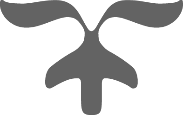


A Level Project

COMPUTER SCIENCE Component 3



June 18, 2018

Sam Macdonald

Mildenhall College Academy Sixth Form

Contents

[ANALYSIS - 1 -](#_Toc524271861)

[What is the problem? - 1 -](#_Toc524271862)

[Why is a Computational approach the most amenable? - 1 -](#_Toc524271863)

[Stakeholders - 2 -](#_Toc524271864)

[Research - 2 -](#_Toc524271868)

[Features of the computational solution - 5 -](#_Toc524271869)

[Limitations - 5 -](#_Toc524271870)

[Requirements - 6 -](#_Toc524271871)

[Success Criteria - 7 -](#_Toc524271872)

[DESIGN - 1 -](#_Toc524271873)

[Objectives / modules - 1 -](#_Toc524271875)

[Program Specifications - 2 -](#_Toc524271876)

[Solution Structure: Top-Down Design - 1 -](#_Toc524271877)

[Smaller computational solutuions - 1 -](#_Toc524271879)

[Alogorithms - 2 -](#_Toc524271880)

[Usability features - 3 -](#_Toc524271882)

[Key variables, data structures and classes - 3 -](#_Toc524271884)

[the test data used in the Iterative development - 5 -](#_Toc524271886)

[further data to be used in post development stage - 5 -](#_Toc524271887)

[SOLUTION DESIGN - 5 -](#_Toc524271888)

[software requrements - 5 -](#_Toc524271889)

# **ANALYSIS**

## What is the problem?

After having an interview with my client and getting the requirements, the problem they have is relatively straight forward to solve. My client works for a company who deals with border control, surveying maritime routes for search and rescue as well as illegal activity. Currently, my client uses a notepad and pen to manually record the variables of the flight during each mission. This is not very efficient or useful because the data cannot be manipulated to produce reports or statistical modelling, for example. As the data is input manually, there is no validation resulting in a lack of data consistency or integrity. As well as being time consuming, my client may have to search through the data manually, which is very inefficient. Due to human error, the client also makes mistakes and crosses out information, making it hard to read and understand what data there is when reviewing at a later date. This current manual input of data, from the notepad and then into a computer is laborious and repetitive which can be made redundant, therefore taking out the notepad stage along with the inherent issues associated with this method of recording.

## Why is a Computational approach the most amenable?

This problem needs a computational approach to solve it, rather than a manual/human approach, because the program would need to store a lot of information; if there are, for arguments sake, 100 different records in the database, it would be difficult to search for a particular attribute because a hand-written database doesn’t give you the privilege of using queries to search throughout the database. Therefore, it is much more efficient to store the data in an electronic database. Having a hand-written database can be prone to human error, because each day (a record) will have similar data and can be very repetitive, allowing the user to make mistakes; a computational approach to the problem will allow the user to edit and change data if they ever make a mistake.

The program also needs to import a Keyhole Markup Language (KML) and a computer is needed in order to do this – doing this manually would be virtually impossible. A KML is way of visualizing a map route geographically. Having a computational approach gives my client more scope, allowing them to do more with the raw data. Furthermore, if the user enters a note in the timeline for 10:00, and then decides to add another note at 09:00, the program can rearrange the table timeline so it is in chronological order. This would be difficult to amend on a manual record requiring re-written data and poor use of time and resources. In addition to this, the problem I will be trying to solve will be high client-side, because it will have a GUI to visualise the form, KML, print layout etc. in a clear, easy-to-read format.

A computational approach is suitable for this problem, because decomposition can allow me to break down the problem systematically, into its smaller components/modules, making it easier to code and understand each of the modules. Thinking about this problem with an abstraction approach would be useful, because it will allow me to take out unnecessary material that would not be effected in the program, just leaving the variables. The problem requires a lot of decisions made (e.g. if the user has entered text or if they have imported a KML) therefore taking this program with a logical approach would be beneficial.

## Stakeholders

### Those interested in this program would include individuals, and/or companies who have the need to log their flight missions. This could be the military, aerial imagery sectors or border control agencies such as Frontex. My client, who is also my end user, is familiar with flight missions and the key terminology which would be required in this program (e.g. ETD, ETA, MSN Time, Eng On, Estimated Time of Departure, Estimated Time of Arrival, Mission Time and Engine On respectively.)

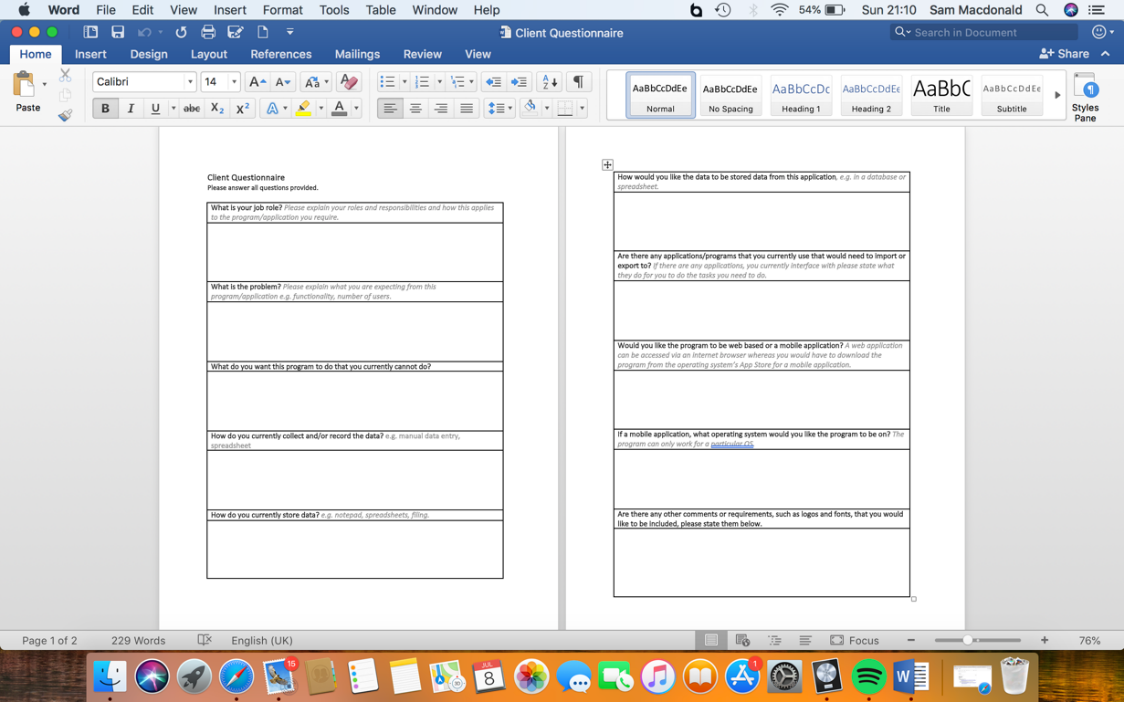
### My stakeholder(s) will have a major input on the program’s development giving me feedback at each stage and suggestions for improvement. They will also directly influence the specification points that the program must meet to be successful.

### In addition to this, I will need to produce a prototype of the program that is on course with the development timeline, to meet the client’s needs and may require adaptation during the process. Regular contact with and feedback from the client will ensure the project remains on track or allows for changes, which would need to be controlled to allow adjustments to scope, timeline or budget to give enhanced functionality to the client. Because my end user is my only stakeholder, I will not need to worry about disagreements between stakeholders, regarding their requirements or outcomes.

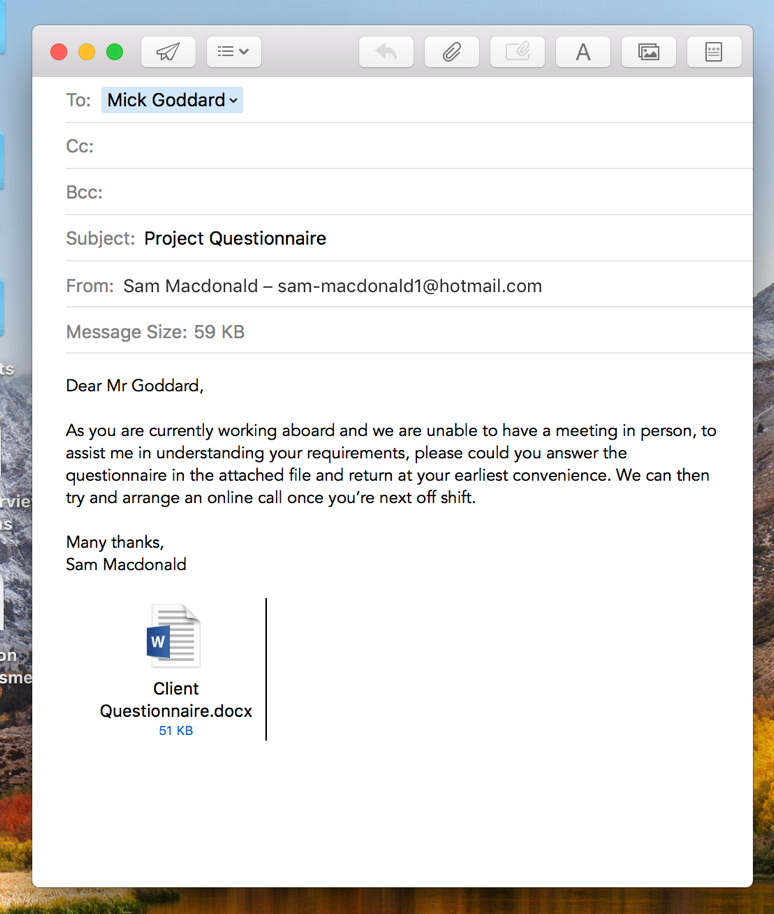
## Research

**First interview**

I need to arrange an interview with the client, finding out specifically what the problem is and how I, the programmer, can make a product for them. Because I cannot currently meet the client in person, I have emailed them a questionnaire, providing questions to get enough information as possible. Here is a screenshot of the questionnaire document:



Here is a screenshot of the email I sent to the client, asking them to complete the questionnaire and to get back to be at their earliest convenience.



\*Insert screenshot of completed questionnaire\*

**Existing Products**

Adobe Acrobat is a family of application software, developed by Adobe, which allows the user to view, convert, edit, sign and print Portable Document Format (PDF). This application has some useful aspects for when the end users export a file as a PDF. As with all applications, there are some advantages and disadvantages. One advantage is that it is an Adobe product, meaning that it is legitimate software, providing conformity and reliability: Acrobat stores every detail (such as spacing, pictures and fonts) within the file itself, making it easier to share documents - if the provider has a font that the recipient has not got, the recipient doesn’t need to install the font. On the other hand, Adobe requires the user to pay for the full version of Acrobat: its £13.14 per month for the “Adobe Standard DC” package or £15.17 per month for the “Adobe Pro DC” package. However, Acrobat does offer a free version which has limited and restricted features.

Google Maps and OpenStreetMap are mapping software in which the user can view the world from the streets. However, this requires the internet, because it uses the uses GPS and the user’s location. An advantage of both applications is that it allows the user to import a KML (which is what the client would like out of my program.) However, some disadvantages of Google Maps and OpenStreetMap is the limited accuracy and the loss of privacy. Moreover, Google Earth is a similar application to both Google Maps and OpenStreetMap and is more beneficial for the client’s needs: the application can provide an up-to-date, detailed and accurate map of the world, providing support and resources for the client in what they currently do. When coding the program, I was going to import a KML separately and then the user imports a screenshot of the map to then place the KML over the top, but after researching the application, I found that not only can the application import a KML, but it can also be embedded into a webpage – being more efficient to code and giving the client a professional end program.

Microsoft Excel is a spread-sheeting software, that my client can create tables as well as importing a KML. An advantage of using Excel is that is a generic, easy to use, application that my client can already use. However, using a table or database in excel is inefficient, inconsistent and unreliable (if there is the same attribute in multiple tables, and I update an attribute in one table, the other table wont update). On the other hand, with Excel, you can create graphs from the tables, so, for example, my client could draw graphs of the average mission time in a week. Microsoft Access is another application, similar to Excel, but deals with databases more, allowing search queries and to sort/filter records. phpMyAdmin is a software tool that handles with the administration of MySQL over the internet. A programmer that can use PHP can connect their code to phpMyAdmin, giving them the facility to create, manipulate and store data in tables. However, the programmer would need to download and run XAMPP, an open source web server, to be able to connect to phpMyAdmin

Currently, my client uses notepad and pen to record the data for each of the flight missions. This may be cheap and easy to use, it is very inefficient to store because they use a manual, paper database. This means they would have to manually search through the database if they want to search for an attribute, being time consuming as well as inefficient.

**Bibliography**

* <http://smallbusiness.chron.com/advantages-disadvantages-using-acrobat-77144.html>
* <https://acrobat.adobe.com/uk/en/acrobat/pricing.html?mv=search&sdid=1FJDDN6S&s_kwcid=AL!3085!3!193496851174!b!!g!!acrobat&ef_id=WzXswwAABKLQJVza:20180629082541:s>
* <https://www.buzzfeed.com/nirdesh191/disadvantages-and-advantages-of-using-google-maps-2q8ao?utm_term=.sq2mDMPqP#.ra0JWbOlO>
* <https://sites.google.com/a/teacherportal.ca/exploring-earth/ge-advantages-disadvantages>
* <https://www.phpmyadmin.net/>

**Second Interview**

## Features of the computational solution

The client requires some sort of software that allows data input electronically, giving the option to update information. They have approached me to produce them a program which logs a flight mission in an easy-to-read table format. They want the data to be stored in a database, allowing them to access previously inputted data beforehand. The client would also like the program to import a KML which can be placed onto a mapping software. Once the client has inputted the data, they would like the option to see this data in a table format, and have the option to create a timeline of events which will take place for the duration of the mission. The client would like the option to have the facility to print the data inputted for a particular date, the KML on top of a geographical system and a timeline of events, as a PDF.

## Limitations

Whilst I aim to provide exactly what my client wanted, there are several limitations affecting the development of the solution; one of which is that my client has a busy schedule working abroad of 4 weeks on and 2 weeks off. If I need to contact them for advice/questions or to arrange an interview, then I would have to wait until they are off shift remotely, or meet when they are back in the UK. This could potentially slow down the development of the solution.

For arguments sake, if the client requested a phone application, rather than a web application, I would need to get the application certified by Apple and Android: it is $99 a year and $25 for a one-off payment, respectively. The solution must be for a specific operating system; if my client initially wanted their program on iOS and in the near future changed over to Android, then the program would not be available because the code would be complied differently due to the type of operating system. In addition to this, if my client would like a web application, then an internet connection would be needed in order to connect the web solution to the database on phpMyAdmin.

My client only wants the solution to work for him, meaning that having a log-in section would seem pointless and not really benefit them because it is for one user. However, if in the near future my client has a team in which multiple users can log on simultaneously, then I would adapt the program to allow for this. I would need to produce a new module in which users can sign up with a username and password and in order to access the program they would have to sign in with the correct data.

My client would like the solution to import a KML, and print off a map showing the KML flight path. This module will probably be the most complex part of the solution, but after researching into the problem, I can embed Google Earth into the program, which will allow me to import a KML, hoping to simplify the development stage of this module.

## Requirements

The client would like a program that records the different variables related to a flight mission, which will calculate the flight duration, then saving the inputs into a database. My client can have the option to upload a KML which can then be placed over a geographical software such as Google Earth. Once setting the values into the database, my client would like view the records and have the ability to select one, displaying all the data in a table format. My client can have the option to upload a KML which can then be placed over a geographical software such as Google Maps or Google Earth. There will be an option to input and view data in a timeline for each date where the client can a time and the event that took place. The client can have the option to print the table of data, KML, and timeline as a PDF, having the table of data and KML on one side, and the timeline on the other. My client also wants the DEA (Diamond Executive Aviation) logo printed on the PDF so I’ll need to somehow import that image, storing it permanently.

If my client decides to have a web based application, rather than a mobile application, then the language I will use is PHP and HTML, whereas if my client uses a mobile application then I can either use Java or Python, or use their database application – JBDC for Java and SQLite for Python respectively.

## Success Criteria

For my program to be successful, it must meet all the requirements below. To find out if all the requirements are met, I will produce a questionnaire to give to the client, after the development and testing stage of the project. Their response of the criteria will tell me if the program has met their needs and satisfaction and if any improvements/maintenance are needed, giving me an indication on how successful the program is. Some of the questions I will include in the questionnaire will be as follow:

Does the program…

* Open/load up with no faults?
* Allow the user to select different sections from the home screen?
* Allow the user to create a new mission?
* Allow the user to enter data into the text fields?
* Allow the user to submit the data?
* Allow the user to view all the dates and information?
* Allow the user to add a timeline?
* Allow the user to import a KML?
* Allow the user to export as a PDF?
* Print the document in the format/layout you intended?
* Meet your satisfaction?

Is the program…

* Accessible?
* Clear and easy to use?
* Clean and simplistic?
* Fast to load pages?
* Meeting your needs?
* Are there any improvements needed to be incorporated into the solution?

These criteria will be quantified with a scale of 0 to 10, where 0 is poor and 10 is excellent.

# **DESIGN**

### In order to produce a working solution effectively, I must first think about the design stage of the process. In this stage, I would need to produce flow charts, pseudo code and a prototype model for the client.

## Top-Down Design

### For this project, I have used a free, online, flow diagram application called *draw.io* to produce a top-down overview of my problem. Here is a screenshot of my initial design:

## Objectives / modules

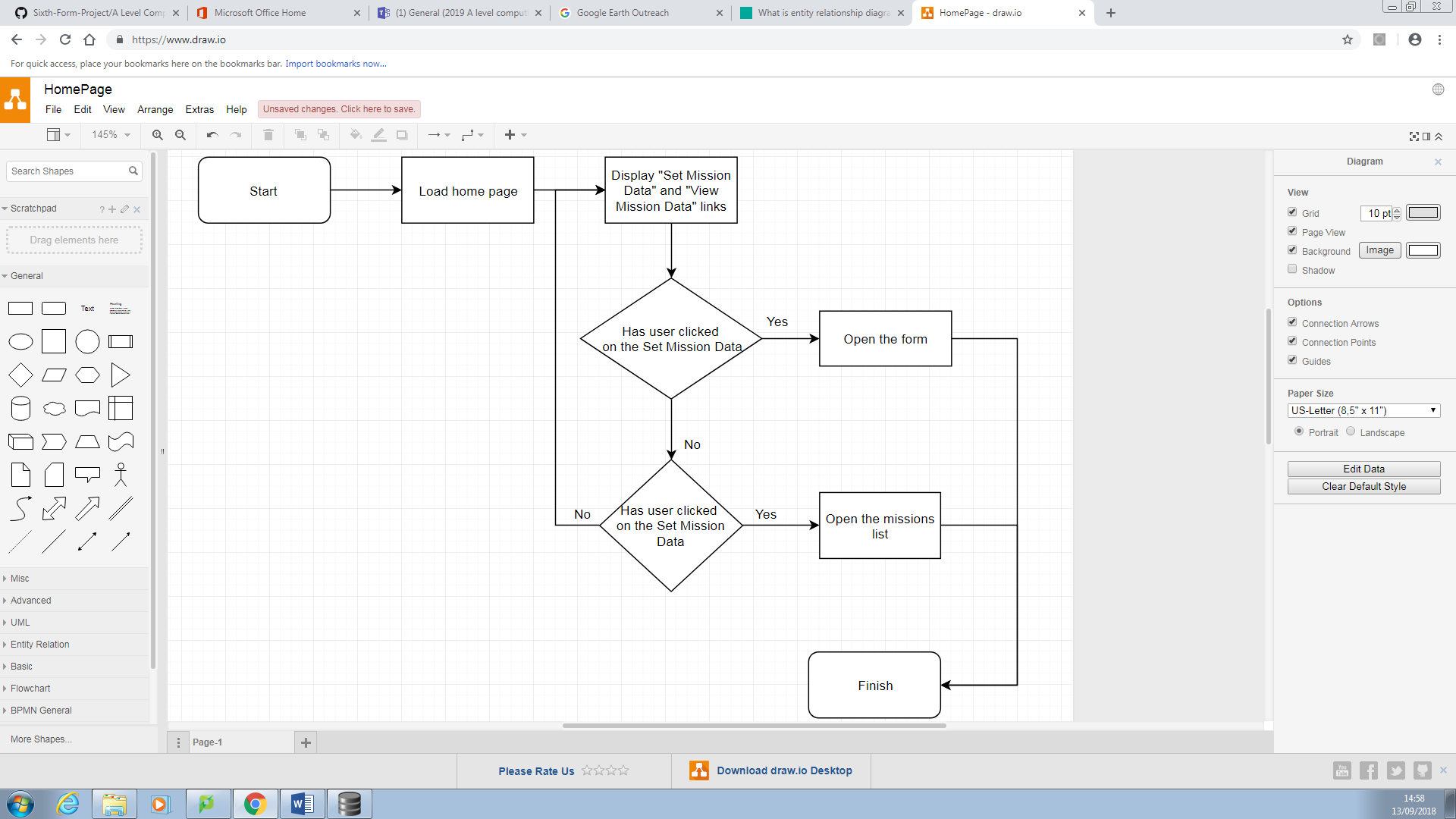
**Objective 1 – Creating the home screen**

The home screen is the first page the end user sees, so I need to make this appealing, simplistic and easy to navigate, but retaining functionality. The home screen would need to have the application title at the top of the page, with two headings for the user to access the application; they will be “Set Mission Data” and “View Missions”.

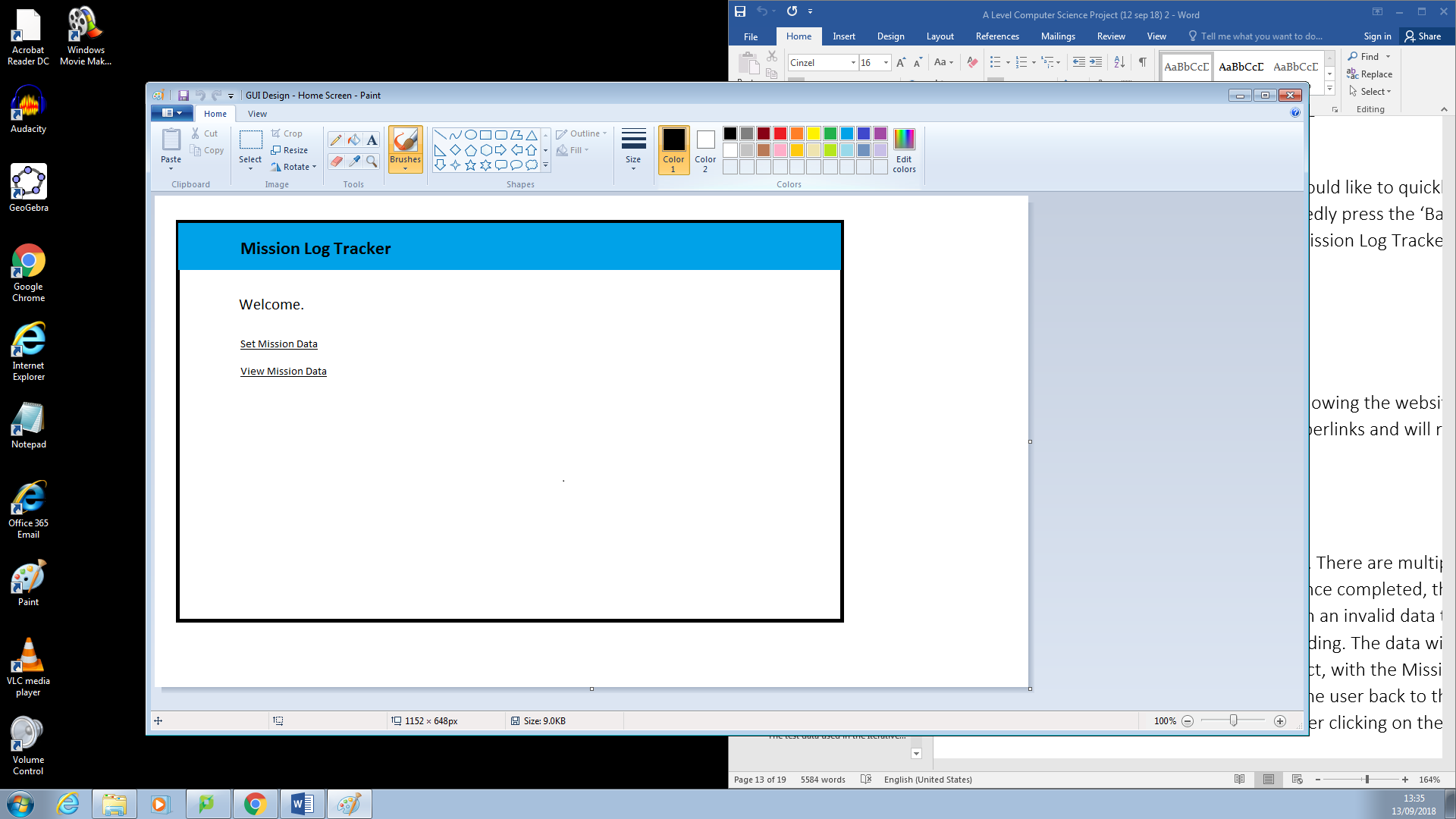
Success Criteria:

* Objective 1 must:
  + Display the home screen to this user, with the appropriate colours, font, title and options.
  + Allow the user to select the two options and carrying out an action once clicked.

Flow Chart:

Flow Charts are the starting point for the design of my project; they will help me code each section by giving brief step-by-steps. However, they may change due to problems I might encounter whilst coding.

GUI Diagram:

I need to create initial ideas of the user interface and what it is going to look like on the user’s end. I would like the website to be simplistic and easy to navigate, but still retaining its functionality.

This is the initial design for the homepage, showing the website title and the two hyperlinks for the user to click on, redirecting the user if they click on them.

Data Structure:

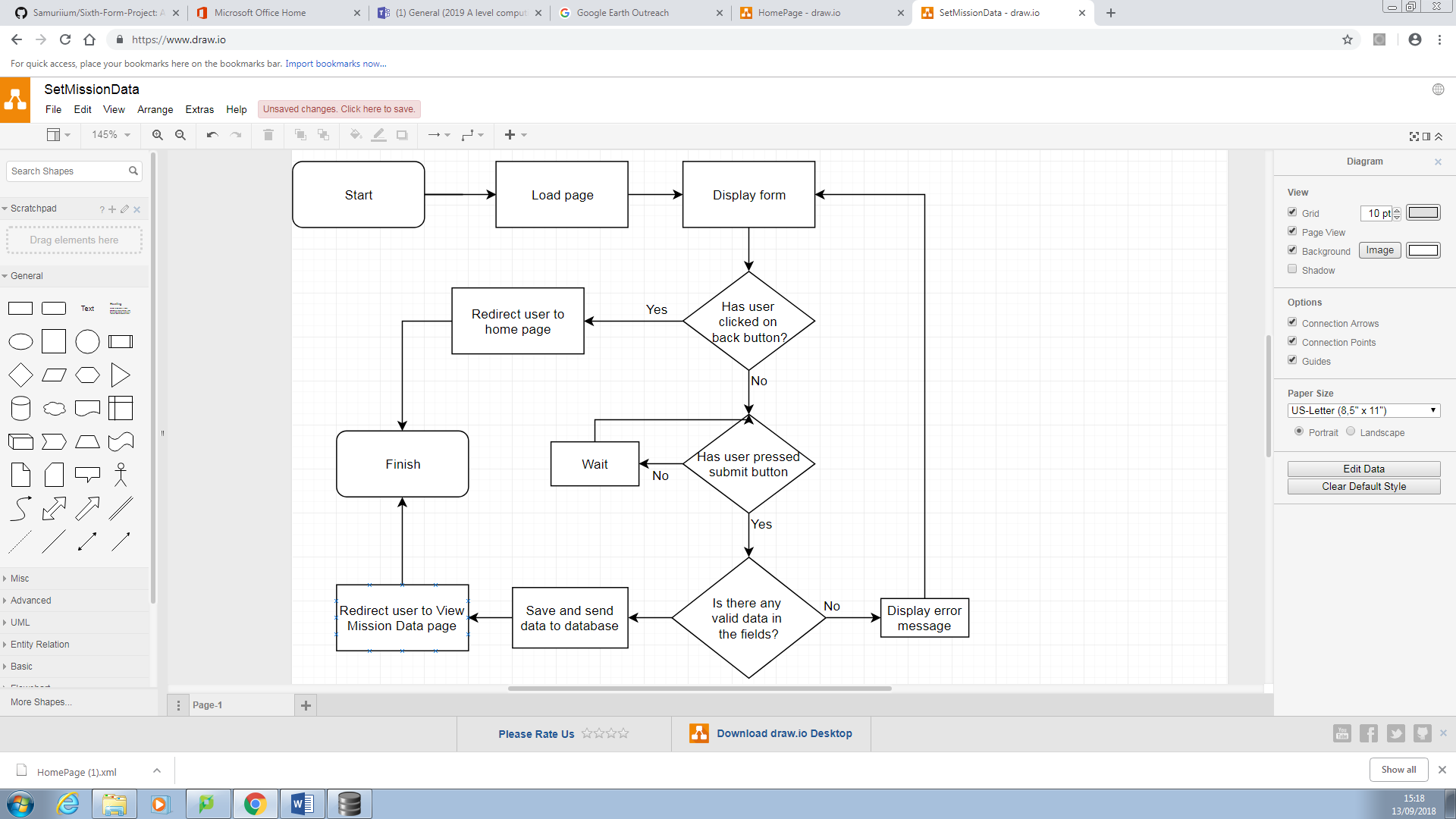
**Objective 2 – Setting mission data**

If the user clicks on “Set Mission Data” they will be redirected to a form and require to enter specific data for a particular mission. It will include various types of data validation, for example correct date format, drop-down selection and verified data input. There will be a submit button where the user can submit their data to the database.

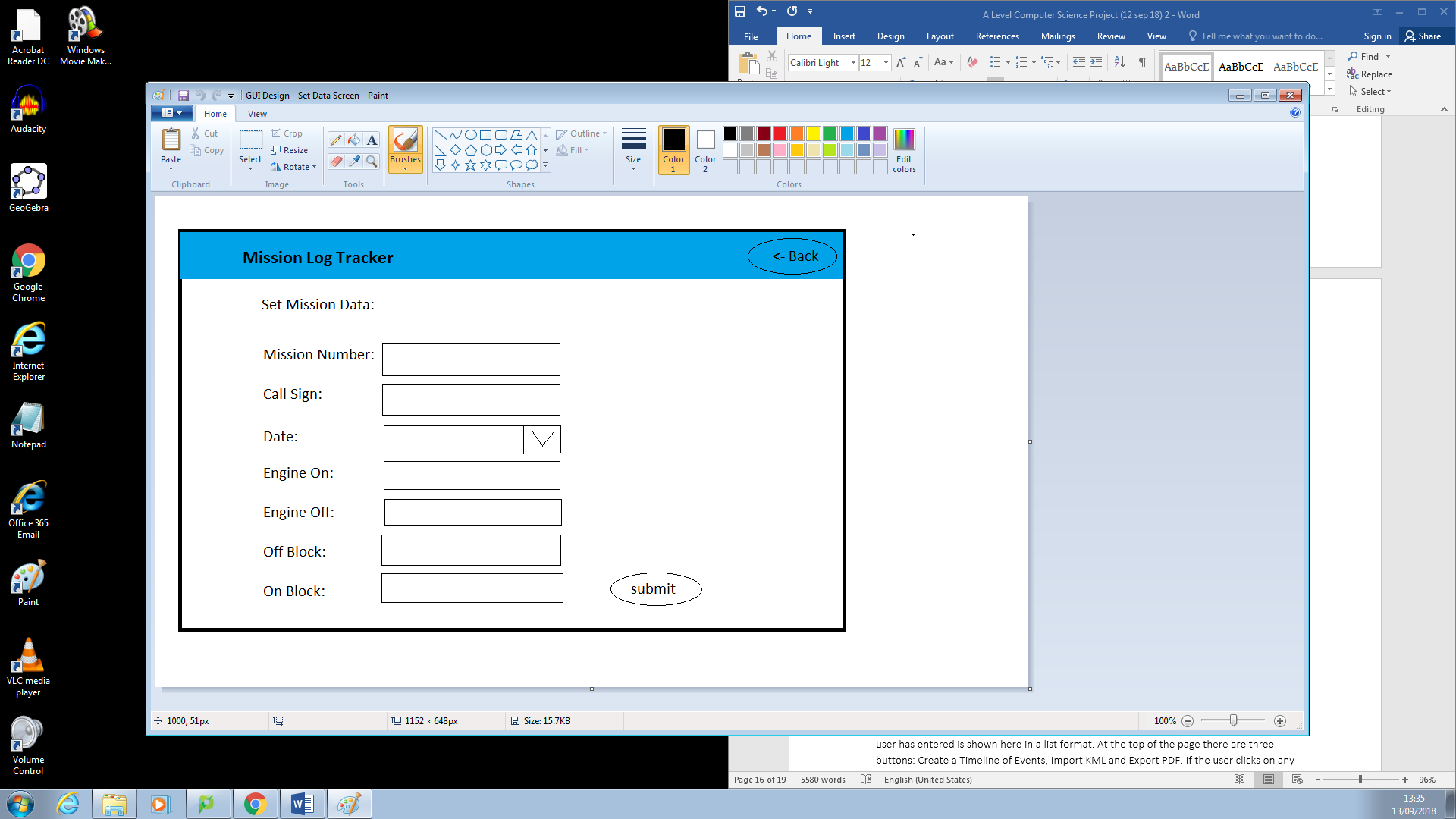
Success Criteria:

* Objective 2 must:
  + Allow the user to view the text boxes
  + Enter data in the text boxes
  + Present a “Submit” button to the user, allowing them to click it. S
  + Display an error message if the user is trying to submit invalid or extreme data
  + Save and send the valid data to the database
  + Redirect the user to the View Mission Data page once submitted valid data
  + Allow the user to return to the home screen by clicking the “Back” button

Flow Chart:



GUI Diagram:



This page allows the user to create a mission. There are multiple text boxes allowing the user to input data for a particular mission. Once completed, they can click the submit button; if there are empty fields or fields with an invalid data type then the user will be alerted and needing to fix this before proceeding. The data will be sent to the database on phpMyAdmin once all the fields are valid, with the Mission Number being the primary key. The user will be redirected to the View Mission Data page. The ‘Back’ button will redirect the user back to the previous page, the home screen, which can also be accessed by the user clicking on the website title.

Data Structure:

* This section will be where most of the user-input will take place. I need to use a database to store the data so it can be accessed/viewed at a later date. For this project, I will use phpMyAdmin to store data in multiple tables, linked together via a primary key (missionNumber).
* A relational database is needed, instead of a flat-file, because I will be creating more than one table which would need to be linked together.
* I will create a variable table called ‘mission\_table’ which will help me decide on what the datatype is and how it will be validated.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| **Variable Name** | **What is it** | **What it stores** | **Data Type** | **Validation** |
| missionNumber | Mission Number | The number allocated to a particular mission | Integer | Checks if the user enters is an integer. It will also check if a mission number has not been stored twice. |
| date | Date | The data of the mission | Date | Checks if the user enters a valid date (e.g. a date that is not after the current date.) |
| callSign | Call Sign | The call sign of the mission | String | Checks if the user enters is a string. |
| engOn | Engine On | The time of which the engine is on | Time | Checks if the user has entered a time. |
| engOff | Engine Off | The time of which the engine is off | Time | Checks the time the user enters is higher/later than the time for engOn. |
| offBlock | Off Block | The time of which the wheels are off the ground | Time | Checks the time the user enters is later than engOn but earlier than engOff. |
| onBlock | On Block | The time of which the wheels are back on the ground | Time | Checks the time the user enters is later than engOn and offBlock but earlier than engOff. |
| takeOffTime | Take Off | The time of which the aircraft takes off | Time | Checks the time the user enters is later than engOn and offBlock but earlier than engOff and offBlock. |
| landTime | Land | The time of which the aircraft lands | Time | Checks the time the user enters is later than engOn offBlock and takeOffTime but earlier than engOff and offBlock. |
| etd | Estimated Time of Departure | The time of which the aircraft is estimated to depart | Time | This is needed to work out msnTime (eta – etd = msn). |
| eta | Estimated Time of Arrival | The time of which the aircraft is estimated to land | Time | This is needed to work out msnTime (eta – etd = msn). |
| streamOn | Stream On | The time of which the video starts streaming | Time | Checks whether the user has entered in the correct format. |
| StreamOff | Stream Off | The time of which the video finishes streaming | Time | Checks whether the user has entered in the correct format. |
| msnTime | Flight Duration | The duration of the mission. The computer would work out this operation, making sure no mistakes are made. | Time | This is needed because it will tell the user how long the plane should have been in the air, and compare it to the real airtime (landTime – takeOffTime). |
| TaskSpecialist | Task Specialist | Name of the task specialist | String | Checks if the user enters is a string. |
| pilotName | Name of Pilot | The name of the pilot flying the aircraft | String | Checks if the user enters is a string. |
| ipAddress | IP Address | The IP address of the flight mission | String | Checks whether the user has entered in the correct format. |
| trackLength | Track Length | The total distance travelled in the flight mission | Integer (nautical miles) | Checks if the user enters is an integer. |
| opsBox | Area of operation | The total area covered in a mission | String | Checks if the user enters is a string. |
| aimsID | Aeronautical Information Management System ID |  | String | Checks if the user enters is a string. |
| weather | Weather | The weather during the mission | String | Checks if the user enters is a string. |

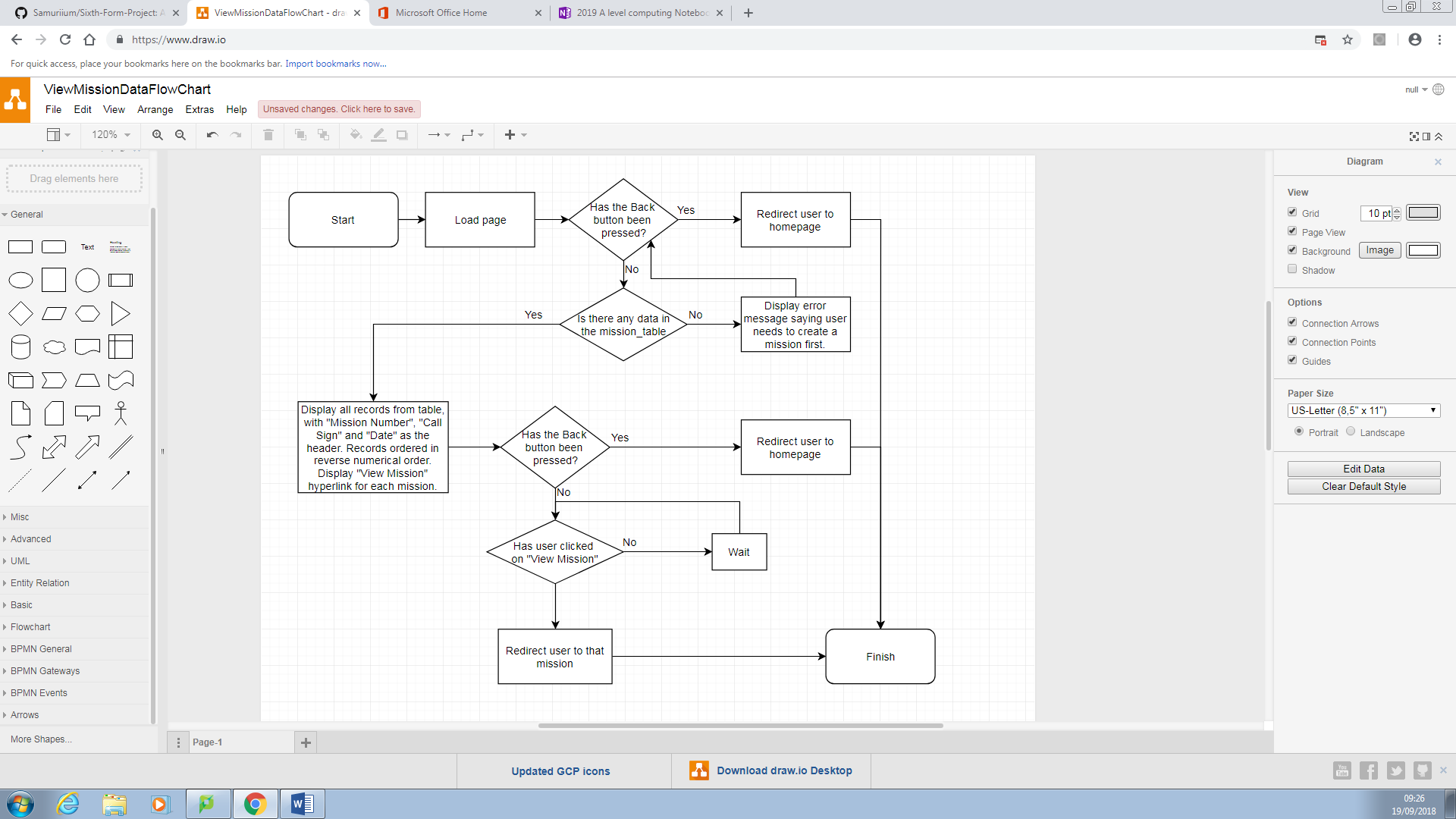
**Objective 3 – Viewing a list of missions**

When the user clicks on “View Missions”, then they would be redirected to a page where all of the previously inputted data is stored. Each of the missions will be in reverse numerical order of mission number, having the date and call-sign as the header, which will allow the end user know what mission the header is referring to. However, if there has been no data inputted, then a message will appear saying “No missions stored, please create a mission first.” There will also be a back button, directing the user back to the home screen.

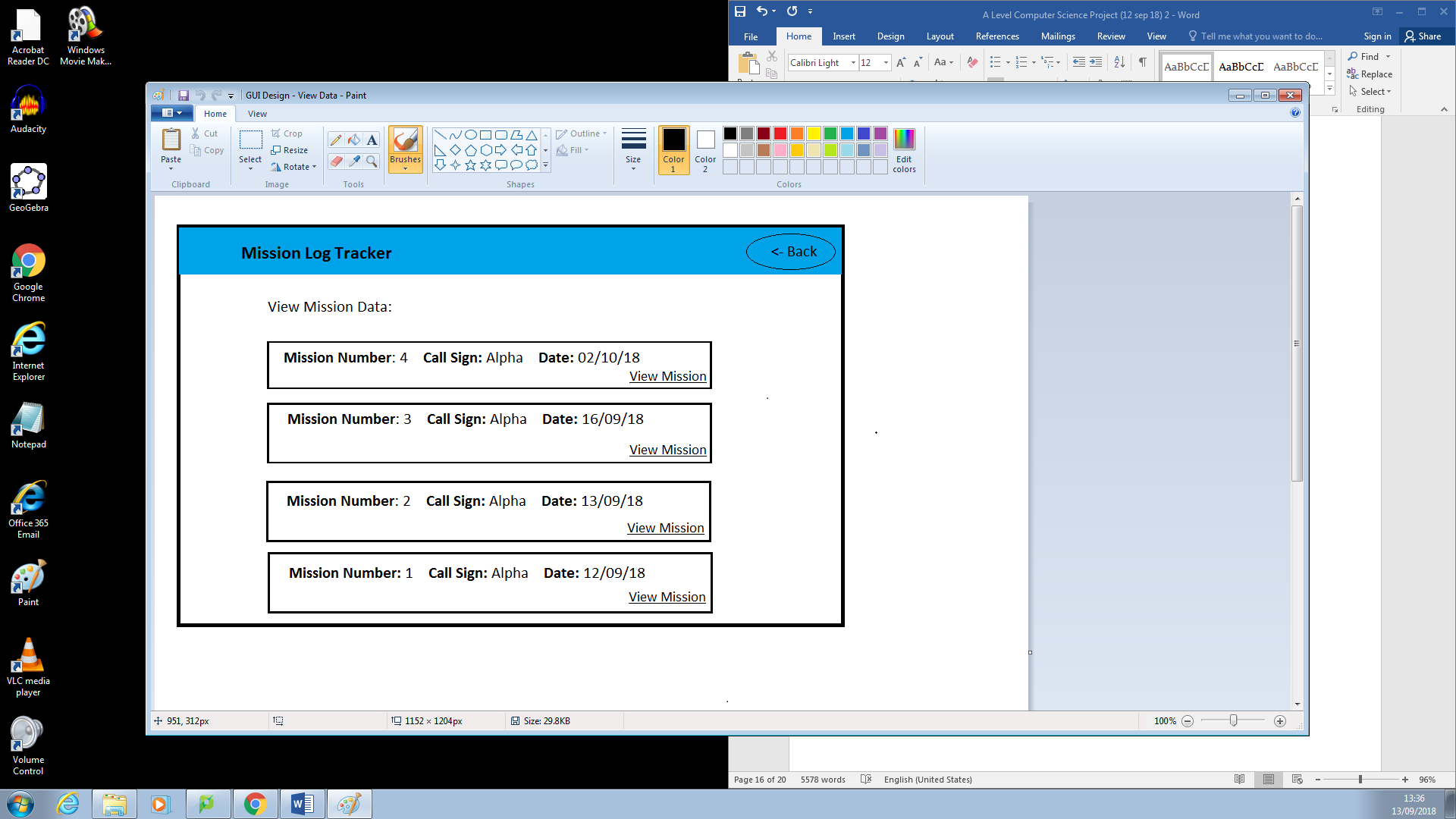
Success Criteria:

* Objective 3 must:
  + Display an error message if there the user has not created any missions
  + Display each mission in reverse numerical order
  + Allow the user to scroll through the missions
  + Allow the user to select a mission by clicking on “View Mission”
  + Allow the user to return to the home screen by clicking the “Back” button or by clicking on the websites title.

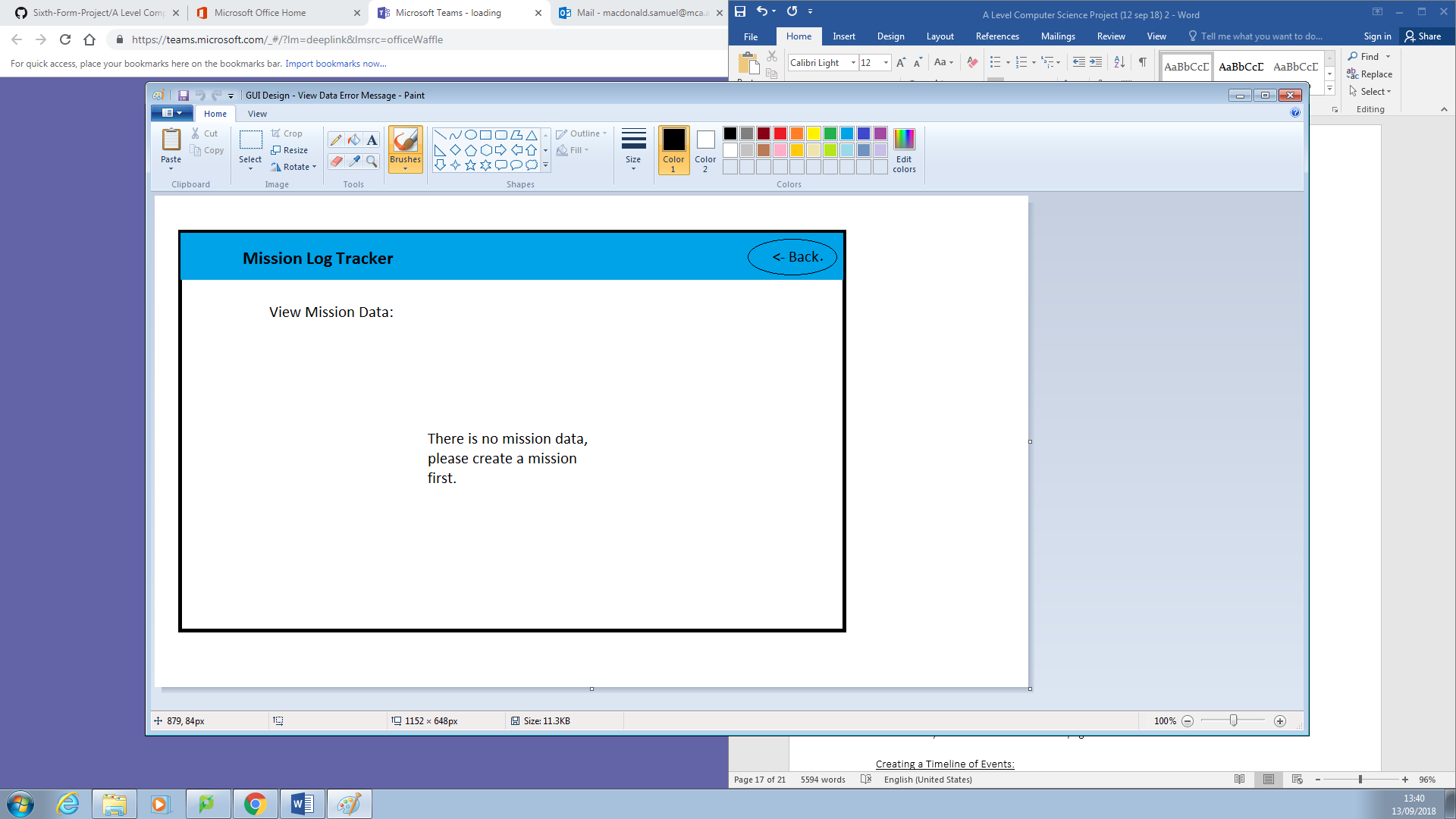
Flow Chart:



GUI Diagram:



This is where all of the stored missions will be displayed to the user, having the Mission Number, Call Sign and Date as the header. The missions will be displayed in reversed numerical order, therefore the most recent mission at the top of the page (the user is most likely to create a timeline of events, import and export date to a recent mission than an older one). The ‘View Mission’ button will direct the user to a new page, showing the metadata for that particular mission. The ‘Back’ button redirects the user back to the home page if the click on it. However, if the user has not created any missions and clicks on the View Mission Data page, then this screen will appear:



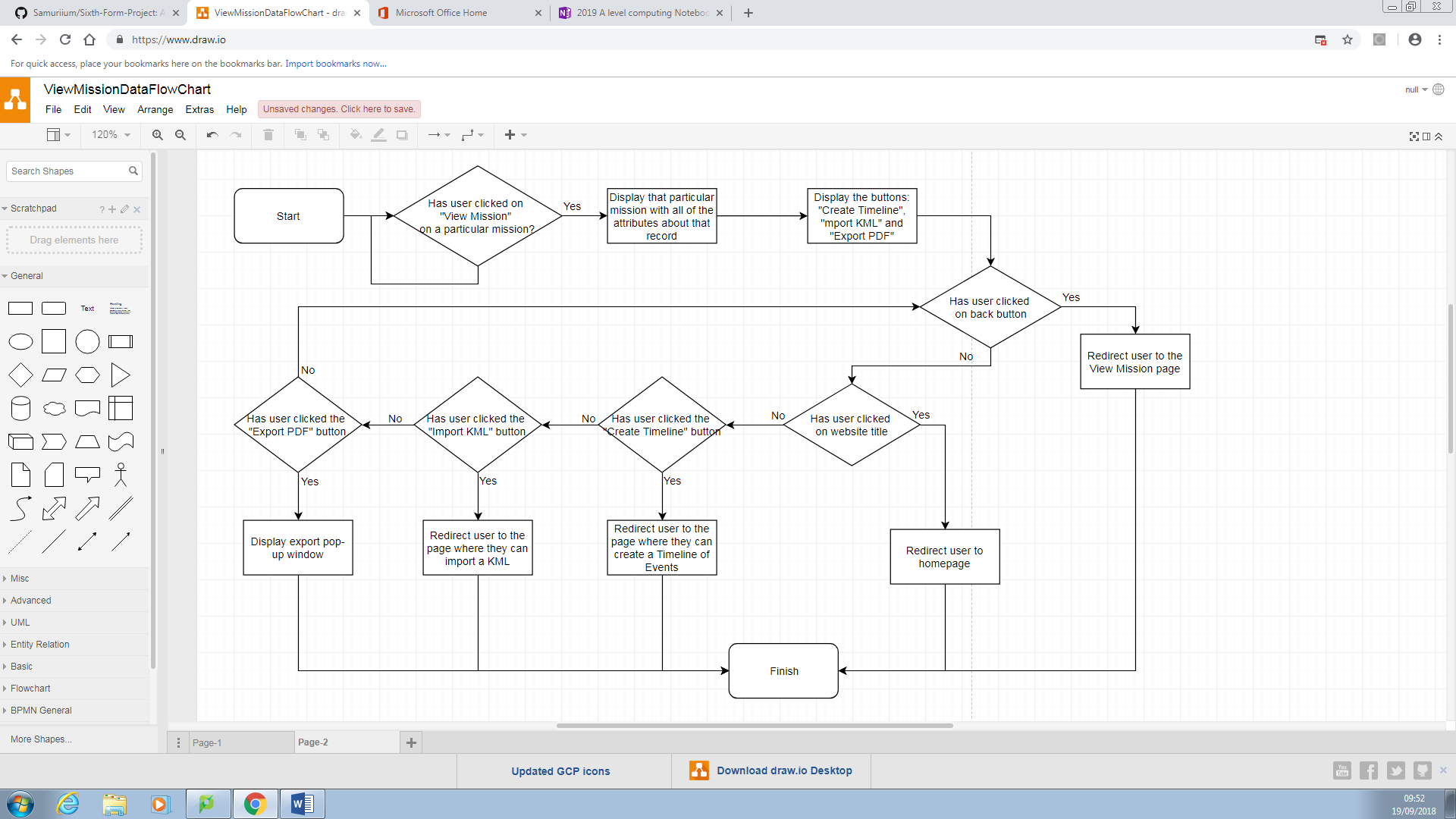
Data Structure:

**Objective 4 – Viewing a particular mission**

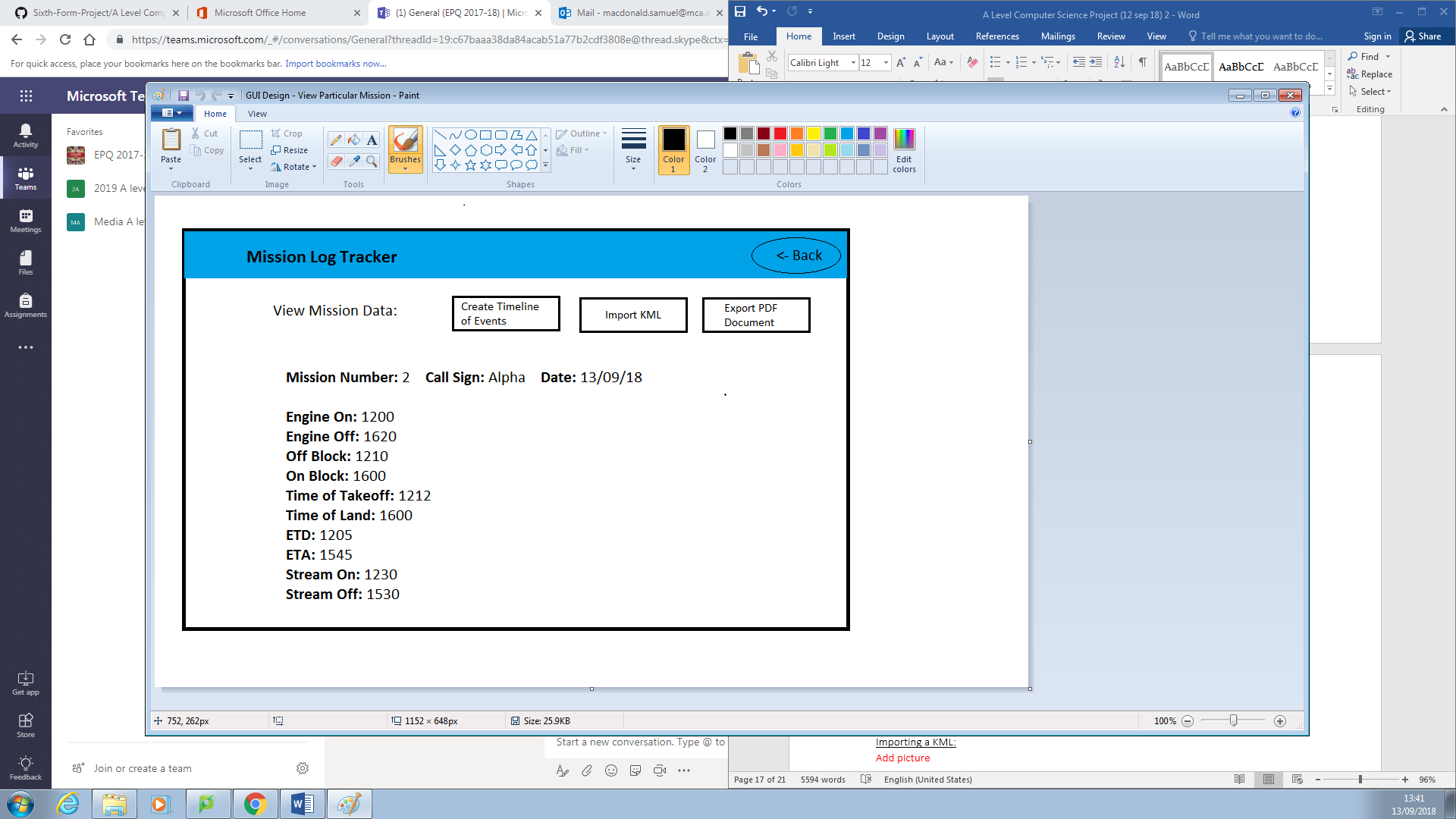
If the user then decides to select a particular mission by clicking on the header, then they would be directed to a new page where all of the stored data would be displayed in a table format. The date, mission number and call-sign would be at the top of the page, and all the other attributes underneath. At the top of the page, the user has the option to create a timeline of events, import a KML or export as a PDF. There will also be a back button, directing the user back to the list of mission’s page.

Success Criteria:

* Objective 4 must:
  + Allow the user to select a particular mission by clicking “View Mission”
  + Display all of the attributes of that mission, such as the mission number, time of takeoff and the pilots name
  + Allow the user to interact with the buttons and take the appropriate action if clicked
  + Allow the user to go back to the previous page, if they click “Back”, or the home screen, if they click on the website title.

Flow Chart:

GUI Diagram:



This page is displayed when the user clicks on ‘View Missions’. All the attributes that the user has entered is shown here in a list format. At the top of the page there are three buttons: Create a Timeline of Events, Import KML and Export PDF. If the user clicks on any of these they will be redirected to a new page.

Data Structure:

* The data that the user stores in the variables in Objective 1 will be displayed here – this is why a database is needed.

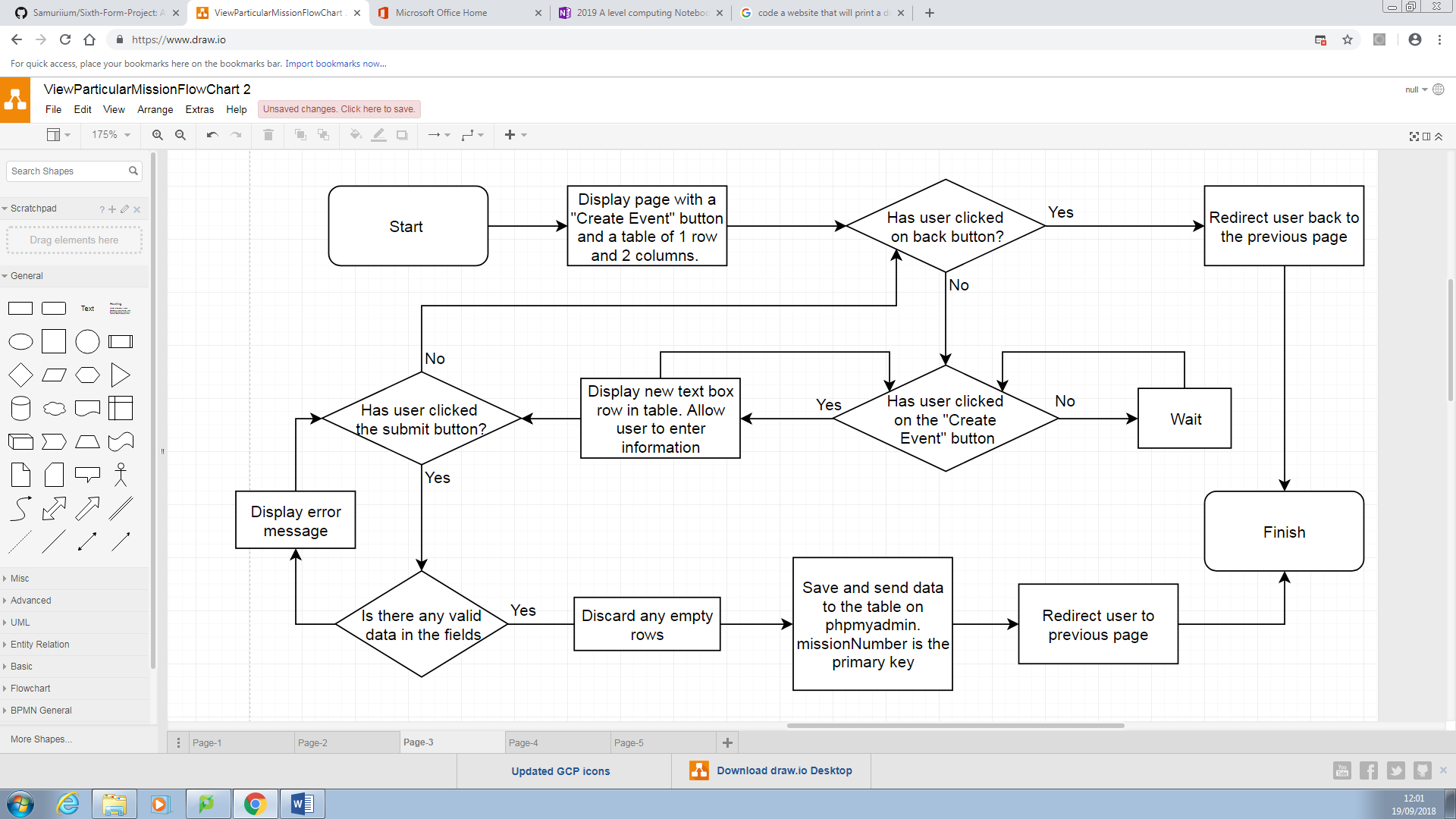
**Objective 5 – Creating a timeline of events**

This will be achieved by allowing the user to click a button to input the time of the event, as well as any notes. Once inputted, the user can submit the event which can be stored in a database. The end user can have unlimited events for a particular mission. These events will be stored in chronological order.

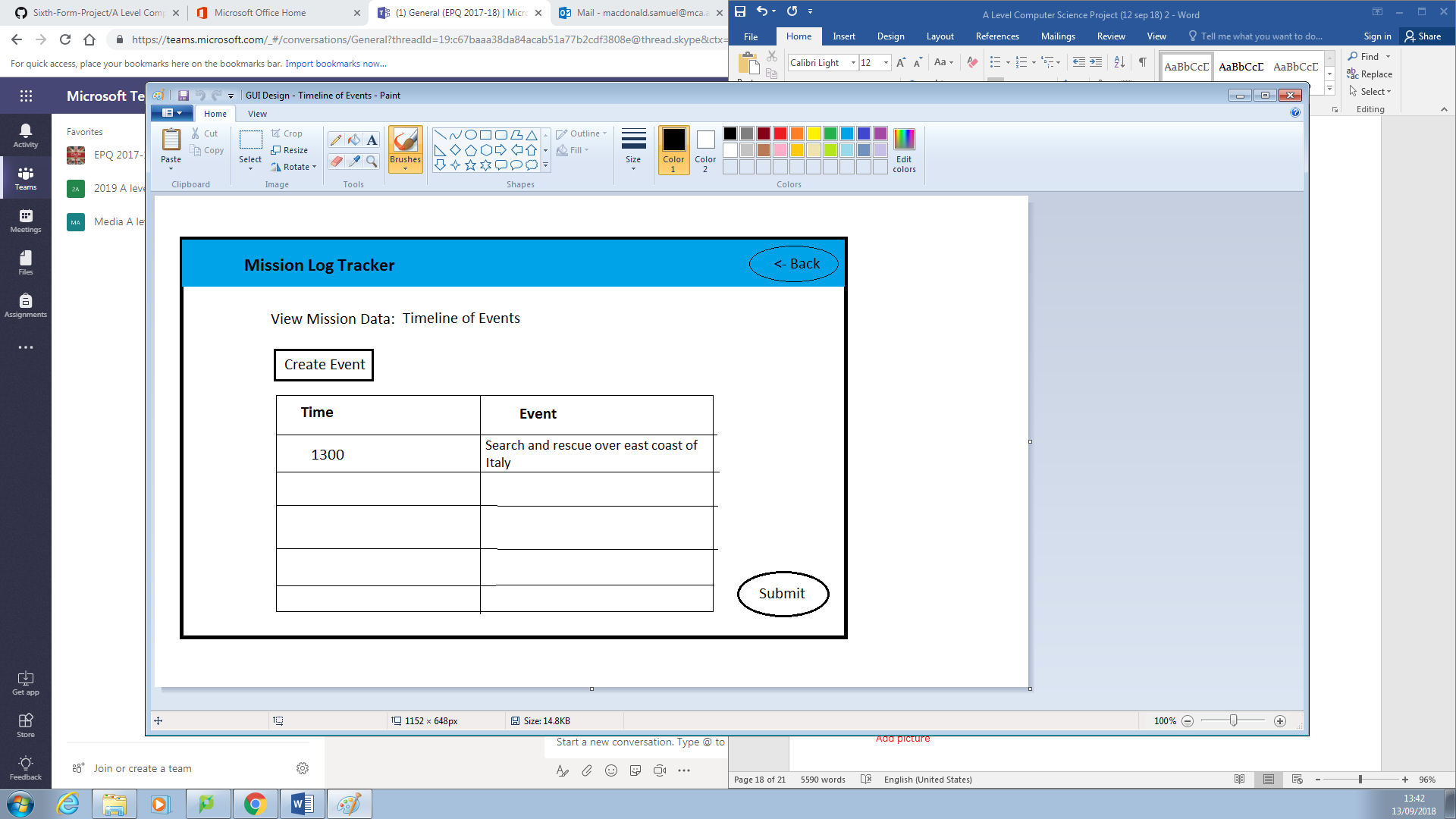
Success Criteria:

* Objective 5 must:
  + Display the “Import”, “Submit”, “Back” button and the table
  + Increase the table by one row if the user clicks the “Create Event” button
  + Allow the user to enter text into the table’s fields
  + Allow the user to submit the data
  + Alert the user if there is any invalid data
  + Save sand send the data to the database if ALL data is valid
  + Discard any empty rows and do not send them to the database
  + Redirect the user to the previous page once data is stored in database

Flow Chart:



GUI Diagram



When the user first accesses this page, they’ll be prompted to create an event; to do this, the user can click on the “Create Event”. This will display a table of two rows and two columns, the first row will be the table header (as shown) and all the rows below that will be text boxes for the user to enter data. The user can keep adding events by clicking a button – there is no limit to how many they can have. Once finished, the user can click the submit button which will send the data to the database on phpMyAdmin, if all of the data is valid. If there are empty rows in the table, then they will be discarded before the data is sent to the database. The program will then redirect them to the previous page.

Data Structure:

* I will create a variable table called ‘event \_table’ which will help me decide on what the datatype is and how it will be validated.
* The event\_table will store the data for the Timeline of Events. This will be linked to the mission\_table via a primary key, missionNumber.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| **Variable Name** | **What is it** | **What it stores** | **Data Type** | **Validation** |
| missionNumber | Mission Number | The number allocated to a particular mission | Integer | Checks if the user enters an integer. |
| timeOfEvent | The time of the event | The time in HHMM format | Integer | Checks if the user enters is an integer. |
| eventDetail | Event information | Details about what occurred during the event | String | Checks if the user enters is an string. |

* An ERD
  + This will provide a visual relationship between the multiple tables in the database, stating the primary keys linking the tables together, as well as the type of relationship. In this case, the relationship is one-to-many because one mission can have more than one event and more than one event can only be assigned to one mission.



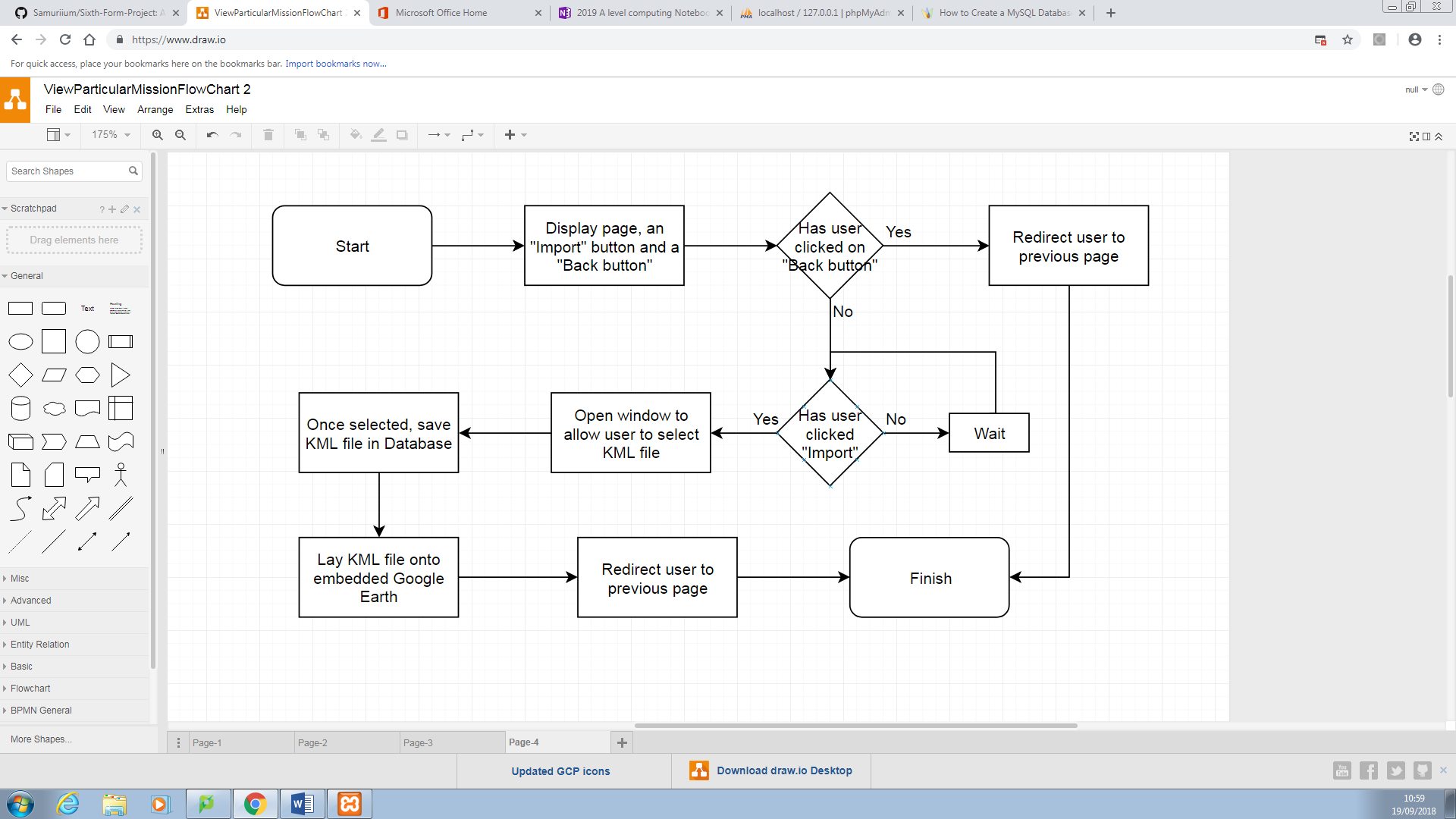
**Objective 6 – Importing a KML**

The user can import a KML, by clicking on “Import KML”, which will give them a pop-up menu of their documents. The user can select the file and click “Import”. This will then embed Google Earth into the program, and the KML on top of the map, allowing the user to view the route of the flight mission.

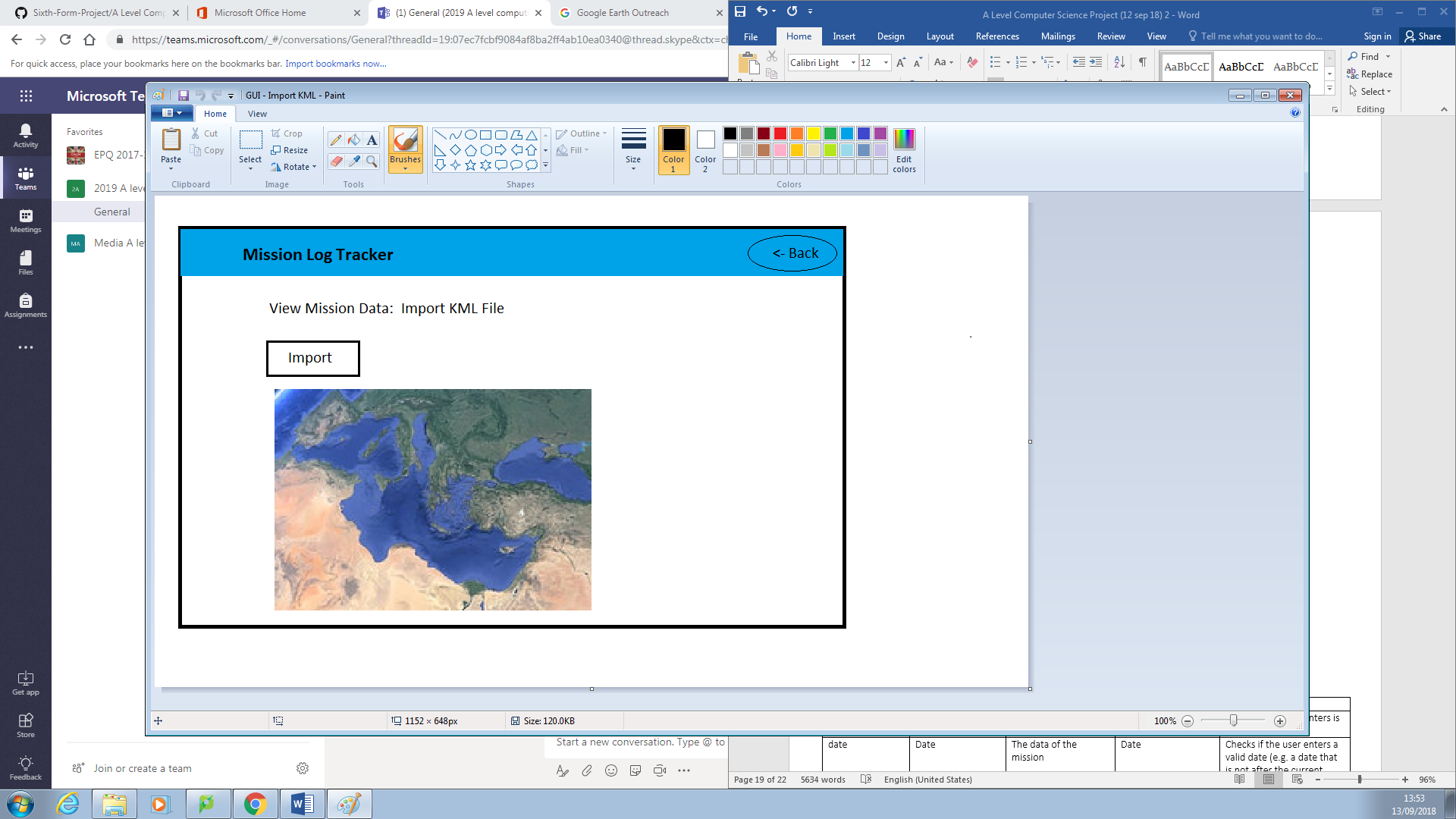
Success Criteria:

* Objective 6 must:
  + Display the “Import”, “Submit” and “Back” button
  + Allow the user to search and select for a KML file once they have clicked on “Import”
  + Lay the KML file on top of the embedded Google Earth
  + Save the file for later reference

Flow Chart:



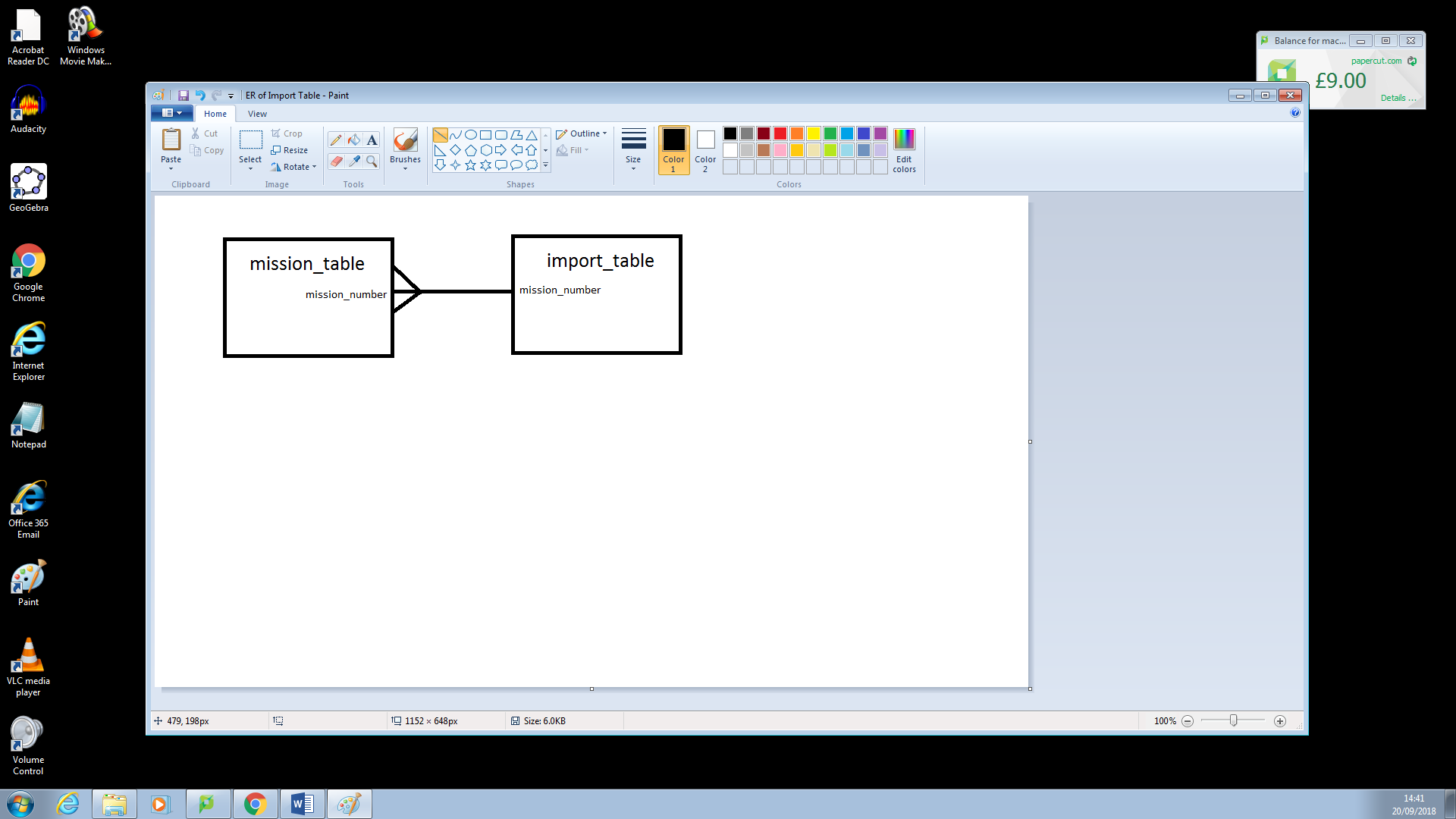
GUI Diagram:



When the user clicks on the ‘Import KML’ button, they will be redirected to a new page where the user can find and import the KML. The KML data will be placed on top of the Google Earth map, which will be embedded into the webpage.

Data Structure:

* If the user would to print off the document at a later date, then I would need to use another table to store the KML file.
* The table connecting the mission\_table to the import\_table is a many-to-one relationship because one mission can only have one KML file whereas one KML file can be for many missions.



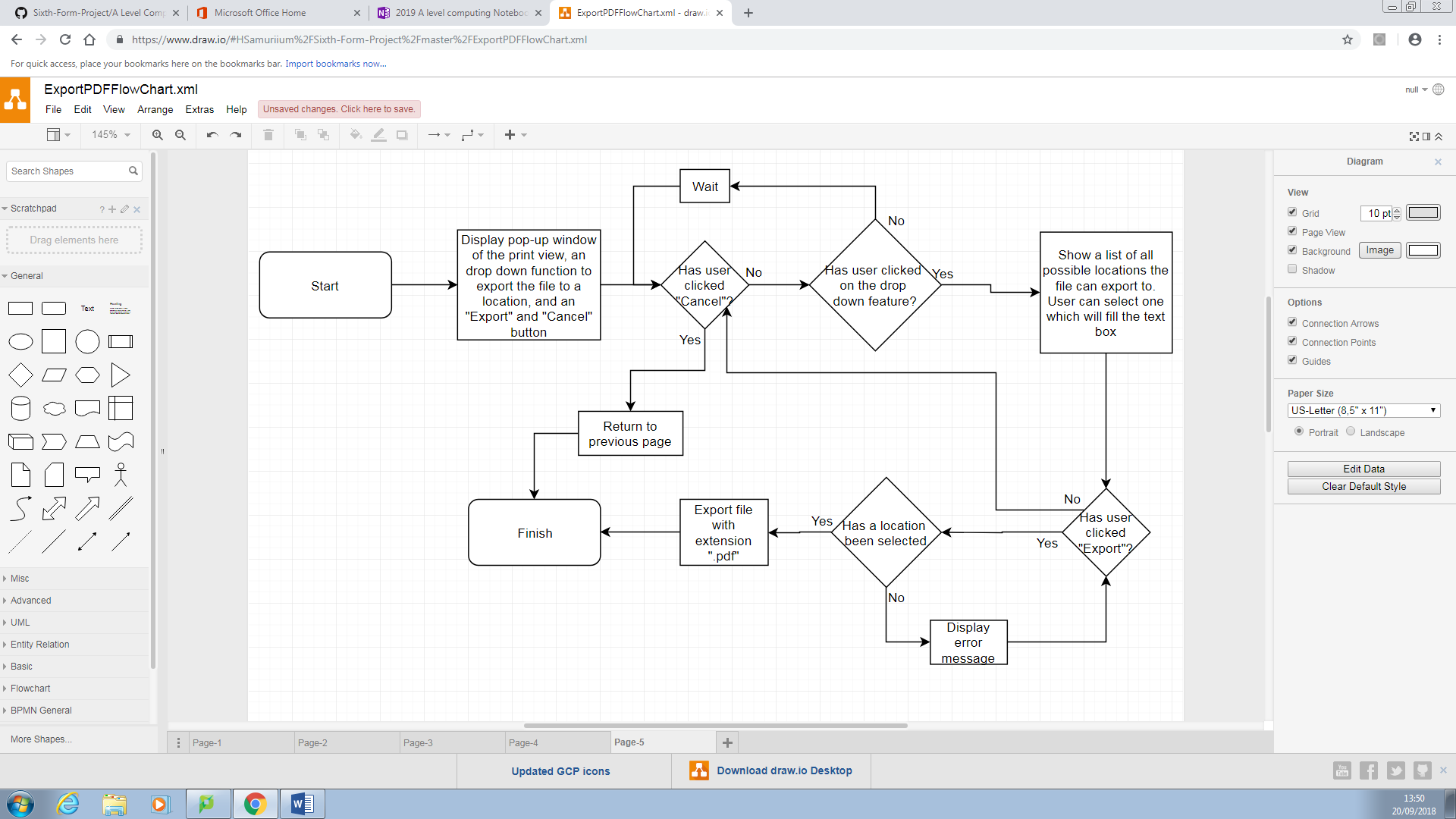
**Objective 7 – Exporting as a PDF**

To achieve this, there will be an “Export as PDF” button. If the user clicks on this button, the program will display a print menu, selecting the size of the paper, if it was going to be double-sided and any connected printers. I will try and make this similar to the print function on a computer in Word Processing. Once the user clicks “Print” then the document will print.

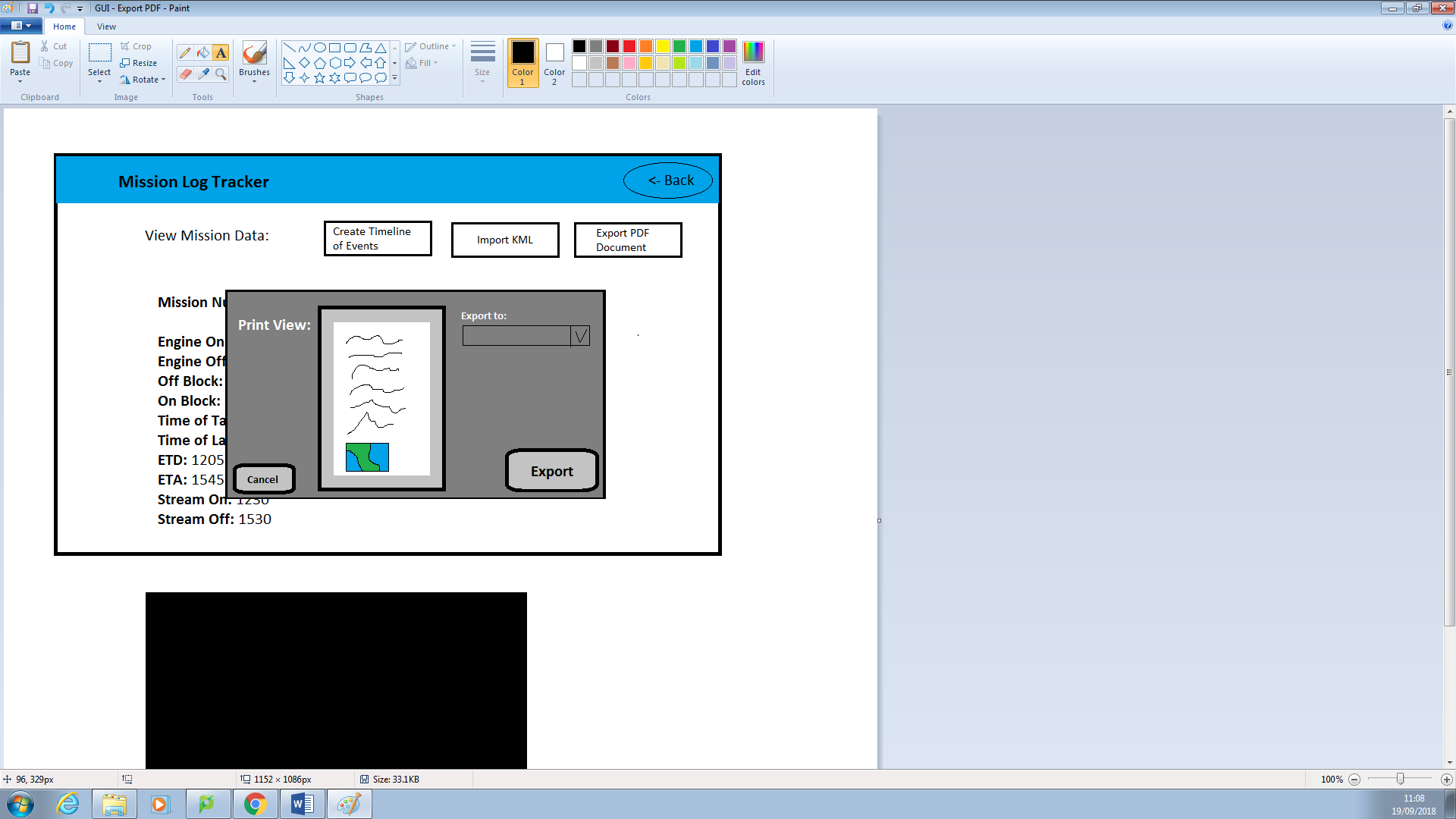
Success Criteria:

* Objective 7 must:
  + Display a pop up menu when the user clicks on “Export PDF Document”
  + Display the print view of the document before its printed, making sure the user is satisfied with the end product
  + Allow the user to cancel the pop-up menu by clicking the “Cancel” button
  + Allow the user to use a drop down feature to select a location to export the file to
  + Check whether the user has selected a location after they have clicked “Export”

Flow Chart:



GUI Diagram:



This pop-up window will appear once the user has clicked the export button. This is where the user can see the document in print view, and select a location to export to. Once they are satisfied, the can click the export button.

Data Structure:

* The location would need to be stored in a variable because the location would be used later on when the program will go to export the file. A table would not be necessary because

## Program Specifications

The program must:

1. Have a simplistic main menu to allow the user to switch between setting the data and viewing the data
   * To prevent unorganised, confusing aspects of the solution.
2. Allow the user to create daily data
   * So the end user can keep a daily record of flight missions.
3. Store the daily data into a table in a database
   * Storing data keeps a record of flight missions, allowing the end user to come back to it at a later date.
4. Allow the end user to access and view previously created data, viewing all of the mission’s metadata
   * So the end user can import a KML or export as a PDF at a later data.
5. Allow the user to import a KML
   * So the end user can physically see the route of the mission.
6. Allow the user to create a timeline of events
   * This allows the end user to make notes and document any data during the mission.
7. Allow the user to export the document and print as a PDF
   * This allows the user to have a physical copy of a daily record.
8. Contain validation
   * This will check whether the user has entered anything valid, or anything at all into the textboxes
9. Be simple to use and navigate
   * Having a simplistic design to navigate through will be measured using the end user’s feedback

In the homepage module, there will be different sections allowing the user to access different parts of the program; this will be simplistic and easy to use, meeting Program Specification 1. When adding data, there will be a form which the user can enter information. Validation will check whether the user has clicked the text box and if any fields have been entered, satisfying Program Specifications 2 and 3. Within the module that allows the user to view the data, Program Specification 4 is met because the program will allow the user to view the data in an ordered list. The module that imports a KML meets Program Specification 5 because …

## Smaller computational solutuions

Broken the problem down systematically into a series of smaller problems suitable for computational solutions, explaining and justifying the process.

Decomposition can be applied to the problem by breaking it down into smaller, individual modules. Each module can be tested separately. One module, is allowing the user to enter and upload the data to a database. This can be broken down from the main solution because I can test it independently from the other modules, checking whether data has been added to the database after I click the submit button.

Furthermore, another module is calculating the flight duration (MSN) from subtracting the ETD variable from the ETA variable, which uses decomposition because I can test this function independently from the main program.

When the user selects a particular Mission, they have the option to import a KML, export as a PDF or go back to the previous page. Each one of these I can program and test the functionality of the module independently from one another. When the user wants to import a KML file, I can use decomposition to tests importation as this would have embed a Google Earth file within the program to work.

## Alogorithms

## Described the solution fully using appropriate and accurate algorithms justifying how these algorithms form a complete solution to the problem.

One of the main algorithms I would need to incorporate into this program is a sorting function, either using a bubble sort and/or an insertion sort. This would be used for when the program needs to order the timeline of events in chronological order, or when the program lists the inputted data via the mission number in the “View Mission Data” function.

I will use a conditional algorithm to validate the “Set Mission Data” function. This will check whether each of the attributes store a value when the user clicks the submit button. If there are any attributes with missing data or containing an invalid data type (e.g. user enters a string instead of an integer), then the program will alert the user before they can proceed.

For the Timeline of Events, I will use a WHILE loop which will keep asking the user to enter the time of the event and the event data until they no longer want to input data. I could use a FOR loop but if the user does not know how many records would be required for each mission, then this would seem pointless. With a WHILE Loop, the program can keep asking the user to enter data until they say otherwise which fits well with the client’s requirements.

## Usability features

Described, justifying choices made, the usability features to be included in the solution.

## Within the homepage, there will be two options: adding and viewing the data. The user can only select to add the data; the viewing section will be currently inactive because they have been no data added beforehand. Once the user has access to viewing the data, there will be a back button which takes them back to the home screen.

When the user creates a mission, they will have to enter the date but in the correct format. If, for example, the user enters ‘12th September 2018” then the program would not recognise this format – they would have to enter “12/08/18”. To solve this issue, I will incorporate a drop-down feature for the date, allowing the user to select a date in the correct format, preventing any errors from occurring.

One attribute the user would need to calculate us the duration of the flight (msnTime) but human error will result in false data. To overcome this issue, the program will calculate the msnTime variable by subtracting the takeOffTime variable from the landTime variable.

If the user is deep within the program and would like to quickly go back to the homepage, then they can do one of two options: repeatedly press the ‘Back’ button until they return to the homepage or click the website title ‘Mission Log Tracker’ which will redirect them back to the homepage.

## Key variables, data structures and classes

## Identified and justified the key variables / data structures / classes (as appropriate to the proposed solution) justifying and explaining any necessary validation.

## The test data used in the Iterative development

Identified and justified the test data to be used during the iterative development of the solution.

## further data to be used in post development stage

Identified and justified any further data to be used in the post development phase.

# **SOLUTION DESIGN**

## software requrements

My client, who currently uses a Windows computer, would like a web application so I need to figure out how I would get PHP Hypertext Preprocessor (PHP) on my client’s computer, and after doing some research, I found that there are several ways to do so. My client would need to install Microsoft Web Platform Installer (Web PI) which can allow them to install Internet Information Services (IIS). Web PI is a package management system that helps users install non-commercial development tools whereas IIS is a web software package used to host websites and other content on the internet. I’m making a solution for a single client – not a company – so my client probably does not have Web PI or IIS installed already (if I was making this for a company who has their own servers, most likely they would have already installed Web PI and IIS and therefore they could integrate my solution into their working environment.)

Alternatively, I could install PHP by using Web PI. Coding in PHP will allow me to use a database to store information on phpMyAdmin. I can also install XAMPP which will allow me to test the solution before I release it to the client. However, with XAMPP, I would need to open the application, run Apache and MySQL (in that order) every time I would want to access the solution via the browser and phpMyAdmin, and when I would want to shut down the computer, I would need to stop Apache and MySQL.

Moreover, my client can download and install PHP and WinCache. WinCache stands for Windows Cache Extension for PHP is a PHP accelerator which makes PHP applications run faster on Windows and Windows Server. The extension consists of PHP opcode cache, user data cache, session cache, file system cache and relative path cache. However, this may be more of a simplistic way of doing this, I would still need to install IIS so either way of doing this wouldn’t differ from each other.

**Bibliography**

* <https://docs.microsoft.com/en-us/iis/application-frameworks/scenario-build-a-php-website-on-iis/configuring-step-1-install-iis-and-php>
* <https://searchwindowsserver.techtarget.com/definition/IIS>
* <https://editrocket.com/articles/php_windows.html>