linear algebra problems through computing. Additionally, the first implementation of the node.cs class file has been developed, taking in user inputs of node location, which node connects to which node (member variable), the forces applied as well as material properties have been implemented.
11/1/2018	2	Further research into the best way of solving this system. In addition, several key implementations of in the program have been added, in the main function, including; nodes, members as well as the start of the; support function, about function and exit application function.
11/3/2018	3	Continued programming of the civil engineering piece of software, after further research I've concluded that the best method of solving this system is utilizing the finite element method. Work on the solve function has begun with the creation of several math's functions including; DeltaX (which finds the difference between 2 points), as well as the implementation of a selection sort algorithm to hopefully find the maximum range and eventually length of a beam when implemented with a distance formula solver.
11/3/2018	3	Continued programming for the civil engineering software, however, I've gotten stuck with logic errors regarding solving the reaction forces, due to the restraint of me not wanting to utilize any additional libraries excluding the compulsory system library, I've gotten stuck determining a way to solve an equation.
11/4/2018	2	Began research on simultaneous equations to help solve the systems forces in each member, additionally found a method of solving simultaneous equations through matrices. This seems like the most efficient way and quickest way to solve a simultaneous equation. In addition, after discussing this with my math's teacher Mr Keagan he has, in addition, concluded that this would be the most efficient way. Due to this, I've begun watching Khan Academy videos on matrices and how to utilize these to solve simultaneous equations.
11/4/2018	1	After going out on Saturday I met a civil engineering and in the discussion, we began talking about my project. He recommended me to attempt to build a system similar to this in excel prior to programming it as it has several solve functions as well as the built-in implementation of matrices. Assisting me in modelling the structure prior to programming it. In addition, he also recommended me a video on how to solve simple truss questions with excel https://www.youtube.com/watch?v=CVSlVTvoMmA.

11/5/2018	1	Documentation has been lagging behind as I've continued to research methods of solving this problem. Today documentation again has been started. Work on the scope of this project has been started and due to a discussion with an engineering teacher an issue was highlighted to me regarding this project, regarding liability. There is rules & regulations regarding civil engineering software and if on the off chance an engineer uses my program and a bridge crashes due to the result of my software, I may be held liable. Therefore the scope has been changed to be specifically targeted towards students studying engineering studies in the HSC syllabus and not for real-life applications of structural analysis, in addition this has also changed the type of license this piece of software operates under, thus now falling under the **Mozilla Public License 2.0** stating that we are not liable for the misuse or use of this program.
11/5/2018	1	Further research and learning of matrices and how to solve simultaneous equations regarding matrices. So far have learnt how to multiply matrices and inverse matrices using the Gauss Jordon elimination method as well as the adjacent method. After careful consideration and the operations required to inverse the matrices, I've determined the easiest method to inverse the matrix would be the adjoint method.
11/10/2018	2	After further looking into the adjoint method and the coding of several functions, I've determined in turn to calculate the determinate using a matrix of minors, via the assistance of my math's teacher, the most efficient method most likely would be the utilization of a recursive function. Therefore, I've decided to research recursion and practice some recursive problems prior to the working on the determinant solver.
11/11/2018	3	Started the programming of the determinant function utilization recursive functions. The basic idea is to start from a big matrix and then split them up into smaller matrices and then split them up into smaller matrices until we get into the base form of a 1*1 matrix. Therefore, starting by a (n*n) then to a ((n-1) * (n-1)) going down to a (1*1).
11/11/2018	5	After a long 10-hour coding spree, finally finished the implementation of the determinant solver utilizing recursive functions. In addition, after further testing of the function have added catch functions allowing the process to only function if the matrix is square and in addition have solved a random issue regarding stack overflow error if the matrix is inputted with negative values.
12/11/2018	4	Further development of the math's functions, still focusing on programming a method to find the inverse of an n*n matrix. Full implementation of a function to remove rows & columns listed on a matrix and changing it to a ((n-1) * (n-1)) matrix has been developed. In addition, the creation of an (n*n) row column remover which takes a matrix of rows and columns has been developed, this function removes the related rows and columns listed simultaneously to avoid the changing of the global matrix's positions, thus changing the index removed.
13/11/2018	1	Completed the full implementation of the adjoint solver. The adjoint solver has been made by changing node locations and then using a matrix of cofactors on it.
20/11/2018	2	Finally finished all key matrix operations. Finished the full implementation of the matrix function, in addition, a key function completed was the inverse matrices solver, with catching if the matrix is not square or there is no determinant for example in the matrix {1,2,3}{4,5,6}{7,8,9} where the determinant, in this case, is 0.
21/11/2018	3	Continued work on additional matrix operations and additional functions. Created a matrix multiplication method between 2 matrices as well as a matrix multiplication method between a cofactor and a matrix. In addition, the creation of a matrix display function has been added, allowing quick ease of debugging as well as an aesthetically pleasing display format of matrix data.
21/11/2018	3	Started further work on the finite element method, specifically on modelling the structure to work with the finite element method. After careful consideration several key variables have been made including; Nodes, Members, Support, Load, Density, Young Modulus & Degrees of Freedom.
24/11/2018	4	Started further research on the finite element analysis method to solve simultaneous equations regarding the stiffness matrix. The first portion of it is understandable however I do not fully understand the global stiffness matrix. I've picked up a book regarding finite element method using this called Finite Element Procedures on the MIT website.
25/11/2018	4	Continued research on the finite element method, still encountering issues regarding the construction of the global stiffness matrix. Continued research on finite element analysis regarding alternative methods of constructing this matrix.

26/11/2018	4	Continued research on the finite element analysis method regarding the element stiffness matrix, after having another discussion with the Civil Engineer who previously recommended me with the matrix method, I've finally figured out a way to create the global stiffness matrix.
2/12/2018	2	Continued work on the feasibility study, focused heavily on the functional requirements of the program; full creation of the mandatory requirements, desirable requirements, optional requirements and possible future enhancements have been created.
3/12/2018	1	Started work on the generation of ideas and alternative solution phase including; idea generations, solution idea generations, possible alternative solutions, existing alternative solutions as well as possible alternative solutions.
5/12/2018	1	Continued work on the feasibility standard and finished the writing up of the convention/standard section as well as the Scope & Financial section. Issues, however, arise through the development in with the cost-benefit analysis.
6/12/2018	1	Continued work on; cost estimate, cost-benefit analysis in addition to operational, technical as well as project plan and Gantt chart.
8/12/2018	1	Finishing off on cost estimate, cost-benefit analysis, operational, technical and project plan and Gantt chart, additionally started and finished the section regarding social & ethical considerations.
10/12/2018	2	Starting work on the planning & designing phase; finished the user interface & screen design in addition to user storyboards and the input process output diagram.
12/12/2018	3	Started and finished the context diagram.
13/12/2018	3	Started and finished the decision trees as an optional requirement.
14/12/2018	4	Started and finished the Data Flow Diagram (DFD), Structure Chart, System Flowchart as well as Hardware & Software Requirements.
16/12/2018	5	Work delayed due to extreme rain conditions, many trees fell, and power was out in West Pennant Hills & Thornleigh, after driving to a friend's house started and finished the Data Dictionaries as well as Test Data.
16/12/2018	0	Finished Section 1 of the SDD major project, going to office work to print out project & get it bound.

Bibliography

- Skyciv.com. (2018). *SkyCiv | Cloud Structural Analysis Software*. [online] Available at: https://skyciv.com/ [Accessed 16 Dec. 2018].
- En.wikipedia.org. (2018). *Area density*. [online] Available at: https://en.wikipedia.org/wiki/Area_density [Accessed 16 Dec. 2018].
- En.wikipedia.org. (2018). *Software license*. [online] Available at: https://en.wikipedia.org/wiki/Software_license [Accessed 16 Dec. 2018].
- CAD Intentions. (2017). *Best Workstation for AutoCAD & Engineering in 2017? – Custom @Xi Computer Workstation*. [online] Available at: http://cadintentions.com/best-workstation-for-autocad-engineering-in-2017-custom-xi-computer-workstation/ [Accessed 16 Dec. 2018].
- YouTube. (2018). *Method of Joints Truss Analysis Matrix Method using MS Excel*. [online] Available at: https://www.youtube.com/watch?v=uvHGCZO1agM&t=216s [Accessed 16 Dec. 2018].
- YouTube. (2018). *Lecture 20 : Matrix Method of Analysis of Trusses(Contd.)*. [online] Available at: https://www.youtube.com/watch?v=aDPaofpQt5g [Accessed 16 Dec. 2018].
- YouTube. (2018). *Trusses - Method of Joints (Matrix Methods) - Structural Analysis*. [online] Available at: https://www.youtube.com/watch?v=ukPw8xh31n8&t=1545s [Accessed 16 Dec. 2018].
- YouTube. (2018). *English - Truss Analysis Using Method of Joints Part 1 of 2*. [online] Available at: https://www.youtube.com/watch?v=xrU3l1RvyiE [Accessed 16 Dec. 2018].
- YouTube. (2018). *Method of Joints Truss Analysis Matrix Method using MS Excel*. [online] Available at: https://www.youtube.com/watch?v=uvHGCZO1agM&t=511s [Accessed 16 Dec. 2018].
- YouTube. (2018). *2D Truss Analyzer with C# .NET (SefatherLAB Analysis)*. [online] Available at: https://www.youtube.com/watch?v=xydkkhwB4ro&t=5s [Accessed 16 Dec. 2018].
- YouTube. (2018). *Method of Joints Truss Analysis Matrix Method using MS Excel*. [online] Available at: https://www.youtube.com/watch?v=uvHGCZO1agM&t=275s [Accessed 16 Dec. 2018].
- YouTube. (2018). *Method of Joints Truss Analysis Matrix Method using MS Excel*. [online] Available at: https://www.youtube.com/watch?v=uvHGCZO1agM&t=373s [Accessed 16 Dec. 2018].
- YouTube. (2018). *SA48: Matrix Displacement Method: Truss Analysis*. [online] Available at: https://www.youtube.com/watch?v=iqdxu8gEZ4c&t=2s [Accessed 16 Dec. 2018].
- YouTube. (2018). *Method of Joints Truss Analysis Matrix Method using MS Excel*. [online] Available at: https://www.youtube.com/watch?v=uvHGCZO1agM&t=41s [Accessed 16 Dec. 2018].
- Romeo, F. and Padoan, G. (2008). JTruss: A CAD-oriented educational open-source software for static analysis of truss-type structures. *Computer Applications in Engineering Education*, 16(4), pp.280-288.
- Au.mathworks.com. (2018). *2D - Truss Analysis help - MATLAB Answers - MATLAB Central*. [online] Available at: https://au.mathworks.com/matlabcentral/answers/140349-2d-truss-analysis-help [Accessed 16 Dec. 2018].
- YouTube. (2018). *SA48: Matrix Displacement Method: Truss Analysis*. [online] Available at: https://www.youtube.com/watch?v=iqdxu8gEZ4c [Accessed 16 Dec. 2018].
- YouTube. (2018). *English - Truss Analysis Using Method of Joints Part 1 of 2*. [online] Available at: https://www.youtube.com/watch?v=xrU3l1RvyiE&t=32s [Accessed 16 Dec. 2018].

- YouTube. (2018). *Method of Joints Truss Analysis Matrix Method using MS Excel*. [online] Available at: https://www.youtube.com/watch?v=uvHGCZO1agM [Accessed 16 Dec. 2018].
- YouTube. (2018). *ENGR 570 Lecture 05: Truss Analysis Using Microsoft Excel (2015.01.26)*. [online] Available at: https://www.youtube.com/watch?v=I2UU6ozjDsE [Accessed 16 Dec. 2018].
- YouTube. (2018). *ENGR 570 Lecture 05: Truss Analysis Using Microsoft Excel (2015.01.26)*. [online] Available at: https://www.youtube.com/watch?v=I2UU6ozjDsE [Accessed 16 Dec. 2018].
- YouTube. (2018). *2D Truss Analyzer with C# .NET (SefatherLAB Analysis)*. [online] Available at: https://www.youtube.com/watch?v=xydkkhwB4ro [Accessed 16 Dec. 2018].
- YouTube. (2018). *Analysis of Truss (Method of Sections) the EASY WAY!!*. [online] Available at: https://www.youtube.com/watch?v=2uSQPm3_4mY [Accessed 16 Dec. 2018].
- YouTube. (2018). *How to Find Adjoint A of a 2x2 matrix {SHORT_CUT}*. [online] Available at: https://www.youtube.com/watch?v=nZBJ9B7HtKY [Accessed 16 Dec. 2018].
- YouTube. (2018). *Multiplication of Two Matrix in C (HINDI)*. [online] Available at: https://www.youtube.com/watch?v=WtmMrHIqfUQ [Accessed 16 Dec. 2018].
- YouTube. (2018). *Inverse of 3x3 matrix*. [online] Available at: https://www.youtube.com/watch?v=pKZyszzmyeQ [Accessed 16 Dec. 2018].
- YouTube. (2018). *Transpose of a matrix | Matrices | Precalculus | Khan Academy*. [online] Available at: https://www.youtube.com/watch?v=TZrKrNVhbjI&t=72s [Accessed 16 Dec. 2018].
- YouTube. (2018). *Adjacency Matrix*. [online] Available at: https://www.youtube.com/watch?v=98vpw3hHVrs [Accessed 16 Dec. 2018].
- YouTube. (2018). *Simpler 4x4 determinant | Matrix transformations | Linear Algebra | Khan Academy*. [online] Available at: https://www.youtube.com/watch?v=QV0jsTiobU4 [Accessed 16 Dec. 2018].
- YouTube. (2018). *Finding the Determinant of a 3 x 3 matrix*. [online] Available at: https://www.youtube.com/watch?v=21LWuY8i6Hw&t=3s [Accessed 16 Dec. 2018].
- YouTube. (2018). *Classic video on inverting a 3x3 matrix part 2 | Matrices | Precalculus | Khan Academy*. [online] Available at: https://www.youtube.com/watch?v=obts_JDS6_Q&t=7s [Accessed 16 Dec. 2018].
- YouTube. (2018). *Idea behind inverting a 2x2 matrix | Matrices | Precalculus | Khan Academy*. [online] Available at: https://www.youtube.com/watch?v=iUQR0enP7RQ&t=2s [Accessed 16 Dec. 2018].
- YouTube. (2018). *Matrix multiplication (part 2)*. [online] Available at: https://www.youtube.com/watch?v=OAh573i_qn8 [Accessed 16 Dec. 2018].
- YouTube. (2018). *Matrix multiplication (part 1)*. [online] Available at: https://www.youtube.com/watch?v=aKhhYguY0DQ [Accessed 16 Dec. 2018].
- People.duke.edu. (2018). [online] Available at: http://people.duke.edu/~hpgavin/cee421/truss-method.pdf [Accessed 16 Dec. 2018].
- Ce.siue.edu. (2018). [online] Available at: http://www.ce.siue.edu/examples/Worked_examples_Internet_text-only/Data_files-Worked_Exs-Word_&_pdf/Trusses_Method_of_Joints.pdf [Accessed 16 Dec. 2018].
- Eng.cu.edu.eg. (2018). [online] Available at: http://www.eng.cu.edu.eg/users/amamansour/MatrixSA.pdf [Accessed 16 Dec. 2018].

- Nptel.ac.in. (2018). [online] Available at: https://nptel.ac.in/courses/Webcourse-contents/IIT%20Kharagpur/Structural%20Analysis/pdf/m4l28.pdf [Accessed 16 Dec. 2018].
- Unm.edu. (2018). [online] Available at: http://www.unm.edu/~bgreen/ME360/Finite%20Element%20Truss.pdf [Accessed 16 Dec. 2018].
- Engineering.purdue.edu. (2018). [online] Available at: https://engineering.purdue.edu/~aprakas/CE474/CE474-Ch5-StiffnessMethod.pdf [Accessed 16 Dec. 2018].
- Web.mit.edu. (2018). [online] Available at: http://web.mit.edu/kjb/www/Books/FEP_2nd_Edition_4th_Printing.pdf [Accessed 16 Dec. 2018].
- Adomako, Mr. (2014). Matrix Stiffness Method in Objective-C for Structural Analysis.
- En.wikipedia.org. (2018). *Ray transfer matrix analysis*. [online] Available at: https://en.wikipedia.org/wiki/Ray_transfer_matrix_analysis [Accessed 16 Dec. 2018].